

Chapitre 6 : Apprentissage Synergique Multi-Échelle et Structures Fractales

6.1. Introduction

Le **chapitre 6** prolonge les fondements et l'architecture posés dans les chapitres précédents (notamment le chapitre 5 sur la mise en place technique d'un SCN) pour montrer comment on peut **gérer** et **exploiter** la diversité d'échelles dans le DSL. À l'issue de ce chapitre, on comprendra en quoi la multi-échelle ouvre la voie à une **coordination** plus souple entre divers niveaux (local, global...) et comment la fractalité peut modéliser ou expliquer l'émergence de motifs répétitifs au sein du réseau de synergies.

6.1.1. Contexte et Motivation

Le **DSL** (Deep Synergy Learning) repose sur des entités \mathcal{E}_i et leurs liens $\omega_{i,j}$, guidés par la fonction de synergie $S(i,j)$. Jusqu'ici, nous avons souvent envisagé la formation de clusters ou la dynamique d'auto-organisation à une **même** échelle (entités ou sous-ensembles d'entités). Or, dans de nombreux systèmes naturels (cérébraux, écologiques, sociaux) ou artificiels (réseaux complexes, infrastructures distribuées), on observe des **structures** à **plusieurs niveaux** de résolution :

- Des **micro-clusters** (petits groupes fortement cohérents),
- Des **macro-ensembles** plus vastes,
- Des **niveaux intermédiaires** jouant un rôle de pivot.

De plus, il se peut qu'un **même motif** (par exemple, un cluster en "étoile" ou un schéma cyclique) se **reproduise** à diverses échelles, suggérant une **auto-similarité** ou un comportement dit *fractal*.

6.1.1.1. Rappel : Le DSL (Deep Synergy Learning) fonctionne sur la base de synergies entre entités, mais l'échelle à laquelle on étudie ces synergies peut varier (local, global, intermédiaire)

La formulation fondamentale du **Deep Synergy Learning (DSL)** repose sur une **règle d'auto-organisation** qui met à jour les liaisons $\omega_{i,j}(t)$ en fonction de la **synergie** $S(i,j)$ évaluée entre les entités \mathcal{E}_i et \mathcal{E}_j . Cette dynamique s'exprime par l'équation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où η représente le taux d'apprentissage et τ un facteur de régularisation. La **règle DSL** ainsi définie peut être généralisée – qu'elle adopte une forme additive, multiplicative ou autre – conformément aux paradigmes exposés au **chapitre 4**. Notons que cette **mise à jour** ne dépend pas intrinsèquement de la taille du système étudié, ce qui permet de l'appliquer à divers **sous-ensembles** d'entités, et donc d'examiner le système à plusieurs **échelles** : **micro**, **intermédiaire** et **macro**.

À l'échelle **micro**, l'analyse se concentre sur la formation de petits **clusters** d'entités fortement interconnectées. Par exemple, un groupe de robots collaborant sur une tâche spécifique ou un ensemble de concepts quasi synonymes dans un SCN linguistique constitue un cas typique d'application. En revanche, à l'échelle **macro**, l'émergence de vastes ensembles regroupant plusieurs sous-clusters permet d'appréhender la structure globale du **SCN**, où la **synergie** se déploie de façon plus étendue.

Il est important de souligner que, bien que la formule de mise à jour

$$\omega_{i,j}(t+1) = F(\omega_{i,j}(t), S(i,j))$$

reste identique, son application à différents niveaux induit des comportements distincts. En effet, l'**analyse multi-échelle** permet soit de « zoomer » sur un ensemble restreint d'entités et leurs interconnexions, soit de « dézoomer » pour englober la totalité du **SCN**. Ce choix de perspective ouvre la voie à l'étude des interactions entre les différents niveaux et à la mise en évidence de structures récurrentes, parfois de nature **fractale**.

Lorsque la logique d'**auto-organisation** – intégrant **synergie**, renforcement et inhibition – est appliquée simultanément sur plusieurs **échelles**, un phénomène d'**auto-similarité** peut émerger. Par exemple, le motif local, tel qu'un « cœur dense » entouré de liaisons moins fortes, se retrouve à une échelle supérieure, traduisant ainsi une **invariance d'échelle**. Mathématiquement, cela signifie que la fonction $F(\omega, S)$ ne dépend pas explicitement de la dimension du système, permettant à un **cluster local** de reproduire le comportement observé dans le **réseau complet**.

Les bénéfices d'une approche **multi-échelle** sont multiples. D'une part, elle autorise une analyse précise d'un **cluster local** tout en préservant une vision d'ensemble du **SCN**, renforçant ainsi sa **modularité** et sa **robustesse**. D'autre part, dans le cadre du **DSL**, cette organisation hiérarchique facilite la gestion de systèmes de grande envergure, notamment via l'introduction d'un **meta-SCN** (cf. **chapitre 5.7.2**), qui constitue un niveau supérieur d'agrégation et applique la même dynamique DSL aux sous-blocs locaux pour garantir une cohérence globale.

L'harmonisation de la **mise à jour** des liaisons sur différentes **échelles** constitue un pilier central du **Deep Synergy Learning (DSL)**. Ce mécanisme assure non seulement l'**invariance fonctionnelle** de la règle d'auto-organisation, mais offre également une perspective unifiée pour appréhender la complexité des interactions hiérarchiques au sein des systèmes synergiques.

6.1.1.2. Nécessité de s'intéresser aux différents niveaux (micro-clusters vs. macro-structures) et à la possible auto-similarité (concept fractal) qui peut émerger dans la répartition des entités

Le **Deep Synergy Learning (DSL)** repose sur un paradigme d'**auto-organisation** dans lequel le renforcement – ou l'inhibition – des liaisons $\omega_{i,j}$ ne se limite pas à une unique **échelle** d'observation. En de nombreuses configurations pratiques, l'ensemble des entités $\{\mathcal{E}_i\}$ et leurs interconnexions s'**ordonnent** simultanément à divers **niveaux**. Ainsi, on peut identifier, d'un côté, un **micro-cluster** constitué d'un petit nombre d'entités fortement interconnectées, et de l'autre, un **macro-cluster** plus étendu dans lequel la cohésion moyenne, quoique notable, est moins marquée. Cette **pluralité d'échelles** soulève des problématiques tant méthodologiques que mathématiques, notamment concernant la manière dont les processus de renforcement, définis par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

se déploient tant au niveau **local** (pour quelques nœuds) qu'au niveau **global** (pour des regroupements massifs).

Les **micro-clusters** désignent ainsi de petits noyaux dans lesquels un **faible** nombre d'entités est relié par une **densité** importante de liaisons, c'est-à-dire par des valeurs élevées de $\omega_{i,j}$. Par exemple, dans un **SCN** dédié à la gestion d'un essaim de robots, un micro-cluster pourrait correspondre à 2 ou 3 robots partageant une **synergie** intense – par exemple, en raison d'une mission commune dans une zone limitée. De même, dans un **SCN** cognitif, il s'agirait de 4 ou 5 concepts étroitement liés par leur co-occurrence ou leur complémentarité sémantique. En revanche, un **macro-cluster** inclut un **grand** nombre d'entités, formant un ensemble plus dispersé mais jouant un rôle systémique – tel qu'un sous-réseau d'agents répartis sur l'ensemble du système.

Sur le plan **mathématique**, un micro-cluster se manifeste comme un sous-graphe où les liaisons $\omega_{i,j}$ atteignent des niveaux particulièrement élevés, tandis qu'un macro-cluster s'étend à un niveau supérieur en englobant plusieurs micro-clusters internes. Cette coexistence de niveaux multiples traduit une **variabilité** inhérente aux rôles, aux proximités et aux tâches assignées aux entités, illustrant la capacité du système à s'adapter de manière flexible à des contraintes hétérogènes.

A. Mécanismes locaux vs. globaux et hiérarchie émergente

La même **règle** d'évolution

$$\omega_{i,j}(t+1) = F(\omega_{i,j}(t), S(i,j))$$

s'applique a priori de manière uniforme à l'ensemble du **SCN**. Toutefois, le **renforcement** (ou l'inhibition) peut se manifester de façon plus **rapide** au niveau **local**. En effet, quelques entités présentant une synergie élevée renforcent rapidement leurs liaisons, donnant lieu à la formation d'un micro-cluster en quelques itérations. À l'échelle **global**, la formation ou la stabilisation d'un grand cluster exige une durée plus importante ou un enchaînement de fusions successives entre micro-clusters, traduisant une dynamique temporelle étendue.

On observe ainsi l'émergence d'une véritable **hiérarchie** dans le **DSL**. Le système ne converge pas uniquement vers « un cluster global » ou vers un simple ensemble de micro-groupes, mais tend à produire des **strates** intermédiaires reliant les niveaux **micro** et **macro**. Cette **stratification** n'est

pas un artefact passager ; elle illustre la capacité du **SCN** à gérer différentes **échelles** de granularité. Les micro-groupes, caractérisés par une forte cohérence interne, coexistent avec des macro-ensembles qui, bien que présentant une affinité moyenne, sont suffisants pour définir une « communauté » large et structurée.

Sur le plan **ingénierie**, cette hiérarchie est exploitée pour segmenter des **tâches** multiples, en permettant à des sous-groupes autonomes de fonctionner tout en conservant une *vue* d'ensemble sur le système. Par ailleurs, du point de vue de l'**analyse**, cette organisation hiérarchique souligne la **modularité** d'un SCN, une propriété particulièrement précieuse dans la gestion de grands systèmes distribués (cf. Chap. 5.7).

B. Auto-similarité et phénomène fractal

La persistance des mêmes **mécanismes** de renforcement et d'inhibition à chaque **niveau** conduit inévitablement à l'émergence de **motifs** qui se **répètent** à différentes échelles. Ce phénomène, désigné par l'**auto-similarité** – ou **fractalité** – se traduit, sur le plan **mathématique**, par une invariance d'échelle. Autrement dit, la *topologie* ou la *distribution* des liens $\omega_{i,j}$ observée dans un petit cluster peut *ressembler* à celle du motif global, à une **taille** différente. Une telle invariance apparaît fréquemment dans des systèmes biologiques ou cognitifs, où des *lois uniformes* – comme le renforcement local et le feedback global – induisent des structures *self-similar*.

Pour illustrer ce concept, considérons un micro-cluster composé de 5 entités qui adopte une configuration quasi « biclique », c'est-à-dire deux sous-ensembles fortement interconnectés par des poids élevés. On peut alors constater que, dans un grand cluster de 100 entités, se reproduit un motif analogue : un sous-ensemble principal et un second sous-ensemble, reliés par un pont central. L'origine de ce *pattern fractal* réside dans le fait que la **règle DSL**, exprimée par

$$\omega_{i,j}(t+1) = F\left(\omega_{i,j}(t), S(i,j)\right),$$

n'exclut pas, à petite ou grande échelle, l'apparition de comportements similaires. Ainsi, la **fractalité** se révèle être un indicateur essentiel d'une dynamique cohérente, caractérisée par la répétition de la forme d'organisation tant localement que globalement.

C. Pourquoi étudier ces multiples échelles ?

D'un point de vue **fondamental**, la pluralité d'échelles témoigne de la **richesse** intrinsèque du DSL. La plasticité du SCN permet d'organiser simultanément le système à différents niveaux.

Au **microniveau**, les entités se regroupent en micro-clusters caractérisés par des interconnexions intenses et une forte cohésion locale. Au **macroniveau**, plusieurs micro-clusters s'agrègent pour former de vastes ensembles, dont la densité est moindre mais dont la signification reste tout aussi importante.

La **robustesse** du système se manifeste dans la capacité d'un micro-cluster à rester stable même lorsque la configuration du macro-cluster évolue. Par ailleurs, la dynamique globale demeure résiliente face aux variations locales.

L'unité énergétique qui gouverne l'ensemble du réseau permet une observation différenciée entre les niveaux, révélant des phénomènes de coopération locale et une organisation communautaire globale.

Enfin, ces propriétés offrent de précieuses **clés d'analyse**. Lorsque la structure présente une auto-similarité marquée, il devient possible de quantifier son degré fractal à l'aide d'indices topologiques ou géométriques, ce qui éclaire la dynamique d'auto-organisation.

6.1.2. Objectifs du Chapitre

Après avoir souligné en (6.1.1) l'importance d'observer plusieurs **échelles** – **micro**, **macro** et intermédiaires – dans un **SCN** et la possibilité d'un **phénomène fractal**, il est essentiel de préciser les objectifs du chapitre.

La **multi-échelle** ne se réduit pas à une curiosité, mais constitue un **cadre conceptuel** et **opérationnel** qui étend le champ du **DSL (Deep Synergy Learning)** et soulève de nouvelles questions en **modélisation**, **dynamique** et **implémentation**.

6.1.2.1. Expliquer comment la *multi-échelle* se traduit dans le DSL, et en quoi elle permet d'organiser et de coordonner différents niveaux d'abstraction

Le **Deep Synergy Learning (DSL)**, dans son expression la plus simple, décrit un réseau d'entités $\{\mathcal{E}_i\}$ dont les liaisons $\omega_{i,j}$ évoluent par le biais d'une règle d'**auto-organisation** (additive, multiplicative, etc.).

Au cours des chapitres précédents, on a principalement envisagé cette dynamique dans le cadre d'un “**niveau unique**”. Chaque entité interagit avec les autres via des synergies $S(i,j)$, et l'on observe un **SCN** global où l'on repère la formation de clusters (ou la répartition en sous-structures).

Toutefois, dans des systèmes complexes, cette perspective **mono-niveau** peut se révéler insuffisante, car l'**organisation** émerge souvent à **plusieurs échelles**. C'est précisément l'enjeu de cette section (6.1.2.1) : clarifier en quoi la *multi-échelle* s'intègre dans la logique du DSL et comment elle améliore la **coordination** entre divers niveaux d'abstraction.

Traduction de la multi-échelle dans le DSL

Sur le plan **mathématique**, la mise à jour des pondérations suit une équation générale du type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j)\tau \omega_{i,j}(t)] \quad (\text{ou autre forme}).$$

Quand on introduit la *multi-échelle*, on considère que la **même** forme de règle DSL peut s'appliquer à plusieurs **niveaux** simultanément ou successivement. Autrement dit, chaque entité \mathcal{E}_i peut apparaître à un niveau “micro” (représentation plus fine) ou à un niveau “macro” (regroupement plus large de sous-entités), et la **dynamique** $\omega(t) \rightarrow \omega(t+1)$ peut être itérée aussi bien sur des micro-groupes que sur des macro-groupes.

Cette approche repose sur deux principes fondamentaux.

Le premier principe consiste à redéfinir la notion d'« entité » selon l'échelle d'observation. À cet égard, on distingue une **micro-entité** — par exemple, une petite brique de connaissances ou un

groupe restreint de neurones — d’une **macro-entité**, qui correspond à un cluster plus étendu ou à une aire fonctionnelle plus large.

Le second principe consiste à maintenir la **règle d’auto-organisation** dans ses fondements. Ainsi, la mise à jour des liaisons $\omega_{i,j}$ s’effectue toujours en fonction de la synergie $S(i, j)$ et d’un terme de décroissance (ou d’inhibition).

Enfin, l’acte de « changer d’échelle » revient, sur le plan analytique, à définir un second SCN dont les nœuds sont des **sous-ensembles** du premier réseau et dont la **synergie** entre ces macro-nœuds est évaluée à partir d’agrégations locales des $\{\omega_{i,j}\}$.

Organisation et coordination des différents niveaux

En introduisant la notion de **multi-échelle**, il ne s’agit pas seulement de contempler plusieurs niveaux de granularité ; on veut également que ces **niveaux s’influencent** mutuellement. Cela se traduit par :

- Des **flux ascendants** (bottom-up) : un **micro-cluster** émergeant (un petit ensemble fortement connecté à l’échelle locale) peut “remonter” vers une **macro-organisation**, se faisant reconnaître comme une entité cohérente dans le niveau supérieur.
- Des **flux descendants** (top-down) : le **macro-niveau**, doté d’une vue globale, peut imposer ou suggérer des contraintes aux micro-nœuds, par exemple en renforçant la liaison entre deux micro-entités si leur appartenance à un même macro-cluster s’avère cruciale pour la cohérence globale.

Ce processus de **coordination** multi-niveau confère au DSL une capacité à gérer des systèmes de **taille** et de **complexité** potentiellement beaucoup plus élevées. On peut, par exemple, répartir la **mise à jour** $\omega(t)$ selon des blocs locaux (micro) supervisés par un “meta-SCN” (macro), suivant les principes énoncés en Chapitre 5.7.

Chaque **échelon** demeure fidèle à la règle DSL, mais opère sur des entités (ou clusters) d’échelle différente. On obtient alors une **architecture** hiérarchique et **modulaire** du SCN, évitant l’explosion combinatoire ou le chaos d’une organisation totalement “plate”.

Exemples concrets

Dans un **système cognitif** inspiré du cerveau, on peut assimiler le niveau “micro” à des micro-colonnes corticales (ou petits sous-réseaux de neurones) et le niveau “macro” à de grandes aires (voies sensorielles, cortex associatif). Les **mêmes** principes de synergie (co-activation neuronale) et d’update $\omega_{i,j}$ (synaptique) se répètent, de sorte que ce qui est acquis localement se reflète globalement. Dans un tel cadre, la *multi-échelle* illustre l’émergence et la stabilisation d’un schéma “macro” (ex. “aire visuelle connectée à aire frontale”) tout en conservant un ajustement local de micro-groupes.

En **robotique** multi-agent, le niveau “micro” peut consister en quelques robots géographiquement proches s’échangeant un flot de données et accroissant $\omega_{i,j}$ si leur collaboration s’avère fructueuse, tandis que le niveau “macro” traite la **mission** d’ensemble et “vois” un cluster englobant ces robots plus un autre groupe. Le SCN multi-échelle intègre alors la capacité de chaque micro-cluster à se spécialiser, tout en unifiant la planification globale.

Regrouper plusieurs **niveaux** dans un **SCN** soulève quelques questions spécifiques :

- **Comment** modéliser la synergie entre macro-entités ? Faut-il calculer la **moyenne** ou le **maximum** des synergies entre leurs micro-composantes ?
- **Comment** gérer la mise à jour ω si l'on bascule d'un niveau à l'autre ? On peut imaginer un *meta-SCN* (Chap. 5.7.2) où la synergie macro \hat{S} est engendrée par l'agrégation des synergies micro $\{S(i, j)\}$.
- **Comment** éviter la désynchronisation entre micro- et macro-niveaux ? Cela rejoint les principes de coordination décrits plus haut : la dynamique est plus riche qu'un simple renforcement local, car un macro-cluster (englobant plusieurs micro-clusters) peut imposer ou suggérer une consolidation de certaines liaisons $\omega_{i,j}$.

6.1.2.2. Introduire la *fractalité* comme un cadre pour appréhender l'*auto-similarité* des patterns synergiques à diverses échelles

Lorsque l'on considère la **multi-échelle** dans un **SCN** (Synergistic Connection Network), il n'est pas rare de voir apparaître des **motifs** qui se **répètent** ou se **reproduisent** à plusieurs **niveaux** d'observation. Dans des systèmes complexes, cette forme de **répétition** ou d'**invariance d'échelle** est précisément décrite par la notion de **fractalité**.

Par ailleurs, lorsqu'un micro-cluster local « ressemble » — dans sa structure ou sa répartition de pondérations — à un macro-cluster plus vaste, on suspecte l'existence d'un **processus** fractal au sein du DSL. Le but de cette section (6.1.2.2) est donc de situer la *fractalité* dans la logique multi-échelle du **Deep Synergy Learning** et de montrer en quoi elle offre un *cadre* théorique pour analyser et comprendre l'**auto-similarité** qui peut émerger dans la répartition des entités.

A. Rappel du Concept de Fractalité

Une **fractalité** (ou structure **fractorisée**) se caractérise par l'**auto-similarité** à des échelles multiples. Géométriquement, dans un ensemble fractal au sens classique (Cantor, Von Koch, Sierpinski, etc.), la même “forme” se reproduit à plus petite échelle, parfois par homothétie ou transformations plus compliquées, d'où la notion d'**invariance d'échelle**. Sur le plan **mathématique**, une définition usuelle s'appuie sur :

$$\text{Auto-similarité} \quad \Leftrightarrow \quad \exists \text{ transformations } f_1, \dots, f_m: E = \bigcup_{k=1}^m f_k(E),$$

pour un ensemble E dit fractal. Les notions de **dimension fractale** ou **loi de puissance** (règle de type $\text{prob}(\text{taille} > s) \sim s^{-\alpha}$) apparaissent comme des marqueurs typiques.

B. Lien avec l'Auto-Similarité dans un Réseau

La **fractalité** s'applique aux réseaux ou graphes. On considère qu'un réseau exhibe une **structure fractale** lorsque, en zoomant sur un sous-réseau ou un cluster interne, la topologie obtenue est équivalente – à un changement d'échelle près – à celle du réseau global. D'un point de vue

statistique, ce comportement se manifeste par des **lois de puissance** (comme la distribution des degrés ou la distribution des tailles de clusters) et par la conservation de certaines **mesures** topologiques lorsque le réseau est reconfiguré à une échelle réduite (par exemple, en regroupant des nœuds en super-nœuds).

Dans un SCN, la **forme** du réseau émane des pondérations $\{\omega_{i,j}\}$. Les *patterns* synergiques typiques, tels que la formation de sous-ensembles denses et des arêtes intergroupes plus faibles, se répètent d’un micro-groupe à un macro-groupe. Ce phénomène traduit une **auto-similarité** dans la distribution des $\omega_{i,j}$, les *clusters* se formant à divers niveaux selon le même schéma de renforcement local et d’inhibition externe.

Si, en pratique, cette logique conduit à l’apparition de structures analogues à différents degrés de zoom, on peut alors parler d’une **fractalité** ou d’un **mécanisme fractal** dans la répartition des entités $\{\mathcal{E}_i\}$.

C. Fractalité comme Cadre d’Analyse pour le DSL

1. Auto-similarité dans les Patterns Synergiques

Dans un DSL, un *pattern synergique* correspond à un certain agencement de liaisons $\{\omega_{i,j}\}$ – par exemple, un cycle dense, un noyau fortement relié ou un clustering modulaire. L’**auto-similarité** signifie que la **même forme** se retrouve à plusieurs niveaux. Ainsi, un micro-cluster pourrait présenter le même type de répartition de poids (forts au centre, décroissants en périphérie) qu’un macro-cluster englobant 50 entités, simplement agrandi ou répliqué à une échelle supérieure.

Au niveau **mathématique**, ce phénomène se formalise en étudiant la “**dimension fractale**” (au sens box-counting ou similaire) du SCN. Si cette dimension présente un comportement indiquant un scaling law, cela témoigne d’une invariance d’échelle dans l’organisation. De manière heuristique, on peut également observer si la distribution des tailles de clusters $\{C_k\}$ suit une **loi de puissance**

$$P(\text{taille} = s) \sim s^{-\alpha}.$$

Ce comportement est un **signe** qu’il n’y a pas de taille préférentielle, ce qui constitue souvent un indice d’une fractalité latente.

2. Notion de “Multi-niveau” fractal

Le **multi-niveau** présenté au Chap. 6.1.2.1 dépasse la simple stratification en se transformant en une **structure fractale** lorsque, en passant d’un niveau d’agencement micro à un niveau d’agencement macro, la même dynamique DSL se reconstitue et reproduit la topologie à une échelle démultipliée.

Ce phénomène trouve des analogies dans des systèmes biologiques – où une logique identique se répète dans des modules de plus en plus grands – ainsi que dans des systèmes sociaux, où les mécanismes de collaboration s’expriment de manière similaire à la fois localement et globalement, engendrant des hiérarchies auto-similaires.

3. Bénéfices de l'Analyse Fractale

La fractalité confère au DSL une remarquable robustesse et invariance. En effet, elle permet qu'une perturbation locale soit absorbée par la répétition des structures, de sorte que si l'échelle micro subit une variation, l'échelle macro conserve son mode d'organisation et réciproquement.

Par ailleurs, en quantifiant la dimension fractale ou en repérant un exposant de loi de puissance, il est possible de caractériser la manière dont le réseau s'étale sur les diverses échelles. Ces mesures fournissent des indices précis sur l'invariance d'échelle et sur la répartition structurelle du système.

Enfin, pour favoriser une structure fractale au sein du SCN, notamment pour des raisons de modularité ou de robustesse, il est envisageable de paramétrer le DSL. En ajustant des paramètres tels que le taux η ou l'inhibition γ , on peut encourager un schéma répétitif d'organisation, orientant ainsi le système vers une configuration auto-similaire à travers les différents niveaux d'observation.

6.1.2.3. Aborder les *modèles mathématiques*, les *dynamiques associées* et l'*implémentation* de l'apprentissage multi-échelle

Lorsque l'on traite la **multi-échelle** et la possible **fractalité** au sein d'un **SCN** (Synergistic Connection Network), il ne suffit pas de se limiter aux intuitions ou aux observations qualitatives. Il est nécessaire de proposer des **modèles mathématiques** explicites qui décrivent l'émergence d'échelles.

Il convient également d'**analyser** en profondeur les dynamiques impliquées, telles que les lois de mise à jour et les couplages inter-niveaux, afin de comprendre le déroulement de l'**implémentation** concrète dans un système capable de manipuler de manière hiérarchique des entités et des clusters.

Le **DSL** (Deep Synergy Learning) doit s'adapter pour gérer des **niveaux multiples** tout en conservant la logique de renforcement et d'inhibition exposée jusqu'ici.

A. Modèles Mathématiques pour la Multi-Échelle

La première étape consiste à formuler un **SCN** à plusieurs **niveaux**. Au niveau de base (micro), l'ensemble d'entités $\mathcal{E}_1, \dots, \mathcal{E}_n$ et leurs liaisons $\omega_{i,j}^{(0)}$ sont définis. Lorsqu'un **cluster** local $\mathcal{C}_\alpha \subseteq \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ se dégage, une *macro-entité* \mathcal{M}_α est alors définie à l'échelle suivante (niveau 1) et une **nouvelle** matrice $\omega_{\alpha,\beta}^{(1)}$ est construite pour décrire les synergies entre ces macro-entités. Cette démarche se poursuit pour des niveaux supérieurs ($k = 2, 3, \dots$), aboutissant à une hiérarchie.

D'un point de vue purement mathématique, on peut procéder ainsi :

- Poser un ensemble d'entités $\mathcal{E}^{(0)}$ au niveau 0 (micro).
- Définir, par agrégation, un ensemble $\mathcal{E}^{(1)}$ de macro-nœuds, où chaque $\mathcal{M}_\alpha^{(1)} \subseteq \mathcal{E}^{(0)}$ correspond à un cluster repéré.
- Répéter cette logique en créant des ensembles $\mathcal{E}^{(k)}$ pour $k = 2, \dots, K$.

Cette approche se rapproche des techniques de **coarse-graining** en physique statistique, où l'on regroupe des sites en super-sites, ainsi que des approches de quotient de graphes en théorie des réseaux, où chaque cluster devient un super-nœud. La mise à jour $\omega(t+1) = F(\omega(t))$ doit alors être déclinée à chaque palier.

Pour les liaisons entre macro-nœuds, une fonction d'agrégation Ψ des liaisons micro est introduite. Par exemple, on peut écrire

$$\omega_{\alpha,\beta}^{(k)} = \Psi \left(\left\{ \omega_{i,j}^{(k-1)} \mid i \in \mathcal{M}_\alpha^{(k)}, j \in \mathcal{M}_\beta^{(k)} \right\} \right).$$

Une fois ces macro-pondérations définies, la règle DSL est **réappliquée** pour ajuster $\omega_{\alpha,\beta}^{(k)}$ en fonction d'une synergie $S^{(k)}(\alpha, \beta)$. Ainsi se construit un modèle multi-échelle strictement formalisé.

B. Dynamiques associées : couplage entre niveaux

En ayant défini des niveaux successifs ($k = 0, \dots, K$), il convient de préciser comment s'effectuent les mises à jour au sein du système. Trois approches principales se dégagent.

1. Dynamique indépendante

Chaque niveau applique la même règle DSL (additive, multiplicative, etc.) comme si l'on disposait de $K+1$ SCN indépendants. Le lien entre eux repose simplement sur la projection, les macro-nœuds du niveau k étant issus de sous-ensembles du niveau $k-1$.

2. Rétroaction inter-niveau

Une dynamique plus complexe autorise un feedback descendant. Ainsi, si, au niveau macro, on constate qu'un lien $\omega_{\alpha,\beta}^{(k)}$ est particulièrement fort, on incite, au niveau $k-1$, les liaisons $\{\omega_{i,j}^{(k-1)}\}$ reliant les entités micro internes aux clusters α, β à se consolider. Inversement, un cluster local du niveau $k-1$ peut remonter un signal vers la macro-liaison $\omega_{\alpha,\beta}^{(k)}$.

3. Schéma cyclique ou séquentiel

Dans certains algorithmes, on effectue d'abord la mise à jour DSL sur le niveau micro jusqu'à stabilisation, puis on agrège pour constituer des macro-nœuds et on calcule $\omega^{(k+1)}$ pour exécuter quelques étapes de mise à jour macro. Ce processus cyclique permet à chaque palier de se stabiliser partiellement avant de passer au suivant.

Ces dynamiques couplées favorisent l'auto-similarité dans la répartition des clusters, puisque la même logique de renforcement/inhibition se répercute du niveau local jusqu'au niveau global. L'équation de mise à jour $\omega(t+1) = F(\omega(t))$ se généralise ainsi en un système multi-échelle formalisé par

$$\omega^{(k)}(t+1) = F_k \left(\omega^{(k)}(t), \omega^{(k-1)}(t), \omega^{(k+1)}(t) \right),$$

introduisant potentiellement un couplage bidirectionnel entre les niveaux $k, k-1$ et $k+1$.

C. Implémentation de l'apprentissage multi-échelle : méthodes pratiques

Sur le plan **ingénierie**, on peut mettre en place une **architecture** logicielle modulaire. Chaque palier k dispose d'un module de mise à jour et d'un module de synergie qui gèrent respectivement la mise à jour $\omega^{(k)}$ via la règle DSL et le calcul de $S^{(k)}(\alpha, \beta)$ pour les macro-nœuds $\mathcal{M}_\alpha, \mathcal{M}_\beta$.

Une passerelle d'agrégation Ψ relie le niveau k au niveau $k - 1$ tandis qu'un mécanisme de feedback (optionnel) renvoie des consignes top-down au niveau inférieur. Selon la complexité du SCN, plusieurs schémas peuvent être appliqués :

- **Séquentiel**
On met d'abord à jour $\omega^{(0)}$ (micro), on regroupe en macro-nœuds, puis on met à jour $\omega^{(1)}$ (macro), et ainsi de suite.
- **Parallèle**
Chaque niveau évolue simultanément et une synchronisation est effectuée à intervalles réguliers.
- **Distribué**
Si ω est déjà réparti (cf. Chap. 5.7), la hiérarchie est répliquée par bloc local puis intégrée dans un meta-niveau global.

Des algorithmes concrets, par exemple un “hierarchical SCN”, consistent à détecter automatiquement des clusters stables au niveau k , à les étiqueter pour créer $\mathcal{E}^{(k+1)}$, à initialiser $\omega^{(k+1)}$ par agrégation, puis à appliquer la règle DSL à ce nouveau niveau. Ce procédé représente une fusion entre le concept de clustering itératif (de type agglomératif) et la dynamique DSL.

6.2. Principes de l'Apprentissage Multi-Échelle

Lorsqu'on souhaite doter le **DSL** d'une capacité à se déployer sur plusieurs échelles (micro, intermédiaire, macro), il est indispensable de saisir l'idée des **hiérarchies synergiques** et des niveaux d'abstraction qu'elles génèrent. La section 6.2.1 introduit cette notion en posant les bases d'une organisation multi-niveau. La section 6.2.2 l'illustre par la théorie des systèmes emboîtés, qui conceptualise l'organisation en couches successives, et la section 6.2.3 décrit comment ces niveaux interagissent par des flux ascendants (bottom-up) et descendants (top-down). L'objectif est de poser les fondations mathématiques permettant à un **SCN** (Synergistic Connection Network) de fonctionner harmonieusement à diverses granularités, du micro-cluster au macro-cluster, de manière coordonnée.

6.2.1. Hiérarchies Synergiques et Niveaux d'Abstraction

Dès qu'on dépasse un certain degré de complexité (grand nombre d'entités, de liens $\omega_{i,j}$), on constate souvent que les entités s'**auto-organisent** en **clusters** de tailles diverses, de petits regroupements locaux et, potentiellement, de grands ensembles plus étendus. Cette diversité de "tailles de clusters" correspond à la **pluralité** des échelles dans le réseau. On parle de **hiérarchies** quand ces clusters de niveau inférieur peuvent se regrouper (ou s'agréger) pour former des "super-clusters" plus vastes, suggérant différents **niveaux d'abstraction** — on peut alors manipuler des entités "brutes" (micro-niveau) ou des entités "agréées" (macro-niveau).

6.2.1.1. Notion de micro-cluster vs. macro-cluster

Dans de nombreux contextes d'**auto-organisation** via un **SCN**, il se forme simultanément des **groupes** à différentes **tailles** et différents **degrés de spécialisation**. Il est alors opportun d'introduire la distinction entre **micro-clusters** et **macro-clusters**. Les premiers correspondent à des **noyaux** restreints d'entités intimement reliées, tandis que les seconds, plus vastes, englobent parfois plusieurs micro-clusters et présentent une cohésion au niveau global.

Cette section (6.2.1.1) précise ces deux concepts, en montrant comment ils s'articulent dans le cadre d'un **SCN** et comment leur co-existence fonde l'idée même de **multi-échelle**.

Définition de micro-cluster et de macro-cluster

Un **micro-cluster** est un groupe de taille modeste, par exemple $\{\mathcal{E}_{i_1}, \dots, \mathcal{E}_{i_k}\}$, dans lequel les pondérations ω_{i_p, i_q} sont **fortement** élevées pour chaque couple (i_p, i_q) appartenant au groupe. On y retrouve généralement une **densité** ou une **somme** de liaisons particulièrement importante par rapport au nombre k d'entités. Ainsi, un micro-cluster a souvent vocation à remplir une **tâche locale** — quelques robots peuvent partager une mission spécialisée, certains concepts en sémantique forment un champ lexical soudé, ou encore quelques neurones, dans un contexte neuroscientifique, constituent un module fortement interconnecté. La force d'un tel cluster se mesure par

$$\Omega(\mathcal{C}) = \sum_{i,j \in \mathcal{C}} \omega_{i,j},$$

ce qui montre que $\Omega(\mathcal{C})$ peut être élevé malgré la taille réduite du groupe \mathcal{C} .

Un **macro-cluster**, en revanche, désigne un sous-ensemble du SCN plus large, pouvant inclure plusieurs micro-clusters en son sein. Les entités y sont globalement **cohésives** – au sens d’une synergie moyenne non négligeable – même si toutes n’entretiennent pas nécessairement des liens très forts entre elles. Au niveau d’un macro-cluster, on observe plutôt une forme de **convergence** globale qui rassemble un ensemble conséquent d’entités, par exemple en vue d’un objectif commun, d’une thématique plus générale ou d’une mission d’envergure. La mesure $\Omega(\mathcal{C})$ pour un macro-cluster peut également être importante, mais dans ce cas, la taille $|\mathcal{C}|$ est grande et la densité moyenne des liaisons est plus modérée, tout en demeurant suffisamment élevée pour distinguer le macro-cluster du reste du réseau.

Exemples et Co-existence

Dans un SCN comportant un grand nombre nn d’entités, la coexistence de micro-clusters et de macro-clusters est fréquente. On observe que le système présente plusieurs micro-clusters, c’est-à-dire des groupes de quelques entités très cohésives, ainsi qu’un ou plusieurs macro-clusters qui, à une échelle plus vaste, rassemblent certaines entités ou micro-clusters proches, formant ainsi un palier d’organisation supérieur.

Cette dualité émerge naturellement de la dynamique DSL, où un renforcement rapide et local conduit à la formation de petits noyaux soudés, tandis qu’une évolution à plus long terme ou à une échelle plus large permet l’agrégation de plusieurs micro-groupes en macro-clusters, reflétant ainsi la réalité observée dans de nombreux systèmes biologiques, cognitifs ou sociotechniques dans lesquels des structures de tailles différentes coexistent et se superposent.

Vers une Structure Hiérarchique

La présence simultanée de micro- et macro-clusters évoque naturellement une **hiérarchie** d’échelles. On peut, par exemple, poser :

- Le **niveau 0** (ou échelle micro) : chaque entité \mathcal{E}_i en détail, permettant de repérer des groupes restreints,
- Le **niveau 1** (ou échelle méso) : on regroupe certains micro-clusters pour former des sous-réseaux plus grands,
- Le **niveau 2** (ou échelle macro) : on identifie un ou plusieurs macro-clusters dominants, englobant éventuellement plusieurs niveaux inférieurs.

Chacun de ces niveaux peut s’analyser, dans le **SCN**, par une matrice de pondérations $\omega^{(k)}$ spécifique (voir Section 6.2.2), reflétant des *super-nœuds* (macro-entités) si l’on agrège les micro. L’**apprentissage** multi-échelle (chap. 6.4–6.5) s’appuiera alors sur des **règles** d’agrégation, de rétroaction (top-down, bottom-up) et de synchronisation inter-niveau pour maintenir la cohérence globale.

6.2.1.2. Rôles et Fonctions des Différents Niveaux (local, intermédiaire, global)

Dans l'étude des SCN appliqués au DSL, il apparaît essentiel de distinguer plusieurs échelles d'organisation qui permettent de moduler la complexité du système de manière hiérarchique. La présente section expose la finalité et le rôle de trois niveaux d'abstraction distincts, à savoir le **niveau local** (ou micro-niveau), le **niveau intermédiaire** (ou méso-niveau) et le **niveau global** (ou macro-niveau). Chacun de ces niveaux possède une fonction propre dans la dynamique d'auto-organisation du réseau, comme nous le développerons ci-après.

A. Niveau Local (Micro-Niveau)

Le niveau local correspond à la granularité la plus fine du SCN. Considérons un ensemble d'entités $\mathcal{E} = \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ et la matrice de pondérations locale $\omega^{(0)}(t) = (\omega_{i,j}^{(0)}(t))$ qui décrit les interactions directes entre ces entités au temps t . La dynamique d'auto-organisation au niveau local est régie par l'équation

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_{\text{loc}} \left(S(i,j) - \tau_{\text{loc}} \omega_{i,j}^{(0)}(t) \right),$$

où η_{loc} représente le taux d'apprentissage local et τ_{loc} un paramètre d'inhibition. La fonction $S(i,j)$ exprime la synergie entre les entités \mathcal{E}_i et \mathcal{E}_j .

Si $S(i,j)$ est supposé constant, l'équation admet un point d'équilibre donné par

$$\omega_{i,j}^{(0)*} = \frac{S(i,j)}{\tau_{\text{loc}}}.$$

La stabilité de cet équilibre peut être analysée en posant $\delta\omega_{i,j}(t) = \omega_{i,j}^{(0)}(t) - \omega_{i,j}^{(0)*}$, ce qui conduit à la relation linéarisée

$$\delta\omega_{i,j}(t+1) \approx (1 - \eta_{\text{loc}} \tau_{\text{loc}}) \delta\omega_{i,j}(t).$$

Si $0 < \eta_{\text{loc}} \tau_{\text{loc}} < 2$, l'équilibre est stable et les pondérations convergent vers $\omega_{i,j}^{(0)*}$.

La formation de micro-clusters se traduit par une homogénéisation locale des pondérations. On définit la synergie locale d'un cluster \mathcal{C} comme

$$\Omega(\mathcal{C}) = \sum_{i,j \in \mathcal{C}} \omega_{i,j}^{(0)},$$

ce qui permet de mesurer la densité des interactions au sein de \mathcal{C} . Un micro-cluster est identifié lorsque $\Omega(\mathcal{C})$ dépasse un seuil déterminé θ_{loc} . L'analyse spectrale de la matrice $\omega^{(0)}$ peut également être utilisée pour détecter ces structures en examinant les valeurs propres qui reflètent la cohésion interne.

Ces propriétés mathématiques illustrent comment, au niveau local, la dynamique DSL permet une détection rapide et une spécialisation fine grâce à des paramètres optimisés. Les micro-clusters

ainsi formés, caractérisés par une forte cohésion (grâce à une synergie locale élevée), constituent la base sur laquelle se construiront les niveaux supérieurs dans la hiérarchie du SCN.

B. Niveau Intermédiaire (Méso-Niveau)

Le niveau intermédiaire intervient en regroupant les micro-clusters issus du niveau local pour former des super-nœuds, généralement notés $\{\mathcal{M}_\alpha\}$. À ce palier, la dynamique d'agrégation vise à condenser les interactions locales, rendant ainsi plus lisibles et gérables les relations complexes qui se tissent à l'échelle du réseau. La nouvelle matrice des pondérations, notée $\omega^{(\text{inter})}$, codifie la synergie entre ces super-nœuds. Cette opération s'exprime par

$$\omega_{\alpha,\beta}^{(\text{inter})} = \Psi \left(\{\omega_{i,j}^{(\text{local})}\}_{i \in \mathcal{M}_\alpha, j \in \mathcal{M}_\beta} \right),$$

où Ψ est une fonction d'agrégation adaptée aux données locales.

À ce niveau, le rôle principal est à la fois d'interface et de coordination. Le niveau intermédiaire informe le niveau global de l'état des sous-ensembles tout en transmettant, par rétroaction descendante, des consignes susceptibles d'ajuster la dynamique locale. La mise à jour des pondérations intermédiaires s'inscrit dans la même logique que celle appliquée au niveau local et se formalise par

$$\omega_{\alpha,\beta}^{(\text{inter})}(t+1) = \omega_{\alpha,\beta}^{(\text{inter})}(t) + \eta_{\text{inter}} \left[S_{\text{inter}}(\alpha, \beta) - \tau_{\text{inter}} \omega_{\alpha,\beta}^{(\text{inter})}(t) \right].$$

Cette loi assure une homogénéité dans l'approche d'auto-organisation sur plusieurs échelles, garantissant que la dynamique d'agrégation et de mise à jour reste cohérente tout au long de la hiérarchie du SCN.

C. Niveau Global (Macro-Niveau)

Au sommet de la hiérarchie se situe le niveau global, qui intègre l'ensemble des super-nœuds afin de former une vision unifiée du réseau. Ce niveau est représenté par des macro-clusters, notés $\{\mathcal{M}_\alpha^{(\text{global})}\}$, et il a pour objectif de dégager la structure ultime du SCN en fournissant une représentation condensée de la configuration globale du système.

La matrice des pondérations globales, notée $\omega^{(\text{global})}$, est mise à jour selon une dynamique analogue à celle des niveaux inférieurs. Par exemple, la loi de mise à jour s'exprime par

$$\omega_{\alpha,\beta}^{(\text{global})}(t+1) = \omega_{\alpha,\beta}^{(\text{global})}(t) + \eta_{\text{glob}} \left[S_{\text{glob}}(\alpha, \beta) - \tau_{\text{glob}} \omega_{\alpha,\beta}^{(\text{global})}(t) \right],$$

où $S_{\text{glob}}(\alpha, \beta)$ représente la synergie évaluée entre de grands ensembles, et η_{glob} ainsi que τ_{glob} sont des paramètres ajustés spécifiquement à cette échelle.

Au niveau global, l'objectif est double. D'une part, il sert à fournir une vision d'ensemble qui stabilise l'organisation du réseau en consolidant les interactions entre les macro-nœuds. D'autre part, il peut jouer le rôle d'interface avec des systèmes externes, par exemple en connectant le SCN à des modules de visualisation ou de prise de décision. De plus, en verrouillant la convergence de la configuration globale lorsque la synergie atteint un certain seuil, le niveau global participe

activement à la stabilisation du SCN. Ce mécanisme de stabilisation repose sur l'hypothèse que, lorsque $\omega_{\alpha,\beta}^{(\text{global})}(t)$ converge vers une valeur limite

$$\omega_{\alpha,\beta}^{(\text{global})*} = \frac{S_{\text{glob}}(\alpha, \beta)}{\tau_{\text{glob}}},$$

le réseau atteint une configuration stable qui se reflète dans la cohérence de l'ensemble des interactions macro-échelonnées.

D. Complémentarité et Couplage des Niveaux

La force d'un **SCN multi-échelle** réside dans l'interaction étroite entre les différents niveaux du système. Les rétroactions bottom-up permettent aux micro-clusters de s'agréger en super-nœuds, tandis que les rétroactions top-down diffusent des contraintes depuis le niveau global vers le niveau local afin d'orienter la dynamique. Cette couverture hiérarchique réduit la complexité du problème en divisant le réseau en sous-ensembles gérables, accélère la convergence en isolant les interactions locales des influences globales excessives et favorise l'émergence d'une auto-similarité fractale, puisque la même règle DSL se répète sur plusieurs niveaux d'abstraction.

D'un point de vue mathématique, cette approche hiérarchique permet de passer d'un problème dont la complexité est de l'ordre de $\mathcal{O}(n^2)$ à des sous-problèmes de dimension réduite. Pour chaque niveau, la dynamique de mise à jour peut être traitée indépendamment, puis intégrée dans une fonction d'agrégation, par exemple via la fonction Ψ , qui combine les interactions locales en une représentation globale. Ce découpage facilite l'analyse de la stabilité et la conception de régulateurs adaptés à chaque échelle, en garantissant que les interactions locales, une fois traitées et agrégées, contribuent à une dynamique cohérente et stable à l'échelle globale.

6.2.1.3. Exemples : micro-réseaux d'entités sensorielles vs. macro-ensembles pour une vue conceptuelle

Les rôles des différents niveaux (micro, intermédiaire, macro) dans un SCN s'appréhendent de manière plus concrète lorsqu'on considère des cas exemplaires de la **coexistence** de micro-réseaux d'entités très locales (capteurs, données brutes, etc.) et de macro-ensembles plus englobants (concepts, missions, sous-systèmes de grande dimension). La présente section illustre comment un **SCN** peut à la fois gérer des **micro-clusters** — typiquement quelques entités fortement reliées — et des **macro-clusters** représentant un agrégat plus large et plus abstrait. Deux exemples sont développés : (A) la formation de micro-réseaux sensoriels, (B) la formation de macro-ensembles conceptuels.

A. Micro-réseaux d'entités sensorielles

Un scénario fréquemment rencontré est celui d'un grand nombre de **capteurs** ou **petites sources** d'information (capteurs IoT, neurones sensoriels, etc.). Chaque capteur \mathcal{E}_i constitue une entité locale avec laquelle on peut former un micro-réseau. Dans un **SCN**, ces capteurs établissent des liens $\omega_{i,j}$ qui se renforcent s'ils détectent une similarité ou corrélation forte. On parle alors de

micro-cluster dès lors qu'un petit ensemble de capteurs $\{\mathcal{E}_{i_1}, \dots, \mathcal{E}_{i_k}\}$ se découvre une synergie ω_{i_p, i_q} élevée.

Soit $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S_{\text{corr}}(i,j) - \tau \omega_{i,j}(t)]$, où $S_{\text{corr}}(i,j)$ exprime le degré de **corrélation** (ou autre similarité) entre les signaux sensoriels de i et j . Les micro-clusters se repèrent lorsqu'un groupe \mathcal{C} présente une somme interne

$$\Omega(\mathcal{C}) = \sum_{i,j \in \mathcal{C}} \omega_{i,j}$$

assez forte pour dépasser un seuil θ . Ces clusters, de taille modeste, reflètent des **interactions fines** et très **spécifiques** (par ex. capteurs proches géographiquement ou capteurs partageant la même tendance de mesure).

Au niveau **micro**, les entités réagissent de manière rapide et locale. Un capteur peut instantanément renforcer ses liens avec un voisin détectant le même événement, entraînant une **dynamique DSL** très réactive. Une fois ces micro-clusters stabilisés, ils peuvent être **agrégés** (Section 6.2.2) pour former un **super-nœud** au niveau méso. Cette agrégation allège la complexité du SCN en considérant chaque petit groupe cohésif comme une **brique** structurante pour les niveaux supérieurs.

B. Macro-ensembles pour une vue conceptuelle

À l'autre extrême, on observe parfois dans un SCN la formation de **macro-ensembles** englobant un large ensemble d'entités pour représenter un **concept** global ou une **mission** d'envergure. Dans un système cognitif, cela pourrait correspondre à un *champ sémantique* ; dans un système multi-robot, à un *groupe de grande taille* coopérant à un objectif.

Sur le plan **mathématique**, un macro-ensemble \mathcal{G} peut rassembler plusieurs dizaines, voire centaines d'entités. La force globale $\Omega_{\mathcal{G}} = \sum_{i,j \in \mathcal{G}} \omega_{i,j}$ n'a pas besoin d'être extrêmement dense (comme dans un micro-cluster) mais reste **significative** pour qualifier un **macro-cluster**. Ce super-groupe agit souvent comme un **super-nœud** dans un niveau macro, en lien avec d'autres macro-ensembles, formalisé par $\omega_{\Gamma, \Delta}^{(\text{global})}$.

Exemples

1. Dans une **architecture neuronale inspirée du cerveau**, un “macro-ensemble” peut symboliser une grande aire associant des fonctions cognitives plus abstraites.
2. Dans un **réseau sémantique**, un macro-cluster incarne un **concept** (ex. “géométrie”, “mécanique quantique”) regroupant des micro-clusters de définitions, de synonymes, etc.
3. Dans la **robotique** multi-agent, un **macro-ensemble** représente une **équipe étendue**, où plusieurs **sous-équipages** plus restreints coopèrent en vue d'atteindre un **objectif commun**.

C. Articulation : du micro au macro

Le **niveau micro** (petits clusters sensoriels) se trouve à la base. Le **niveau macro** (grands ensembles conceptuels) se trouve au sommet. Entre les deux, un **méso-niveau** (chap. 6.2.2) peut

regrouper des sous-groupes intermédiaires. Chaque palier propose une **granularité** distincte mais reliée :

- Le micro-niveau, réactif et spécialisé,
- Le méso-niveau, coordinateur ou “articulateur”,
- Le macro-niveau, plus abstrait, offrant une synthèse plus large ou une “vision d’ensemble”.

Les **micro-réseaux sensoriels** produisent des signaux qui s’agrègent en **macro-ensembles conceptuels**, suivant un **flux ascendant** (*bottom-up*). En retour, le niveau macro peut imposer des **contraintes** ou **orientations descendantes** (*top-down*). Par exemple, un **macro-objectif** comme “réduire la consommation d’énergie” influence le niveau micro en ajustant les pondérations $\omega_{i,j}$ pour privilégier certains capteurs plus pertinents à cet objectif.

6.2.2. Théorie des Systèmes Emboîtés

Lorsque l’on manipule des **SCN** (Synergistic Connection Networks) appelés à se structurer en **niveaux** (micro, intermédiaire, macro), il est souvent utile d’invoquer la **théorie des systèmes emboîtés**. Cette théorie, inspirée de modèles biologiques ou complexes, décrit comment des **entités de base** (cellules, micro-clusters) se regroupent en « tissus » (super-nœuds de niveau intermédiaire), puis en « organes » (macro-nœuds plus vastes). La notion d’**emboîtement** formalise la hiérarchie progressive aboutissant, in fine, à un **organisme** global (un niveau macro englobant la plupart ou la totalité des entités).

6.2.2.1. Systèmes emboîtés : cellule → tissu → organe → organisme (analogie)

L’idée de **multi-échelle** et d’**emboîtement** dans un **SCN** (Synergistic Connection Network) trouve un écho naturel dans la **biologie**, où l’on décrit une hiérarchie Cellule → Tissu → Organe → Organisme. Chaque palier agrège des entités de taille ou de complexité plus faible pour former un ensemble supérieur doté de nouvelles fonctions. Cette analogie éclaire comment, dans un **SCN** multi-échelle, on peut progressivement construire, de bas en haut, des **niveaux** de plus en plus **macro** tout en maintenant un lien avec les **interactions** plus fines du niveau inférieur.

1. Analogie Biologique

Il est courant, en biologie, de formaliser l’échelle **cellule** comme l’unité de base, dont la coordination et l’agrégation forment un **tissu** spécialisé (musculaire, nerveux, épithélial). Plusieurs tissus interdépendants constituent alors un **organe**, et la conjonction d’organes distincts mais coopérants donne naissance à un **organisme** entier. Sur le plan **hiérarchique**, on peut poser :

Cellule → Tissu → Organe → Organisme.

Chacune de ces étapes repose sur l’**agrégation** ou l’**encapsulation** des entités de l’étape précédente, tout en donnant naissance à de **nouvelles** propriétés émergentes comme la régulation hormonale ou la vascularisation. Cet enchaînement illustre un **système emboîté**, où chaque niveau

structurel contient et organise les niveaux inférieurs, un tissu regroupant des cellules, un organe intégrant plusieurs tissus, et ainsi de suite.

2. Formalisation en Systèmes Emboîtés

D'un point de vue **mathématique**, on peut représenter la hiérarchie cellulaire–tissulaire–organique par des **sous-ensembles** successifs $\{\mathcal{E}_i^{(k)}\}$ où k identifie le niveau. Par exemple :

- Niveau $k = 0$: $\mathcal{E}_i^{(0)}$ correspond aux **cellules** (entités de base).
- Niveau $k = 1$: on agrège ces cellules en **tissus** $\mathcal{T}_\alpha^{(1)}$.
- Niveau $k = 2$: on réunit plusieurs tissus en **organes** $\mathcal{O}_\beta^{(2)}$.
- Niveau $k = 3$: la **somme** des organes interdépendants constitue un **organisme** $\mathcal{O}^{(3)}$.

Chaque niveau s'**emboîte** dans le suivant, les éléments de niveau k se regroupant pour constituer les entités du niveau $k + 1$. Sur le plan formel, on peut définir une **application** Φ_k :

$$\Phi_k: \{\mathcal{E}_i^{(k)}\} \rightarrow \{\mathcal{E}_\alpha^{(k+1)}\},$$

qui regroupe (ou *coarse-grain*) les entités $\mathcal{E}_i^{(k)}$ en super-nœuds $\mathcal{E}_\alpha^{(k+1)}$. Cela **reproduit** la logique de passage de la cellule au tissu, puis du tissu à l'organe, etc.

3. Maintien des Interactions

Dans la biologie, on sait que les **tissus** n'annulent pas les interactions **locales** entre cellules ; elles se reflètent sous la forme de **communications** (signaux chimiques, électriques, etc.) à un **niveau supérieur**. Pareillement, un organe dépend de la cohérence entre ses tissus, et l'organisme de la synergie entre ses organes.

Sur le plan **mathématique** et plus spécifiquement dans un **SCN** multi-niveau, cette interdépendance se traduit :

- Par l'existence de **fonctions d'agrégation** (Section 6.2.2.2) qui synthétisent les liaisons ω d'un niveau vers le suivant.
- Par des **rétroactions** top-down (Section 6.2.3.1) dans lesquelles l'état macro peut influencer les pondérations micro.

La structure hiérarchique ainsi formée n'annule pas la réalité des interactions locales, mais les reflète dans des *super-pondérations* $\omega^{(k)}$ entre macro-nœuds, tout en laissant la possibilité que le niveau inférieur conserve sa propre dynamique DSL.

Exemple : cellule \rightarrow tissu \rightarrow organe \rightarrow organisme

Prenons un organisme simple, par exemple un invertébré. Les **cellules** (niveau 0) se spécialisent et se coordonnent pour bâtir un **tissu** musculaire (niveau 1). À ce stade, Φ_0 associe un ensemble de cellules “musculaires” à un super-nœud “tissu musculaire”. Parallèlement, d'autres cellules deviennent un “tissu nerveux”. Au niveau 2, on réunit ces tissus en un organe (muscle + nerf +

vaisseaux) via l'application Φ_1 . Et au sommet, l'**organisme** complet (niveau 3) résulte de la **cohésion** entre organes.

Formellement, si on note Γ_k la fonction de regroupement de niveau k à $k + 1$, on a :

$$\Gamma_0(\{\text{cellules}\}) = \{\text{tissus}\}, \quad \Gamma_1(\{\text{tissus}\}) = \{\text{organes}\}, \quad \Gamma_2(\{\text{organes}\}) = \{\text{organisme}\}.$$

C'est cette **progression hiérarchique** que l'on cherche à transposer dans le **DSL** multi-échelle, en passant d'entités fines (micro-clusters) à des super-nœuds (méso) puis à des macro-entités globales (macro-niveau), tout en préservant la mémoire des interactions locales et intermédiaires.

6.2.2.2. Transposition au DSL : agrégation progressive d'entités vers des super-nœuds (macro-nœuds)

Le passage de **niveaux** inférieurs (micro) à des **super-nœuds** plus abstraits (macro-nœuds) est un élément clef lorsqu'on applique l'idée de **systèmes emboîtés** (6.2.2.1) dans un **SCN** (Synergistic Connection Network) reposant sur le **DSL** (Deep Synergy Learning). Il s'agit de formaliser, dans un cadre mathématique et algorithmique, la manière dont un *ensemble* d'entités localement cohésives (k -ième niveau) se **fusionnent** pour former des **super-nœuds** au niveau $k + 1$. Cette section décrit les **fondements** de ce mécanisme d'**agrégation progressive**, en expliquant comment on repère les clusters, comment on "condense" leurs pondérations, et de quelle manière la **dynamique DSL** peut se poursuivre à un niveau supérieur.

A. Notation Générale : Agrégation de Clusters

1. Ensemble initial (niveau 0)

On part d'un **niveau 0** composé d'entités "microscopiques" $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$, qui interagissent via une matrice $\omega^{(0)}$. La mise à jour se fait suivant la **règle DSL** :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_0 [S_0(i,j) - \tau_0 \omega_{i,j}^{(0)}(t)].$$

Chaque $\omega_{i,j}^{(0)}$ peut s'interpréter comme la synergie (ou le degré de corrélation, ou de compatibilité) entre deux entités **basiques** du réseau.

2. Détection ou formation de clusters

Lorsque la **dynamique DSL** a suffisamment évolué (ou de manière continue), on identifie un **ensemble** de *micro-clusters* $\{\mathcal{C}_1^{(0)}, \dots, \mathcal{C}_r^{(0)}\} \subseteq \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. En pratique, on repère ces clusters à l'aide :

- D'un **seuil** sur $\omega_{i,j}^{(0)}$: si $\omega_{i,j} \geq \theta$, on les considère comme reliés.
- D'un **algorithme** de détection de communautés (modulaire, composantes fortement connectées).
- D'une **analyse** topologique (loi des distances, etc.) ou d'une **fonction** $\Omega(\mathcal{C}) = \sum_{i,j \in \mathcal{C}} \omega_{i,j}$.

Dès lors, chaque micro-cluster $\mathcal{C}_\alpha^{(0)}$ regroupe un certain nombre d'entités **très** cohésives. Ce sont ces regroupements que l'on va "**remplacer**" par des super-nœuds pour construire le niveau 1.

3. Agrégation en super-nœuds

Concrètement, on "condense" chaque micro-cluster $\mathcal{C}_\alpha^{(0)}$ en un **super-nœud** $\mathcal{N}_\alpha^{(1)}$. Ainsi, le *niveau 1* du SCN se compose :

$$\{\mathcal{N}_1^{(1)}, \dots, \mathcal{N}_r^{(1)}\}.$$

On parle alors d'une application de **coarse-graining** ou de "macro-agrégation" Γ_0 , telle que :

$$\Gamma_0: \{\mathcal{C}_\alpha^{(0)}\} \rightarrow \{\mathcal{N}_\alpha^{(1)}\}.$$

Les *indices* α, β indiquent que chaque super-nœud du niveau 1 regroupe un *ensemble* de micro-entités du niveau 0. La question essentielle est alors de déterminer **comment** définir les pondérations $\omega_{\alpha, \beta}^{(1)}$ entre les super-nœuds $\mathcal{N}_\alpha^{(1)}$ et $\mathcal{N}_\beta^{(1)}$.

B. Définition des Pondérations entre Super-nœuds

1. Fonction d'agrégation Ψ

Pour **transposer** les pondérations $\omega_{i,j}^{(0)}$ au niveau 1, on se sert d'une **fonction** Ψ . Par exemple, une **moyenne** :

$$\omega_{\alpha, \beta}^{(1)} = \frac{1}{|\mathcal{C}_\alpha^{(0)}| \cdot |\mathcal{C}_\beta^{(0)}|} \sum_{i \in \mathcal{C}_\alpha^{(0)}, j \in \mathcal{C}_\beta^{(0)}} \omega_{i,j}^{(0)}.$$

Ou une **somme** ou un **maximum**. L'idée est de condenser les multiples liaisons $\omega_{i,j}^{(0)}$ internes à α (ou entre α et β) en une unique $\omega_{\alpha, \beta}^{(1)}$. Ce nouveau "réseau" (niveau 1) présente alors beaucoup moins de nœuds que le réseau de base.

2. Règle DSL au niveau 1

Une fois $\omega_{\alpha, \beta}^{(1)}$ initialisées par Ψ , on peut **appliquer** à ce niveau la même **dynamique** DSL (ou une version paramétrée différemment) :

$$\omega_{\alpha, \beta}^{(1)}(t+1) = \omega_{\alpha, \beta}^{(1)}(t) + \eta_1 \left[S_1(\alpha, \beta) - \tau_1 \omega_{\alpha, \beta}^{(1)}(t) \right],$$

afin que les super-nœuds α, β interagissent *entre eux* selon une logique similaire. De cette manière, un "**cluster de super-nœuds**" peut encore se former au niveau 1, ce qui **prépare** l'étape suivante (passage à un niveau 2).

C. Répétition jusqu'au Macro-Niveau

Si le **niveau 1** engendre lui aussi des clusters $\mathcal{C}_\gamma^{(1)} \subseteq \{\mathcal{N}_\alpha^{(1)}\}$, on **répète** le processus en agrégeant ces sous-groupes en **macro-nœuds** $\mathcal{N}_\gamma^{(2)}$. Les pondérations $\omega_{\gamma,\delta}^{(2)}$ sont définies en “relisant” $\omega_{\alpha,\beta}^{(1)}$ à l’aide d’une fonction d’agrégation Ψ . On obtient ainsi :

$$\Gamma_1: \{\mathcal{C}_\gamma^{(1)}\} \rightarrow \{\mathcal{N}_\gamma^{(2)}\}.$$

Ce mécanisme se **poursuit** jusqu’à un certain **niveau K**, où l’on considère avoir atteint un **macro-nœud** quasi final (ou un petit nombre de macro-nœuds dominants). Chaque itération réduit la granularité et crée une **hiérarchie**.

Si, à chaque niveau, on applique la même forme de la **règle DSL** et la même fonction Ψ , on observe une **auto-similarité**. La formation des clusters à un niveau se réplique aux niveaux supérieurs. Cette propriété suggère une **fractalité** où apparaissent des lois de puissance et des motifs récurrents. L’analyse de cette structure multiniveau dans la théorie du DSL devient alors essentielle.

L’approche multiniveau optimise la gestion des interactions dans un **SCN** en structurant les entités à différentes échelles. Ses principaux avantages sont :

- **Réduction de complexité** en remplaçant un ensemble massif de micro-pondérations $\omega_{i,j}^{(0)}$ par un ensemble plus restreint $\omega_{\alpha,\beta}^{(1)}$, facilitant ainsi le traitement des liens.
- **Modularité** où chaque niveau gère ses propres pondérations sans interférer avec le niveau inférieur, garantissant une organisation plus stable et structurée.
- **Adaptabilité à grande échelle** permettant d’étendre la hiérarchie sur plusieurs niveaux si le nombre n de micro-entités devient trop important, assurant ainsi une structuration efficace et évolutive du réseau.

Processus d’Agrégation

Typiquement, un **algorithme** multi-échelle fonctionnant par itérations “agrégation–mise à jour” :

1. **Clusterisation** (niveau k) : On repère les sous-ensembles $\mathcal{C}_\alpha^{(k)}$.
2. **Agrégation** : On crée des super-nœuds $\mathcal{N}_\alpha^{(k+1)}$ et initialise $\omega_{\alpha,\beta}^{(k+1)}$ via Ψ .
3. **Mise à jour** : On applique la règle DSL au niveau $k + 1$, construisant la structure $\omega^{(k+1)}$.
4. **Répétition** : Tant que $\omega^{(k+1)}$ suscite de nouvelles agrégations, on recommence (sinon on s’arrête).

Pseudo-code

```
function buildHierarchy(level=0, listOfEntities E^(0)):  
  # 1) Repérer clusters micro
```

```

clusters = detectClusters(E^(0), w^(0))
# 2) Construire super-nœuds
superNodes = []
for each cluster C in clusters:
    superNode M = new SuperNode(C)
    superNodes.add(M)

# 3) Agréger pondérations
w^(level+1) = computeWeights(superNodes, w^(level)) # via Psi

# 4) Mise à jour DSL
applyDSL(w^(level+1))

# 5) S'il reste de la structure interne
if new merges can appear:
    buildHierarchy(level+1, superNodes)
else:
    return superNodes

```

Cet algorithme illustre un schéma **ascendant** (bottom-up). Des variantes peuvent incorporer des **feedback** top-down (section 6.2.3).

6.2.2.3. Multi-niveau comme “pilier” pour la gestion de la complexité

Lorsque la taille n d’un **SCN** (Synergistic Connection Network) devient très grande (dizaines, centaines de milliers ou plus d’entités), on se retrouve face à une croissance potentielle de la complexité en $O(n^2)$ si l’on s’en tient à un *niveau unique*. Une telle situation rend difficile la mise à jour de la matrice ω et la détection de clusters significatifs. Le **multi-niveau** (ou multi-échelle), décrit dans les sections précédentes, se révèle alors un **pilier** incontournable pour contenir cette complexité et l’organiser de manière **hiérarchique**.

A. Réduction de la complexité via l’agrégation

Dans un **SCN** de niveau “zéro” (micro) avec n entités, la matrice $\omega^{(0)}$ peut compter jusqu’à $O(n^2)$ pondérations (chaque couple (i, j) peut a priori disposer d’un lien $\omega_{i,j}$). Chaque itération de la règle DSL pourrait, dans le pire cas, nécessiter $O(n^2)$ mises à jour si l’on ne pratique ni **parsimonie** (seuil, kNN, etc.) ni **multi-niveau**.

Lorsque l’on introduit un **niveau supérieur** (voir section 6.2.2.2) qui regroupe les entités en **super-nœuds**, la taille de la matrice $\omega^{(k+1)}$ peut chuter de $O(n^2)$ à $O(m^2)$ pour $m \ll n$. Cela signifie que la charge de gestion des pondérations au niveau $k + 1$ est désormais bien plus réduite. Dans un système hiérarchique allant jusqu’à un macro-niveau, on obtient plusieurs matrices $\omega^{(k)}$ de tailles décroissantes n, n_1, n_2, \dots

Si un système compte un **niveau 0** à n entités, on détecte des **clusters** et on regroupe ces entités en n_1 super-nœuds pour le **niveau 1**. Puis on applique le même principe pour passer de n_1 à n_2 au **niveau 2**, etc., produisant une suite $n_0 = n > n_1 > n_2 > \dots > n_K$. Souvent, $n_K \approx 1$ ou un petit nombre, correspondant à un nombre réduit de macro-nœuds.

Cette **hiérarchie** rend possible un **traitement** plus local au niveau inférieur, alors que le niveau supérieur opère sur moins de super-nœuds, entraînant un coût algorithmique plus faible ($O(n_k^2)$ au niveau k).

Supposons $n_0 = 10^5$ au niveau micro. On pourrait aisément se retrouver avec $O(10^{10})$ liens théoriques. Si on agrège en $n_1 = 10^3$ super-nœuds (niveau 1), on ne gère plus que $O((10^3)^2) = 10^6$ pondérations à ce niveau. En passant à un niveau 2 avec $n_2 \approx 10^2$, la complexité tombe encore à $O((10^2)^2) = 10^4$. Chaque **niveau** manipule donc une structure plus “légère”, et l’ensemble forme un **organigramme** multi-niveau de l’auto-organisation.

B. Organisation hiérarchique et contrôle local vs. global

Le **niveau micro** (entités de base, éventuellement groupées en petits micro-clusters) s’occupe des **réactions** les plus fines ou rapides, avec un taux d’apprentissage η_{loc} potentiellement plus élevé. Cela s’explique par la nature **proche** et **réactive** des interactions locales, où deux entités très similaires renforcent rapidement leurs liens sans affecter l’ensemble du réseau.

L’avantage de cette adaptation réside dans son impact **localisé** : seules les entités directement concernées subissent des modifications en temps réel, tandis que le reste du SCN demeure relativement stable. Ce mécanisme facilite un *contrôle local*, qui est non seulement plus efficace mais aussi plus adapté à une **exécution parallèle** ou à une **distribution** sur plusieurs nœuds.

Au niveau **macro**, ou global, la **structure** du SCN se décrit en termes de super-nœuds plus vastes, ce qui permet de piloter de *grandes portions* de l’ensemble sans avoir à gérer chaque entité individuellement.

Cette approche offre un **contrôle plus large** en limitant le nombre d’interactions à $O(m^2)$ liens si l’on ne considère que m macro-nœuds. Bien que cette vision soit moins détaillée, elle simplifie la gestion des dynamiques globales du SCN et favorise une **régulation efficace** à grande échelle.

Ce macro-niveau peut fixer des **contraintes** ou envoyer des **feedback** descendant (top-down) pour orienter la dynamique micro, par exemple imposer des seuils minimaux de synergie pour que des entités participent à un macro-objectif.

Cette structuration (micro, méso, macro) évite qu’un **seul** algorithme doive gérer simultanément des centaines de milliers (ou millions) de liens. On crée plutôt **plusieurs** paliers, chacun manipulant un **réseau** de taille moindre, tout en assurant la **cohérence** grâce à la **rétroaction** (section 6.2.3). Sur le plan mathématique, on se dote d’une suite de matrices $\{\omega^{(0)}, \omega^{(1)}, \dots\}$ reliant des entités de *plus en plus agrégées*, et on applique la mise à jour DSL (additive, multiplicative, etc.) à chacun de ces niveaux.

C. Maintien de la cohérence multi-échelle

Le **couplage** multi-échelle se concrétise par des **fonctions d’agrégation** (pour passer du niveau k au niveau $k + 1$) et par des **rétroactions** top-down (le niveau macro imposant ou suggérant des ajustements au niveau micro). Chaque matrice $\omega^{(k)}$ conserve alors une **trace** de la structure d’interaction héritée du niveau inférieur, sous forme “coarse-grained”.

On aboutit à un **système** de poids $\omega^{(k)}(t)$ sur $O(n_k^2)$ liens, avec n_k (le nombre de super-nœuds au niveau k). Un opérateur Ψ_k définit $\omega^{(k+1)}$ à partir de $\omega^{(k)}$. Chaque $\omega^{(k)}$ suit sa propre dynamique DSL :

$$\omega_{\alpha,\beta}^{(k)}(t+1) = \omega_{\alpha,\beta}^{(k)}(t) + \eta_k [S_k(\alpha, \beta) - \tau_k \omega_{\alpha,\beta}^{(k)}(t)].$$

Le multi-niveau permet de restreindre η_k ou τ_k à des valeurs plus appropriées à la taille du niveau. On gère ainsi un **réseau** à $O(n_k^2)$ plutôt qu'à $O(n^2)$, tout en conservant une **vision** hiérarchique plus flexible.

En outre, un **événement** local (un micro-cluster s'effondre, un sabotage local, etc.) n'a pas à se propager instantanément au niveau macro, tant que l'agrégation au niveau supérieur n'y voit pas d'impact majeur. Il en résulte une **protection top-down** ou un **isolement partiel**, qui renforce la robustesse du SCN en évitant que le macro-niveau ne soit submergé par des fluctuations mineures.

En parallèle, la **modularité** simplifie la maintenance et l'évolution du réseau. Il devient possible de remplacer un bloc micro sans perturber l'ensemble de la structure macro, garantissant ainsi une meilleure **flexibilité** et une adaptation progressive du SCN.

6.2.3. *Rétroactions et Anticipations Multi-Niveau*

Dans une **organisation** hiérarchique (micro, intermédiaire, macro) — telle qu'exposée en (6.2.1) et (6.2.2) —, il est essentiel de comprendre **comment** ces niveaux interagissent. Les **flux ascendants** (bottom-up) font “remonter” l'information du micro-cluster vers des super-nœuds plus larges (macro-ensembles) ; les **flux descendants** (top-down) permettent au niveau supérieur d'exercer un **contrôle** (ou une **influence**) sur la dynamique locale. Cette section (6.2.3) en décrit les mécanismes, à commencer par le flux ascendant (6.2.3.1).

6.2.3.1. Flux ascendants (bottom-up) : micro \rightarrow macro

Dans un **SCN** (Synergistic Connection Network) multi-niveau, le **flux ascendant** (ou bottom-up) représente la façon dont la **structure** et la **dynamique** établies à un **niveau** local (micro) se répercutent vers les **niveaux supérieurs** (macro). Concrètement, c'est l'opération qui permet à des *micro-clusters* ou *super-nœuds* d'être “remontés” et agrégés pour former des macro-nœuds plus vastes. Cette section (6.2.3.1) détaille le **principe**, les **motifs** et les **bénéfices** de ce flux ascendant, en l'illustrant avec des exemples de robotique et de systèmes cognitifs.

A. Principe du flux ascendant

Le flux bottom-up suppose que des **groupes** d'entités (au niveau k) — par exemple un ensemble de micro-entités —, une fois **fortement cohésifs**, vont être **promus** ou **agrégés** en un **super-nœud** au niveau $k+1$. Cela incarne mathématiquement la **transition** :

$$\mathcal{C}_\alpha^{(k)} \xrightarrow{\text{agrég.}} \mathcal{N}_\alpha^{(k+1)},$$

où $\mathcal{C}_\alpha^{(k)}$ est un cluster de niveau k et $\mathcal{N}_\alpha^{(k+1)}$ le super-nœud correspondant au niveau $k+1$.

Dans la dynamique DSL, si $\mathcal{C}_\alpha^{(k)} \subset \{\mathcal{E}_i^{(k)}\}$ regroupe un ensemble d'entités ou de nœuds de niveau k , on définit les pondérations $\omega_{\alpha,\beta}^{(k+1)}$ par agrégation des pondérations $\omega_{i,j}^{(k)}$. Généralement, on emploie une fonction Ψ , telle que

$$\omega_{\alpha,\beta}^{(k+1)} = \Psi\left(\omega_{i,j}^{(k)} \mid i \in \mathcal{C}_\alpha^{(k)}, j \in \mathcal{C}_\beta^{(k)}\right).$$

L'essence du **flux ascendant** tient à ce que des **clusters** “stables” du niveau inférieur *remontent* et se placent comme **unités** (super-nœuds) au niveau supérieur.

Condition de cohésion

Pour qu'un cluster $\mathcal{C}_\alpha^{(k)}$ soit jugé “stable”, on impose souvent une **condition** de type :

$$\Omega\left(\mathcal{C}_\alpha^{(k)}\right) = \sum_{i,j \in \mathcal{C}_\alpha^{(k)}} \omega_{i,j}^{(k)}(t) > \theta_{\text{coh}},$$

où θ_{coh} est un seuil de cohésion. Une fois ce seuil dépassé, on *verrouille* $\mathcal{C}_\alpha^{(k)}$ et on crée un super-nœud $\mathcal{N}_\alpha^{(k+1)}$. C'est ce **verrouillage** qui incarne la décision de “promouvoir” un cluster micro en macro.

B. Motivation et Bénéfices

La **réduction de la complexité** constitue le premier atout majeur. Le flux ascendant filtre l'information en ne conservant que les éléments les plus denses ou les plus stables au niveau micro, ce qui permet de ne remonter que des agrégats cohésifs vers les niveaux supérieurs. Cela dispense le niveau macro de manipuler directement toutes les entités une à une, puisqu'une **compression** ou *coarse-graining* est opérée sur les pondérations ω dans la transition micro \rightarrow macro. Dans cet esprit, les blocs formés à l'échelle inférieure servent de briques de construction pour l'étape suivante, allégeant la matrice $\omega^{(k+1)}$.

La **spécialisation locale** joue un rôle essentiel. Au niveau micro, chaque entité ou petit cluster opère avec une réactivité accrue, par exemple en adoptant un taux d'apprentissage plus élevé η , ce qui permet des ajustements rapides et ciblés. Cette configuration permet de repérer rapidement une synergie forte. Ce n'est qu'une fois la cohésion atteinte que le cluster ainsi stabilisé est promu au niveau supérieur. En d'autres termes, le “détail” micro demeure géré en local, puis seul le résultat final de l'auto-organisation (un groupe cohésif) se transmet à l'échelle macro, favorisant l'efficacité et la robustesse de la construction hiérarchique.

Un **exemple formel** se produit lorsqu'un cluster $\mathcal{C} \subset \{\mathcal{E}_i^{(k)}\}$ a été identifié. Après détection, on construit un **super-nœud** $\mathcal{N}_\alpha^{(k+1)}$. Les pondérations $\omega_{\alpha,\beta}^{(k+1)}$ de ce nœud vis-à-vis d'un autre super-nœud $\mathcal{N}_\beta^{(k+1)}$ découlent alors d'une agrégation, typiquement :

$$\omega_{\alpha,\beta}^{(k+1)} = \Psi\left(\omega_{i,j}^{(k)} \mid i \in \mathcal{C}_\alpha^{(k)}, j \in \mathcal{C}_\beta^{(k)}\right),$$

où Ψ combine les poids $\omega_{i,j}^{(k)}$ (moyenne, maximum, minimum, etc.) selon les besoins. De cette manière, le **niveau** $k + 1$ ne conserve qu'un nombre restreint de super-nœuds, ce qui allège considérablement la matrice $\omega^{(k+1)}$ et participe à la flexibilité globale du SCN.

C. Exemples Illustratifs

Robotique : micro-clusters

Dans un essaim de robots, le niveau local (micro) peut repérer un **mini-groupe** très coopérant (quelques robots). Ce *micro-cluster* ascendant devient alors un *super-robot* au niveau supérieur, consolidant ses liaisons envers d'autres mini-groupes. On obtient un “**groupe de groupes**” au niveau macro, construit *bottom-up*.

Système cognitif : patterns neuronaux

Dans une architecture cognitive, le micro-niveau (p. ex. des *feature maps* en vision) détecte des **clusters** de neurones (ou filtres) s'activant fortement. Une fois stabilisé, ce cluster “remonte” en tant que *macro-patch* pour le niveau associatif, permettant une **synthèse** plus globale. De cette manière, l'information *micro* (détails sensoriels) s'organise en *proto-représentations* qui deviennent des super-nœuds conceptuels ou symboliques.

6.2.3.2. Flux descendants (top-down) : macro \rightarrow micro

Après avoir vu en section 6.2.3.1 comment les *micro-clusters* (ou ensembles locaux) “remontent” vers les niveaux supérieurs via le flux **ascendant**, il est tout aussi fondamental de comprendre le **flux descendant** (top-down). Celui-ci décrit la manière dont le **niveau macro** (ou intermédiaire, s'il est plus élevé) peut rétroagir et exercer une influence directe sur les **pondérations** ou la **dynamique** au niveau micro. Ce mécanisme top-down assure une **cohérence** globale du SCN (Synergistic Connection Network) en permettant aux intentions, visions ou constats du niveau supérieur de se **répercuter** localement dans les liens $\omega_{i,j}$.

A. Rôle du niveau macro vis-à-vis du niveau micro

Au **niveau macro**, on manipule un plus petit nombre de *super-nœuds* $\{\mathcal{M}_\alpha^{(k)}\}$, chacun agrégé depuis un niveau inférieur. Ce palier peut détecter une structure globale, par exemple l'émergence de deux **macro-clusters** se rapprochant ou la nécessité de les fusionner. Pour que cette **fusion** s'opère effectivement en bas (niveau micro), il peut être **nécessaire** que le macro-nœud envoie un signal top-down demandant d'**augmenter** certaines liaisons $\omega_{i,j}^{(\text{micro})}$ ou d'en **diminuer** d'autres, en fonction d'un objectif général.

D'un point de vue **mathématique**, on modélise la mise à jour au **niveau micro** par une équation augmentée :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_0 \left[S_0(i,j) - \tau_0 \omega_{i,j}^{(0)}(t) \right] + \gamma_{\text{top-down}} h_{i,j}^{(\text{macro})}(t),$$

où le terme $h_{i,j}^{(\text{macro})}(t)$ encode l'**influence** descendante du niveau macro (par exemple, un **boost** si on souhaite rapprocher deux entités i, j appartenant à des macro-nœuds censés fusionner).

Imaginons un macro-niveau $\omega^{(k)}$ détectant que deux grands ensembles $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$ s'avèrent synergiques, et qu'il est donc logique de les **fusionner**. Cependant, la **réalité** micro peut comporter des liens $\omega_{i,j}^{(0)}$ (pour $i \in \mathcal{N}_\alpha^{(k)}, j \in \mathcal{N}_\beta^{(k)}$) trop faibles. Le macro-niveau envoie alors un message top-down pour **accroître** ces $\omega_{i,j}^{(0)}$, guidant la dynamique micro vers un regroupement effectif.

Inversement, si un conflit est détecté, un message top-down peut **abaisser** certaines synergies locales, évitant des ambiguïtés ou chevauchements contradictoires.

Sans ce flux descendant, la **logique** de chaque niveau local risquerait d'être **myope**, focalisée sur de petites agrégations sans percevoir l'intérêt d'une fusion plus large. Le **macro-niveau** peut imposer ou suggérer des orientations qui, sinon, ne surgiraient pas localement. Cela donne une **harmonisation** entre la réactivité fine en bas et la vision globale en haut.

B. Forme générale du flux descendant

On peut définir, pour chaque niveau k , une fonction ou un opérateur Λ_k qui transforme l'état $\omega^{(k)}(t)$ du réseau macro (ses liens, ses clusters) en un **terme** de correction $\Delta^{(k-1)}$ pour le niveau $k - 1$. Concrètement :

$$\Lambda_k: \omega^{(k)}(t) \mapsto \Delta_{i,j}^{(k-1)}(t),$$

et on rajoute $\Delta_{i,j}^{(k-1)}(t)$ dans l'équation de mise à jour $\omega_{i,j}^{(k-1)}$. Cela permet d'effectuer des ajustements locaux sur $\omega_{i,j}$ sans affecter l'ensemble du réseau.

Dans la pratique, cette **rétroaction** top-down se manifeste par un **renforcement** des liens $\omega_{i,j}^{(0)}$ jugés stratégiques pour l'objectif global, comme la fusion, et par une **atténuation** des connexions lorsque le macro-niveau identifie un conflit ou une incompatibilité.

On peut programmer un **module** top-down qui, chaque X itérations, regarde $\omega^{(k)}$ et en déduit des $\Delta_{i,j}^{(k-1)}$ à appliquer. Ainsi, on obtient :

$$\omega_{i,j}^{(k-1)}(t+1) = \omega_{i,j}^{(k-1)}(t) + \Delta_{i,j}^{(k-1)}(t),$$

où $\Delta_{i,j}$ est *calculé* en fonction du macro-niveau.

Exemple

Soit un robot multi-niveau :

- Niveau **micro** : quelques robots $\{r_1, r_2, \dots\}$.
- Niveau **macro** : un "essaim" global défini par $\omega_{\alpha,\beta}^{(k)}$.

Si le macro-niveau souhaite que deux sous-essaims se coordonnent, il affecte une valeur positive Δ_{r_i,r_j} pour les robots r_i, r_j situés dans deux ensembles censés fusionner. Au cycle suivant, $\omega_{i,j}^{(0)}$ au niveau micro s'en voit **augmentée**, favorisant la coopération.

C. Finalité : Couplage Macro \rightarrow Micro pour une cohérence globale

Le macro-niveau peut “voir” des **corrélations** ou **objectifs** plus vastes. S’il veut encourager une convergence entre macro-nœuds, mais remarque des liens micro trop faibles, il agit en top-down pour relever ces liens. Ce faisant, il “induit” de nouvelles synergies locales.

Le flux ascendant (micro \rightarrow macro) consolide ce qui s’est stabilisé au niveau local. Le flux descendant (macro \rightarrow micro) renforce ou corrige ce qui serait souhaitable pour la vision d’ensemble. Ensemble, ils établissent un **cercle de rétroaction** où l’information remonte, est synthétisée au niveau macro, puis redescend sous forme de consignes. Cette boucle permet à la **dynamique** multi-niveau d’être plus stable et efficace, réduisant les essais-erreurs limités au niveau local.

En dernier lieu, ce mécanisme empêche l’**émiettement** excessif d’un SCN en multiples micro-groupes indépendants. Dès que le macro-niveau perçoit un intérêt commun, il enjoint aux entités de se rapprocher. Sur le plan **mathématique**, cela se formalise par des *termes d’influence* $\Delta_{i,j}$ insérés dans l’équation micro. Sur le plan **pratique**, c’est une manière de *diriger* ou *guider* localement l’auto-organisation vers une **harmonie** plus globale.

6.2.3.3. Communication entre niveaux synergiques et stabilisation

L’**organisation multi-niveau** d’un SCN repose sur une communication verticale entre les paliers micro, intermédiaire et macro. Le **flux ascendant** regroupe les interactions locales pour former des structures globales, tandis que le **flux descendant** ajuste la dynamique en fonction des objectifs globaux. Cette section illustre la **coexistence** des deux sens de circulation et décrit de quelle manière ils aboutissent à une **stabilisation** globale, où chaque niveau coopère avec les autres pour garantir la cohérence d’ensemble du réseau.

A. Schéma Général de Communication

On suppose un **système hiérarchique** à K paliers, indexés par $k = 0, 1, \dots, K$.

- Le niveau k dispose d’un ensemble de nœuds (entités ou super-nœuds) et d’une matrice $\omega^{(k)}$.
- Le passage $\omega^{(k)} \rightarrow \omega^{(k+1)}$ constitue un **flux ascendant** (bottom-up), noté Γ_k ou Ψ (6.2.3.1).
- Le passage $\omega^{(k+1)} \rightarrow \Delta^{(k)}$ (une correction appliquée à $\omega^{(k)}$) s’interprète comme un **flux descendant** (top-down), noté Λ_{k+1} .

Cette structure forme un **couplage** entre niveaux :

$$\begin{cases} \omega^{(k+1)}(t) = \Gamma_k \left(\omega^{(k)}(t) \right), \\ \omega^{(k)}(t+1) = \omega^{(k)}(t) + \Delta^{(k)} \left(\omega^{(k+1)}(t) \right). \end{cases}$$

Dans les faits, la mise à jour $\omega^{(k+1)}(t+1)$ peut aussi dépendre du nouvel état $\omega^{(k)}(t+1)$, ou être réalisée en mode **asynchrone** ; l'important est qu'il y ait un aller-retour permanent garantissant la cohésion.

Selon l'**implémentation**, on adopte un protocole :

- **Synchrone** : après chaque itération au niveau k , on agrège (Γ_k), met à jour $\omega^{(k+1)}$, puis calcule la correction $\Delta^{(k)}$ à renvoyer.
- **Asynchrone** : chaque niveau avance à son rythme, et on synchronise moins fréquemment, créant un *retard* possible entre les informations montantes et descendantes.

L'objectif reste d'assurer un **échange continu** entre les niveaux. Lorsque des micro-clusters émergent, le niveau macro en prend connaissance. Inversement, si le niveau macro identifie une structure globale, il renvoie un signal pour influencer certaines liaisons locales.

B. Stabilisation Multi-Niveau

Un SCN multi-niveau est dit **stabilisé** s'il existe un *point fixe* (ou un cycle) $\{\omega^{(k)*}\}_{k=0}^K$ tel que :

$$\Gamma_k(\omega^{(k)*}) = \omega^{(k+1)*}, \quad \Delta^{(k)}(\omega^{(k+1)*}) = 0, \quad \forall k.$$

Autrement dit, les agrégations bottom-up Γ_k reproduisent la configuration $\omega^{(k+1)*}$ déjà existante, tandis que les corrections top-down $\Delta^{(k)}$ sont nulles (pas de besoin d'ajustement). Cela signifie que la hiérarchie ne modifie plus ni les clusters ni les liens microscopiques.

La **convergence** vers un tel point fixe peut ne pas être triviale à démontrer. Des conditions de **stabilité** (taux d'apprentissage, seuils de cohésion, etc.) doivent être satisfaites. En pratique, on observe souvent que le SCN multi-niveau se stabilise ou aboutit à un *cycle* (un mouvement périodique restreint), plutôt que de rester en fluctuations incessantes. Dans certains cas, de légères oscillations peuvent persister, reflétant un **équilibre dynamique**.

Il peut se produire qu'un niveau **macro** "veille" un certain regroupement alors que le niveau **micro** persiste à séparer les entités. Cela peut entraîner des **conflits** ou des oscillations lorsque le niveau macro **impose** une augmentation de certains liens, tandis que le niveau micro **choisit** ensuite de les affaiblir, créant ainsi un cycle d'ajustements contradictoires. Pour éviter cela, on peut recourir à des *mécanismes de temporisation* ou un *paramétrage* plus modéré ($\gamma_{\text{top-down}}$ pas trop élevé, etc.) assurant la douceur de la rétroaction.

C. Communication et Rétroaction Continue

Il est possible que *chaque* niveau applique la **même** règle DSL (additive, multiplicative, etc.), assurant une sorte d'**auto-similarité** du mécanisme. On peut également envisager des paramètres η_k, τ_k distincts pour chaque niveau, avec un niveau micro plus **réactif** et un niveau macro plus **stable**, évitant ainsi les oscillations excessives.

Cette modularité engendre un **système** multi-niveau dans lequel la **fréquence** et la **vitesse** de mise à jour diffèrent, encourageant une stabilisation plus locale en bas, puis une adaptation plus globale au sommet.

En pratique, l'information **monte** par agrégation en condensant $\omega^{(k)}$ en $\omega^{(k+1)}$, comme décrit en **6.2.3.1**. Puis elle **redescend** par corrections (6.2.3.2), ce qui se matérialise par des **messages** top-down affectant $\omega^{(k)}$. Un système **synchrone** peut gérer ça en “phases” successives, un système **asynchrone** le fait plus irrégulièrement. D'un point de vue **mathématique**, on peut introduire des *délais* δ_k simulant une communication “à la volée”.

Le résultat est une **pyramide** ou un “arbre” hiérarchique :

- le niveau 0 (micro) manipule $\omega^{(0)}$ et détecte des clusters,
- le niveau 1 construit $\omega^{(1)}$ en agrégeant,
- etc.

Chaque palier envoie **rétroactions** (top-down) et “résultats” (bottom-up) dans un **cycle** continu. Quand ce cycle se **calme** (ou s'équilibre), on a une **organisation** stable du SCN multi-niveau.

6.3. Fractalité et Auto-Similarité dans le DSL

Après avoir exploré la hiérarchie multi-niveau (6.2) et les mécanismes de communication entre paliers (bottom-up, top-down), nous abordons maintenant la **possibilité** qu'un **SCN** (Synergistic Connection Network) puisse présenter une **fractalité** — c'est-à-dire une forme d'**auto-similarité** à travers différentes échelles. Dans l'histoire des systèmes complexes, la notion de fractales, venue de la géométrie et de la modélisation de phénomènes naturels, s'avère un **paradigme** puissant pour décrire les invariances d'échelle. La section 6.3.1 introduit le concept de fractales en IA, puis 6.3.2 et 6.3.3 détaillent comment cette auto-similarité peut se manifester et se mesurer dans le cadre du DSL, et enfin 6.3.4 discute les avantages et les limites de la fractalité appliquée aux réseaux synergiques.

6.3.1. Concept de Fractales en IA

Les **fractal(e)s** renvoient à des objets ou des systèmes dans lesquels on retrouve, à plusieurs échelles d'observation, une **structure** (ou un "motif") essentiellement identique, éventuellement modulo un facteur d'échelle. C'est cette idée d'**invariance d'échelle** ou d'**auto-similarité** qui caractérise la fractalité, qu'il s'agisse d'ensembles géométriques (courbe de Koch, ensemble de Cantor, etc.) ou de phénomènes physiques et biologiques (réseaux vasculaires, littoraux, ramifications dans un arbre, etc.).

6.3.1.1. Historique : Fractales dans la Modélisation de Phénomènes Naturels, Auto-similarité à Plusieurs Échelles

La notion de fractale, introduite et popularisée dans les années 1970–1980 notamment par Benoît Mandelbrot, a profondément transformé notre compréhension des phénomènes naturels complexes. En effet, nombre d'objets ou de processus, traditionnellement jugés irréguliers ou chaotiques, révèlent une structure sous-jacente caractérisée par l'auto-similarité, c'est-à-dire la répétition de motifs à différentes échelles. Cette propriété permet de modéliser, d'une manière relativement concise, des objets dont la complexité apparente dépasse les approches classiques de la géométrie euclidienne.

A. Émergence du Concept de Fractale

1. Les Travaux Pionniers de Mandelbrot

Au cœur des travaux de Mandelbrot se trouve l'observation que de nombreux phénomènes naturels — comme les côtes marines, les nuages, les réseaux d'agrégation ou même certains aspects des systèmes biologiques — présentent une irrégularité apparente qui, lorsqu'on les observe à des échelles différentes, révèle une structure récurrente. Dans *The Fractal Geometry of Nature*, Mandelbrot illustre comment la mesure d'une côte varie avec la résolution d'observation. Plus on affine la mesure, plus de détails émergent, entraînant une augmentation non linéaire de la longueur calculée. Cette propriété a conduit à la conceptualisation d'objets dont la dimension, dans le sens

de Hausdorff ou de Minkowski, est non entière. Pour le flocon de Koch, l'un des exemples classiques de fractale, la dimension fractale est donnée par :

$$\dim_H(\text{Flocon de Koch}) = \frac{\ln 4}{\ln 3} \approx 1.2619,$$

ce qui indique que cet objet, bien que contenu dans le plan, possède une complexité qui dépasse la simple dimension linéaire.

2. Auto-similarité et Invariance d'Échelle

La propriété d'auto-similarité est l'un des piliers de la géométrie fractale. Un objet est dit auto-similaire s'il se recoupe avec lui-même à différentes échelles de grandeur, c'est-à-dire que si l'on effectue un zoom sur une portion de l'objet, cette portion présente la même structure (à un facteur d'échelle près) que l'ensemble. Mathématiquement, on peut exprimer cette propriété à l'aide de similitudes. Soit $F \subset \mathbb{R}^d$ un ensemble fractal, alors il existe un nombre fini de fonctions contractantes $\{f_1, f_2, \dots, f_n\}$ telles que :

$$F = \bigcup_{i=1}^n f_i(F).$$

Ce formalisme, qui fait partie de la théorie des systèmes itératifs par fonction (IFS), traduit l'invariance d'échelle propre aux fractales. Par ailleurs, de nombreux phénomènes naturels se caractérisent par des lois de puissance, telles que :

$$P(X > x) \sim x^{-\alpha},$$

ce qui illustre que la distribution de certaines quantités (taille des amas, fréquence des phénomènes sismiques, etc.) reste identique lorsqu'on change d'échelle d'observation.

B. Exemples de Phénomènes Naturels

1. Côtes Maritimes et Littoraux

L'un des exemples les plus emblématiques est celui de la mesure des côtes. Lorsque l'on mesure la longueur d'un littoral, la valeur obtenue dépend de la précision de la règle utilisée. Si l'on se limite à une règle de 100 mètres, on obtient une certaine valeur, mais en utilisant une règle de 1 mètre, la longueur mesurée augmente de manière significative. Ce phénomène, illustré par Mandelbrot, montre que les côtes possèdent une dimension fractale comprise entre 1 et 2, reflétant ainsi leur complexité géométrique.

2. Réseaux Vasculaires et Bronchiques

Les systèmes biologiques, tels que les réseaux vasculaires ou bronchiques, exhibent également des propriétés fractales. Dans ces systèmes, les structures se ramifient de manière récurrente, chaque branche se subdivisant en sous-branches selon des ratios constants. Cette organisation optimise la distribution des fluides ou de l'air. La structure auto-similarité assure que l'ensemble du réseau, du plus petit capillaire jusqu'à l'organe complet, obéit à des règles géométriques similaires.

3. Lois de Puissance dans les Agrégations

En physique statistique et en géophysique, on observe fréquemment que la distribution des tailles d'agrégats ou la fréquence des phénomènes (comme les séismes) suit une loi de puissance. Ces lois de puissance traduisent une invariance d'échelle dans l'agrégation des éléments, suggérant que la même dynamique de formation se répète quelle que soit l'échelle d'observation.

C. Fractales et Auto-similarité dans les SCN et l'IA

1. Réseaux Complexes et Dimension Fractale

Les réseaux complexes, qu'ils soient sociaux, biologiques ou informatiques, présentent souvent des distributions de degrés et des structures hiérarchiques qui rappellent les propriétés fractales. Par exemple, la répartition des degrés dans un réseau dit « scale-free » suit typiquement une loi de puissance, indiquant que la structure du réseau est auto-similaire. Cette propriété peut être mise en parallèle avec la dimension fractale des objets géométriques, suggérant que le même type de processus d'agrégation opère à différentes échelles.

2. Auto-organisation dans les SCN

Dans un SCN, la règle d'auto-organisation (par exemple, la mise à jour des pondérations $\omega_{i,j}$ selon la règle DSL) peut se répéter à plusieurs niveaux hiérarchiques. Si la même dynamique s'applique localement dans des clusters et se retrouve également au niveau global, on peut alors qualifier le système de fractal dans le sens où il présente une **auto-similarité** d'une échelle à l'autre. Cette observation permet de modéliser la formation de clusters et d'agrégats dans un SCN par des lois de puissance et d'invariance d'échelle, donnant ainsi une dimension supplémentaire à l'analyse des réseaux auto-organisés dans le domaine de l'IA.

3. Impact et Applications

L'étude des fractales et de l'auto-similarité a des implications profondes pour la modélisation de phénomènes naturels et pour la conception de systèmes d'intelligence artificielle. Dans les SCN, elle permet de mieux comprendre comment des dynamiques locales peuvent se répéter pour générer une organisation globale cohérente. De plus, la notion de dimension fractale offre un outil quantitatif pour mesurer la complexité d'un réseau et pour évaluer la robustesse de son auto-organisation. Ces concepts trouvent des applications dans des domaines variés, allant de la modélisation des réseaux neuronaux à la prédiction des comportements dans les systèmes complexes.

6.3.1.2. Intérêt pour le DSL : la Récurrence de Patterns Synergiques à Différentes Granularités

L'idée d'auto-similarité, centrale dans l'étude des fractales, se retrouve également dans le cadre du Deep Synergy Learning (DSL) lorsqu'on observe que les mêmes motifs d'organisation apparaissent à plusieurs niveaux de granularité, du micro au macro. Autrement dit, la dynamique qui régit l'évolution locale des pondérations $\omega_{i,j}$ se réplique à différentes échelles, conduisant à une structuration hiérarchique qui se caractérise par l'auto-similarité. Nous développons ci-après, en enrichissant l'argumentation par des éléments mathématiques, comment cette récurrence de patterns synergiques s'exprime et quels en sont les avantages pour un SCN.

A. Notion de Patterns Synergiques et Auto-similarité

1. Définition Mathématique d'un Motif

Considérons un petit graphe ou sous-réseau, noté \mathcal{P} , qui représente un motif synergique typique dans un SCN. Ce motif peut être défini par un ensemble de nœuds $\{1, 2, \dots, r\}$ et une matrice de pondérations associée $\omega^{(\text{patt})} = \left(\omega_{a,b}^{(\text{patt})} \right)_{1 \leq a, b \leq r}$. L'idée fondamentale est que, si l'on peut trouver, dans un grand réseau G (représenté par la matrice ω d'un SCN), un sous-ensemble $\mathcal{C} \subset \{1, \dots, n\}$ qui possède une topologie similaire à celle de \mathcal{P} , alors il existe une application bijective (ou une homothétie approchée)

$$\phi: \mathcal{C} \rightarrow \mathcal{P},$$

telle que pour tout $i, j \in \mathcal{C}$, la relation suivante est approximativement vérifiée :

$$\omega_{i,j} \approx \lambda \omega_{\phi(i), \phi(j)}^{(\text{patt})},$$

où λ est un facteur d'échelle positif. Cette relation traduit l'**auto-similarité**, où la structure du motif \mathcal{P} réapparaît dans le sous-groupe \mathcal{C} à une échelle différente. Ainsi, le même arrangement de liens se répète, ce qui est caractéristique d'un comportement fractal dans un réseau.

2. Dimension Fractale et Couverture en Boîtes

Un autre outil mathématique issu de l'analyse des fractales est la **dimension de Hausdorff** (ou dimension fractale) qui, dans le contexte d'un SCN, peut être interprétée en examinant comment le nombre de sous-groupes distincts évolue en fonction de l'échelle de résolution. Plus précisément, en couvrant le réseau par des "boîtes" de taille ϵ (où ϵ représente ici une tolérance dans la similarité des valeurs de ω ou dans la distance physique entre les agents), on observe que le nombre de boîtes nécessaires $N(\epsilon)$ satisfait typiquement une relation de type puissance :

$$N(\epsilon) \propto \epsilon^{-\dim_f},$$

où \dim_f est la dimension fractale du réseau. L'invariance d'échelle implique que, quelle que soit la granularité choisie (micro, méso ou macro), la structure du réseau conserve une distribution caractéristique des liaisons, reflétant l'uniformité de la dynamique d'auto-organisation.

B. Récurrence de Motifs via la Dynamique DSL

1. La Règle de Mise à Jour Uniforme

La dynamique du DSL s'exprime typiquement par une règle de mise à jour telle que :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ est la synergie mesurée entre les entités i et j , et η et τ sont des paramètres constants (taux d'apprentissage et coefficient de décroissance, respectivement). Le fait que cette même règle soit appliquée à chaque niveau du réseau signifie que le même opérateur F agit sur les pondérations, que l'on observe au niveau d'un micro-cluster ou d'un macro-cluster.

Autrement dit, l’auto-organisation locale (à l’échelle d’un petit groupe de robots par exemple) se fait selon une mécanique identique à celle du système global. Cela se traduit par la **répétition** de la même forme de dynamique :

$$\omega^{\text{micro}}(t + 1) = F(\omega^{\text{micro}}(t), S^{\text{micro}}),$$

$$\omega^{\text{macro}}(t + 1) = F(\omega^{\text{macro}}(t), S^{\text{macro}}),$$

avec S^{micro} et S^{macro} étant les scores de synergie évalués à des échelles différentes. Si ces scores sont obtenus par des mécanismes analogues – par exemple via des fonctions de similarité normalisées – alors la **forme** du système (et, en particulier, la distribution des ω) se révèle invariante d’échelle, c’est-à-dire fractale.

2. Lois de Puissance et Distribution de la Taille des Clusters

Une conséquence de l’auto-similarité dans un SCN est que la distribution des tailles de clusters peut suivre une **loi de puissance**. Plus formellement, si l’on note s la taille d’un cluster (le nombre d’entités qui le composent), on peut observer que :

$$P(s) \sim s^{-\alpha},$$

où α est un exposant caractéristique. Ce comportement scale-free indique que la même règle d’agrégation, appliquée à différentes échelles, donne lieu à la formation d’un petit nombre de grands clusters et à une majorité de clusters de taille modeste. Cette propriété fractale assure une **robustesse** du système puisque le mécanisme de formation des clusters n’est pas sensible à l’échelle, mais reste invariant, favorisant ainsi une hiérarchie cohérente d’organisations locales et globales.

3. Fusion et Invariance d’Échelle

Dans le cadre du DSL, le score de synergie $S(i, j)$ est souvent le résultat d’un processus de fusion de divers sous-scores (par exemple, issus de différentes modalités ou d’interactions à différents niveaux). Si l’on définit un score hybride comme :

$$S(i, j) = \alpha S_{\text{sub}}(i, j) + \beta S_{\text{sym}}(i, j),$$

et si ce même mécanisme de fusion est appliqué dans chaque sous-système du SCN, alors la même structure de calcul apparaît quel que soit le niveau d’agrégation. En particulier, si l’on peut identifier un motif \mathcal{P} à une échelle micro tel que

$$\omega_{i,j}^{\text{micro}} \approx \lambda \omega_{\phi(i),\phi(j)}^{(\mathcal{P})},$$

alors, en agréant ces micro-clusters, le macro-cluster résultant exhibera une structure similaire (avec éventuellement un facteur d’échelle différent). Cela traduit l’invariance d’échelle et renforce l’idée que l’auto-organisation s’effectue de manière fractale.

C. Avantages pour l’Apprentissage Multi-échelle dans le DSL

L’intérêt de la récurrence des patterns synergiques dans un SCN se manifeste sur plusieurs plans :

5. Homogénéité des Mécanismes d'Auto-organisation

Lorsque la même règle d'agrégation–inhibition s'applique à toutes les échelles, le système gagne en simplicité conceptuelle. Un unique mécanisme, qu'on peut exprimer par $F(\omega(t))$, suffit pour régir la dynamique du réseau, de la plus petite unité à l'agrégation globale. Cette homogénéité simplifie l'analyse mathématique et la mise en œuvre algorithmique.

6. Robustesse et Résilience

La présence d'un comportement fractal – où la structure d'un micro-cluster se répète à l'échelle macro – signifie que le système est intrinsèquement résilient. En effet, si une partie du réseau est perturbée ou isolée, la dynamique locale qui reste intacte permet de reconstruire la structure globale par agrégation, sans nécessiter de réinitialisation complète.

7. Lois de Puissance et Détection d'Émergences

L'apparition de lois de puissance dans la distribution des tailles de clusters permet non seulement de quantifier la structure du réseau mais aussi d'identifier des seuils critiques dans l'évolution de l'auto-organisation. Par exemple, si la distribution des tailles de clusters se stabilise selon $P(s) \sim s^{-\alpha}$, alors la valeur de l'exposant α peut servir d'indicateur pour adapter dynamiquement les paramètres du DSL (comme η ou τ) afin de préserver la stabilité du système.

8. Simplicité du Modèle

Du fait que la même dynamique se répète à chaque niveau, le modèle global se réduit à une itération de la même règle. Cela permet de construire des simulations et des théories analytiques qui, en se concentrant sur une échelle donnée, s'appliquent ensuite par récurrence aux autres échelles, facilitant ainsi la compréhension de l'ensemble du processus d'auto-organisation.

6.3.2. Auto-Similarité dans les Réseaux Synergiques

La fractalité, telle que discutée en (6.3.1), trouve un **terrain d'expression** privilégié dans un SCN (Synergistic Connection Network) lorsque l'on constate que les *patterns* d'auto-organisation se répètent, à la fois **localement** (dans de petits clusters) et **globalement** (à l'échelle d'un cluster plus vaste), sans nécessiter de règles qualitativement différentes. C'est cette **auto-similarité** à travers les échelles, ou *invariance d'échelle*, qui caractérise un **réseau fractal**.

6.3.2.1. Exemple : Un Cluster Local Reproduit la Même Structure que le Cluster Global (Formes d'Invariance d'Échelle)

Dans le contexte d'un SCN (Synergistic Connection Network), l'auto-organisation repose sur l'évolution locale des pondérations $\omega_{i,j}$ via des règles d'actualisation telles que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i, j)$ est le score de synergie entre les entités i et j . Une des propriétés remarquables de cette dynamique est la possibilité d’observer des structures similaires à des échelles différentes – un phénomène que l’on qualifie d’**invariance d’échelle** ou d’**auto-similarité**. Autrement dit, un petit cluster (micro-cluster) de robots ou d’agents, qui se forme par des interactions locales fortes, peut présenter une configuration similaire à celle d’un cluster global (macro-cluster) englobant un nombre beaucoup plus important d’entités. Nous détaillons ci-après cette idée en intégrant des arguments mathématiques et en exposant les implications pour le DSL (Deep Synergy Learning).

A. Structure d’un Cluster Local et d’un Cluster Global

Supposons qu’on ait identifié, à l’échelle locale, un cluster \mathcal{C}_{loc} composé de n_{loc} entités qui sont fortement interconnectées. La topologie de ce cluster se caractérise par une distribution interne des pondérations

$$\{\omega_{i,j}^{(\text{loc})} \mid i, j \in \mathcal{C}_{\text{loc}}\},$$

qui présente par exemple une densité élevée, une répartition particulière (cycle, étoile, réseau complet partiel) ou encore une symétrie dans l’organisation des liens.

Au niveau global, on considère un cluster $\mathcal{C}_{\text{glob}}$ de taille beaucoup plus importante, $n_{\text{glob}} \gg n_{\text{loc}}$. L’hypothèse d’auto-similarité consiste à constater que la structure (ou le “motif”) de \mathcal{C}_{loc} se retrouve, à un facteur d’échelle près, dans $\mathcal{C}_{\text{glob}}$. En d’autres termes, il existe une application ou un isomorphisme (éventuellement approché)

$$\phi: \mathcal{C}_{\text{glob}} \rightarrow \mathcal{P},$$

où \mathcal{P} représente un motif de référence qui est isomorphe (à un facteur d’échelle près) au motif local. Plus formellement, pour toute paire d’entités $i, j \in \mathcal{C}_{\text{glob}}$, la relation suivante s’exprime :

$$\omega_{\phi(i), \phi(j)}^{(\text{glob})} \approx \lambda \omega_{i,j}^{(\text{loc})},$$

où $\lambda > 0$ est un facteur d’échelle qui permet de “rescaler” la distribution des pondérations du cluster local afin de la faire correspondre à celle du cluster global. Ce résultat exprime l’**invariance d’échelle**, où la topologie des connexions se réplique de manière similaire, indépendamment de la différence d’ordre de grandeur du nombre d’entités.

B. Invariance d’Échelle et Modélisation Mathématique

1. Homothétie et Isomorphisme

Pour exprimer mathématiquement l’auto-similarité, on peut considérer que le motif local \mathcal{P} est obtenu par une transformation homothétique d’un sous-ensemble du cluster global. Soit $\omega^{(\text{loc})}$ la matrice des pondérations dans le cluster local et $\omega^{(\text{glob})}$ celle du cluster global. L’existence d’une fonction de transformation ϕ et d’un facteur λ s’exprime par :

$$\forall i, j \in \mathcal{C}_{\text{loc}}, \quad \omega_{\phi(i), \phi(j)}^{(\text{glob})} = \lambda \omega_{i,j}^{(\text{loc})} + \varepsilon_{i,j},$$

où $\varepsilon_{i,j}$ représente une erreur ou une perturbation qui tend à zéro dans l'idéal. Ce cadre mathématique, proche de la notion d'isomorphisme entre graphes pondérés, permet de caractériser l'auto-similarité par la présence d'une *structure résiliente* et récurrente. En outre, en couvrant le cluster global par des boîtes de taille ϵ (dans le sens de la méthode de box-counting utilisée pour calculer la dimension fractale), on obtient une relation du type :

$$N(\epsilon) \propto \epsilon^{-\dim_f},$$

où $N(\epsilon)$ est le nombre de boîtes nécessaires pour couvrir le cluster global et \dim_f la dimension fractale du réseau. Cette relation illustre que la structure du réseau est invariante par changement d'échelle, caractéristique des systèmes fractals.

2. Loi de Puissance dans la Distribution des Clusters

L'auto-similarité dans un SCN se traduit également par une distribution en loi de puissance des tailles de clusters. Si l'on note $s = |\mathcal{C}|$ la taille d'un cluster, on peut observer empiriquement que :

$$P(s) \sim s^{-\alpha},$$

ce qui signifie que, quelle que soit l'échelle, la probabilité d'obtenir un cluster de taille s décroît selon une loi de puissance. Ce comportement scale-free est une caractéristique essentielle des réseaux fractals, indiquant que le même mécanisme d'agrégation et de séparation s'applique indépendamment de la taille des clusters.

C. Interprétations dans le DSL et Conséquences pour l'Auto-organisation

1. Unicité du Mécanisme d'Auto-organisation

L'application uniforme de la règle DSL, telle que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

indique que la même dynamique opère sur tous les niveaux du réseau. Cette uniformité signifie que le processus d'agrégation (renforcement des liens forts) et d'inhibition (affaiblissement des liens faibles) se répète de manière identique, qu'on observe un petit groupe local ou l'ensemble global des agents. Ainsi, un micro-cluster se forme localement par la même mécanique que celle qui gouverne la formation d'un macro-cluster. Le fait que la même règle puisse être appliquée de manière récursive dans le réseau est un gage de robustesse et de simplicité du modèle.

2. Avantages pour l'Apprentissage Multi-échelle

L'auto-similarité fractale dans le DSL offre plusieurs avantages majeurs :

- **Simplicité du modèle** : La possibilité de décrire le comportement global du réseau par une seule règle d'auto-organisation permet d'éviter la complexité inhérente à la gestion de multiples mécanismes distincts pour chaque échelle. Un modèle unique s'applique localement et se propage globalement par rescaling.

- **Robustesse** : La redondance des motifs d'organisation, reproduits à différentes échelles, assure que le système reste stable même en cas de perturbations locales. Ainsi, un micro-cluster qui se déstabilise peut être compensé par la stabilité d'un macro-cluster dont la structure est identique.
- **Facilité d'analyse** : Les lois de puissance associées à la distribution des tailles de clusters et la relation homothétique entre les niveaux permettent d'exploiter des outils mathématiques (dimension fractale, box-counting) pour prédire et contrôler l'évolution du réseau.
- **Adaptabilité** : Dans un environnement dynamique, la capacité de reproduire la même structure d'auto-organisation à différentes granularités permet d'adapter le modèle sans recalibrer l'ensemble des paramètres pour chaque niveau d'agrégation.

6.3.2.2. Conditions Mathématiques : Si la Distribution des Entités et leurs Synergies Respectent Certaines Lois de Puissance, un Aspect Fractal Peut Émerger

Dans l'étude des systèmes complexes, et plus particulièrement dans le cadre des Synergistic Connection Networks (SCN) appliqués au Deep Synergy Learning (DSL), il apparaît que la dynamique d'auto-organisation peut, sous certaines conditions, donner lieu à un comportement fractal. Ce phénomène se manifeste lorsque la distribution des pondérations $\omega_{i,j}$, la taille des clusters $|C|$ ou d'autres indicateurs statistiques suivent des lois de puissance. Nous détaillons ici les fondements mathématiques de ces conditions, en insistant sur l'invariance d'échelle et les mécanismes qui conduisent à l'émergence d'un aspect fractal dans un SCN.

A. Notion de Lois de Puissance et d'Invariance d'Échelle

Une loi de puissance se caractérise par l'expression générale

$$\text{Prob}(X > x) \sim x^{-\alpha},$$

pour x suffisamment grand, où α est l'exposant caractéristique de la distribution. Cette relation implique l'absence d'une échelle caractéristique pour la variable X , où la probabilité que X dépasse une valeur x suit une décroissance selon une loi de puissance. Une propriété clé des lois de puissance est leur invariance d'échelle. En effet, pour tout facteur de redimensionnement $\lambda > 0$, on a :

$$\text{Prob}(\lambda X > x) \approx \lambda^{-\alpha} \text{Prob}(X > x).$$

Ce comportement signifie que, si l'on “zoome” sur la distribution, la forme de la courbe demeure identique à une constante multiplicative près. Dans le contexte d'un SCN, si la distribution des valeurs de $\omega_{i,j}$, ou la répartition des tailles de clusters, suit une telle loi de puissance, alors le réseau est dit *invariant d'échelle*, une propriété fondamentale pour caractériser la fractalité.

B. Génération de Loïs de Puissance par la Dynamique DSL

Dans un SCN, la dynamique d'auto-organisation est régie par une règle de mise à jour du type :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ représente le score de synergie entre les entités i et j , η est le taux d'apprentissage, et τ le coefficient de décroissance. Ce schéma de mise à jour présente plusieurs similitudes avec le mécanisme de renforcement préférentiel (« rich-get-richer ») étudié dans la théorie des réseaux complexes. En effet, lorsqu'un lien est légèrement supérieur à la moyenne, la dynamique tend à le renforcer, tandis que les liens faibles tendent à s'effacer. Ce processus de sélection naturelle des interactions conduit, sous certaines conditions, à l'émergence d'une distribution très hétérogène des pondérations.

Lorsque l'on ne contraint pas artificiellement la distribution (par exemple, par un seuil strict de saturation), la compétition entre les liens peut conduire à une répartition des valeurs de $\omega_{i,j}$ qui suit une loi de puissance. De même, dans l'agrégation hiérarchique des clusters – où les micro-clusters sont regroupés pour former des macro-clusters – si le même mécanisme de renforcement et d'inhibition s'applique à chaque niveau, on observe que la distribution des tailles de clusters, notée par exemple

$$\text{Prob}(|\mathcal{C}| > s) \sim s^{-\alpha},$$

reste invariante d'échelle. Cela signifie que le même schéma statistique apparaît tant à petite qu'à grande échelle, une signature typique d'un comportement fractal.

C. Conditions d'Invariance d'Échelle et Fractalité Statistique

Pour qu'un aspect fractal émerge dans un SCN, il faut que plusieurs conditions soient réunies :

9. Invariance de la Forme de la Distribution

Si l'on considère une variable X (qui peut être, par exemple, la force interne d'un cluster $\Omega(\mathcal{C}) = \sum_{i,j \in \mathcal{C}} \omega_{i,j}$ ou le degré d'un nœud $\deg(i) = \sum_j \omega_{i,j}$), l'invariance d'échelle se traduit par la propriété

$$\text{Prob}(\lambda X > x) \approx \lambda^{-\alpha} \text{Prob}(X > x),$$

ce qui indique que la distribution est une loi de puissance sans échelle caractéristique. Cette propriété doit être observée sur plusieurs ordres de grandeur, c'est-à-dire pour $x \in [x_{\min}, x_{\max}]$.

10. Répétition du Mécanisme d'Auto-organisation

La même règle de mise à jour des pondérations, appliquée de manière itérative, doit générer une dynamique qui se répète à toutes les échelles. Autrement dit, si l'on observe un micro-cluster \mathcal{C}_{loc} de petite taille et un macro-cluster $\mathcal{C}_{\text{glob}}$ beaucoup plus grand, il doit exister une transformation d'homothétie ϕ et un facteur λ tel que :

$$\omega_{\phi(i),\phi(j)}^{(\text{glob})} \approx \lambda \omega_{i,j}^{(\text{loc})} \quad \text{pour } i, j \in \mathcal{C}_{\text{loc}}.$$

Cette relation est au cœur de l'auto-similarité fractale et permet d'affirmer que le même schéma d'interconnexion se reproduit, simplement rescalé, d'un niveau à l'autre.

11. Aggregation et Box-Counting

Pour quantifier l'invariance d'échelle, on peut utiliser la méthode de box-counting. Si l'on recouvre le réseau par des boîtes de taille ϵ , le nombre de boîtes nécessaires $N(\epsilon)$ devrait suivre une loi de la forme :

$$N(\epsilon) \propto \epsilon^{-\dim_f},$$

où \dim_f est la dimension fractale du réseau. La constance de \dim_f à différentes échelles indique que la structure du réseau est auto-similaire.

D. Illustrations Concrètes et Implications

Plusieurs indicateurs permettent d'observer ces comportements dans un SCN :

- **Distribution de la Force Interne des Clusters** : La somme des pondérations à l'intérieur d'un cluster, $\Omega(\mathcal{C})$, si elle suit une loi de puissance,

$$\text{Prob}(\Omega(\mathcal{C}) > x) \sim x^{-\alpha},$$

témoigne d'une invariance d'échelle dans la façon dont les interactions se répartissent au sein des clusters.

- **Distribution des Degrés** : Si le degré d'un nœud, défini par $\deg(i) = \sum_j \omega_{i,j}$, suit une loi de puissance, cela indique que le SCN est *scale-free*, avec quelques nœuds hubs et une majorité de nœuds à faible degré. Ce comportement se retrouve généralement dans les réseaux fractals.
- **Répétition des Motifs** : La présence d'un motif structurel \mathcal{P} dans un micro-cluster, qui se retrouve dans un macro-cluster après application d'une transformation homothétique, confirme l'auto-similarité. Cette observation permet d'affirmer que la même logique d'agrégation et d'inhibition, codifiée par la dynamique DSL, opère de manière récurrente.

Ces constats mathématiques et statistiques confirment que, dans un SCN où la mise à jour des pondérations ne présente pas d'échelle caractéristique, l'auto-organisation produit des structures fractales. L'avantage est double. D'une part, il devient possible d'analyser et de prévoir le comportement du réseau à partir d'un sous-ensemble représentatif. D'autre part, cette invariance d'échelle garantit une robustesse et une homogénéité dans l'apprentissage multi-échelle, ce qui facilite l'extension du modèle à des systèmes de grande taille.

6.3.2.3. Mesures Fractales Possibles (Dimension Fractale d'un Graphe, etc.)

L'analyse de la structure fractale dans des systèmes complexes repose sur l'idée d'**invariance d'échelle**, c'est-à-dire que la même forme ou le même comportement se répète à différentes échelles de mesure. Dans le contexte d'un SCN (Synergistic Connection Network), cette approche permet de quantifier l'auto-similarité de la répartition des pondérations $\omega_{i,j}$ et, par extension, de la structure des clusters qui en résultent. Pour ce faire, plusieurs mesures fractales, issues de la géométrie fractale, ont été adaptées à l'étude des graphes pondérés et des réseaux complexes. Nous présentons ici en détail les principales méthodes, leurs fondements mathématiques et leur application au DSL (Deep Synergy Learning).

A. Rappels sur le Box-Counting en Géométrie Fractale

L'une des méthodes les plus classiques pour estimer la **dimension fractale** d'un ensemble $F \subset \mathbb{R}^d$ est la méthode du box-counting. Pour un tel ensemble, on définit $N(\varepsilon)$ comme le nombre minimal de boîtes (ou hypercubes) de côté ε nécessaires pour couvrir l'ensemble F . Si, pour $\varepsilon \rightarrow 0$, on observe que

$$N(\varepsilon) \sim \varepsilon^{-\dim_f},$$

alors la valeur \dim_f est appelée la **dimension fractale** (ou dimension de Minkowski–Bouligand) de l'ensemble F . Plus précisément, on définit

$$\dim_f = \lim_{\varepsilon \rightarrow 0} \frac{\ln N(\varepsilon)}{-\ln \varepsilon},$$

ce qui exprime le fait que la couverture de l'ensemble se fait suivant une loi de puissance. Dans le cas des objets géométriques classiques (ensemble de Cantor, flocon de Koch), cette méthode aboutit à des valeurs non entières, traduisant la complexité inhérente à ces structures.

B. Adaptation du Box-Counting à un SCN ou Graphe Pondéré

Dans un SCN, les entités sont représentées par des nœuds et les interactions par des liaisons pondérées $\omega_{i,j}$. Ces éléments ne sont pas nécessairement disposés dans un espace euclidien de manière canonique ; cependant, on peut définir une **métrique adaptée** à partir des pondérations. Par exemple, on peut définir la distance entre deux nœuds i et j par :

$$d(i, j) = \min_{\gamma(i \rightarrow j)} \sum_{(u,v) \in \gamma} \frac{1}{\omega_{u,v}},$$

où $\gamma(i \rightarrow j)$ désigne un chemin reliant i à j . L'inversion de la pondération permet d'interpréter un lien fort (grand ω) comme une distance courte. À partir de cette métrique, la méthode de box-counting peut être adaptée de la manière suivante :

1. Définition d'une Boîte dans le Graphe

Pour un nœud central c et un paramètre $\ell > 0$, on définit la "boîte" $B(c, \ell)$ comme l'ensemble des nœuds j tels que

$$B(c, \ell) = \{j \mid d(c, j) \leq \ell\}.$$

2. Recouvrement Minimal du Graphe

On cherche ensuite le nombre minimal $N(\ell)$ de tels blocs $B(c, \ell)$ nécessaire pour couvrir l'ensemble des nœuds du graphe. Si la relation

$$N(\ell) \sim \ell^{-\dim_f}$$

est vérifiée pour $\ell \rightarrow 0$, alors le graphe possède une **dimension fractale** \dim_f qui caractérise son auto-similarité. La pente de la courbe obtenue en traçant $\ln N(\ell)$ en fonction de $\ln(1/\ell)$ donne une estimation de \dim_f .

C. Méthodes Alternatives pour Quantifier la Fractalité

1. Dimension de Corrélation

La dimension de corrélation est une autre mesure employée dans l'analyse des attracteurs fractals, notamment via l'algorithme de Grassberger–Procaccia. Dans le contexte d'un SCN, on définit la fonction de corrélation $C(\ell)$ comme la probabilité qu'une paire de nœuds soit à une distance inférieure à ℓ :

$$C(\ell) = \frac{2}{N(N-1)} \sum_{i < j} \chi(d(i, j) \leq \ell),$$

où χ est la fonction indicatrice. Pour ℓ petit, on peut s'attendre à ce que

$$C(\ell) \sim \ell^{\dim_{\text{corr}}},$$

donnant ainsi une **dimension de corrélation** \dim_{corr} qui est souvent proche de la dimension fractale obtenue par box-counting. Cette méthode offre l'avantage de ne pas nécessiter un recouvrement explicite du graphe et s'appuie directement sur les distances entre paires de nœuds.

2. Analyse des Lois de Puissance

L'observation empirique des **lois de puissance** dans la distribution de certaines variables du réseau est également un indicateur d'auto-similarité. Par exemple, si l'on note $P(s)$ la probabilité qu'un cluster ait une taille s et que

$$P(s) \sim s^{-\alpha},$$

alors le réseau présente un comportement **scale-free**. De même, la distribution des degrés $\deg(i) = \sum_j \omega_{i,j}$ peut suivre une loi de puissance, ce qui est typique des réseaux fractals. Ces observations statistiques renforcent l'idée que, quelle que soit l'échelle, la structure du réseau ne présente pas d'échelle caractéristique, mais obéit à une dynamique de type "rich get richer".

D. Implications et Conditions pour l'Émergence de l'Aspect Fractal

Pour qu'un SCN présente un aspect fractal, il faut que la dynamique d'auto-organisation (le processus DSL) n'introduise pas d'échelle artificielle dans la distribution des pondérations. Cela signifie que les paramètres de mise à jour, tels que le taux d'apprentissage η et le coefficient de décroissance τ , doivent être réglés de manière à favoriser un comportement "échelle-libre". Dans un tel régime, les règles d'agrégation (renforcement des liens forts et inhibition des liens faibles) s'appliquent de manière uniforme et génèrent des distributions de type power law pour des indicateurs tels que :

- La **force interne** d'un cluster, $\Omega(\mathcal{C}) = \sum_{i,j \in \mathcal{C}} \omega_{i,j}$,
- Le **degré** d'un nœud, $\deg(i) = \sum_j \omega_{i,j}$,
- La **taille des clusters**, mesurée en nombre d'entités.

Si, lors de l'agrégation des clusters, on observe que ces distributions restent invariantes à un facteur multiplicatif près, c'est-à-dire que pour une échelle de regroupement donnée, la forme de la distribution reste la même, on peut conclure à une invariance d'échelle dans le SCN.

L'apparition d'un cut-off naturel dans la distribution, c'est-à-dire une borne inférieure x_{\min} et une borne supérieure x_{\max} sur lesquelles la loi de puissance est valide, est également attendue dans des systèmes réels. Cela indique que la fractalité est présente sur plusieurs ordres de grandeur, même si elle n'est pas globale.

6.3.3. Modélisation et Indicateurs

Pour passer du **concept** d'auto-similarité (6.3.1–6.3.2) à une **quantification** de la fractalité dans un SCN (Synergistic Connection Network), on recourt à des **indicateurs** formels. Parmi eux, l'**approche par boîtes** (aussi appelée *box-counting*) est l'une des plus répandues pour estimer la **dimension fractale**. Elle s'adapte assez bien à un réseau (ou graphe), à condition de définir un **critère** de recouvrement en "boîtes" (ou "blocs") pertinent.

6.3.2.3. Mesures Fractales Possibles (Dimension Fractale d'un Graphe, etc.)

Dans cette section, nous exposons de manière détaillée et rigoureuse le cadre mathématique permettant de quantifier l'**auto-similarité** dans un SCN (Synergistic Connection Network). L'objectif est de mettre en œuvre des méthodes issues de la géométrie fractale – telles que la méthode du **box-counting**, la dimension de corrélation et l'analyse des lois de puissance – afin de caractériser la structure auto-similaire d'un réseau synergétique. Cette démarche s'inscrit dans le contexte du **Deep Synergy Learning** (DSL) et vise à démontrer que, si la distribution des pondérations ou des tailles de clusters suit une loi de puissance, alors le réseau présente une invariance d'échelle, qui est la marque d'un comportement fractal.

A. Définitions Fondamentales et Cadre Conceptuel

Soit un ensemble d'entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ constituant un SCN, où chaque paire d'entités (i, j) est reliée par une pondération $\omega_{i,j}$ qui reflète le degré de **synergie** entre elles. Pour appliquer une méthode

de **box-counting** à ce réseau, il est indispensable de définir une **métrie** sur l'ensemble des nœuds. Une approche naturelle consiste à introduire une fonction κ qui associe à chaque lien une distance « locale » inversée, c'est-à-dire que l'on définit :

$$\kappa(\omega_{i,j}) = \frac{1}{\omega_{i,j}},$$

ce qui traduit l'intuition selon laquelle un lien fort (c'est-à-dire un $\omega_{i,j}$ élevé) correspond à une distance courte entre les entités. Par conséquent, la **distance** entre deux nœuds i et j peut être définie par la longueur du plus court chemin reliant ces deux nœuds :

$$d(i,j) = \min_{\gamma(i \rightarrow j)} \sum_{(\mu,v) \in \gamma} \frac{1}{\omega_{\mu,v}},$$

où $\gamma(i \rightarrow j)$ désigne un chemin reliant i à j dans le graphe du SCN. Cette distance $d(i,j)$ remplit le rôle de métrique dans notre approche et nous permet de transposer l'idée de couverture par des boîtes dans le contexte d'un graphe.

Pour une échelle donnée $\ell > 0$, nous définissons une **boîte** (ou une boule) centrée sur un nœud c par :

$$B(c, \ell) = \{j \in \{1, \dots, n\} \mid d(c, j) \leq \ell\}.$$

Le problème consiste alors à déterminer le **nombre minimal** de ces boîtes, noté $N(\ell)$, qui permet de recouvrir l'ensemble des nœuds du SCN. Ce nombre, en fonction de l'échelle ℓ , joue un rôle analogue au nombre de cubes nécessaires pour recouvrir un objet géométrique dans \mathbb{R}^d .

B. Procédure du Box-Counting et Estimation de la Dimension Fractale

La méthode du **box-counting** adaptée à un SCN s'articule autour de plusieurs étapes essentielles. D'une part, il faut établir une couverture du graphe par des boules de rayon ℓ . Pour une valeur d'échelle fixée, on choisit successivement des nœuds centraux, construit les boules correspondantes $B(c, \ell)$ et retire les nœuds ainsi couverts du graphe jusqu'à ce que tous les nœuds soient inclus dans au moins une boule. Le nombre minimal de boules obtenues dans cette procédure est noté $N(\ell)$.

Si, lorsque l'échelle ℓ tend vers zéro, on observe que

$$N(\ell) \sim \ell^{-\dim_f},$$

la **dimension fractale** du SCN est définie par la relation :

$$\dim_f = \lim_{\ell \rightarrow 0} \frac{\ln N(\ell)}{-\ln \ell}.$$

En pratique, on trace la courbe logarithmique $\ln N(\ell)$ en fonction de $\ln(1/\ell)$ et la pente de la droite de régression (sur un intervalle $[\ell_{\min}, \ell_{\max}]$) fournit une estimation de \dim_f . La linéarité de cette courbe dans cet intervalle est le témoignage de l'**invariance d'échelle** qui caractérise la structure fractale.

Il est à noter que la méthode du box-counting peut être complétée par d'autres approches, telles que la **dimension de corrélation**, où l'on définit la fonction

$$C(\ell) = \frac{2}{n(n-1)} \sum_{i < j} \chi(d(i, j) \leq \ell),$$

avec χ la fonction indicatrice, et l'on examine la dépendance de $C(\ell)$ en fonction de ℓ pour estimer la dimension de corrélation \dim_{corr} par la relation $C(\ell) \sim \ell^{\dim_{\text{corr}}}$. De même, l'analyse des **lois de puissance** dans la distribution des tailles de clusters ou des degrés des nœuds permet de détecter une invariance d'échelle. Par exemple, si la distribution des tailles de clusters satisfait

$$\text{Prob}(|\mathcal{C}| > s) \sim s^{-\alpha},$$

on identifie une caractéristique scale-free, qui renforce l'hypothèse d'auto-similarité dans l'agrégation des clusters.

C. Implications pour le Deep Synergy Learning (DSL)

L'application de ces méthodes fractales dans le cadre d'un SCN, et plus particulièrement du DSL, offre plusieurs avantages importants. Lorsque la même dynamique d'auto-organisation, exprimée par la règle de mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

s'exerce à toutes les échelles, on obtient une auto-similarité qui se traduit par une invariance de la distribution des pondérations et des tailles de clusters. Cette invariance permet de simplifier la modélisation multi-échelle, car une même loi (ou un même algorithme) peut être utilisée pour décrire les interactions tant au niveau local que global. De plus, la robustesse du système est renforcée puisque l'auto-organisation, en se répétant à différentes granularités, tend à corriger les perturbations locales par une réorganisation à plus grande échelle. La dimension fractale \dim_f quantifiée par la méthode du box-counting constitue ainsi un indicateur précieux pour l'analyse du comportement global du SCN et pour le réglage des paramètres du DSL.

6.3.3.2. Représentation d'échelles multiples : la “scale invariance” si la structure se répète

Dans le cadre de l'analyse des **réseaux synergétiques**, il apparaît qu'un même **motif** d'organisation peut se reproduire à différentes échelles, ce que l'on désigne par le terme d'**invariance d'échelle** ou **scale invariance**. Cette propriété, intrinsèque aux systèmes fractals, se manifeste dans un SCN (Synergistic Connection Network) lorsque la même répartition des liaisons, la même configuration topologique, se retrouve de manière auto-similaire que ce soit au niveau local (micro-clusters) ou global (macro-clusters). L'objectif de cette section est de présenter rigoureusement comment la représentation d'échelles multiples permet de quantifier et d'exploiter cette invariance, et d'en démontrer les implications pour la dynamique du DSL (Deep Synergy Learning).

A. Représentation Multiscale du SCN

Considérons un SCN composé d'un ensemble d'entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ reliées par des pondérations $\omega_{i,j}$. La dynamique d'auto-organisation, régie par une règle de mise à jour telle que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

s'exerce uniformément sur le réseau. Dès lors, il est naturel de considérer que, par des procédés d'**agrégation hiérarchique** ou de *coarse-graining* (voir notamment la section 6.2.2), le SCN peut être représenté à plusieurs niveaux de granularité. Ainsi, on introduit une suite d'agrégations

$$\omega^{(0)} \rightarrow \omega^{(1)} \rightarrow \omega^{(2)} \rightarrow \dots \rightarrow \omega^{(K)},$$

où $\omega^{(0)}$ représente la matrice de pondérations du réseau initial (au niveau micro) et $\omega^{(K)}$ correspond à celle obtenue après K niveaux d'agrégation, chacun étant obtenu par une opération de regroupement des entités en « super-nœuds ». À chaque niveau, la même dynamique d'auto-organisation est appliquée aux agrégats, de sorte que la **structure** (la répartition des liens et la topologie) reste, à un facteur d'échelle près, identique à celle du niveau précédent.

Pour formaliser cette idée, on suppose qu'il existe un **isomorphisme approché** ϕ et un **facteur de rescaling** $\lambda > 0$ tels que, pour tout couple d'entités i, j appartenant à un cluster au niveau ℓ , la relation

$$\omega_{\phi(i), \phi(j)}^{(\ell+1)} \approx \lambda \omega_{i,j}^{(\ell)}$$

soit vérifiée. Cette relation exprime que la *même* forme structurelle est retrouvée au niveau $\ell + 1$ (macro-niveau) à partir du niveau ℓ (micro-niveau) une fois les pondérations multipliées par un facteur d'échelle constant. En d'autres termes, la configuration topologique du SCN se reproduit de manière **auto-similaire** à travers les niveaux d'agrégation.

B. Détection et Quantification de la Scale Invariance

L'approche par boîtes (ou **box-counting**), présentée en 6.3.3.1, offre un cadre permettant de mesurer la **dimension fractale** du SCN. Dans un contexte multiscale, cette méthode s'applique non seulement au niveau initial mais également sur les versions agrégées du réseau. En recouvrant le réseau par des « boîtes » ou boules de rayon ℓ (définies via une métrique adaptée $d(i, j)$), par exemple $d(i, j) = \min_{\gamma(i \rightarrow j)} \sum_{(\mu, \nu) \in \gamma} \frac{1}{\omega_{\mu, \nu}}$, on détermine le nombre minimal $N(\ell)$ de boîtes nécessaires pour couvrir l'ensemble des entités. Si l'on trouve que

$$N(\ell) \sim \ell^{-\dim_f},$$

alors la pente de la droite obtenue en traçant $\ln N(\ell)$ en fonction de $\ln(1/\ell)$ donne la **dimension fractale** \dim_f du réseau. Le fait que cette relation reste valable à différents niveaux d'agrégation atteste de l'**invariance d'échelle** du SCN. De surcroît, une analyse statistique complémentaire basée sur la distribution en **lois de puissance** des tailles de clusters ou des degrés des nœuds renforce l'idée que le même schéma organisationnel se retrouve quelle que soit l'échelle.

La **dimension de corrélation**, obtenue par l'analyse de la fonction de corrélation

$$C(\ell) = \frac{2}{n(n-1)} \sum_{i < j} \chi(d(i, j) \leq \ell),$$

est une autre méthode qui permet d'évaluer l'auto-similarité du réseau. Si $C(\ell)$ satisfait une relation de la forme

$$C(\ell) \sim \ell^{\dim_{\text{corr}}},$$

alors \dim_{corr} fournit une estimation alternative de la dimension fractale. Ces approches, en s'appliquant de manière répétée sur les différentes couches d'agrégation du SCN, permettent de vérifier que la **structure** du réseau est effectivement **scale-invariant**.

C. Implications pour le Deep Synergy Learning (DSL)

L'existence d'une invariance d'échelle dans un SCN signifie que la même **règle d'auto-organisation** peut être appliquée de manière itérative à différents niveaux, du micro au macro, sans nécessiter d'ajustements spécifiques pour chaque palier. Si le mécanisme DSL (par exemple, la règle additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

) est la même à toutes les échelles, alors un micro-cluster se structure de la même manière qu'un macro-cluster, à un facteur de rescaling près. Cette **homogénéité** du mécanisme d'auto-organisation assure une **robustesse** intrinsèque, car toute perturbation locale peut être compensée par la réapparition du même motif à une échelle supérieure. Cette propriété fractale simplifie l'analyse du comportement multi-échelle du système, car la connaissance de la dynamique à une échelle permet de prédire le comportement à d'autres échelles grâce à des relations de type

$$\omega_{\phi(i),\phi(j)}^{(\ell+1)} \approx \lambda \omega_{i,j}^{(\ell)}.$$

Par ailleurs, l'obtention d'une dimension fractale non entière (par exemple, $\dim_f = 1.5$ ou 2.3) offre un indicateur quantitatif du degré de complexité et d'irrégularité du réseau. Cette mesure peut être utilisée pour adapter dynamiquement les paramètres du DSL, afin de maintenir la stabilité et la cohérence des structures émergentes.

6.3.3.3. Exemples de phénomènes fractals dans la nature (p. ex. réseaux vasculaires) et analogies avec le DSL

La compréhension des **phénomènes fractals** dans la nature repose sur l'observation que la structure des systèmes biologiques et géophysiques se répète à différentes échelles, de manière hiérarchique et auto-similaire. Cette propriété d'**invariance d'échelle** se manifeste notamment dans les **réseaux vasculaires**, les **ramifications bronchiques** et les **systèmes racinaires**. Ces exemples offrent des analogies riches avec le **Deep Synergy Learning (DSL)** lorsque celui-ci se déploie en mode multi-niveau, permettant d'illustrer que la même dynamique d'auto-organisation peut s'appliquer tant au niveau micro qu'au niveau macro, assurant une cohérence structurelle du réseau.

A. Réseaux Vasculaires et Bronchiques : Une Ramification Auto-similaire



Dans le domaine biologique, le système cardiovasculaire présente une organisation fractale remarquable. L'**aorte** se subdivise en artères, lesquelles se transforment en artérioles, avant de se ramifier en capillaires. À chaque palier de cette hiérarchie, la même **loi de bifurcation** se reproduit de manière quasi-identique, ce qui se traduit mathématiquement par des relations telles que celles observées dans les modèles de **Murray** ou les travaux de **West-Brown-Enquist**. Par exemple, une loi de conservation typique dans ces systèmes se formalise par

$$r_{\text{parent}}^3 \approx r_{\text{enfant}_1}^3 + r_{\text{enfant}_2}^3,$$

ce qui implique qu'à chaque bifurcation, le diamètre des vaisseaux décroît d'un facteur constant, et que la distribution des rayons suit une loi de puissance. L'auto-similarité dans ce contexte se manifeste par la répétition d'un même motif de subdivision, de sorte que la structure globale demeure invariante d'échelle, à l'exception d'un facteur multiplicatif constant. Dans un **SCN fractal**, les pondérations $\omega_{i,j}$ peuvent être interprétées comme des "flux" ou intensités de connexion, et si la même règle de mise à jour (renforcement/inhibition) s'applique à chaque niveau, on observe que la structure d'un petit groupe de nœuds (micro-cluster) se retrouve, à un facteur d'échelle près, dans le réseau global (macro-cluster). Cette correspondance est formellement exprimée par la relation

$$\omega_{\phi(i),\phi(j)}^{(\text{glob})} \approx \lambda \omega_{i,j}^{(\text{loc})},$$

où ϕ désigne une application de correspondance entre les nœuds du micro-cluster et ceux du macro-cluster, et λ représente le facteur de rescaling. L'existence d'un tel facteur souligne l'**invariance d'échelle** dans le réseau, analogue à la manière dont le système vasculaire conserve sa forme malgré la diminution progressive des diamètres des vaisseaux.

B. Systèmes Racinaires et Réseaux de Rivières

Les **systèmes racinaires** constituent un autre exemple marquant d'auto-similarité dans la nature. La racine principale se divise en racines secondaires, lesquelles se subdivisent à leur tour en radicelles. Mathématiquement, ce processus de division peut être modélisé par une relation de type

$$L_{\text{tot}} \sim \sum_{k=1}^m r_k^{-\beta},$$

où r_k représente le diamètre des branches à chaque niveau, et l'exposant β caractérise l'agencement fractal du système. Une structure similaire se retrouve dans les **réseaux de rivières**. Ces derniers sont organisés selon des règles telles que celles de **Horton-Strahler** ou de **Tokunaga**, qui décrivent comment chaque sous-affluent se structure de manière récurrente par rapport au cours d'eau principal. Le petit cours d'eau possède la même topologie que la rivière principale, mis à part un facteur d'échelle en termes de longueur et de débit. Cette récurrence topologique signifie que l'auto-organisation observée dans un SCN, par le biais de la mise à jour des pondérations, est comparable à ces processus naturels où l'agrégation et la division se font de manière régulière et prévisible.

C. Analogies Essentielles avec le DSL

L'analogie entre les phénomènes fractals naturels et le **Deep Synergy Learning** se fonde sur plusieurs aspects fondamentaux. D'une part, les **réseaux vasculaires** ou **systèmes racinaires** présentent une **absence d'échelle caractéristique** ; toutes les échelles participent de manière équivalente à la formation de la structure globale. De même, dans un SCN où la dynamique DSL ne privilégie aucune échelle – c'est-à-dire qu'il n'existe pas de seuil arbitraire imposant une taille fixe pour un cluster – la distribution des pondérations et la formation des clusters suivent une **loi de puissance**. D'autre part, l'**auto-organisation** dans ces systèmes se caractérise par une **hiérarchie récurrente**. Un petit groupe local qui se structure selon un certain motif se retrouve, lorsqu'on l'agrége, à l'échelle macro, en reproduisant le même schéma, comme si l'ensemble du réseau était le reflet d'une même dynamique, simplement dilatée par un facteur constant. Cette **self-similarity** se traduit dans le DSL par l'application uniforme de la règle d'actualisation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

à toutes les échelles, ce qui garantit que le même motif de renforcement et d'inhibition apparaît du micro au macro. Par ailleurs, la **robustesse** inhérente à une structure fractale – où l'absence d'un point critique unique permet à l'ensemble du système de se réorganiser même en cas de perturbation locale – se transpose directement dans le DSL, rendant le modèle plus résilient face aux variations et aux imprévus.

6.3.4. Avantages et Limitations

Après avoir exploré les principes de la fractalité (6.3.1 et 6.3.2) et comment la *quantifier* (6.3.3), il convient d'évaluer ce que **signifie** concrètement, pour un **SCN** (Synergistic Connection Network), d'être (ou de tendre vers) un **système fractal**. La fractalité n'est pas une finalité **systématique** dans le DSL (Deep Synergy Learning), mais plutôt un **cas particulier** susceptible d'offrir certains avantages tout en présentant des limites.

6.3.4.1. **Avantage : la fractalité suggère une “unité” ou une cohérence à toutes les échelles**

Dans l’étude des réseaux complexes, la notion de **fractalité** apparaît comme une caractéristique essentielle, notamment parce qu’elle suggère l’absence d’une taille caractéristique imposée aux **clusters** ou aux **liens**. Cette propriété d’**invariance d’échelle** se manifeste lorsque, quelle que soit la granularité d’observation – du niveau **micro** jusqu’au niveau **macro** – les motifs d’organisation se répètent, de manière auto-similaire, à un facteur d’échelle près. En d’autres termes, la même structure de distribution des **pondérations** $\omega_{i,j}$ et la même logique d’auto-organisation se retrouvent, que l’on “zoome” sur un petit sous-groupe ou que l’on “dézoome” sur l’ensemble du réseau. Ce phénomène, qui constitue l’un des piliers théoriques de la modélisation fractale, présente plusieurs avantages tant sur le plan mathématique que sur le plan opérationnel dans le cadre du **Deep Synergy Learning (DSL)**.

Soit un réseau synergétique $G = (V, E)$ où chaque nœud représente une entité \mathcal{E}_i et chaque arête (i, j) est associée à une pondération $\omega_{i,j}$ qui reflète le degré de **synergie** entre les entités i et j . La dynamique d’auto-organisation dans le DSL est gouvernée par une règle de mise à jour, souvent exprimée sous la forme

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d’apprentissage** et τ est le **coefficient de décroissance**. Lorsque cette même règle est appliquée de manière uniforme à toutes les échelles du réseau, elle produit des motifs récurrents, ce qui conduit à une structure fractale. En effet, dans un tel réseau, la distribution des **clusters** ne présente pas de taille caractéristique ; au contraire, un *micro-cluster* local peut présenter exactement la même topologie qu’un *macro-cluster*, à un facteur d’échelle près. Formellement, on peut exprimer cette idée par l’existence d’un isomorphisme approché ϕ et d’un facteur $\lambda > 0$ tels que, pour tout i, j appartenant à un cluster local \mathcal{C}_{loc} , la relation suivante est satisfaite :

$$\omega_{\phi(i), \phi(j)}^{(\text{glob})} \approx \lambda \omega_{i,j}^{(\text{loc})}.$$

Cette équation illustre que, si l’on agrège les entités du niveau micro pour obtenir un niveau macro, la *forme* des liaisons se conserve, seule l’échelle change.

Du point de vue **mathématique**, cette invariance d’échelle se manifeste par la présence de **lois de puissance** dans la distribution des tailles de clusters ou des degrés des nœuds. Par exemple, si l’on note $s = |\mathcal{C}|$ la taille d’un cluster, on peut observer que la probabilité qu’un cluster dépasse une taille s se comporte selon

$$\text{Prob}(s > x) \sim x^{-\alpha},$$

ce qui indique que le même mécanisme d’agrégation s’applique indépendamment de l’échelle considérée. De même, une analyse via la méthode du **box-counting** (voir la section 6.3.3.1) permet d’extraire une dimension fractale dim_f du réseau en montrant que

$$N(\ell) \sim \ell^{-\text{dim}_f},$$

où $N(\ell)$ représente le nombre minimal de boîtes de rayon ℓ nécessaires pour recouvrir l’ensemble des nœuds. La constance de dim_f sur un intervalle d’échelles indique que le réseau est **auto-similaire**.

D'un point de vue **ingénierique**, le fait que la même logique d'auto-organisation (les mêmes paramètres η et τ dans la dynamique DSL, ainsi que les mêmes mécanismes d'inhibition ou de saturation) se répète de manière identique du niveau micro au niveau macro confère une **unité** au système. Cette **cohérence** structurelle se traduit par une simplification de l'implémentation, puisqu'un unique ensemble de règles s'applique à toute la hiérarchie. Par ailleurs, en cas de perturbation locale, le système peut se reconfigurer en s'appuyant sur les mêmes schémas fractals à d'autres niveaux, renforçant ainsi sa **robustesse** et son **adaptabilité**. La capacité à analyser un petit sous-ensemble du réseau et à extrapoler ses propriétés à l'ensemble du SCN constitue un avantage considérable pour l'analyse et la conception des systèmes d'auto-organisation.

En somme, la fractalité suggère que le réseau, quelle que soit son échelle d'observation, présente une **unité** intrinsèque ; la **même forme** de distribution des liens et la même logique d'auto-organisation se retrouvent aussi bien dans un micro-cluster que dans un macro-cluster, modulo un simple changement d'échelle par un facteur λ . Cette propriété renforce la **cohérence** du DSL et facilite l'analyse multi-échelle, tout en simplifiant la mise en œuvre et le réglage des paramètres du système. L'universalité de la dynamique DSL, qui se répète à tous les niveaux, est ainsi l'une des raisons majeures pour lesquelles les réseaux fractals sont particulièrement recherchés dans la modélisation des systèmes complexes et auto-organisés.

6.3.4.2. Limitation : tous les systèmes DSL ne sont pas fractals ; c'est un cas particulier où les synergies multi-niveau se recoupent fortement

Dans l'étude des **réseaux synergétiques** et du **Deep Synergy Learning (DSL)**, il est tentant d'identifier des structures fractales lorsque la même dynamique d'auto-organisation se répète à différentes échelles. Toutefois, il apparaît que l'émergence d'un comportement fractal n'est pas universelle au sein des systèmes DSL, mais se manifeste seulement dans des conditions particulières où les synergies multi-niveau s'alignent de façon très homogène. Cette limitation doit être examinée à la lumière de plusieurs aspects théoriques et pratiques, et il est essentiel d'en préciser les conditions, afin de ne pas extrapoler de manière abusive l'idée de fractalité à l'ensemble des systèmes DSL.

Une première condition requise pour qu'un SCN puisse exhiber une structure fractale est que la **fonction d'agrégation** utilisée dans la dynamique d'auto-organisation, souvent notée Ψ dans le cadre des opérations de coarse-graining (cf. section 6.2.2), soit appliquée de manière quasi uniforme à tous les niveaux. Il faut que la règle d'actualisation des pondérations, typiquement donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

reçoive exactement le même traitement que l'on applique à des sous-ensembles du réseau, de sorte que la distribution des valeurs $\omega_{i,j}$ à chaque niveau d'agrégation soit comparable, modulo un simple facteur d'échelle. Dès lors, si les **paramètres** η et τ ne varient pas ou varient proportionnellement d'un niveau à l'autre, la dynamique risque de se maintenir, produisant ainsi une auto-similarité au sens strict. En revanche, si ces paramètres sont ajustés indépendamment à chaque palier – par exemple, en introduisant un taux d'apprentissage local η_{loc} distinct d'un taux global η_{glob} – l'uniformité de la règle d'auto-organisation est rompue, et le réseau ne peut plus présenter la récurrence des motifs caractéristiques d'un système fractal.

Un deuxième facteur réside dans la **nature des données** et la **distribution initiale** des synergies. Dans de nombreux systèmes DSL, les entités sont hétérogènes, et la répartition des pondérations $\omega_{i,j}$ peut être influencée par des contraintes externes ou par des mécanismes de saturation qui imposent une échelle caractéristique. Par exemple, lorsqu'un seuil de saturation est appliqué pour éviter que les pondérations ne croissent indéfiniment, ou lorsque des règles spécifiques de formation des clusters induisent une taille moyenne prédéfinie, la distribution des $\omega_{i,j}$ ne suit plus une loi de puissance pure. Cette rupture de la loi de puissance empêche l'apparition d'un comportement véritablement **scale-free**, et par conséquent, la fractalité du système ne s'exprime que de manière partielle ou localisée.

Un troisième aspect concerne la **diversité topologique** inhérente au réseau. Même si l'on applique une règle DSL uniforme, la topologie du SCN peut être influencée par des facteurs externes ou par des propriétés intrinsèques des interactions entre entités.

Par exemple, dans un système multimodal, certains sous-groupes d'entités se formeront naturellement selon des motifs distincts en fonction de leurs caractéristiques spécifiques, tandis que d'autres adopteront une organisation plus homogène. Cette hétérogénéité dans l'**organisation** empêche la réapparition d'un motif unique à toutes les échelles, et seule une partie du réseau pourra être qualifiée de fractale, tandis que d'autres portions présenteront des distributions plus ordinaires.

En résumé, la fractalité dans un SCN n'émerge que dans le cas particulier où (i) la même **fonction d'agrégation** Ψ est appliquée de manière uniforme à tous les niveaux, (ii) les paramètres d'actualisation η et τ restent constants ou sont ajustés de manière proportionnelle, et (iii) la distribution initiale des synergies et des données est suffisamment homogène pour ne pas introduire de coupures d'échelle.

Lorsque ces conditions ne sont pas réunies – par exemple, en présence de **paramètres divergents** entre les niveaux, de contraintes de saturation strictes ou de données hétérogènes – le SCN ne présente pas de comportement fractal global, mais seulement une **fractalité partielle** ou même l'absence d'auto-similarité.

Cette limitation est essentielle à comprendre, car elle indique que l'apparition d'une **structure fractale** dans un système DSL est un cas particulier, illustrant l'unité et la robustesse de la dynamique d'auto-organisation, mais qui n'est pas généralisable à tous les réseaux synergétiques.

D'un point de vue théorique, cela signifie que les outils mathématiques permettant de quantifier la fractalité, tels que le box-counting, la dimension de corrélation ou l'analyse des lois de puissance, ne s'appliquent que lorsque le réseau ne présente pas de contraintes d'échelle imposées par la nature des données ou par les mécanismes d'inhibition et de saturation.

D'un point de vue opérationnel, il convient de reconnaître que, dans la pratique, de nombreux systèmes DSL ne sont que partiellement fractals et que leur analyse doit intégrer cette hétérogénéité afin d'éviter des interprétations trop simplistes de la dynamique d'auto-organisation.

6.3.4.3. Applicabilité : ex. grands réseaux hétérogènes, IA inspirée du cerveau

Dans le contexte des systèmes complexes, l'idée d'**invariance d'échelle** se retrouve naturellement dans de nombreux phénomènes naturels et inspire la modélisation de systèmes artificiels tels que

le **Deep Synergy Learning (DSL)**. Cette section examine de manière approfondie l'applicabilité des concepts fractals à de grands réseaux hétérogènes et à des approches d'**IA inspirée du cerveau**, en mettant en évidence les avantages mais également les limites de l'hypothèse d'auto-similarité dans ces domaines.

A. Grands Réseaux Hétérogènes

Dans des environnements comportant des millions d'entités et une multiplicité de types de relations – par exemple dans les **réseaux sociaux** ou les systèmes multi-domaines – aucune taille de cluster n'est imposée a priori. En l'absence de contraintes externes fortes, la dynamique d'auto-organisation, pilotée par une règle DSL uniforme, permet l'émergence de clusters de toutes tailles. D'un point de vue mathématique, cette liberté se traduit souvent par des distributions de taille de clusters qui suivent une **loi de puissance** :

$$\text{Prob}(|\mathcal{C}| > s) \sim s^{-\alpha},$$

ce qui signifie qu'il n'existe pas de taille caractéristique dominante dans le réseau. L'**absence de scale** se reflète ainsi dans une distribution continue et heavy-tail des agrégations, indiquant que le même mécanisme d'agrégation – notamment le renforcement des liens forts et l'inhibition des liens faibles – opère de manière homogène sur toutes les échelles. Par conséquent, dans ces grands réseaux hétérogènes, le DSL, lorsqu'il est laissé relativement libre de ses paramètres, peut générer ou révéler une **structure fractale**. Cette propriété assure une **robustesse** et une **flexibilité** accrues de l'organisation. L'ajout ou la suppression de nœuds ne perturbe pas l'équilibre global, car la loi de puissance reste valide, permettant ainsi une auto-organisation stable sans point critique fixe.

B. IA Inspirée du Cerveau

La biologie neuronale offre un exemple emblématique d'**organisation multi-échelle**. Le cerveau humain, par exemple, présente une structure hiérarchique allant des micro-colonnes corticales aux aires fonctionnelles et jusqu'aux lobes. Des études empiriques suggèrent que les connexions neuronales, tant au niveau des synapses que dans la configuration des réseaux neuronaux, peuvent exhiber des propriétés d'auto-similarité et d'invariance d'échelle. Mathématiquement, de telles structures se caractérisent par des mesures comme la **dimension fractale** ou par des lois de puissance dans la distribution des degrés de connexion, lesquelles indiquent que les motifs de connectivité se répètent à toutes les échelles.

Dans une approche DSL inspirée du cerveau, la règle d'auto-organisation – par exemple

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

– s'applique de manière uniforme du niveau micro (groupes de neurones ou micro-colonnes) jusqu'au niveau macro (aires corticales interconnectées). La réutilisation des mêmes mécanismes d'apprentissage et de plasticité synaptique, associés à des schémas de connectivité auto-similaire, confère au réseau une **robustesse** comparable à celle observée dans le cortex. Un tel système permet, par analogie avec les réseaux vasculaires ou bronchiques, de maintenir une cohérence structurelle malgré l'hétérogénéité des modules fonctionnels, facilitant ainsi la transmission d'informations et l'adaptation aux perturbations.

C. Synthèse et Implications

La fractalité, lorsqu'elle se manifeste dans un SCN, offre un cadre conceptuel puissant pour l'analyse de grands réseaux hétérogènes ainsi que pour la conception d'architectures d'**IA inspirée du cerveau**. Dans ces domaines, l'absence d'une taille caractéristique imposée aux clusters – traduite mathématiquement par des lois de puissance telles que

$$\text{Prob}(|\mathcal{C}| > s) \sim s^{-\alpha}$$

– et la récurrence de la même dynamique d'auto-organisation à toutes les échelles, permettent d'envisager un modèle robuste et évolutif. L'universalité de la règle DSL assure que, même en présence d'une hétérogénéité extrême, les motifs d'interconnexion se reproduisent de façon auto-similaire, garantissant une répartition efficace des ressources et une coordination harmonieuse. Toutefois, il convient de noter que la fractalité est souvent un cas particulier qui se révèle pleinement lorsque les conditions d'homogénéité des paramètres et d'absence de contraintes externes prédominantes sont réunies. Dans de nombreux systèmes réels, des facteurs tels que des seuils de saturation ou des différences de paramètres entre les niveaux peuvent limiter l'étendue de cette invariance d'échelle. Néanmoins, l'inspiration tirée des structures fractales observées dans la nature – des réseaux vasculaires aux systèmes racinaires – offre des perspectives prometteuses pour concevoir des SCN capables de s'auto-organiser de manière à la fois flexible et résiliente, en reproduisant des motifs d'organisation qui transcendent les échelles d'observation.

6.4. Interactions Multi-Niveau et Coordination

Le **DSL** (Deep Synergy Learning) opère de manière particulièrement riche lorsqu’il est déployé sur un **SCN** (Synergistic Connection Network) organisé en **plusieurs niveaux** (micro, méso, macro). Dans ce cadre, les **flux** ascendants (bottom-up) et descendants (top-down) assurent une **coordination** essentielle entre les niveaux, les entités (ou petits clusters) **remontent** leur organisation vers les paliers supérieurs, tandis que les paliers supérieurs **rétroagissent** pour influencer ou “piloter” la configuration des liens au palier inférieur. La section 6.4.1 récapitule ces deux flux de base et souligne l’importance de la cohérence entre niveaux.

6.4.1. Bottom-Up vs. Top-Down

Dans un **SCN** multi-niveau, on distingue au moins deux grands axes de **communication** :

- **Flux ascendants (bottom-up)** : depuis les entités brutes (niveau micro), on agrège progressivement des clusters vers un niveau macro (ou intermédiaire).
- **Flux descendants (top-down)** : le macro-niveau, disposant d’une vue plus globale, impose un **feedback** (contrainte ou renforcement) sur les pondérations ω au niveau inférieur.

6.4.1.1. Rappel des flux ascendants (entités \rightarrow cluster local \rightarrow cluster macro)

Les **flux ascendants** dans un **SCN** (Synergistic Connection Network) multi-échelle désignent le mécanisme par lequel on **construit** progressivement, à partir d’entités élémentaires au niveau micro, des **clusters** toujours plus vastes au fil des niveaux supérieurs, jusqu’à atteindre un niveau macro. Cette démarche correspond à un processus de *coarse-graining* itératif, où l’on consolide des groupes de bas niveau en “super-nœuds” plus globaux, optimisant ainsi la représentation et la gestion de la complexité.

On considère un ensemble d’entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$, il s’agit du **niveau 0** ou micro-niveau. Les **pondérations** $\omega_{i,j}^{(0)}$ décrivent les affinités (ou synergies) entre ces entités. Dans un **DSL** (Deep Synergy Learning), ces pondérations évoluent selon une règle itérative, par exemple de type additive :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_0 \left[S_0(i,j) - \tau_0 \omega_{i,j}^{(0)}(t) \right],$$

avec η_0, τ_0 pour le niveau micro. À un instant donné, certains groupes d’entités peuvent présenter une cohésion interne suffisamment élevée (somme des $\omega_{i,j}$ importante).

Dès lors que l’on détecte un **cluster** $\mathcal{C}_\alpha^{(0)}$ (quelques entités fortement reliées), on mesure par exemple :

$$\Omega(\mathcal{C}_\alpha^{(0)}) = \sum_{i,j \in \mathcal{C}_\alpha^{(0)}} \omega_{i,j}^{(0)}.$$

Un **seuil** θ_0 (cohésion minimale) permet de décider si ce regroupement est *actif*. Chaque cluster local $\mathcal{C}_\alpha^{(0)}$ acquiert une identité propre, possiblement appelée “super-nœud de niveau 1”, notée $\mathcal{N}_\alpha^{(1)}$.

Une fois qu’on a constitué plusieurs clusters $\{\mathcal{C}_1^{(0)}, \dots, \mathcal{C}_r^{(0)}\}$ au niveau 0, chacun se transforme en **super-nœud** $\{\mathcal{N}_1^{(1)}, \dots, \mathcal{N}_r^{(1)}\}$. Les pondérations entre ces super-nœuds s’obtiennent via une **fonction** d’agrégation Ψ appliquée aux $\omega_{i,j}^{(0)}$ des entités qu’ils contiennent :

$$\omega_{\alpha,\beta}^{(1)} = \Psi \left(\left\{ \omega_{i,j}^{(0)} \mid i \in \mathcal{C}_\alpha^{(0)}, j \in \mathcal{C}_\beta^{(0)} \right\} \right).$$

Une fois ce niveau 1 établi, on reproduit une **même** dynamique DSL (ou adaptée) pour $\omega_{\alpha,\beta}^{(1)}$. Ainsi, si plusieurs super-nœuds de niveau 1 s’attirent fortement, ils peuvent se fondre en un **cluster** $\mathcal{C}_\gamma^{(1)}$ de taille supérieure, que l’on réifie au niveau 2, et ainsi de suite.

Ce **flux ascendant** se répète itérativement. Au niveau k , on identifie des *clusters* $\{\mathcal{C}_\alpha^{(k)}\}$. Chaque $\mathcal{C}_\alpha^{(k)}$ devient un super-nœud $\mathcal{N}_\alpha^{(k+1)}$. Les pondérations $\omega_{\alpha,\beta}^{(k+1)}$ entre super-nœuds s’agrègent via Ψ . Ainsi, on converge vers des **macro-nœuds** de plus en plus globaux, caractérisant le niveau K (ultime palier, voire unique macro-nœud).

On peut schématiser la montée d’échelle ainsi :

$$\omega^{(k+1)} = \Gamma_k(\omega^{(k)}),$$

où Γ_k agrège les **clusters** de $\omega^{(k)}$. On définit des **clusters** $\mathcal{C}_\alpha^{(k)} \subseteq \{\mathcal{N}_1^{(k)}, \dots\}$ et les pondérations $\omega_{\alpha,\beta}^{(k+1)}$ via

$$\omega_{\alpha,\beta}^{(k+1)} = \Psi \left(\left\{ \omega_{\alpha',\beta'}^{(k)} \mid \alpha' \in \mathcal{C}_\alpha^{(k)}, \beta' \in \mathcal{C}_\beta^{(k)} \right\} \right).$$

Ici, $\mathcal{C}_\alpha^{(k)}$ désigne un regroupement au **niveau** k . Après agrégation, on obtient un ensemble de “super-nœuds” $\{\mathcal{N}_\alpha^{(k+1)}\}$. Si le SCN se **stabilise**, on pourra aboutir à un nombre réduit de super-nœuds (voire un unique macro-nœud).

Ce **flux ascendant** s’accompagne d’un **gain** en :

1) Réduction de la taille

Si au niveau 0 il existe $O(n^2)$ liens, le niveau 1 n’en comporte plus que $O(m^2)$ (si $m \ll n$), libérant une partie de la charge de calcul et facilitant un traitement plus rapide au palier supérieur.

2) Hiérarchie fonctionnelle

Chaque **cluster local** se spécialise (regroupe des entités proches), puis devient un super-nœud dans l'étape supérieure, reflétant un **rôle** plus large. La progression micro→macro donne une **vision** d'ensemble en successive agrégation.

3) Couplage avec le flux descendant

Les macro-nœuds, une fois construits, peuvent influencer sur les règles locales (chap. 6.4.1.2). Ce couplage (bottom-up + top-down) renforce l'auto-organisation hiérarchique.

6.4.1.2. Étude Mathématique Approfondie : Flux Descendants (Cluster Macro → Entités) et Contraintes sur les Pondérations

La construction hiérarchique de *clusters* en flux ascendant (cf. 6.4.1.1) ne suffit pas à elle seule pour assurer la cohérence globale d'un **SCN** (Synergistic Connection Network) à plusieurs niveaux. Il est souvent nécessaire que les **niveaux supérieurs** — où sont agrégés de larges super-nœuds ou clusters macro — puissent **redescendre** des signaux ou contraintes vers les liaisons $\omega_{i,j}$ du niveau inférieur, afin de réorienter ou de stabiliser la dynamique locale. Cette section expose une formulation analytique de ce **flux descendant**, montre comment l'inclure dans l'équation de mise à jour du **DSL** et discute son rôle dans la cohérence globale.

A. Cadre Mathématique et Définition du Flux Descendant

On considère un **SCN** multi-niveau indexé par $k = 0, 1, \dots, K$. Le **niveau 0**, ou micro-niveau, rassemble les **entités élémentaires** $\{\mathcal{E}_i\}$. À chaque palier k , on définit un ensemble de *super-nœuds* $\{\mathcal{N}_\alpha^{(k)}\}$. On note $\omega_{\alpha,\beta}^{(k)}$ les liaisons entre ces super-nœuds au niveau k .

Le **flux descendant** part du niveau k (macro ou intermédiaire) pour imposer un **feedback** sur le niveau $k - 1$. Si $\mathcal{N}_\alpha^{(k)}$ se décompose en un ensemble de super-nœuds du niveau $k - 1$, noté $\mathcal{C}_\alpha^{(k-1)} \subseteq \{\mathcal{N}_\gamma^{(k-1)}\}$, alors le niveau k peut envoyer un **terme** correctif dans la mise à jour des pondérations $\omega_{i,j}^{(k-1)}$.

Dans un DSL purement local, on a généralement

$$\omega_{i,j}^{(k-1)}(t+1) = \omega_{i,j}^{(k-1)}(t) + \eta_{k-1} \left[S_{k-1}(i,j) - \tau_{k-1} \omega_{i,j}^{(k-1)}(t) \right],$$

où η_{k-1}, τ_{k-1} sont des paramètres (taux d'apprentissage, facteur d'oubli) et $S_{k-1}(i,j)$ décrit la synergie locale entre les super-nœuds (ou entités) i, j au niveau $k - 1$.

Pour tenir compte du **flux descendant**, on **rajoute** un terme $\gamma_{td} h_{i,j}^{(k)}(t)$ dans cette équation :

$$\omega_{i,j}^{(k-1)}(t+1) = \omega_{i,j}^{(k-1)}(t) + \eta_{k-1} \left[S_{k-1}(i,j) - \tau_{k-1} \omega_{i,j}^{(k-1)}(t) \right] + \gamma_{td} h_{i,j}^{(k)}(t).$$

Le **signal** descendant $h_{i,j}^{(k)}(t)$ est défini par le niveau k . Il reflète par exemple la volonté de **renforcer** ou **d'affaiblir** certains liens $\omega_{i,j}$ au niveau $k - 1$ pour rendre cohérente une macro-structure pressentie au palier k . Le coefficient $\gamma_{td} > 0$ règle l'intensité globale de cette rétroaction.

Exemples de Signaux Descendants

- **Renforcement** : si le niveau k décide de fusionner deux super-nœuds $\mathcal{N}_\alpha^{(k)}, \mathcal{N}_\beta^{(k)}$, il peut fixer $h_{i,j}^{(k)}(t) = \delta^+$ (un scalaire positif) pour tous les liens (i,j) situés “sous” ces deux macro-nœuds. Ainsi, la mise à jour $\omega_{i,j}^{(k-1)}$ devient positivement biaisée et encourage leur croissance.
- **Séparation** : si le macro-niveau veut scinder un bloc trop hétérogène, il peut imposer un $h_{i,j}^{(k)}(t) = \delta^- < 0$ sur les liens (i,j) internes, incitant la dynamique locale à **affaiblir** $\omega_{i,j}$.

B. Formulation Analytique du Feedback : Un Modèle

Soit un niveau k avec super-nœuds $\mathcal{N}_\alpha^{(k)}$. Chacun $\mathcal{N}_\alpha^{(k)}$ recouvre un ensemble $\mathcal{C}_\alpha^{(k-1)} \subseteq \{\mathcal{N}_\gamma^{(k-1)}\}$. De la même manière, un super-nœud $\mathcal{N}_\gamma^{(k-1)}$ recouvre lui-même des entités d’un niveau antérieur, jusqu’au micro-niveau si on poursuit le dépliage. Alors

$$h_{i,j}^{(k)}(t) = \theta_{\alpha,\beta}^{(k)}(t) \quad \text{dès que } i \in \mathcal{N}_\alpha^{(k)}, j \in \mathcal{N}_\beta^{(k)}.$$

En d’autres termes, le signal descendant agit sur *toutes* les paires (i,j) dont les entités relèvent des mêmes macro-nœuds α et β au niveau k .

On obtient

$$\Delta \omega_{i,j}^{(k-1)}(t) = \eta_{k-1} \left[S_{k-1}(i,j) - \tau_{k-1} \omega_{i,j}^{(k-1)}(t) \right] + \gamma_{td} \theta_{\alpha,\beta}^{(k)}(t).$$

Même si la synergie locale $S_{k-1}(i,j)$ est faible, un *terme macro* positif $\theta_{\alpha,\beta}^{(k)}$ peut maintenir ou renforcer $\omega_{i,j}$. Inversement, un signal négatif peut annihiler $\omega_{i,j}$. Le niveau macro **façonne** ainsi la structure locale pour parvenir à un objectif d’ensemble.

C. Existence de Contraintes et Cohérence Globale

Dans de nombreux scénarios, imposer par le niveau macro un *minimum* ou un *maximum* à la somme des $\omega_{i,j}$ sur un bloc est réécritable sous forme d’un *multiplieur de Lagrange*. Pour un cluster macro $\mathcal{M}_\alpha^{(k)}$, on peut vouloir

$$\sum_{i \in \mathcal{M}_\alpha^{(k-1)}, j \in \mathcal{M}_\beta^{(k-1)}} \omega_{i,j}^{(k-1)} \geq T_{\alpha,\beta}.$$

Si on transcrit ceci en une maximisation (ou minimisation) du potentiel DSL soumis à cette contrainte, la **solution optimale** introduit un $\lambda_{\alpha,\beta} > 0$, qui s’avère **équivalent** à un signal descendant dans $\Delta \omega_{i,j}^{(k-1)}$. En somme, la rétroaction top-down se formalise comme la réponse d’un problème contraint cherchant à préserver (ou atteindre) un lien global entre \mathcal{M}_α et \mathcal{M}_β .

Pour intégrer cette contrainte dans un problème de minimisation (ou de maximisation) d’un **potentiel DSL** $V(\omega)$, on forme la fonction Lagrangienne

$$\mathcal{L}(\omega, \lambda_{\alpha, \beta}) = V(\omega) - \lambda_{\alpha, \beta} \left(\sum_{i \in \mathcal{M}_\alpha^{(k-1)}, j \in \mathcal{M}_\beta^{(k-1)}} \omega_{i,j}^{(k-1)} - T_{\alpha, \beta} \right),$$

où $\lambda_{\alpha, \beta}$ est le multiplicateur de Lagrange associé à cette contrainte, avec $\lambda_{\alpha, \beta} > 0$ pour une contrainte d'inégalité. L'optimisation consiste à trouver ω et $\lambda_{\alpha, \beta}$ tels que les dérivées partielles de \mathcal{L} par rapport à ces variables soient nulles, c'est-à-dire

$$\frac{\partial \mathcal{L}}{\partial \omega_{i,j}^{(k-1)}} = 0 \quad \text{pour tout } i, j \quad \text{et} \quad \frac{\partial \mathcal{L}}{\partial \lambda_{\alpha, \beta}} = 0.$$

La dérivée par rapport à $\lambda_{\alpha, \beta}$ impose directement la contrainte

$$\sum_{i \in \mathcal{M}_\alpha^{(k-1)}, j \in \mathcal{M}_\beta^{(k-1)}} \omega_{i,j}^{(k-1)} = T_{\alpha, \beta},$$

dans le cas limite où la contrainte est active. La dérivée par rapport à $\omega_{i,j}^{(k-1)}$ fournit une condition d'optimalité qui lie la variation du potentiel $V(\omega)$ aux pondérations elles-mêmes et à $\lambda_{\alpha, \beta}$.

Calculons cette dérivée. La fonction \mathcal{L} se décompose en un **potentiel** $V(\omega)$ et un **terme de contrainte** modulé par $\lambda_{\alpha, \beta}$. Ainsi, pour un lien donné, la dérivée partielle s'écrit :

$$\frac{\partial \mathcal{L}}{\partial \omega_{i,j}^{(k-1)}} = \frac{\partial V(\omega)}{\partial \omega_{i,j}^{(k-1)}} - \lambda_{\alpha, \beta} \frac{\partial}{\partial \omega_{i,j}^{(k-1)}} \left(\sum_{p \in \mathcal{M}_\alpha^{(k-1)}, q \in \mathcal{M}_\beta^{(k-1)}} \omega_{p,q}^{(k-1)} - T_{\alpha, \beta} \right).$$

Notons que la somme

$$\sum_{p \in \mathcal{M}_\alpha^{(k-1)}, q \in \mathcal{M}_\beta^{(k-1)}} \omega_{p,q}^{(k-1)}$$

est linéaire par rapport à chaque variable $\omega_{i,j}^{(k-1)}$. Par conséquent, la dérivée partielle de cette somme par rapport à $\omega_{i,j}^{(k-1)}$ est simplement 1 (lorsque $i \in \mathcal{M}_\alpha^{(k-1)}$ et $j \in \mathcal{M}_\beta^{(k-1)}$); de plus, la dérivée de la constante $T_{\alpha, \beta}$ est nulle. On obtient donc :

$$\frac{\partial \mathcal{L}}{\partial \omega_{i,j}^{(k-1)}} = \frac{\partial V(\omega)}{\partial \omega_{i,j}^{(k-1)}} - \lambda_{\alpha, \beta}.$$

Pour que cette dérivée soit nulle, il faut que

$$\frac{\partial V(\omega)}{\partial \omega_{i,j}^{(k-1)}} = \lambda_{\alpha, \beta}.$$

ce qui implique que l'augmentation de la pondération $\omega_{i,j}^{(k-1)}$ doit être compensée par un signal descendant proportionnel à $\lambda_{\alpha,\beta}$. En d'autres termes, le multiplicateur $\lambda_{\alpha,\beta}$ agit comme un **signal rétroactif top-down** qui ajuste les mises à jour locales $\Delta \omega_{i,j}^{(k-1)}$ afin de satisfaire la contrainte globale imposée sur le cluster.

Ce formalisme montre que la rétroaction top-down dans le DSL peut être interprétée comme la solution optimale d'un problème d'optimisation sous contrainte. Le **multiplicateur de Lagrange** $\lambda_{\alpha,\beta}$ introduit une pénalité ou une récompense qui guide la mise à jour des liens pour atteindre ou préserver un niveau de cohésion global, exprimé par le seuil $T_{\alpha,\beta}$. Autrement dit, si les interactions entre les clusters doivent dépasser un certain seuil pour garantir la formation d'un super-cluster cohérent, le multiplicateur $\lambda_{\alpha,\beta}$ se positionne de manière à influencer les mises à jour locales, assurant ainsi la **rétroaction** nécessaire pour maintenir la structure hiérarchique du réseau.

Une fois ce **terme** $\theta_{\alpha,\beta}^{(k)}$ inséré, on peut étudier la stabilité de la dynamique couplée (macro + micro). On obtiendra en général un **point fixe hiérarchique** si le flux descendant n'est ni trop fort ni en contradiction permanente avec la synergie locale. L'analyse linéaire (ou non-linéaire) autour d'un état stationnaire montrerait comment ce *terme descendant* modifie les conditions d'équilibre.

D. Conséquences Dynamiques : Stabilisation ou Oscillation

1) Stabilisation Accélérée

Lorsque le niveau macro repère que deux blocs $\mathcal{M}_\alpha^{(k)}, \mathcal{M}_\beta^{(k)}$ devraient logiquement se rapprocher (ou se fusionner) selon un macro-critère, il envoie un **feedback positif** $\theta_{\alpha,\beta}^{(k)}(t) > 0$. Cela *force* la consolidation au niveau $k - 1$ (voir ex. 5.1 dans 6.4.1.2). La **convergence** peut alors se faire plus vite, le macro-niveau évitant des tâtonnements au palier inférieur.

2) Oscillations ou Conflits

Une rétroaction trop massive ou contradictoire engendre possiblement des **oscillations**. Le **niveau micro** cherche à suivre sa *synergie intrinsèque*, pendant que le macro-niveau force un *schéma* incompatible. On peut ainsi observer un *cycle* si le feedback inverse le sens local à un moment inopportun, tant que la synergie micro n'a pas encore atteint un état stable. Ce se traduit par une valeur propre (ou un module > 1) dans le *jacobien* du système couplé, signe d'un régime oscillatoire.

E. Exemple Pratique de Fusion Macro

Considérons deux **macro-clusters** $\mathcal{M}_\alpha^{(k)}, \mathcal{M}_\beta^{(k)}$ au niveau k . On souhaite les "unir" pour constituer un plus grand bloc, mais au palier $k - 1$, leurs entités (i, j) ne sont pas encore suffisamment reliées. Le macro-niveau définit alors :

$$\theta_{\alpha,\beta}^{(k)}(t) = \delta^+ > 0,$$

et $\theta_{\alpha',\beta'}^{(k)} = 0$ pour tout autre couple (α', β') . Automatiquement, la formule

$$\Delta \omega_{i,j}^{(k-1)}(t) = \eta_{k-1} [S_{k-1}(i, j) - \tau_{k-1} \omega_{i,j}(t)] + \gamma_{td} \delta^+$$

se retrouve > 0 pour (i, j) appartenant à $\mathcal{M}_\alpha^{(k-1)} \times \mathcal{M}_\beta^{(k-1)}$. En itérant, ces liens $\omega_{i,j}$ se renforcent jusqu'à atteindre un point stable élevé, *verrouillant* (lock-in) la fusion macro souhaitée.

Pour analyser l'état d'équilibre, supposons que la dynamique converge vers une valeur stable $\omega_{i,j}^*$. À l'équilibre, nous avons

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta_{k-1} [S_{k-1}(i, j) - \tau_{k-1} \omega_{i,j}^*] + \gamma_{td} \delta^+.$$

En simplifiant, nous obtenons

$$\eta_{k-1} [S_{k-1}(i, j) - \tau_{k-1} \omega_{i,j}^*] + \gamma_{td} \delta^+ = 0.$$

Isolons $\omega_{i,j}^*$:

$$S_{k-1}(i, j) - \tau_{k-1} \omega_{i,j}^* = -\frac{\gamma_{td} \delta^+}{\eta_{k-1}},$$

ce qui conduit à

$$\omega_{i,j}^* = \frac{S_{k-1}(i, j)}{\tau_{k-1}} + \frac{\gamma_{td} \delta^+}{\eta_{k-1} \tau_{k-1}}.$$

Cette expression montre que l'équilibre atteint par la pondération $\omega_{i,j}^{(k-1)}$ est supérieur à celui que l'on aurait obtenu sans la rétroaction macro (c'est-à-dire sans le terme $\gamma_{td} \delta^+$). Le terme supplémentaire $\frac{\gamma_{td} \delta^+}{\eta_{k-1} \tau_{k-1}}$ est positif, ce qui force la pondération à converger vers une valeur élevée. En conséquence, lorsque l'on itère la mise à jour, les liens entre les entités des sous-clusters qui constituent les macro-clusters $\mathcal{M}_\alpha^{(k)}$ et $\mathcal{M}_\beta^{(k)}$ se renforcent progressivement jusqu'à ce qu'ils atteignent ce niveau élevé, assurant ainsi la fusion verrouillée des macro-clusters. Ce phénomène de **lock-in** garantit que, dès lors que le signal macro est positif, la rétroaction descendante ajuste la dynamique locale de manière à fusionner de manière cohérente les clusters concernés.

Ainsi, la **rétroaction top-down** s'avère indispensable pour qu'un SCN multi-niveau *ne se cantonne pas* à un simple auto-apprentissage local, mais bénéficie d'une **vision** plus large, garantissant une **cohérence** entre les micro-liens et les *objectifs* ou configurations macro. Cette connexion "macro \rightarrow micro" rend possible une véritable **hiérarchie** où *chaque* palier est actif, et non seulement un agrégateur passif (flux ascendant).

6.4.1.3. Rôle crucial de la cohérence pour éviter le conflit entre niveaux

Les **flux ascendants** (bottom-up) et **descendants** (top-down) (cf. 6.4.1.1 et 6.4.1.2) assurent la double direction de la hiérarchie dans un **SCN** (Synergistic Connection Network) à plusieurs échelles. Cependant, l'existence simultanée de ces deux flux n'est pas exempte de difficultés. Un **conflit** peut survenir si les décisions ou la logique d'auto-organisation d'un **niveau** (micro ou macro) viennent s'opposer de manière répétée à celles d'un autre. Pour maintenir la **stabilité** et permettre une véritable coordination multi-niveau, on a besoin d'une **cohérence**, c'est elle qui

garantit que les modifications imposées en haut (macro) et la dynamique locale en bas (micro) ne se **détruisent** pas mutuellement.

A. Notion de “Conflit” Entre Niveaux

Dans la formulation du **DSL** (Deep Synergy Learning) au niveau micro (ou local), des entités $\{\mathcal{E}_{ij}\}$ se regroupent en clusters via la règle $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$, appuyée par des seuils ou conditions de cohésion. Le flux ascendant envoie au niveau supérieur l’information “tel sous-ensemble forme un cluster local stable”.

Le **niveau macro** peut, inversement, vouloir briser ou fusionner ces clusters pour un objectif global. S’il identifie qu’un ensemble local \mathcal{C} est “mal aligné” avec le schéma plus vaste, il envoie un signal descendant pour casser la cohésion $\sum_{i,j \in \mathcal{C}} \omega_{i,j}$. Si cette décision contrarie trop fortement la synergie locale $S(i,j)$, un **conflit** naît. On a alors, par exemple, un **macro-niveau** dictant $\theta_{\text{desc}} < 0$ sur tous les liens de \mathcal{C} , tandis que localement, $\omega_{i,j}$ continue d’augmenter sous l’effet d’une forte affinité.

Sans un **alignement** suffisant, le micro-niveau peut “verrouiller” un cluster alors que le macro-niveau tente de le disloquer, causant des **oscillations** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)] + \gamma_{\text{topdown}} h_{i,j}^{(\text{macro})}(t).$$

Si γ_{topdown} est trop grand et s’oppose en permanence à la synergie $S(i,j)$, on peut assister à un “tir à la corde” où $\omega_{i,j}$ ne converge pas, passant successivement au-dessus et au-dessous du seuil critique.

B. Formulation Mathématique du Conflit et de la Cohérence

Dans un SCN hiérarchique, on définit les pondérations :

$$\omega_{\alpha,\beta}^{(k)}(t+1) = \omega_{\alpha,\beta}^{(k)}(t) + \eta_k[S_k(\alpha,\beta) - \tau_k \omega_{\alpha,\beta}^{(k)}(t)] + \Delta_{\text{desc}}^{(k)}(\omega^{(k+1)}),$$

pour le niveau k . La fonction $\Delta_{\text{desc}}^{(k)}$ correspond au flux descendant venant de $\omega^{(k+1)}$. Parallèlement,

$$\omega_{\alpha,\beta}^{(k+1)}(t) = \Gamma_k(\omega^{(k)}(t)) \quad (\text{agrégation ascendante}).$$

Si la *direction* imposée par $\Delta_{\text{desc}}^{(k)}$ contredit la logique locale $S_k(\alpha,\beta)$, il y a risque de boucle contradictoire.

Un moyen de **limiter** la discorde est de postuler une **contrainte** du type

$$\|\Delta_{\text{desc}}^{(k)}(\omega^{(k+1)})\| \leq \alpha \|\eta_k[S_k(\cdot) - \tau_k \cdot]\|,$$

avec $\alpha < 1$. Cela impose que le flux descendant ne soit pas plus grand (en norme) que la mise à jour locale. Sur un plan conceptuel, cela **force** le macro-niveau à *respecter* la logique micro au lieu de la balayer, prévenant les conflits.

Nous démontrons ci-dessous, de manière rigoureuse, en quoi cette contrainte garantit la cohérence de la dynamique.

Pour chaque paire (α, β) considérée, la mise à jour totale au niveau k est la somme du terme local et du flux descendant :

$$U_{\text{total}}^{(k)} = \eta_k \left[S_k(\alpha, \beta) - \tau_k \omega_{\alpha, \beta}^{(k)}(t) \right] + \Delta_{\text{desc}}^{(k)}(\omega^{(k+1)}).$$

Supposons que la norme du terme local soit notée

$$\| U^{(k)} \| = \| \eta_k \left[S_k(\alpha, \beta) - \tau_k \omega_{\alpha, \beta}^{(k)}(t) \right] \|.$$

La contrainte imposée se traduit par

$$\| \Delta_{\text{desc}}^{(k)} \| \leq \alpha \| U^{(k)} \| \quad \text{avec} \quad \alpha < 1.$$

Par l'inégalité triangulaire, nous avons

$$\| U_{\text{total}}^{(k)} \| \geq \| U^{(k)} \| - \| \Delta_{\text{desc}}^{(k)} \| \geq \| U^{(k)} \| (1 - \alpha).$$

Comme $1 - \alpha > 0$, il s'ensuit que le signe (et par conséquent la direction) du terme total est dominé par celui du terme local. En d'autres termes, la rétroaction descendante ne peut renverser la logique de la mise à jour locale si sa contribution reste inférieure à une fraction déterminée de celle-ci. Cette propriété garantit que le mécanisme de mise à jour au niveau k continue à évoluer dans la direction dictée par $S_k(\alpha, \beta)$, évitant ainsi la formation de boucles contradictoires.

Nous pouvons analyser ce phénomène en considérant deux cas extrêmes. D'une part, si la rétroaction descendante est parfaitement nulle, la mise à jour est entièrement dictée par la logique locale, et le système converge vers l'équilibre défini par $S_k(\alpha, \beta) = \tau_k \omega_{\alpha, \beta}^{(k)}(t)$. D'autre part, si la rétroaction descendante devient trop imposante (c'est-à-dire que $\| \Delta_{\text{desc}}^{(k)} \|$ dépasse la limite fixée par la contrainte), la direction du signal total pourrait être inversée par rapport à celle du terme local, conduisant potentiellement à une oscillation ou à une instabilité du système. Le choix de $\alpha < 1$ est donc essentiel pour forcer la hiérarchie à respecter la logique locale, en empêchant le niveau macro de submerger la dynamique micro.

Dans ce contexte, la contrainte agit comme un **filtre** ou un **modulateur** qui assure que l'influence descendante reste subordonnée à l'influence locale. Cela permet d'obtenir une convergence stable du système et d'éviter les comportements indésirables, tels que des cycles oscillatoires ou un comportement chaotique, qui pourraient résulter d'une rétroaction trop forte ou mal synchronisée.

Si, malgré tout, on laisse γ_{topdown} ou $\Delta_{\text{desc}}^{(k)}$ devenir très imposant, le système peut :

- **Osciller** (cycle),
- **Stagner** dans un compromis inutile,
- Produire un **comportement chaotique** si la rétroaction n'est pas phasée.

C. Stratégies d'Évitement du Conflit

Une **solution** est d'**ordonner** clairement les phases d'apprentissage local et les phases de correction macro. Par exemple, on laisse le micro-niveau faire θ_1 itérations, puis le macro-niveau intervient, puis on reboucle. Cela évite que le feedback macro ne se déclenche en même temps que les dynamiques locales, réduisant les risques d'oscillation.

On peut calibrer η_{micro} et γ_{topdown} de sorte que le niveau macro ne renverse pas violemment la cohésion locale. On peut également intégrer un *filtrage* :

$$h_{i,j}^{(\text{macro})}(t+1) = \lambda h_{i,j}^{(\text{macro})}(t) + (1-\lambda) \tilde{h}_{i,j}^{(\text{macro})}(t),$$

où \tilde{h} est le signal brut. Ainsi, on lisse la correction descendante, évitant des secousses brusques.

Dans certains *algorithmes DSL*, on peut surveiller la **norme** d'évolution $\|\Delta\omega\|$. Si on détecte une amplitude de mises à jour trop forte signant un conflit, on restreint temporairement la rétroaction top-down ou on revoit les paramètres d'agrégation ascendante pour calmer la discorde.

Considérons un SCN hiérarchique dans lequel la mise à jour des pondérations au niveau local (micro-niveau) est effectuée selon la dynamique classique

$$\omega(t+1) = \omega(t) + \eta_{\text{micro}}[S_{\text{local}}(\omega(t)) - \tau_{\text{micro}} \omega(t)],$$

où S_{local} représente le signal de synergie issu des interactions locales. Supposons que l'on laisse le micro-niveau opérer pendant θ_1 itérations consécutives, ce qui conduit à une certaine convergence ou stabilisation locale. Après ces θ_1 itérations, un niveau macro intervient en fournissant un signal de rétroaction, que nous notons $\Delta\omega_{\text{topdown}}$. L'ordonnancement séquentiel consiste alors à définir une alternance temporelle, telle que :

$$\begin{aligned} \text{Pour } t \in [t_0, t_0 + \theta_1]: \quad \omega(t+1) &= \omega(t) + \eta_{\text{micro}}[S_{\text{local}}(\omega(t)) - \tau_{\text{micro}} \omega(t)], \\ \text{Pour } t = t_0 + \theta_1: \quad \omega(t+1) &= \omega(t) + \Delta\omega_{\text{topdown}}, \end{aligned}$$

puis le cycle se répète. Cette phase de séparation temporelle garantit que le micro-niveau atteint d'abord un état quasi-stationnaire avant que le macro-niveau n'intervienne, évitant ainsi que le signal top-down ne vienne perturber simultanément la dynamique locale.

La stabilité du système dépend fortement du calibrage des paramètres η_{micro} et γ_{topdown} (ce dernier étant le coefficient associée à la rétroaction macro). Pour empêcher le signal macro de renverser violemment la cohésion locale, il convient d'exiger que, pour tout instant t , la norme du signal de rétroaction $\|\Delta\omega_{\text{topdown}}(t)\|$ soit limitée par rapport à la norme de la mise à jour locale. Formellement, on impose la contrainte

$$\|\Delta\omega_{\text{topdown}}(t)\| \leq \alpha \|\eta_{\text{micro}}[S_{\text{local}}(\omega(t)) - \tau_{\text{micro}} \omega(t)]\|, \quad \text{avec } \alpha < 1.$$

Cette inégalité assure que, quelle que soit l'amplitude du signal macro, sa contribution reste strictement inférieure à celle du mécanisme local. En d'autres termes, même en présence d'un feedback top-down, la mise à jour totale

$$\Delta\omega(t) = \eta_{\text{micro}}[S_{\text{local}}(\omega(t)) - \tau_{\text{micro}} \omega(t)] + \Delta\omega_{\text{topdown}}(t)$$

vérifie, d'après l'inégalité triangulaire,

$$\begin{aligned} \|\Delta\omega(t)\| &\geq \|\eta_{\text{micro}} [S_{\text{local}}(\omega(t)) - \tau_{\text{micro}} \omega(t)]\| - \|\Delta\omega_{\text{topdown}}(t)\| \geq (1 - \alpha) \\ &\quad \|\eta_{\text{micro}} [S_{\text{local}}(\omega(t)) - \tau_{\text{micro}} \omega(t)]\|. \end{aligned}$$

Ainsi, la direction de la mise à jour reste dominée par le signal local, ce qui empêche le macro-niveau d'imposer un changement radical dans la cohésion locale.

Afin d'éviter des variations brusques du signal macro, il est judicieux d'intégrer un mécanisme de filtrage qui lisse la rétroaction descendante. Pour ce faire, on définit un signal filtré $h_{i,j}^{(\text{macro})}(t)$ qui se met à jour selon la relation de lissage exponentiel

$$h_{i,j}^{(\text{macro})}(t+1) = \lambda h_{i,j}^{(\text{macro})}(t) + (1 - \lambda) \tilde{h}_{i,j}^{(\text{macro})}(t),$$

où $\tilde{h}_{i,j}^{(\text{macro})}(t)$ représente le signal top-down brut, et $\lambda \in (0,1)$ est le coefficient de lissage. Ce procédé agit comme un **passe-bas** qui atténue les fluctuations rapides du signal macro, assurant que la correction descendante se fasse de manière progressive. Par conséquent, la mise à jour effective devient

$$\Delta\omega_{\text{topdown,eff}}(t) = \gamma_{\text{topdown}} h_{i,j}^{(\text{macro})}(t),$$

ce qui garantit que les variations brusques du feedback sont atténuées par le facteur $(1 - \lambda)$.

En complément des stratégies précédentes, il est possible d'introduire un mécanisme de surveillance de la norme d'évolution des pondérations, notée $\|\Delta\omega(t)\|$. L'objectif est de détecter en temps réel des mises à jour excessivement importantes, qui signaleraient un conflit entre le signal local et le feedback macro. Si, à un instant donné, l'on observe que

$$\|\Delta\omega(t)\| > \varepsilon_{\text{critique}},$$

où $\varepsilon_{\text{critique}}$ est un seuil fixé en fonction des caractéristiques du système, alors une procédure de contrôle est déclenchée, pouvant consister en la réduction temporaire de γ_{topdown} ou en l'ajustement des paramètres d'agrégation ascendante. Ce mécanisme adaptatif permet de ramener la dynamique dans la zone de stabilité définie par la contrainte ci-dessus.

D. Intérêt Fondamental de la Cohérence

Convergence Efficace. Une cohérence multi-niveau bien gérée garantit une **convergence** plus rapide. Le flux descendant consolide ou oriente ce qui remonte au lieu de le déstabiliser. Au final, moins d'itérations suffisent pour qu'un **cluster** local se confirme ou se dissolve selon les vœux macro.

Robustesse. Sans cohérence, un petit aléa peut dégénérer en conflit entre étages, produisant des effets chaotiques ou des réorganisations incessantes. Avec cohérence, on obtient une **résilience**. Si un bloc local est perturbé, le macro-niveau intervient pour le réajuster de manière stable, évitant ainsi toute lutte contradictoire.

Cas d'École : Applications Cognitives ou Robotiques

- **Cognitif** : Le *macro-concept* (aire cérébrale supérieure) incite des groupes de neurones (niveau micro) à se réorganiser, mais ne détruit pas systématiquement leurs associations *stables* ; c'est un **compromis**.
- **Robotique** : Une flotte de robots se coordonne (niveau micro), le commandant (macro) donne une directive globale ; la cohérence évite l'état de confusion où les robots réagissent localement à un plan que l'autorité contredit.

6.4.2. Communication Synergiques entre Niveaux

Au sein d'un **SCN** (Synergistic Connection Network) multi-niveau, la **communication** entre paliers (micro, méso, macro) ne se limite pas à de simples flux d'informations, elle s'inscrit dans la **dynamique** d'auto-organisation, où les liens ω (pondérations) sont mis à jour de manière **synchrone** ou **asynchrone** suivant la règle DSL. La notion de “communication synergiques” évoque spécifiquement la façon dont les **clusters** détectés (ou émergents) à un niveau sont **transmis** sous forme d'agrégation vers le niveau supérieur, et comment, réciproquement, ce niveau supérieur valide ou rétroagit sur ces clusters. Dans la section 6.4.2.1, nous examinons particulièrement l'idée selon laquelle un micro-cluster **stabilisé** (au niveau local) peut être “validé” ou “agrégé” au palier macro, illustrant la prise en compte hiérarchique des informations.

6.4.2.1. Si un micro-cluster se stabilise, le niveau macro peut le “valider” ou l'agréger

La dynamique **multi-échelle** au sein d'un **SCN** (Synergistic Connection Network) permet à des **micro-clusters** (niveau local) de se former, puis de “remonter” progressivement vers des niveaux supérieurs (intermédiaire ou macro), conformément à la logique du **flux ascendant** (cf. 6.4.1.1). Une fois qu'un micro-cluster \mathcal{C}_α est jugé **stable** au niveau 0 (ou local), il revient au **niveau macro** (ou intermédiaire) de **valider** cette stabilisation, puis, si nécessaire, de l'**agréger** sous forme d'un super-nœud. Le texte qui suit expose les bases mathématiques de cette validation et en souligne les bénéfices et précautions.

A. Stabilisation au Niveau Micro et Constitution d'un Cluster

Un **micro-cluster** $\mathcal{C}_\alpha \subseteq \{\mathcal{E}_i\}$ se définit généralement comme un groupe d'entités dont les **pondérations** internes $\{\omega_{i,j}^{(0)}\}$ sont élevées. En pratique, on introduit un **critère** de cohésion tel que

$$\Omega(\mathcal{C}_\alpha) = \sum_{i,j \in \mathcal{C}_\alpha} \omega_{i,j}^{(0)} \geq \theta_{\text{loc}},$$

où θ_{loc} est un **seuil** local. La dynamique DSL (par exemple de type additive) agit au niveau 0 :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_0 [S_0(i,j) - \tau_0 \omega_{i,j}^{(0)}(t)],$$

faisant **croître** les liens $\omega_{i,j}^{(0)}$ entre entités localement synergiques, jusqu'à ce qu'un sous-ensemble \mathcal{C}_α dépasse θ_{loc} . On dit alors que \mathcal{C}_α est un **cluster local stabilisé** (voir aussi 6.4.1.1 pour le flux ascendant).

B. Rôle du Niveau Macro : Valider ou Refuser la Stabilisation

Le **niveau macro** (ou supérieur) reçoit l'information qu'un **cluster local** \mathcal{C}_α s'est formé. Deux possibilités se présentent :

- Le niveau macro **valide** la stabilisation, c'est-à-dire qu'il *accepte* ce micro-cluster comme un bloc fonctionnel.
- Le niveau macro **rejette** (ou “modère”) cette stabilisation, par exemple s'il estime que \mathcal{C}_α interfère avec un **objectif** ou une **cohérence** globale (cf. 6.4.1.3 sur la cohérence).

Dans la première hypothèse, le macro-niveau **agrège** \mathcal{C}_α en un **super-nœud** $\mathcal{N}_\alpha^{(1)}$. On procède à une fonction d'agrégation Ψ (6.2.2) pour définir les liaisons entre $\mathcal{N}_\alpha^{(1)}$ et les autres super-nœuds $\{\mathcal{N}_\beta^{(1)}\}$ déjà reconnus :

$$\omega_{\alpha,\beta}^{(1)} = \Psi\left(\{\omega_{i,j}^{(0)} \mid i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta\}\right).$$

Le cluster local \mathcal{C}_α devient donc un **nœud de niveau 1**. L'information micro est ainsi “condensée” pour que l'échelle macro puisse la manipuler avec un **nombre réduit** de nœuds. Cette **validation** confère au cluster local un statut “officiel” à l'échelle supérieure.

Si le niveau macro **refuse** la stabilisation locale, il peut renvoyer un **feedback** descendant (6.4.1.2) pour affaiblir les liens $\omega_{i,j}^{(0)}$ du cluster \mathcal{C}_α . Par exemple, un **signal** $\theta_\alpha^{(\text{macro})} < 0$ s'ajoute dans la mise à jour :

$$\Delta \omega_{i,j}^{(0)}(t) = \eta_0 \left[S_0(i,j) - \tau_0 \omega_{i,j}^{(0)}(t) \right] + \gamma_{\text{top-down}} \theta_\alpha^{(\text{macro})}.$$

Cela **empêche** le cluster de persister localement et force sa reconfiguration, évitant un conflit durable.

C. Bénéfices de la Validation par le Macro-Niveau

La “validation” d'un micro-cluster par le macro-nœud apporte plusieurs **avantages** dans un SCN multi-échelle :

- **Réduction de la complexité** : On n'a pas à gérer toutes les entités $\{\mathcal{E}_i\}$ individuellement au niveau macro. Une fois validé, \mathcal{C}_α est résumé par $\mathcal{N}_\alpha^{(1)}$.
- **Stabilisation hiérarchique** : Le micro-cluster profite de l'**appui** du macro-niveau. En retour, le macro-niveau sait qu'il peut compter sur l'existence de ce bloc local “fiable” pour bâtir des structures plus vastes.

- **Rapidité de convergence** : Plutôt que d’attendre un consensus global, on **valide** localement et on “remonte” ces blocages fonctionnels. Ainsi, la mise en place d’une structure macro se nourrit immédiatement de la dynamique micro.

D. Exemple : Fusion de Micro-Clusters Réunis

Dans un système DSL robotique (6.1, 6.2), supposons que deux micro-clusters \mathcal{C}_α et \mathcal{C}_β se forment dans des zones proches. Le **macro-niveau** repère qu’ils ont un **objectif** semblable (cohérence spatiale, même mission), et décide de “valider” chacun individuellement, puis de **fusionner** ces deux super-nœuds $\mathcal{N}_\alpha^{(1)}$ et $\mathcal{N}_\beta^{(1)}$ en un mégaccluster $\mathcal{N}_{\alpha,\beta}^{(2)}$. Cette validation en chaîne illustre comment le flux ascendant (entités \rightarrow micro-clusters) s’enchaîne au flux macro (validation \rightarrow super-nœuds).

Sans l’étape de “confirmation” macro, les micro-clusters pourraient rester dans un état incertain, ou, à l’inverse, un flux descendant contradictoire viendrait contester leur existence sans se manifester explicitement. La validation rend la hiérarchie plus explicite et plus stable.

6.4.2.2. Mécanismes de filtrage : ne pas transmettre toute la synergie brute vers le haut, résumer ou agréger

La **multi-échelle** dans un **SCN** (Synergistic Connection Network) induit une série de **flux ascendants** (chap. 6.4.1) où des entités, d’abord organisées localement (niveau 0), s’agrègent ou se stabilisent sous forme de clusters, puis sont transmises à l’échelle supérieure (mésos, macro). Cette **transmission** d’informations soulève immédiatement une question de **volume**. Si l’on transmet la **totalité** des pondérations $\omega_{i,j}$ ou liaisons internes à chaque cluster, le palier supérieur se retrouve surchargé et perd l’avantage de la hiérarchie. D’où la nécessité d’un **filtrage** ou d’une **agrégation** qui, au lieu de transmettre toutes les “synergies brutes”, n’en relaie qu’un **résumé** concis et pertinent.

La présente section clarifie les principes formels de ce **filtrage**. D’un point de vue **mathématique**, il s’agit d’une opération Ψ réduisant la dimension du réseau ; d’un point de vue **opérationnel**, cela limite la **masse** d’informations circulant vers le haut, assurant robustesse et économie.

A. Motivation du Filtrage et de l’Agrégation

Si au niveau local n entités forment potentiellement $O(n^2)$ liaisons, transmettre $\omega_{i,j}^{(0)}$ pour toutes les paires (i,j) vers le niveau 1 annulerait l’intérêt d’une **architecture hiérarchique**. Le filtrage permet de **résumer** la masse des liens en un plus petit nombre de **super-pondérations** ou **indicateurs**, de sorte que le niveau macro n’opère que sur $O(m^2)$ valeurs, où $m \ll n$.

Exemple : si $n = 10^5$, on peut réduire à m
 $= 10^3$ super-nœuds, puis passer de 10^{10} liaisons à 10^6 .

Les dynamiques DSL (ex. la mise à jour additive) génèrent souvent beaucoup de **liens faibles** ou transitoires. Les transmettre « bruts » enliserait le niveau macro dans des détails superflus. Un **filtrage** (seuil, élagage) écarte les liaisons jugées peu contributrices, protégeant l’échelle supérieure contre des variations mineures.

Un cluster local \mathcal{C}_α peut avoir $O(k^2)$ liens internes. En les condensant via une fonction Ψ (moyenne, somme, etc.), on gagne une “seule” valeur $\omega_\alpha^{(1)}$ pour décrire la cohésion interne de \mathcal{C}_α , ce qui **lisse** les aléas locaux. Mathématiquement, la variance de liens internes se “contracte” dans une moyenne, rendant la **signalétique** plus stable aux fluctuations.

B. Principes Formels du Filtrage/Agrégation

Un mécanisme simple :

$$\tilde{\omega}_{i,j}^{(0)} = \begin{cases} \omega_{i,j}^{(0)}, & \text{si } \omega_{i,j}^{(0)} \geq \theta, \\ 0, & \text{sinon.} \end{cases}$$

On ne conserve que les liaisons $\omega_{i,j}$ supérieures à un seuil θ . Ce filtrage peut s’appliquer avant de construire le super-nœud, assurant qu’on n’intègre dans \mathcal{C}_α que les liens réellement “forts”.

Une fois un cluster $\mathcal{C}_\alpha \subseteq \{\mathcal{E}_i\}$ déterminé, la liaison $\omega_{\alpha,\beta}^{(1)}$ entre deux clusters $\mathcal{C}_\alpha, \mathcal{C}_\beta$ se calcule par une **fonction**

$$\omega_{\alpha,\beta}^{(1)} = \Psi \left(\left\{ \omega_{i,j}^{(0)} \mid i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta \right\} \right).$$

Les formes de Ψ peuvent être :

- **Somme** : $\omega_{\alpha,\beta} = \sum_{i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta} \omega_{i,j}$.
- **Moyenne** : $\omega_{\alpha,\beta} = \frac{1}{|\mathcal{C}_\alpha| |\mathcal{C}_\beta|} \cdot \sum_{i,j} \omega_{i,j}$.
- **Max** ou **quantile** : $\omega_{\alpha,\beta} = \max_{i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta} \omega_{i,j}$.

Plutôt que de transmettre un seul nombre, on peut fournir un **vecteur** de statistiques comprenant min, max, moyenne, écart-type, ... afin de mieux caractériser la distribution interne. Le niveau macro reçoit ainsi une signature plus détaillée, tout en limitant l’explosion de la quantité d’informations échangées, puisque seules quelques métriques sont envoyées au lieu de milliers de liens individuels.

C. Processus Algorithmique : du Micro au Macro

1. Détection de Clusters Locaux

On commence par identifier $\{\mathcal{C}_\alpha^{(0)}\}_{\alpha=1,\dots,r}$ au niveau micro, où chaque $\mathcal{C}_\alpha^{(0)}$ correspond à un **bloc** présentant une forte cohésion, vérifiant $\Omega(\mathcal{C}_\alpha^{(0)}) \geq \theta_{\text{loc}}$.

2. Filtrage

On applique un seuil ou un algorithme d’élagage pour ne **retenir** que les liens importants au sein de \mathcal{C}_α ou entre \mathcal{C}_α et \mathcal{C}_β .

3. Construction du Super-Nœud

On agrège \mathcal{C}_α en un *super-nœud* $\mathcal{N}_\alpha^{(1)}$. On calcule $\omega_{\alpha,\beta}^{(1)}, \forall \beta$, grâce à Ψ .

4. Transmission

On envoie vers le niveau macro le **résumé** sous la forme de $\{\omega_{\alpha,\beta}^{(1)}\}_{\alpha,\beta}$ ou d'un ensemble d'indicateurs statistiques décrivant les liaisons internes et externes. Le macro-niveau ne traite pas toutes les $\omega_{i,j}^{(0)}$, mais uniquement ce “filtrat” représentatif.

D. Impact Pratique : Économie et Stabilité

Allègement. Avec le filtrage, le **niveau macro** ne manie plus $O(n^2)$ poids bruts, mais $O(r^2)$ super-pondérations ($r = \text{nombre de clusters} \leq n$). L'économie est dramatique si $r \ll n$.

Robustesse. Les *petites* fluctuations des liens $\omega_{i,j}^{(0)}$ qui n'atteignent pas le seuil θ ne remontent pas. On évite de bouleverser la structure macro pour un “micro-bruit” local. Cela procure une forme de **tolérance** aux perturbations.

Possibilité de Réactualisation. Le filtrage doit rester **réversible** afin de permettre l'évolution des liaisons. Un lien actuellement **faible** peut se renforcer si la synergie $S_0(i,j)$ évolue, garantissant ainsi une adaptation dynamique du réseau. Certaines implémentations DSL réévaluent régulièrement les *liens filtrés*, pour ne pas rater l'opportunité d'un futur cluster.

6.4.2.3. Exemples Concrets : Agent Local en Robotique, Superviseur Macro

Il est souvent utile d'illustrer la **communication synergiques** (section 6.4.2) entre un **niveau micro** et un **niveau macro** dans un contexte de **Deep Synergy Learning**. Deux cas d'usage sont envisagés. Le premier concerne un **agent local** en robotique au niveau micro, qui ajuste ses connexions en fonction des interactions immédiates. Le second met en scène un **superviseur macro**, chargé d'assurer une coordination globale en régulant les dynamiques collectives. Dans ces deux scénarios, on voit comment un *flux ascendant* condense l'information essentielle (évacuant les détails inutiles) et comment un *flux descendant* fournit un **contrôle** ou une **influence** réciproque sur les liaisons du niveau inférieur.

A. Agent Local en Robotique

Considérons un système multi-agent de type robotique, où chaque **robot** est modélisé comme une **entité** \mathcal{E}_i . Les pondérations $\omega_{i,j}^{(0)}$ relient les paires de robots i et j au **niveau 0**, c'est-à-dire le niveau micro. L'**objectif** du **Deep Synergy Learning** est de permettre une auto-organisation locale ; chaque robot peut adapter ses connexions selon une **règle** de plasticité. Une forme linéaire-additive simple est donnée par

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_{\text{loc}} [S_{\text{loc}}(i,j) - \tau_{\text{loc}} \omega_{i,j}^{(0)}(t)],$$

où $\eta_{\text{loc}} > 0$ est un **taux d'apprentissage** et $\tau_{\text{loc}} > 0$ un **terme de décroissance**. La **fonction de synergie** $S_{\text{loc}}(i,j)$ dépend par exemple de la **distance** entre robots, de leur **similarité** de tâches ou de toute autre mesure favorisant la coopération.

Lorsque ce processus d'**auto-organisation** local aboutit à un **micro-cluster**, noté $\mathcal{C}_\alpha \subset \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$, on peut agréger \mathcal{C}_α en un **super-nœud** $\mathcal{N}_\alpha^{(1)}$. Au **niveau 1**, on ne conserve pas la totalité des liaisons individuelles $\omega_{i,j}^{(0)}$, mais une **agrégation** $\omega_{\alpha,\beta}^{(1)}$ calculée typiquement via

$$\omega_{\alpha,\beta}^{(1)} = \Psi\left(\{\omega_{i,j}^{(0)} \mid i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta\}\right),$$

où Ψ est une **opération de synthèse**, par exemple une somme, une moyenne pondérée ou un maximum. De la sorte, le **niveau macro** ne voit plus qu'une poignée d'entités $\mathcal{N}_\alpha^{(1)}$, correspondant chacune à un ensemble local cohésif de robots.

Le **flux ascendant** (micro vers macro) se retrouve donc filtré ; on élimine la majorité des détails et on concentre la communication sur un petit nombre de "super-nœuds" issus de l'auto-organisation des agents locaux. Cette logique **multi-niveau** permet de limiter le coût, au lieu de gérer $O(n^2)$ liaisons élémentaires, le **niveau 1** ne manipule que $O(k^2)$ super-liaisons, où k est le nombre de clusters.

B. Superviseur Macro

Dans un second exemple, on suppose l'existence d'un **superviseur macro**, un **niveau supérieur** qui reçoit les **super-nœuds** du niveau 1 (ou un niveau intermédiaire plus haut encore). Ce superviseur gère la **tâche globale** (répartition de zones à explorer, contraintes de mission, etc.) et peut renvoyer un *feedback* vers le micro-niveau pour orienter le renforcement ou l'inhibition de certaines connexions.

Une façon de modéliser ce *feedback descendant* est d'enrichir l'équation de mise à jour micro par un **terme d'influence** Δ_{down} . On peut alors écrire

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_{\text{loc}} \left[S_{\text{loc}}(i,j) - \tau_{\text{loc}} \omega_{i,j}^{(0)}(t) \right] + \gamma_{\text{macro}} \Delta_{\text{down}},$$

où γ_{macro} est un **facteur** réglant l'intensité de l'intervention du **superviseur**. Le terme Δ_{down} peut être, par exemple, un **signal** incitant à renforcer les liaisons entre deux sous-groupes \mathcal{C}_α et \mathcal{C}_β si le niveau supérieur décide de fusionner leurs efforts.

Dans ce cas, l'**auto-organisation** demeure largement locale (les agents mettent à jour leurs poids selon une synergie locale S_{loc}), mais un **contrôle** partiel par la structure de plus haut niveau permet d'anticiper les besoins de la mission globale (objectif externe imposé, ou simple consigne d'alignement).

C. Avantages de la Communication Synergiques

L'**agent local** peut s'organiser entre ses pairs de manière autonome, créant ainsi des **micro-clusters** robustes et adaptatifs. Cette information "micro" est alors **remontée** au **superviseur macro** sous forme condensée (agrégation en super-nœuds), évitant une surcharge $O(n^2)$. Le **superviseur** peut à son tour moduler la configuration du niveau inférieur via un *flux descendant*, soit pour soutenir certaines synergies, soit pour en restreindre d'autres.

Ces mécanismes hiérarchisés assurent à la fois la **scalabilité** (le macro-niveau n'interagit qu'avec un nombre réduit de super-nœuds) et la **réactivité** (chaque groupe local s'auto-organise rapidement

selon η_{loc}). Les pondérations finales $\omega_{i,j}^{(0)}$ résultent ainsi d’une **double influence** combinant la synergie locale estimée à partir des interactions directes et l’**orientation** régulatrice exercée par le macro-niveau.

Lorsque ce cadre s’applique à l’**industrie** robotique ou à un **système** multi-agent plus générique, on préserve un fort degré de **décentralisation** tout en maintenant un pilotage global cohérent. La communication **synergiques** entre niveaux assure en effet un bon compromis entre l’autonomie des sous-groupes et la coordination stratégique imposée par le sommet de la hiérarchie.

Cette capacité à se reconfigurer, tant localement que globalement, correspond à l’esprit du **Deep Synergy Learning**. Plusieurs *couches* ou *niveaux* interagissent via des **flux** réciproques, filtrés et agrégés, sans pour autant que l’un de ces niveaux ne subisse la totalité des détails issus des autres. Le résultat est un **réseau** où l’intelligence se répartit des agents les plus élémentaires jusqu’aux superviseurs d’échelle plus élevée, ce qui favorise la **robustesse** et l’**adaptation** face aux changements de contexte.

6.4.3. Synchronisation et Clustering

Dans la perspective d’un **DSL** (Deep Synergy Learning) déployé sur un **SCN** (Synergistic Connection Network) multi-niveau, la question de la **synchronisation** entre niveaux et celle du **clustering** progressif se posent en termes concrets :

- *Comment* garantir que les divers paliers (micro, méso, macro) ne sont pas en perpétuel décalage ?
- *Comment* gérer la **coexistence** de clusters à échelles différentes, et *quand* décider de la stabilisation ou de la fusion de plusieurs regroupements ?

La section 6.4.3.1 met l’accent sur le rôle d’“échelles intermédiaires” (semi-globales), agissant comme des **étapes** cruciales dans la stabilisation hiérarchique du SCN.

6.4.3.1. Les Clusters d’Échelle Intermédiaire (Semi-Globale) comme Étapes de Stabilisation

Introduction. Le passage d’un niveau **micro** (petits groupes ou entités isolées) à un niveau **macro** (super-nœuds ou regroupements englobants) peut s’avérer trop rapide ou trop complexe si l’on ne tient pas compte de *paliers intermédiaires*. Dans de nombreux schémas multi-niveau, ces paliers prennent la forme de **clusters semi-globaux**, c’est-à-dire des regroupements de taille moyenne. Ces entités intermédiaires facilitent la stabilisation et la cohérence hiérarchique, tout en régulant la montée progressive vers des macro-structures. Les sections qui suivent détaillent le rôle de ces clusters semi-globaux, les mécanismes de synchronisation inter-paliers et divers cas d’utilisation.

A. Échelle Intermédiaire et Stabilisation Progressive

Dans une architecture multi-niveau, un **micro-niveau** peut produire des micro-clusters $\{\mathcal{C}_\alpha^{(0)}\}$. On agrège ensuite ces micro-clusters en un **niveau 1**, où l’on considère des super-nœuds $\mathcal{N}_\alpha^{(1)}$. Lorsque l’échelle totale du système est grande, il est souvent nécessaire de continuer à agréger ces super-

nœuds pour former un **niveau 2**, aboutissant à des macro-clusters $\{\mathcal{N}_\beta^{(2)}\}$ de dimension plus élevée. Dans ce contexte, le niveau 1 agit comme un **tampon** ou une **étape pivot**. Il consolide des groupes de taille moyenne, sans prétendre atteindre immédiatement la structure macro complète.

La stabilisation progressive s'explique en termes **d'équilibre dynamique**. Les pondérations $\omega^{(k)}$ au palier k suivent une règle de mise à jour de type

$$\omega_{a,b}^{(k)}(t+1) = \omega_{a,b}^{(k)}(t) + \eta_k \left[S^{(k)}(a,b) - \tau_k \omega_{a,b}^{(k)}(t) \right],$$

où η_k et τ_k sont respectivement le **taux d'adaptation** et le **terme de décroissance** associés au niveau k . Il n'est pas nécessaire d'augmenter subitement l'échelle de ces pondérations pour passer directement du micro au macro. Les paliers intermédiaires laissent le temps à chaque niveau d'atteindre un quasi-équilibre local, évitant un chaos dû à un trop grand saut d'échelle.

B. Clusters Semi-Globaux et Régulation de la Complexité

Un cluster semi-global représente un groupe de dimension intermédiaire, par exemple quelques dizaines ou centaines de nœuds sur un total de milliers. Sur le plan **dynamique**, ces regroupements agissent comme un filtre. Les micro-clusters du niveau inférieur $\mathcal{N}_\alpha^{(1)}$ peuvent fusionner ou se scinder pour former des ensembles plus vastes, mais seulement si leur **synergie** demeure assez élevée lors des itérations d'**auto-organisation**. Les clusters semi-globaux ainsi créés ne sont ni trop petits ni trop grands, ce qui facilite leur réorganisation interne et leur éventuelle fusion en un macro-niveau ultérieur.

La compétition ou la collaboration entre clusters semi-globaux peut être modélisée par des pondérations $\omega_{\alpha,\beta}^{(1)}$. Par exemple, si $\mathcal{N}_\alpha^{(1)}$ et $\mathcal{N}_\beta^{(1)}$ exhibent une forte compatibilité, la mise à jour

$$\omega_{\alpha,\beta}^{(1)}(t+1) = \omega_{\alpha,\beta}^{(1)}(t) + \eta_1 \left[S_1(\alpha,\beta) - \tau_1 \omega_{\alpha,\beta}^{(1)}(t) \right]$$

peut conduire, à terme, à l'union de ces deux regroupements si cette compatibilité dépasse un certain seuil. Inversement, de faibles valeurs de synergie pousseront à la désactivation ou au relâchement des liens. Les clusters semi-globaux les plus stables constitueront alors le socle pour la **phase suivante** d'agrégation.

C. Synchronisation des Paliers et Ordonnancement Temporel

Les différents niveaux ne doivent pas être mis à jour simultanément à chaque instant, sous peine d'induire des oscillations ou conflits entre le micro et le macro. Une stratégie courante de **synchronisation** consiste à laisser le micro-niveau évoluer pendant un certain nombre d'itérations, puis à figer ses résultats, à construire les clusters de niveau supérieur et à itérer à ce niveau intermédiaire. Ce découpage temporel peut se formaliser par un calendrier en étapes :

- Le **micro-niveau** (niveau 0) applique la règle de mise à jour durant T_0 itérations, produisant des groupes $\{\mathcal{C}_\alpha^{(0)}\}$.
- On agrège ces groupes en super-nœuds $\{\mathcal{N}_\alpha^{(1)}\}$. Les pondérations $\omega_{\alpha,\beta}^{(1)}$ sont initialisées par une opération de synthèse (somme, moyenne, etc.) sur les liens intergroupes de niveau 0.

- Le **niveau intermédiaire** exécute alors la mise à jour pendant T_1 itérations, aboutissant à la formation ou la stabilisation de clusters semi-globaux.
- Enfin, si nécessaire, un passage au **macro-niveau** (niveau 2) peut se faire après un nombre d'itérations suffisant pour que le niveau 1 se soit **stabilisé**.

Dans cette séquence, chaque échelon (k) transmet ses informations à ($k + 1$) lorsque le système local est parvenu à un état suffisamment cohérent. Cela évite de multiplier des allers-retours entre des niveaux qui n'ont pas eu le temps de converger.

D. Cas d'Utilisation et Exemples Concrets

Les **clusters semi-globaux** apparaissent dans un large éventail de systèmes multi-niveau. Dans le domaine des réseaux distribués (cloud et edge), le palier intermédiaire peut correspondre à des **data centers régionaux** qui rassemblent les ressources de plusieurs sites. En robotique, le niveau 1 regroupe des sous-équipes locales pour former des unités de taille moyenne, avant de s'associer en un méta-groupe macro si une mission globale l'exige. Dans des approches inspirées de la **cognition** ou du **cerveau**, des sous-réseaux peuvent émerger en tant qu'aires corticales locales et coopérer à plus grande échelle par paliers successifs.

Dans tous ces scénarios, la dynamique d'échelle intermédiaire offre une **stabilisation hiérarchique**. Plutôt que de confronter immédiatement l'ensemble des entités à une vision macro, on laisse les synergies locales s'organiser en regroupements robustes. Les paliers semi-globaux constituent ainsi une étape de maturation indispensable, limitant le risque de configurations fragiles ou mal ajustées lorsqu'on atteindra le niveau englobant.

6.4.3.2. Coordination Latérale entre Clusters de Niveaux Proches : Comment Interagir avec le Cluster Voisin

Introduction. Dans un **Deep Synergy Learning** multi-niveau, il est courant de focaliser la communication sur l'axe vertical, que ce soit vers le haut (bottom-up) ou vers le bas (top-down). Toutefois, au sein d'un **même** palier de la hiérarchie (par exemple, un niveau intermédiaire ou semi-global), plusieurs **clusters** peuvent coexister. La question se pose alors de savoir comment ces clusters, qui partagent la même échelle, interagissent **directement** entre eux sans nécessairement faire appel au niveau supérieur ou au niveau micro. Cette **coordination latérale** s'avère cruciale pour résoudre d'éventuels conflits locaux, permettre des fusions partielles ou faciliter des échanges de ressources à l'échelle intermédiaire. Les paragraphes suivants détaillent la nécessité et les modalités de cette interaction horizontale, ainsi que les bénéfices et implications sur la dynamique globale.

A. Raison d'Être de la Coordination Latérale

Au sein d'un palier donné, il est possible que plusieurs super-nœuds $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$ se retrouvent dans une situation de **complémentarité** ou de **compétition**. Des groupes voisins peuvent occuper des régions adjacentes (dans un contexte de robotique) ou partager des caractéristiques similaires (dans une application de clustering de données), et se retrouvent incités à établir des mécanismes de dialogue direct. L'idée est de gérer en local leurs **relations mutuelles**, plutôt que de solliciter le

niveau macro pour chaque ajustement mineur. Cela accélère la réaction à des défis communs et décharge la couche supérieure, qui se concentre alors sur des stratégies plus globales.

S'il n'existe pas de **coordination latérale**, toute communication entre clusters du même palier devrait être relayée à travers le niveau $k + 1$. Un tel détour crée un surcoût, perturbe la réactivité locale et contribue à la surcharge d'information chez le macro-nœud. La mise en place de **liens horizontaux**, avec leurs pondérations $\omega_{\alpha,\beta}^{(k)}$, constitue un moyen efficace de rendre ces échanges plus directs, plus rapides et mieux adaptés à la dimension semi-globale.

B. Formalisation : Liens Latéraux au Même Palier

Au niveau k , chaque super-nœud $\mathcal{N}_\alpha^{(k)}$ est potentiellement connecté aux autres super-nœuds $\mathcal{N}_\beta^{(k)}$. On peut donc définir une **matrice** $\omega^{(k)}$ dont les éléments $\omega_{\alpha,\beta}^{(k)}$ sont sujets à une règle de plasticité du même type que celle utilisée aux niveaux micro. Une loi de mise à jour classique est :

$$\omega_{\alpha,\beta}^{(k)}(t+1) = \omega_{\alpha,\beta}^{(k)}(t) + \eta_k [S_k(\alpha, \beta) - \tau_k \omega_{\alpha,\beta}^{(k)}(t)].$$

Le **taux d'adaptation** η_k et la **constante de décroissance** τ_k peuvent être spécifiques au palier k . La *fonction de synergie* $S_k(\alpha, \beta)$ correspond alors à un **score** reflétant la compatibilité entre les regroupements $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$. Ce score peut être dérivé de la somme ou de la moyenne des liens inter-entités au niveau $k - 1$, de la proximité spatiale, de la complémentarité de compétences, ou de toute autre donnée permettant d'évaluer l'intérêt de coopérer.

L'existence de ces pondérations $\omega_{\alpha,\beta}^{(k)}$ fait du palier k un **réseau** de super-nœuds s'auto-organisant selon le même principe DSL, où les clusters les plus synergiques renforcent leurs liens tandis que les autres s'affaiblissent progressivement. Les super-nœuds peuvent alors se **rapprocher**, se **fuser** ou maintenir une simple relation d'information mutuelle. Cette auto-organisation horizontale rend la couche k plus autonome et plus stable face à des phénomènes locaux.

C. Collaboration, Concurrence et Frontières

D'un point de vue dynamique, deux super-nœuds $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$ peuvent :

- **Fuser** s'ils constatent que leur synergie latérale $\omega_{\alpha,\beta}^{(k)}$ croît jusqu'à dépasser un certain seuil, ou si leurs entités $\{\mathcal{E}_i\}$ montrent un recouvrement élevé.
- **Coexister** paisiblement en maintenant une frontière clairement définie, peut-être par une relation modérée (ni trop basse, ni trop élevée).
- **Entrer en compétition** pour des ressources partagées ou si leurs objectifs se recouvrent, ce qui peut conduire à une réduction de $\omega_{\alpha,\beta}^{(k)}$ via la dynamique de décroissance.

Un **exemple** concret est celui d'un niveau intermédiaire regroupant des équipes robotisées. Deux super-nœuds voisins peuvent être naturellement amenés à échanger des robots, des informations ou à se répartir des sous-zones d'exploration. Dès lors, la pondération $\omega_{\alpha,\beta}^{(k)}$ augmente si le partage se révèle mutuellement bénéfique, reflétant leur **coopération**. Dans le cas contraire, si des **conflits**

surgissent (redondance de ressources, incompréhensions de mission), la valeur de $\omega_{\alpha,\beta}^{(k)}$ tend à diminuer, signifiant un cloisonnement plus net.

D. Avantages Généraux et Implications

Stabilisation latérale. Autoriser cette coordination horizontale évite que chaque super-nœud fonctionne en vase clos. Un cluster semi-global peut résoudre localement ses ajustements avec un voisin, limitant ainsi les remontées de demandes vers le niveau macro. Cette **stabilisation latérale** contribue à ce que la couche k soit plus robuste et cohérente avant de se projeter vers une agrégation de plus grande échelle.

Réduction de la surcharge au niveau supérieur. Dès lors que les clusters parviennent à trouver un consensus sur leur frontière commune, ils n’ont pas besoin de mobiliser le palier macro ($k + 1$). Le niveau supérieur est ainsi déchargé de multiples querelles de détail et peut se consacrer à la vision plus large et à la stratégie globale.

Complexité algorithmique. La gestion des liens latéraux $\omega_{\alpha,\beta}^{(k)}$ introduit une **matrice** de taille proportionnelle au nombre de super-nœuds au palier k . Bien que cela représente un coût en plus, ce coût reste très inférieur à l’échelle micro si le palier k agrège déjà plusieurs entités dans un même nœud. Il s’agit donc d’un **mini-réseau** plus léger à manipuler que le réseau initial complet.

Convergence globale. Sur le plan mathématique, la mise à jour simultanée des liaisons **verticales** (entre niveaux différents) et **horizontales** (à l’intérieur d’un même niveau) s’apparente à un **double mécanisme** d’auto-organisation. Le niveau k peut stabiliser la structure dans sa couche, tandis que la structure agrégée (ou filtrée) est progressivement transmise à ($k + 1$). Une telle gestion hiérarchique, associée à la coordination latérale, favorise une convergence plus fluide et évite les incohérences ou oscillations qui surviendraient si tout devait être uniformément décidé au niveau macro.

6.4.3.3. Détection et gestion de conflits : si deux macro-nœuds aspirent la même entité

Dans la dynamique **multi-niveau** d’un SCN (Synergistic Connection Network), il peut arriver qu’au **niveau macro**, deux super-nœuds $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$ se disputent le **même** ensemble local, c’est-à-dire qu’une ou plusieurs entités $i \in \{\mathcal{E}_i\}$ (au niveau micro) se retrouvent “attirées” par les deux macro-clusters. Ce phénomène engendre un **conflit**, car il devient incertain quel macro-nœud absorbera ou contrôlera réellement l’entité i . Si les synergies ω vers plusieurs macro-nœuds sont comparables, cela peut provoquer des oscillations ou une incohérence dans l’assignation de l’entité. Cette section présente **comment** détecter ce type de conflit (section A) et **comment** le gérer mathématiquement (section B), afin d’assurer la stabilité globale du **DSL** (Deep Synergy Learning).

A. Détection de Conflits : deux macro-nœuds visant la même entité

Au **niveau 0** (micro), une entité \mathcal{E}_i peut avoir un ensemble de pondérations $\{\omega_{i,j}\}$ la reliant aux entités $\mathcal{C}_\alpha^{(0)}$ ou $\mathcal{C}_\beta^{(0)}$.

Après agrégation (chap. 6.2.2), on forme au niveau k deux **macro-nœuds** $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$. Chacun d'eux englobe un certain sous-ensemble d'entités (ou super-nœuds du palier $k - 1$).

Il se peut qu'une *même* entité \mathcal{E}_i (au palier micro) apparaisse partiellement dans la “zone” de $\mathcal{N}_\alpha^{(k)}$ et de $\mathcal{N}_\beta^{(k)}$. Par exemple, ses liens $\omega_{i,j}$ avec les membres de \mathcal{N}_α sont élevés, mais $\omega_{i,j}$ avec certains membres de \mathcal{N}_β le sont aussi.

Si la *même* entité \mathcal{E}_i est simultanément attirée par $\mathcal{N}_\alpha^{(k)}$ et $\mathcal{N}_\beta^{(k)}$, le SCN risque une incohérence. L'entité i ne peut appartenir à deux macro-nœuds distincts en même temps, sauf si le modèle autorise explicitement un recouvrement partiel. Dans la plupart des schémas, on privilégie une partition stricte ou un recouvrement contrôlé afin d'éviter des conflits dans l'assignation et la mise à jour des pondérations.

Sur le plan **mathématique**, on peut poser l'**indicateur** $\zeta(i, \alpha)$ qui vaut 1 si $i \in \mathcal{N}_\alpha$ et 0 sinon. Un conflit se détecte si $\zeta(i, \alpha) = \zeta(i, \beta) = 1$ pour $\alpha \neq \beta$.

Souvent, la **détection** se fait en observant la **somme des pondérations** entre i et chacun des macro-nœuds :

$$\Omega(i, \alpha) = \sum_{j \in \mathcal{N}_\alpha} \omega_{i,j}^{(0)}, \quad \Omega(i, \beta) = \sum_{j \in \mathcal{N}_\beta} \omega_{i,j}^{(0)}.$$

Si $\Omega(i, \alpha) \approx \Omega(i, \beta)$ et les deux dépassent un certain seuil, un conflit d’“aspiration” se produit.

B. Gestion Mathématique des Conflits : Choix, Partage, ou Inhibition

Une entité i ne peut appartenir *exclusivement* qu'à un **macro-nœud**. Lorsqu'on détecte un conflit ($\Omega(i, \alpha) \approx \Omega(i, \beta)$), on applique une **règle** de résolution :

$$\zeta(i, \alpha) = 1, \zeta(i, \beta) = 0 \quad \text{ou} \quad \zeta(i, \alpha) = 0, \zeta(i, \beta) = 1,$$

selon lequel des deux dépasse le plus nettement le seuil, ou selon un *tirage aléatoire pondéré* (si Ω est quasi identique).

On peut introduire un **terme** Δ_{conflict} qui “coupe” la liaison la plus faible, par ex. :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) - \gamma(\cdot),$$

pour les liens orientés vers le macro-nœud qu'on *rejette*.

Dans certains modèles **non** exclusifs, on autorise un recouvrement des macro-nœuds (une entité peut être dans \mathcal{N}_α et \mathcal{N}_β) si c'est *logiquement* possible (ex. un robot fait partie de 2 équipes).

Mais cette *co-appartenance* peut introduire de la **redondance** et compliquer la logique d'agrégation. On implémente alors un **poids d'appartenance** $\zeta(i, \alpha) \in [0,1]$ qui mesure à quel degré i contribue à \mathcal{N}_α . S'il y a un conflit, on *répartit* ζ sur plusieurs macro-nœuds.

La troisième méthode consiste à considérer que si le système **macro** (niveau k) estime que \mathcal{N}_α possède déjà suffisamment d'entités, il impose une **inhibition descendante** ($\Delta_{\text{down}}^{(\alpha)} < 0$) sur les liaisons reliant i à \mathcal{N}_α , ce qui incite i à se replier vers \mathcal{N}_β .

Le flux descendant du macro-niveau α vers l'entité i (au niveau micro) est :

$$\Delta_{\text{down}}^{(\alpha)}(i) = -\varepsilon \quad \text{si conflit.}$$

Cela **affaiblit** la synergie $\omega_{i,j}$ pour $j \in \mathcal{N}_\alpha$, poussant \mathcal{E}_i à basculer vers \mathcal{N}_β .

C. Stabilisation et Résolution Finale

Au fil des itérations, une entité i se **fixe** dans un macro-nœud \mathcal{N}_α ou un autre, au fur et à mesure que les **liens** (micro-liaisons $\omega_{i,j}$) s'ajustent en réponse aux signaux top-down.

Quand la **convergence** s'opère, le conflit disparaît et i s'intègre pleinement ou majoritairement à un cluster macro, stabilisant ainsi la structure du SCN.

S'il n'y a pas de **règle** claire pour briser l'égalité $\Omega(i, \alpha) \approx \Omega(i, \beta)$, on peut voir i osciller entre \mathcal{N}_α et \mathcal{N}_β . D'où l'intérêt d'un *petit* mécanisme de **brisure de symétrie** (ex. un ajout aléatoire, un tirage stochastique) ou d'un *facteur d'hystérésis* (on ne bascule pas si la différence est $< \delta$) pour éviter le “flip-flop”.

Optionnellement, on peut définir une **énergie** locale $E_i(\alpha, \beta)$ associée au fait que i appartient à \mathcal{N}_α et \mathcal{N}_β . Si ce double appartenance coûte cher (haute énergie), le système finit par minimiser l'énergie en choisissant un *unique* cluster macro pour i .

D. Résumé et Impact

Quand deux macro-nœuds aspirent la même entité, cela révèle souvent une **ambiguïté** ; l'entité partage une forte synergie avec deux groupes différents. C'est fréquent si le **réseau** n'impose pas une partition stricte ou si la distribution initiale rend l'entité “hybride”.

La **gestion** de ce conflit est donc un aspect **naturel** de la coordination multi-niveau en DSL.

Une première approche consiste à appliquer un **choix excluant**, qu'il soit forcé ou stochastique, afin que l'entité finisse par appartenir à un seul macro-nœud. Une autre possibilité repose sur le **partage**, si le modèle autorise l'overlapping, ce qui implique l'introduction d'une pondération d'appartenance $\zeta(i, \alpha) \in [0,1]$. Enfin, une troisième stratégie repose sur l'**inhibition**, où l'entité est repoussée hors d'un macro-nœud par un mécanisme de feedback négatif.

En combinant un *terme top-down* et une *loi DSL* locale, on garantit qu'un conflit se **résoudra** généralement, tant que les paramètres (η, τ, γ) sont choisis pour **éviter** les oscillations permanentes. Des petits ajouts de *bruit* ou de *random break ties* aident aussi à la *brisure de symétrie*.

6.5. Dynamique et Algorithmes Multi-Échelle

Dans de nombreux **SCN** (Synergistic Connection Networks), l'organisation **multi-niveau** n'est pas juste un concept statique, elle émerge et se peaufine *dynamiquement* grâce à l'application d'**algorithmes** spécifiques. La section 6.5.1 décrit l'**agrégation progressive** (bottom-up), principe selon lequel on forme des super-nœuds de plus en plus grands (micro \rightarrow méso \rightarrow macro). Ensuite, 6.5.2 verra la **division** (top-down), et 6.5.3 une **approche hybride**. Enfin, 6.5.4 discutera des aspects algorithmiques concrets et des paramètres clés.

6.5.1. Agrégation Progressive (Bottom-Up)

Lorsqu'on parle de **DSL** (Deep Synergy Learning) à plusieurs paliers, l'une des logiques classiques consiste à **agréger** progressivement les entités de bas en haut, d'abord on détecte des **micro-clusters**, puis on fusionne ces micro-clusters pour former des **super-nœuds**, et ainsi de suite jusqu'au "macro-nœud" ou niveau global.

6.5.1.1. Détecter des Micro-Clusters, en Créer un "Super-Nœud" ; puis Détecter des Super-Nœuds Cohérents pour un "Macro-Nœud", etc.

Introduction. Dans une architecture **Deep Synergy Learning** multi-niveau, l'un des mécanismes les plus naturels pour construire une **hiérarchie** consiste à procéder de manière bottom-up ; on commence par détecter, au **niveau micro**, des **micro-clusters** suffisamment cohérents. On agrège ensuite chacun de ces groupes en un super-nœud pour le palier suivant, puis on répète le même principe de détection/agrégation afin de former des *macro-clusters* ou un *macro-nœud* unique. Les paragraphes qui suivent détaillent la démarche et les fondements mathématiques associés.

A. Détection de Micro-Clusters au Niveau 0

On se place au **niveau 0**, celui des entités élémentaires $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. Chaque paire (i, j) est associée à un poids $\omega_{i,j}^{(0)}$ et suit une **règle DSL** inspirée de la plasticité synaptique. Par exemple, une règle additive canonique peut s'écrire :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_0 [S_0(i, j) - \tau_0 \omega_{i,j}^{(0)}(t)],$$

où η_0 est un **taux d'apprentissage**, τ_0 un **terme de décroissance**, et $S_0(i, j)$ une **fonction de synergie** mesurant la similarité ou l'utilité mutuelle des entités \mathcal{E}_i et \mathcal{E}_j . Après un certain nombre d'itérations, on voit émerger des **groupes** (micro-clusters) dont les pondérations internes $\omega_{i,j}^{(0)}$ restent élevées et se stabilisent, tandis que les liens extérieurs faiblissent ou s'annulent.

Lorsqu'un **sous-ensemble** $\mathcal{C}_\alpha^{(0)} \subset \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ présente une **cohésion** suffisamment forte, on peut le reconnaître comme un **micro-cluster** stable. À partir de ce moment, on peut réduire l'information qu'il contient à un **super-nœud** $\mathcal{N}_\alpha^{(1)}$, de telle sorte que le **niveau 0** se voit partiellement "condensé" pour constituer le **niveau 1**.

B. Construction d'un Super-Nœud

Une fois que l'on a détecté un micro-cluster $\mathcal{C}_\alpha^{(0)}$, on crée un **super-nœud** $\mathcal{N}_\alpha^{(1)}$. Pour former l'ensemble des liaisons $\omega_{\alpha,\beta}^{(1)}$ au **niveau 1**, on réalise une **agrégation** des poids $\omega_{i,j}^{(0)}$. Une manière générique de procéder consiste à choisir une fonction Ψ et à poser :

$$\omega_{\alpha,\beta}^{(1)} = \Psi\left(\{\omega_{i,j}^{(0)} \mid i \in \mathcal{C}_\alpha^{(0)}, j \in \mathcal{C}_\beta^{(0)}\}\right).$$

La fonction Ψ peut être une **moyenne**, une **somme**, un **maximum**, ou toute autre mesure appropriée pour condenser les poids $\omega_{i,j}^{(0)}$ liant le cluster $\mathcal{C}_\alpha^{(0)}$ au cluster $\mathcal{C}_\beta^{(0)}$. Cette opération de filtrage ou de synthèse garantit que l'on passe d'un nombre potentiellement élevé de liaisons au niveau micro ($O(n^2)$) à un nombre bien plus faible de liaisons au niveau 1. De plus, cette **réduction** souligne uniquement les liens les plus significatifs entre les clusters du niveau inférieur.

C. Détection de Super-Nœuds Cohérents

Au **niveau 1**, on se retrouve avec un ensemble de super-nœuds $\{\mathcal{N}_1^{(1)}, \dots, \mathcal{N}_{m_1}^{(1)}\}$ et une matrice de pondérations $\omega_{\alpha,\beta}^{(1)}$. On peut appliquer la **même** règle DSL (ou une variante) pour faire évoluer ces poids :

$$\omega_{\alpha,\beta}^{(1)}(t+1) = \omega_{\alpha,\beta}^{(1)}(t) + \eta_1 \left[S_1(\alpha, \beta) - \tau_1 \omega_{\alpha,\beta}^{(1)}(t) \right].$$

Ici, η_1 et τ_1 sont des paramètres d'apprentissage fixés pour le niveau 1, et $S_1(\alpha, \beta)$ est la nouvelle fonction de synergie, reflétant la proximité ou la compatibilité entre super-nœuds $\mathcal{N}_\alpha^{(1)}$ et $\mathcal{N}_\beta^{(1)}$. Cette dynamique auto-organisée permet de repérer à ce **deuxième palier** des regroupements $\{\mathcal{C}_p^{(1)}\}$ que l'on convertit ensuite en super-nœuds $\mathcal{N}_p^{(2)}$ pour le **niveau 2**, et ainsi de suite. En itérant le processus, on peut progressivement atteindre un **macro-niveau** (niveau K) où il ne reste plus qu'un petit nombre de super-nœuds de grande taille, voire un seul **macro-nœud** global.

D. Progression jusqu'au Macro-Nœud

La répétition de l'agrégation forme une **cascade** :

- Le niveau 0 détecte des micro-clusters $\mathcal{C}_\alpha^{(0)}$.
- Les micro-clusters sont condensés en super-nœuds $\mathcal{N}_\alpha^{(1)}$.
- Le niveau 1 détecte à son tour des clusters $\mathcal{C}_p^{(1)}$ parmi les super-nœuds déjà existants.
- Ces regroupements sont convertis en super-nœuds $\mathcal{N}_p^{(2)}$.
- Le niveau 2 poursuit le même mouvement, etc.

Ce **principe** bottom-up s'arrête lorsque le nombre de super-nœuds restants est faible (par exemple, un unique macro-nœud englobant tout), ou lorsque certains **critères** d'arrêt (seuil de similarité, contrainte de taille, etc.) sont atteints. La complexité $O(n^2)$ potentiellement nécessaire pour mettre

à jour toutes les liaisons au niveau micro se résorbe au profit d'un nombre plus restreint de liaisons $O(m^2)$ au fur et à mesure que $m \ll n$.

6.5.1.2. Avantages : Construction Organique, Moins de Paramétrage Initial

Introduction.

La stratégie **bottom-up** d'agrégation progressive, décrite précédemment pour bâtir une hiérarchie de super-nœuds (section 6.5.1.1), comporte des avantages significatifs lorsque l'on souhaite structurer un **Synergistic Connection Network (SCN)** en plusieurs échelles de manière flexible. Contrairement aux approches imposant dès le départ un découpage (top-down) ou un nombre fixe de partitions (clustering global), l'agrégation **bottom-up** offre une **construction organique** qui épouse la dynamique locale du **DSL** et réclame un **paramétrage** minimal. Les développements ci-après mettent en évidence les principaux bénéfices, tout en situant cette démarche par rapport à d'autres méthodes.

A. Construction Organique

La première caractéristique marquante de l'agrégation **bottom-up** tient au fait qu'elle est en phase directe avec la **dynamique** locale du Deep Synergy Learning. Au **niveau micro**, on met en œuvre la règle DSL :

$$\omega_{i,j}^{(0)}(t+1) = \omega_{i,j}^{(0)}(t) + \eta_0 [S_0(i,j) - \tau_0 \omega_{i,j}^{(0)}(t)].$$

Cette équation favorise l'émergence de **micro-clusters** $\{\mathcal{C}_\alpha^{(0)}\}$ qui se forment dès lors que certaines paires ou triplets d'entités présentent une synergie récurrente. Contrairement à une coupe arbitraire, ces groupes se **stabilisent** naturellement sous l'effet des renforcements et des décroissances sélectives de $\omega_{i,j}$. L'agrégation progressive prend alors ces micro-clusters comme points de départ et les élève au palier supérieur sous forme de super-nœuds, sans brusquer la progression de l'auto-organisation locale.

Une fois les micro-clusters détectés, on ne "saute" pas immédiatement à un macro-niveau unique, on laisse la possibilité de créer des **niveaux intermédiaires** (méso-niveaux), répétant la mise à jour DSL. Les fusions successives reproduisent une logique de **croissance progressive**, proche de mécanismes observés dans des contextes biologiques ou sociologiques, où de petits groupements se consolident en sous-communautés avant d'aboutir, le cas échéant, à des structures englobantes plus vastes.

Cette architecture bottom-up se révèle extrêmement **adaptative**. Si de nouveaux liens $\omega_{i,j}$ apparaissent (ou se renforcent) au niveau micro, cela provoque la création (ou la reconfiguration) de clusters. Ces changements locaux se répercutent naturellement aux paliers supérieurs, lesquels peuvent également réajuster leurs super-nœuds ou fusionner des groupes si la synergie l'exige. La **souplesse** qui en découle est particulièrement cruciale dans des environnements évolutifs ou soumis à des flux de données continus.

B. Moins de Paramétrage Initial

Contrairement aux approches top-down, où il faut généralement définir un **nombre de partitions** dès le début (ex. dire “on veut 5 clusters” ou “on découpe en 3 niveaux de taille égale”), l’agrégation **bottom-up** se limite à un **faible** nombre de paramètres.

Les seuls éléments imposés concernent la **règle de mise à jour** (taux η_0 , décroissance τ_0 , etc.) et la **fonction de synergie** $S_0(i, j)$. Éventuellement, on peut définir un **seuil** pour reconnaître qu’un groupe local est assez cohésif, mais sans imposer la forme ou la taille exacte de chaque cluster.

Il n’est pas nécessaire de fixer a priori le nombre exact de clusters, ni même de préciser la profondeur de la hiérarchie. Les niveaux intermédiaires surgissent **spontanément** dès lors que des sous-ensembles \mathcal{C}_α se distinguent et sont jugés suffisamment stables pour être agrégés. La hiérarchie finale – pouvant s’arrêter à 1, 2, 3 ou plus de niveaux – est ainsi **découverte** plutôt qu’imposée.

Lorsque le partitionnement est dicté en amont, on court le danger de forcer des **découpages artificiels**, mal adaptés à la structure réelle des synergies $\omega_{i,j}$. L’approche **bottom-up** évite ces erreurs en laissant la dynamique et les degrés de cohésion déterminer les regroupements. Les grands clivages dans le réseau émergent **naturellement** s’ils sont effectivement présents, tandis que les liens de force intermédiaire aboutissent à des fusions partielles ou restent séparés.

C. Implications Pratiques

La mise en place d’une agrégation **bottom-up** se traduit par une séquence d’étapes :

1. Boucles d’Itération au Niveau Micro.

Le réseau exécute la règle DSL pendant un certain nombre d’itérations, renforce les liens synergiques et affaiblit les autres.

2. Détection/Actualisation des Micro-Clusters.

On identifie les groupes ayant atteint une cohésion suffisante.

3. Construction du Palier Supérieur.

Chaque micro-cluster devient un super-nœud, et l’on agrège les poids $\omega^{(0)}$ pour former $\omega^{(1)}$.

4. Mise à Jour au Niveau Méso.

Le même procédé d’**auto-organisation** se poursuit, cette fois entre super-nœuds, jusqu’à former de nouveaux regroupements $\mathcal{C}_\alpha^{(1)}$.

5. Prolongement ou Arrêt.

On répète les paliers tant qu’il reste un intérêt à poursuivre l’agrégation. S’il ne subsiste qu’un petit nombre de super-nœuds ou un unique **macro-nœud**, on stoppe la progression.

Cette stratégie assure une **scalabilité** naturelle en réduisant progressivement le nombre de nœuds par agrégation des éléments fortement connectés, évitant ainsi un traitement constant à l’échelle

globale du réseau. Elle permet également une **synchronisation ascendante** en attendant la stabilisation du niveau micro avant de transférer la structure au palier supérieur, garantissant ainsi une cohérence hiérarchique renforcée.

6.5.1.3. Inconvénients : Besoin d’Algorithmes “Greedy” ou Heuristiques, Potentiels Merges Successifs

La démarche **bottom-up** (agrégation progressive) exposée dans les sections précédentes (6.5.1.1 et 6.5.1.2) apporte une flexibilité et une réduction du paramétrage initial, mais n’est pas exempte de limites. Elle repose souvent sur des **algorithmes** de fusion successifs, de type *greedy* ou **heuristiques**, et peut ainsi s’avérer délicate dans certaines circonstances. Les paragraphes suivants mettent en évidence deux grandes familles d’inconvénients. D’une part, l’impossibilité de garantir une fusion optimale, ce qui impose le recours à des heuristiques. D’autre part, le risque de **merges** itératifs entraînant des configurations sous-optimales en raison d’une **erreur cumulative** au fil des agrégations successives.

A. Besoin d’Algorithmes Greedy ou Heuristiques

Dans un **SCN** (Synergistic Connection Network) de grande taille, trouver la **meilleure** manière de fusionner les micro-clusters est un problème qui, sous de nombreuses variantes (community detection, clustering hiérarchique optimal, etc.), est **NP-difficile**. Autrement dit, l’exploration exhaustive de toutes les possibilités de partitions ou de merges devient **inabordable** à mesure que le nombre d’entités croît.

Vouloir choisir à chaque étape la fusion localement “idéale” – ou, plus encore, vouloir optimiser globalement l’agencement des fusions – se heurte à une **explosion** combinatoire. Cela rend quasi impossible, en pratique, l’obtention d’une **solution exacte** dès lors que n dépasse quelques dizaines ou centaines.

Pour contourner cette complexité, on recourt typiquement à des **procédures** itératives, dites *greedy merges*, où l’on :

- Cherche les deux clusters \mathcal{C}_α et \mathcal{C}_β qui présentent la **plus forte synergie** (ou dont la fusion entraîne la **meilleure réduction de coût** local),
- Fusionne ces deux ensembles en un **super-cluster** $\mathcal{C}_{\alpha \cup \beta}$,
- Met à jour la structure, puis répète l’opération jusqu’à ce qu’on atteigne un niveau jugé suffisant (arrêt par critère de taille, de niveau, ou de cohésion).

De nombreux schémas d’**agrégation hiérarchique** (single linkage, complete linkage, average linkage, etc.) s’appuient sur cette philosophie, en substituant un **score** de proximité ou de similarité entre clusters. Dans un **SCN**, les **pondérations** $\omega_{\alpha, \beta}$ permettent d’évaluer la force d’attraction entre clusters. Plus $\omega_{\alpha, \beta}$ est élevée, plus la fusion des clusters α et β devient probable.

Il s’agit cependant d’un **choix local**, qui ne garantit pas de maximiser la cohésion globale du réseau (ou de minimiser toute fonction de coût plus étendue).

B. Risques de Merges Successifs et Erreurs Cumulatives

Parce qu'on s'appuie sur une suite de **décisions locales** (fusionner deux clusters \mathcal{C}_α et \mathcal{C}_β si leur lien est jugé suffisant), il y a un risque d'**erreurs** successives qui finissent par s'accumuler dans la configuration finale.

La règle *greedy* peut pousser à fusionner rapidement deux clusters dont la synergie est pour l'instant dominante, mais qui pourraient en réalité être mieux agencés (ou scindés) si l'évolution du réseau se prolongeait. On peut voir cela comme un *verrouillage* où, une fois la fusion réalisée, le super-cluster $\mathcal{C}_{\alpha \cup \beta}$ reste stable et n'est que rarement redécomposé dans le même algorithme bottom-up.

Des “fusions hâtives” de ce type peuvent enfermer la hiérarchie dans un état sous-optimal.

Dans la plupart des implémentations, l'**agrégation progressive** ne prévoit pas de “**dé-fusion**” ou de *split* durant le même cycle d'ascension. Il faudrait recourir à des *extensions* plus complexes (chap. 6.5.2 sur des divisions possibles, ou des mécanismes top-down correctifs) pour rétablir une répartition plus adaptée.

Sans cette possibilité de correction, chaque petit choix local vient s'inscrire dans la structure globale, risquant de créer une **erreur cumulative** au fil des paliers.

Si l'on associe à la partition du réseau une **énergie** (ou “coût”) \mathcal{E} à minimiser, tout algorithme de fusion local vise à réduire $\Delta\mathcal{E}$ sur l'instant. Or, un optimum local n'est pas nécessairement un optimum global, et, en multipliant les “petites optimisations” locales, on peut aboutir à un état final nettement plus élevé en \mathcal{E} que la **configuration** la plus favorable.

C. Conséquences Pratiques et Possibles Solutions

Ces inconvénients ne rendent pas l'agrégation *bottom-up* inopérante, mais soulignent la nécessité de précautions ou de *mécanismes complémentaires*.

Il est fréquent d'introduire un **feedback descendant** (cf. 6.4) au cours duquel un niveau supérieur peut suggérer la re-scission d'un cluster si la macro-analyse repère des incohérences. Cela donne un algorithme plus complet, où la construction bottom-up est **assouplie** par des corrections top-down.

Dans des architectures DSL complexes, ce double flux (ascendant et descendant) permet de retarder ou d'inverser certaines unions malheureuses.

S'il n'existe pas de mécanisme de “re-diviser”, on peut subir d'importants effets d'inertie. Une fusion inappropriée à la première étape se répercutera dans les paliers ultérieurs, menant parfois à une structure loin de l'optimum. À l'inverse, si on autorise trop facilement la division, il y a un risque d'oscillations (fusions défusions récurrentes).

Des **seuils** ou paramètres de stabilisation (hystérésis, temporalité de mise à jour) aident à limiter ces phénomènes.

En dépit de ces limites, un algorithme *greedy* de fusion hiérarchique reste souvent **très rapide** et **simple** à mettre en œuvre. Il donne des résultats satisfaisants dans la majorité des scénarios pratiques, pourvu qu'on accepte une configuration “raisonnablement bonne” plutôt que *parfaitement optimale*.

6.5.2. Division ou Zoom (Top-Down)

En **complément** de la logique d'agrégation (bottom-up) décrite en (6.5.1), certains **algorithmes** multi-niveaux adoptent ou complètent l'approche **top-down**, où l'on **part** d'un **macro-cluster** global (ou d'un ensemble restreint de grands clusters) pour ensuite **segmentation** s'il apparaît que la **cohésion** interne est insuffisante. La section 6.5.2.1 introduit la notion d'une **approche fractale descendante**, qui adopte une perspective inverse : au lieu de fusionner progressivement, on opère un **zoom progressif**, divisant les structures globales en sous-ensembles plus fins.

6.5.2.1. Approche Fractale Descendante : Partir d'un Cluster Global et le Segmenter si des Sous-Groupes se Forment

Contrairement à la logique **bottom-up**, qui agrège progressivement des micro-clusters vers des super-nœuds (section 6.5.1), la démarche **top-down** se propose de partir d'un **grand** ensemble englobant toutes les entités et d'y opérer des **divisions** successives dès que l'on détecte des “failles” ou des “sous-groupes” cohésifs. Cette approche se qualifie parfois de “fractale descendante” parce qu'elle reproduit à chaque échelle la même procédure de scission, révélant des **structures** de plus en plus fines. Les paragraphes suivants présentent le principe général, illustrent le lien avec la notion de fractalité, puis discutent les avantages et inconvénients de cette segmentation top-down.

A. Principe d'une Division/Zoom du Macro vers le Micro

La stratégie **top-down** consiste à partir d'un **unique** cluster (macro) qui inclut la totalité des entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. On considère alors ce bloc global comme un “super-nœud” $\mathcal{N}^{(\text{global})}$. À ce stade, si une mesure de cohésion, d'hétérogénéité ou de modularité indique que l'ensemble est trop hétérogène, on procède à une **division** en séparant $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ en plusieurs **sous-clusters** plus homogènes.

On ne part pas d'entités dispersées, mais d'un bloc unique englobant tout. Cela peut correspondre à la situation où l'on sait d'emblée qu'il existe un *objectif commun* ou qu'un macro-niveau coiffe l'ensemble des agents (par exemple, un centre de commande).

Dès lors qu'un critère d'hétérogénéité (ou de “faible synergie interne”) est détecté, on *scinde* le cluster macro en deux (ou plus) sous-blocs. Chaque sous-bloc \mathcal{C}_α , considéré comme un cluster local, peut faire l'objet du **même** examen, se divisant à son tour si nécessaire.

À chaque niveau de division, la **même règle** s'applique. Si un sous-ensemble demeure hétérogène, une nouvelle scission est effectuée. Ce procédé descend petit à petit vers des regroupements de taille moyenne (mésos-niveau), puis éventuellement de petite taille (micro-niveau), tant que la cohésion interne n'est pas jugée satisfaisante.

B. Allusion à une Logique Fractale

Le terme “fractale” se réfère au fait que la même *loi* de division se reproduit à chaque échelle, dévoilant une **auto-similarité** structurale. On peut définir une **fonction** d'hétérogénéité \mathcal{H} mesurant dans quelle mesure un ensemble est “mélangé”. Si \mathcal{H} s'avère trop élevée pour le cluster global, on le coupe en deux blocs, puis on évalue \mathcal{H} dans chacun de ces blocs, et ainsi de suite :

si $\mathcal{H}(\mathcal{C}) > \theta > \Rightarrow$ division de \mathcal{C} en sous-blocs $\mathcal{C}_\alpha, \dots, \mathcal{C}_\beta$.

Cette itération, ou récursion, illustre le principe selon lequel en “zoomant” à un niveau plus fin, la **même** règle de division s’applique. Cela renvoie au concept d’**auto-similarité fractale**, où le “motif” de segmentation se répète à chaque échelle jusqu’à atteindre un seuil de granularité.

C. Exemple d’Illustration : Cortex, Réseaux, etc.

Cortex. Si on envisage un ensemble de neurones occupant une zone cérébrale globale, on peut constater que certains neurones forment un sous-ensemble très interconnecté, ce qui justifie une **scission** dans la structure globale. On isole ce sous-groupe en tant qu’aire plus restreinte, puis on continue le même examen en son sein.

Réseaux (Graphes). Sur un graphe englobant toutes les entités, la détection de **communautés** peut s’effectuer en repérant des coupes ou des **faiblesses** dans la matrice de pondérations ω . Chaque fois qu’on découvre un sous-ensemble de nœuds mieux reliés entre eux qu’au reste du graphe, on scinde. Les sous-ensembles sont ensuite traités comme des graphes à part entière, dans lesquels on peut dénicher des sous-communautés supplémentaires.

Analogie fractale. Dans certains systèmes, cette hiérarchie descendante révèle des *patterns* identiques à différentes échelles, renforçant l’aspect fractal. Les mêmes principes de “cohésion interne” et de “faibles connexions externes” se répètent du niveau macro jusqu’au niveau micro.

D. Schéma Mathématique d’une Division Descendante

On note $\mathcal{N}_{\text{global}}^{(K)}$ le cluster unique contenant tous les indices. On évalue sa “cohérence” ou son hétérogénéité $\mathcal{H}(\mathcal{N}_{\text{global}}^{(K)})$. Si cette valeur reste en deçà d’un seuil θ_K , on décide de **conserver** le bloc entier. Sinon, on le segmente en plusieurs sous-ensembles.

À chaque découpage, un cluster $\mathcal{N}_\alpha^{(k)}$ est subdivisé en deux (ou plusieurs) sous-clusters $\mathcal{N}_{\alpha_1}^{(k-1)}, \mathcal{N}_{\alpha_2}^{(k-1)}, \dots$. On mesure ensuite $\mathcal{H}(\mathcal{N}_{\alpha_i}^{(k-1)})$ pour voir si on doit à nouveau “descendre” d’un cran. Cette itération se poursuit tant que l’on découvre des divisions nécessaires ou que l’on n’a pas atteint la taille souhaitée (niveau micro).

Le processus s’interrompt lorsque tous les *blocs* résultants satisfont un critère de cohésion, ou bien lorsqu’on considère qu’ils sont déjà suffisamment fins (niveau minimal). On obtient alors une hiérarchie descendante où chaque palier résulte de scissions itérées.

E. Avantages d’une Approche Top-Down

Le fait de commencer par un bloc global procure une **vision** macroscopique d’ensemble. Au lieu d’assembler lentement des micro-éléments, on *contrôle* tout de suite la partition du grand ensemble, ce qui peut être avantageux si le système est déjà régi par un cadre commun ou un gestionnaire central.

Dans une démarche descendante, on peut décider à chaque étape comment scinder le bloc (en deux, en trois, etc.) en fonction des **informations** dont on dispose (score de variance, synergie interne, etc.). On évite le risque d’avoir un trop grand nombre de micro-clusters non désirés au départ.

Si la structure du réseau présente une forme d'**auto-similarité**, la segmentation top-down la met en évidence naturellement. En “zoomant” dans le grand ensemble, on retrouve la même loi de division à chaque niveau. Cela peut être particulièrement intéressant dans des contextes où l’échelle macro est connue pour être hétérogène, mais où l’on s’attend à des *patterns* récurrents à des niveaux plus fins.

F. Limites et Paramétrages

Malgré son intérêt, cette approche **top-down** soulève plusieurs questions :

1. Choix du Critère de Division.

Il faut définir comment on décide qu’un bloc est “trop hétérogène”. Cette définition implique la mise en place d’une **fonction** \mathcal{H} (ou d’un critère d’énergie \mathcal{E}) et d’un **seuil** θ . Ces paramètres doivent être ajustés de manière à éviter soit l’absence de division (si θ est trop élevé), soit une segmentation excessive (si θ est trop bas).

2. Décisions de Découpage.

Une fois qu’on a décrété qu’un cluster se scinde, comment répartir les entités ? Faut-il un simple **bipartitionnement** ou bien une **multipartition** ? Les algorithmes de segmentation peuvent être aussi complexes que ceux employés en “cut” de graphes, nécessitant parfois des heuristiques (ex. minimisation d’une coupe, maximisation de la modularité locale).

3. Risque de Sur-Segmentation.

Si l’on n’y prend pas garde, on peut créer trop de sous-blocs. Certains systèmes peuvent présenter des liaisons d’intensité moyenne et non pas de fortes discontinuités, ce qui conduit à scinder trop finement et à rater la cohérence multi-niveau. Comme en bottom-up, des algorithmes heuristiques peuvent conduire à des configurations globalement sous-optimales.

6.5.2.2. Systèmes “Macro → Micro” : Si la Synergie Interne n’est pas Assez Homogène, On Scinde

Dans la dynamique **top-down** (ou descendante) de la construction multi-niveau, il est possible de partir d’un **ensemble englobant** (un ou plusieurs gros clusters) et de **scinder** ceux-ci dès que leur cohésion interne est jugée insuffisante. Cette logique repose sur une subdivision progressive. Plutôt que de former immédiatement des micro-clusters, on commence par un **bloc global**, que l’on divise progressivement lorsque la synergie interne devient inhomogène. Les sections suivantes détaillent le principe, illustrent la mécanique de division, et exposent les avantages comme les contraintes pratiques de ce mode “macro → micro”.

A. Principe : Partir d’un (ou de quelques) Grand(s) Bloc(s)

Une caractéristique fondamentale de la démarche top-down est qu’elle **commence** par un **très gros cluster** — parfois unique — couvrant la totalité ou une grande portion de l’ensemble $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. D’un point de vue pratique, on peut considérer qu’une entité $\mathcal{N}^{\text{global}}$ réunit tous les indices dans un **super-nœud**. On analyse alors la **cohérence** de ce super-nœud.

On note $\mathcal{C}^{(\text{macro})}$ ce grand bloc. Dans certains contextes, cette configuration s'impose naturellement. Un système robotique opérant sous un même contrôle ou une base de données non encore partitionnée en sont des exemples typiques.

On introduit un **critère** $\mathcal{H}(\mathcal{C}^{(\text{macro})})$ évaluant l'homogénéité ou la cohésion interne. Il peut s'agir d'un indicateur calculé à partir des poids $\omega_{i,j}$ (moyenne, variance, distribution des liaisons, etc.), ou d'une mesure plus large (taux d'accord sémantique, par exemple).

$$\mathcal{H}(\mathcal{C}) = f(\{\omega_{i,j}\}_{i,j \in \mathcal{C}}).$$

Si \mathcal{H} est trop élevé (sous l'hypothèse que “trop élevé” signifie “trop hétérogène”), on entreprend une scission.

B. Si la Synergie Interne n'est pas Assez Homogène, On Scinde

Lorsqu'un cluster macro $\mathcal{C}^{(\text{macro})}$ s'avère **insuffisamment** cohérent, on le **divise**. Cela se traduit souvent par :

Décision de scission.

On identifie, au sein de $\mathcal{C}^{(\text{macro})}$, des sous-groupes plus denses ou des “failles” dans la matrice ω . On partitionne $\mathcal{C}^{(\text{macro})}$ en deux (ou plus) blocs, $\mathcal{C}_\alpha^{(\text{macro}-1)}$, $\mathcal{C}_\beta^{(\text{macro}-1)}$, ..., qui héritent chacun d'une **cohésion** mieux définie.

Par exemple, on peut imposer un gain minimum $\delta > 0$ en termes de réduction de l'hétérogénéité :

$$\mathcal{H}(\mathcal{C}_\alpha)\mathcal{H}(\mathcal{C}_\beta) \leq \mathcal{H}(\mathcal{C}^{(\text{macro})}) - \delta.$$

Descente itérative.

Après cette première scission, on applique la **même** opération à chaque sous-cluster obtenu. Si l'on constate qu'un bloc reste trop hétérogène, on le scinde de nouveau, et ainsi de suite. D'un point de vue fractal, la **même** règle de division se reproduit à chaque niveau, tant que la **cohérence** n'est pas jugée satisfaisante.

Exemple de formulation.

On peut définir $\mathcal{H}(\mathcal{C})$ comme la **variance** des poids $\{\omega_{i,j}\}$ à l'intérieur de \mathcal{C} , ou comme un **écart** normalisé :

$$\mathcal{H}(\mathcal{C}) = \frac{1}{|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} |\omega_{i,j} - \bar{\omega}_{\mathcal{C}}|.$$

Si $\mathcal{H}(\mathcal{C}) > \theta$, on scinde. Les critères peuvent varier selon la nature des données et la signification de la synergie.

C. Avantages de ce Zoom Descendant

La logique “macro \rightarrow micro” recèle plusieurs **points forts** qui la distinguent de l'agrégation ascendante :

On commence par un ensemble unique, ce qui procure une **vue d'ensemble** sur l'intégralité du réseau. Cela évite de créer d'emblée de multiples clusters "éparpillés" qu'il faudrait ensuite fusionner.

Si le **SCN** possède une forme d'auto-similarité (section 6.3), cette descente révèle à chaque niveau des **motifs** comparables aux niveaux supérieurs. Chaque scission reproduit la même règle, "zoomant" dans le bloc pour en extraire des sous-blocs plus cohérents. Il est parfois plus naturel, pour des raisons d'ingénierie ou de supervision, de "partir grand" et de n'effectuer des partitions qu'en cas de nécessité. Cela correspond à un schéma où le macro-niveau règne et ne crée de subdivisions qu'en identifiant des segments mal intégrés.

D. Inconvénients à Garder à l'Esprit

Comme toute méthode descendante, ce schéma n'est pas exempt de **limitations** :

Risque de sur-segmentation. Si le critère d'homogénéité est trop exigeant, on risque de fragmenter exagérément le cluster macro, produisant nombre de petits blocs. Cela reflète une **division excessive** qui peut saper la compréhension globale ou perdre de la connectivité.

Paramétrage délicat. Il faut fixer un **seuil** θ ou un **niveau** de variance acceptable. Un θ trop bas conduit à ne jamais scinder, un θ trop haut peut déclencher des coupes incessantes. On se retrouve face à un problème similaire à la détermination du nombre k de partitions dans les algorithmes classiques de clustering.

Algorithmes non optimaux. Découper $\mathcal{C}^{(\text{macro})}$ de manière "parfaite" (minimisant une fonction de coût globale) est souvent **NP-difficile** si \mathcal{C} est volumineux. On recourt donc à des **heuristiques** (ex. on identifie un couplage faible entre deux groupes et on coupe le long de cette "faille"). Mais rien ne garantit l'optimalité de la répartition obtenue.

Moins de "naissance" organique. Contrairement à la démarche bottom-up, où les micro-clusters émergent localement sans action extérieure, ici la partition est décidée par un **niveau macro**, imposant la coupe depuis le haut. Cela peut être moins "naturel" dans un système où la dynamique DSL locale est la clé principale d'auto-organisation.

6.5.2.3. Risques d'une Division Excessive ou de la Sur-Segmentation

Introduction.

Dans la démarche **top-down** (voir 6.5.2.1 et 6.5.2.2), on part d'un cluster global englobant toutes les entités, puis on le scinde dès que l'on juge sa cohésion interne insuffisante. Bien que cela permette de "zoomer" sur des sous-groupes plus homogènes, on court également le risque d'une **sur-segmentation** lorsque les critères de division sont trop stricts ou appliqués trop précocement. Il s'ensuit une multiplication de petits blocs, ce qui peut compromettre la logique de construction hiérarchique à moyenne ou grande échelle. Cette section examine les principaux **mécanismes** menant à la sur-segmentation, puis discute des **conséquences** et des **pistes** pour l'atténuer.

A. Mécanismes menant à la sur-segmentation

Dans un schéma top-down, on définit un **critère** $\mathcal{H}(\mathcal{C})$ évaluant l'hétérogénéité interne d'un cluster \mathcal{C} , associé à un **seuil** θ . Si θ est placé trop bas (ou la cohésion exigée est trop élevée), alors même

des blocs relativement cohérents apparaissent comme “insuffisamment homogènes”. Cela déclenche des **divisions** successives qui morcellent le réseau en de nombreux **petits** sous-groupes.

Sur le plan formel, la règle

$$\mathcal{H}(\mathcal{C}) > \theta \Rightarrow \text{scission de } \mathcal{C},$$

conduit à un découpage sans fin si θ ne tolère qu’un degré de cohésion interne excessivement élevé.

Dans certains systèmes top-down, on évalue la cohésion d’un cluster à chaque **itération**, sans laisser la dynamique locale (ex. mise à jour DSL) prendre le temps de renforcer les poids $\omega_{i,j}$ à l’intérieur du bloc. En ne laissant pas à la **synergie** le temps de se structurer, on perçoit hâtivement des failles internes. On segmente alors avant même que des liaisons fortes aient pu émerger.

Il en résulte une fragmentation rapide et parfois injustifiée. Un bloc qui aurait pu devenir cohérent se retrouve divisé en deux simplement parce que la stabilisation des $\omega_{i,j}$ n’a pas été suffisamment prise en compte.

Lorsque la division top-down ne s’accompagne d’aucune **possibilité de fusion** (retour arrière), chaque scission devient pratiquement **irréversible**. Ainsi, même si, à un moment ultérieur, des liens plus forts se forment entre deux blocs séparés, on reste avec un découpage trop fin.

Ce phénomène de **fragmentation irréversible** conduit à la formation de nombreux petits clusters. À mesure que la hiérarchie descend, aucun *regroupement* n’est envisagé, même si la dynamique interne du réseau évolue.

B. Conséquences sur la structure multi-niveau

Un des apports majeurs d’une hiérarchie multi-niveau est de révéler des **clusters** de taille intermédiaire (méso) ou des macro-groupes rassemblant plusieurs entités fortement reliées. La **sur-segmentation** fait disparaître ces blocs plus vastes, puisque tout se retrouve scindé en unités trop petites.

Cela diminue la **clarté** de la structure, aboutissant à des “feuilles” de très faible cardinalité plutôt qu’à des “nœuds” cohérents et significatifs.

Au lieu de manipuler quelques dizaines de super-nœuds, on se retrouve avec une kyrielle de mini-blocs. Pour les niveaux supérieurs de la hiérarchie, le gain de simplification attendu (réduction du nombre de ω à gérer) disparaît, car l’excès de partitions entraîne une complexité accrue dans la gestion du réseau.

Dans des applications (ex. IA cognitive, robotique), on souhaite parfois que des sous-groupes partagent des ressources ou des tâches. Une sur-segmentation aboutit à des groupes hyper-spécialisés qui ne communiquent pas ou peu entre eux, et on perd la **coopération** inter-bloc. On sacrifie alors l’effet de mutualisation que vise une structure multi-niveau.

C. Pistes pour y remédier

La première mesure consiste à **redéfinir** le critère de division pour être moins sévère. Autrement dit, laisser plus de marge à la cohésion interne. On peut aussi imposer qu’un bloc ne soit évalué qu’après un certain délai, permettant aux $\omega_{i,j}$ de se renforcer si la dynamique DSL locale le justifie.

Avant de décider d'une scission, on peut laisser le bloc "mûrir" sous la règle de plasticité (renforcement/décroissance) un certain nombre d'itérations. Ce laps de temps permet à la synergie de s'établir, révélant qu'un groupe est en fait cohérent, et évitant une coupe précipitée.

Comme en bottom-up, un mécanisme de **fusion** au même niveau (cf. 6.4.3.2 sur la coordination latérale) ou une boucle de *feedback* descendant (cf. 6.4) peut éviter les scissions définitives. Même si un bloc a été scindé, si l'on constate ultérieurement une interaction forte entre deux mini-blocs, on peut les **ré-agrégé**.

Dans les grandes architectures DSL, on combine souvent **top-down** et **bottom-up** (cf. 6.5.3). Si le **top-down** fragmente excessivement, le **bottom-up** fusionne en parallèle certains blocs trop fins, maintenant ainsi un équilibre dynamique entre division et agrégation.

6.5.3. Hybridation (Approche Mixte)

Dans les sections précédentes (6.5.1 et 6.5.2), nous avons présenté deux grandes **orientations** pour structurer un SCN (Synergistic Connection Network) de manière multi-niveau. L'**agrégation progressive (bottom-up)** permet de fusionner des entités locales en super-clusters, tandis que la **division (top-down)** segmente un ensemble global en sous-groupes plus homogènes. Cependant, dans la pratique, de nombreux **systèmes** tirent parti d'une **approche mixte**. La dynamique locale permet de **fusionner** des entités ou micro-clusters selon une vision **bottom-up**, tout en conservant la possibilité de **scinder** ces regroupements lorsqu'ils deviennent trop hétérogènes, adoptant ainsi une stratégie **top-down** complémentaire. Cette **hybridation** procure davantage de **flexibilité** et évite les limites inhérentes à la pure agrégation (6.5.1.3) ou la pure division (6.5.2.3).

6.5.3.1. Combiner Agrégation Partielle et Division Sélective pour plus de Flexibilité

Les approches **bottom-up** (voir 6.5.1) et **top-down** (voir 6.5.2) présentent chacune des limites. L'agrégation ascendante peut rapidement verrouiller le réseau dans une configuration sous-optimale à cause de fusions irréversibles, tandis que la scission descendante risque d'aboutir à des partitions excessivement fines. Une **stratégie hybride** reposant sur l'alternance de **phases** d'agrégation et de **phases** de division permet de maintenir la construction hiérarchique plus fluide et mieux adaptée aux fluctuations de la **synergie** ou de l'hétérogénéité. Les développements qui suivent décrivent ce principe mixte et ses avantages, en cohérence avec la logique DSL multi-niveau.

A. Motivation

Les démarches purement ascendantes induisent parfois des **verrous irréversibles**. En s'appuyant sur de multiples agrégations (voir section 6.5.1), il est possible de former prématurément de **gros blocs cohésifs**, ce qui complique la redistribution des entités si ces **macro-groupes** se révèlent incohérents par la suite. À l'inverse, une division uniquement descendante (voir section 6.5.2) peut provoquer une segmentation trop poussée et fragmenter la hiérarchie en innombrables petits sous-ensembles. Il devient alors pertinent de recourir à une **hybridation** qui exploite à la fois la fusion lorsqu'une forte synergie est constatée et la division sélective pour corriger les cas d'hétérogénéité

interne. Cette oscillation entre fusion et scission empêche les blocages inhérents aux approches unilatérales.

B. Principe de l'Approche Mixte

Le noyau méthodologique réside dans l'emploi successif de l'agrégation locale et de la détection d'incohérences macro, menant à des divisions ciblées. Dans un premier temps, on laisse la dynamique **bottom-up** rassembler des entités en super-nœuds à l'aide de la règle

$$\Delta \omega_{i,j}(t) = \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

ce qui consolide les liens forts. Ensuite, si l'examen macro met en évidence une hétérogénéité dans un super-nœud $\mathcal{N}_\alpha^{(k)}$, on introduit un mécanisme de division locale qui affaiblit certains liens internes et aboutit à un découpage partiel. Ainsi, le réseau peut se réorganiser en continu et affiner la hiérarchie en fonction des variations de la synergie.

C. Rôle du Feedback Descendant

Pour réaliser la scission dans un super-nœud hétérogène, on fait appel à un **flux descendant** (voir 6.4.1.2) fondé sur un signal inhibiteur visant des liens internes spécifiques. Sur le plan mathématique, on peut l'écrire comme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] - \gamma \Delta_{\text{split}},$$

où $\Delta_{\text{split}} > 0$ réduit progressivement la pondération $\omega_{i,j}$ dans la zone désignée. En affaiblissant graduellement ces liens, on amorce la division ; si la situation évolue, le flux descendant peut être stoppé, ce qui évite les ruptures irréversibles. Ce **feedback** autorise une scission sélective et réversible, guidée par le niveau macro, qui surveille et corrige les configurations trop vastes ou incohérentes.

D. Équilibre Continu et Flexibilité

Avec la faculté de réaliser simultanément des **fusions** (renforcement) et des **divisions** (inhibition sélective), le SCN converge vers un **état stable** garantissant qu'aucun super-nœud n'est à la fois trop vaste et trop hétérogène, ni que la synergie entre blocs distincts soit négligée. Les risques liés à l'agrégation irréversible (voir 6.5.1.3) et à la sur-segmentation (voir 6.5.2.3) se trouvent atténués, car le réseau peut revenir sur ses choix ou affiner la hiérarchie au fur et à mesure des signaux multiblocs et des évaluations macro. On obtient ainsi une forme d'**équilibre continu**, dans la mesure où la topologie hiérarchique peut évoluer librement en réponse à la dynamique des pondérations $\omega_{i,j}$.

E. Exemples de Mise en Œuvre

Dans les **réseaux sociaux**, l'agrégation locale unit des individus partageant un même intérêt ou renforçant leurs liens. En parallèle, une division sélective intervient lorsqu'un **super-groupe** devient conflictuel et que le niveau supérieur modérateur déclenche une **scission partielle** pour séparer les factions émergentes. Le résultat est un réseau social plus flexible, à l'abri d'une unique communauté géante qui manquerait de cohésion.

Dans les **systèmes robotiques**, les robots qui coopèrent forment des équipes par agrégation. Un contrôle global peut néanmoins forcer la scission d’une sous-équipe si elle poursuit un objectif différent ou si la communication interne se dégrade, ce qui produit un SCN adaptatif où les équipes se forment et se défont en fonction des missions.

Dans les **systèmes sensoriels ou cognitifs**, on peut fusionner des capteurs (ou neurones) fortement corrélés, tandis que le niveau macro “coupe” un bloc bruyant ou conflictuel, isolant les signaux perturbateurs et réassignant éventuellement ces entités ailleurs. Cette logique reflète une **auto-organisation** à deux vitesses, ascendante pour la cohérence locale et descendante pour la répartition sélective.

F. Avantages de la Démarche Mixte

Cette approche permet une flexibilité et une adaptation accrues en évitant les erreurs d’une stratégie uniquement ascendante, qui pourrait mener à des agrégations irréversibles, ou d’une stratégie strictement descendante, risquant une fragmentation excessive. Elle garantit également une robustesse face aux évolutions du réseau, qu’il s’agisse de l’ajout de nouvelles entités ou d’un changement dans les synergies, en maintenant la possibilité d’ajuster la structure par fusion ou division.

Cela se révèle essentiel dans un contexte de traitement en temps réel ou en présence de flux de données continus. De plus, le contrôle multi-niveau assure que la dynamique DSL reste focalisée sur le renforcement local des liens, tandis que l’analyse macro agit par rétroaction descendante, permettant une régulation hiérarchique efficace.

G. Points à Surveiller

Un des **inconvenients** majeurs tient au risque d’oscillations, lorsque le réseau oscille entre un bloc fusionné puis scindé, puis refusionné. Pour éviter ce “va-et-vient” perpétuel, on introduit souvent des **seuils d’hystérésis** ou des **périodes de stabilisation**, empêchant la redécoupe immédiate d’un bloc à peine fusionné. La **complexité algorithmique** est également plus élevée, car on doit gérer simultanément les règles ascendantes et descendantes, définir des priorités entre fusion et scission, et paramétrer leur temporalité. Si le réseau n’exige pas un tel degré de flexibilité, la difficulté accrue peut ne pas se justifier. Cependant, la comparaison aux approches pures montre que cette méthode **mixte** gagne en souplesse et n’est que légèrement plus complexe à mettre en œuvre.

6.5.3.2. Applications : Systèmes Complexes à Grande Échelle (Réseaux Sociaux, Sensoriels)

Les **systèmes complexes** de grande envergure, tels que les **réseaux sociaux** massifs ou les **réseaux de capteurs** (IoT), sont des terrains privilégiés pour mettre en œuvre l’**approche mixte** décrite précédemment (voir section 6.5.3.1). Dans ces environnements, la quantité d’entités peut s’élever à des milliers ou millions, et la structure de leurs liens $\omega_{i,j}$ ou de leur synergie évolue rapidement. Il s’avère essentiel d’y autoriser non seulement l’**agrégation ascendante** (mécanismes *bottom-up*) mais aussi la possibilité d’une **division sélective** (mécanismes *top-down*) afin d’ajuster en continu la hiérarchie et d’éviter la stagnation dans une configuration inadaptée. Cette **flexibilité** est d’autant plus cruciale que les liens se forment, se transforment ou disparaissent à un rythme soutenu, selon les interactions sociales ou les corrélations de mesure.

A. Réseaux Sociaux de Grande Échelle

Dans un **réseau social**, les utilisateurs peuvent être représentés par des nœuds $i \in \{1, \dots, n\}$, et les pondérations $\omega_{i,j}(t)$ refléteront la force de leur interaction à l’instant t . On peut définir une **similitude** ou **corrélation** $\text{corr}(X_i, X_j)$, où X_i et X_j sont des vecteurs ou historiques d’activité (échanges, centres d’intérêt, contenus partagés). La **dynamique DSL** appliquée localement en mode *bottom-up* prend souvent la forme :

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [\text{corr}(X_i, X_j) - \tau \omega_{i,j}(t)],$$

ce qui consolide les liens entre utilisateurs fortement corrélés. Les agrégations successives décrites en section 6.5.1 forment des micro-communautés ou des super-groupes, mais elles risquent d’enfermer à long terme des factions distinctes dans un même bloc.

La division sélective présentée en section 6.5.2 intervient pour scinder un macro-groupe devenu trop grand ou trop hétérogène. Un feedback descendant ajuste alors la pondération $\omega_{i,j}$ entre sous-groupes jugés rivaux en y ajoutant un terme négatif $\gamma \Delta_{\text{desc}}$, notamment en cas de divergence idéologique ou de clivage d’intérêt.

Cette dynamique permet au réseau social d’évoluer en fusionnant localement lorsque la synergie est forte, puis en opérant des divisions ciblées lorsque des tensions émergent. Des communautés spontanées formées par un événement viral peuvent ainsi se fragmenter ultérieurement en sous-groupes plus spécialisés.

L’avantage de cette approche réside dans sa réactivité et sa scalabilité, garantissant une hiérarchie modulaire qui équilibre agrégation et segmentation, plutôt que de figer le réseau dans une structure rigide ou de le morceler excessivement.

B. Systèmes Sensoriels (IoT, Réseaux de Capteurs)

Dans un **réseau de capteurs** dispersés (IoT), le nombre d’appareils peut aussi atteindre l’échelle du million. Chaque capteur i fournit une mesure $X_i(t)$, et l’on peut définir la pondération $\omega_{i,j}$ à partir d’une **distance inversée** ou d’une **corrélation** sur les signaux $\{X_i(t)\}$. Une règle DSL *ascendante* :

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [\text{similarite}(X_i, X_j) - \tau \omega_{i,j}(t)]$$

permet la formation progressive de clusters ou zones cohérentes, rassemblant les capteurs dont les mesures sont proches. Cependant, un unique “super-cluster” peut s’avérer trop vaste et contenir des capteurs géographiquement ou fonctionnellement très différents. Le mécanisme de **division** top-down autorise alors la **scission** si un algorithme de contrôle macro détecte un “mélange” inattendu. Cette division, guidée par un signal descendant δ_{split} qui réduit certaines pondérations $\omega_{i,j}$, maintient la possibilité d’une réorganisation spatiale. Certains capteurs pourront ainsi se regrouper différemment, aboutissant à une architecture multi-niveau plus stable et mieux segmentée.

La **flexibilité** de l’agrégation–division se révèle cruciale dans un système IoT où la topologie évolue (nouveaux capteurs, pannes, variations de signaux), et où la qualité de service dépend d’une

segmentation performante. Plutôt que de se figer dans des macro-zones fixes, on s’adapte aux variations locales et globales.

6.5.3.3. Observation des Patterns Fractals si le Cycle Agrégation–Division Suit une Logique de Récurrence

Cadre et Motivation. Les sections précédentes ont montré qu’une **approche hybride** (section 6.5.3.1) combine les mécanismes **bottom-up** (agrégation) et **top-down** (division) dans la construction hiérarchique d’un **SCN** (Synergistic Connection Network). Dans certaines configurations, un **cycle** récurrent d’agrégation et de division peut émerger. Des groupes se forment lorsque leur synergie est élevée, puis se scindent à nouveau dès qu’ils deviennent trop hétérogènes. Lorsqu’une telle dynamique se reproduit à différents paliers (ou échelles) avec la même *loi* d’auto-organisation, on constate parfois l’apparition de **patterns fractals**, traduisant une **auto-similarité** ou **invariance d’échelle**. Les développements qui suivent analysent comment ce cycle agrégation–division engendre des régularités fractales, et quelles hypothèses sont nécessaires pour les observer.

A. Logique de Répétition et d’Échelle

La notion de **fractale** (chapitre 6.3) repose sur une propriété d’**auto-similarité**, où le même schéma de construction ou d’évolution se répète à plusieurs niveaux de **granularité**, sans qu’une échelle caractéristique fixe ne soit imposée. Dans le contexte d’un **cycle** combinant agrégation et division, ce comportement se concrétise si, à chaque itération ou à chaque palier, la *même* règle DSL localement et la *même* procédure de scission globalement se déploient.

On suppose qu’à un moment donné, des entités (ou clusters) se **fusionnent** partiellement si leur pondération $\omega_{i,j}$ est assez élevée, conduisant à la création de super-nœuds. Dès qu’un super-nœud devient lui-même trop grand ou trop hétérogène (critère \mathcal{H} trop élevé), une **division** le scinde en sous-blocs. Les deux (ou plusieurs) blocs nés de la scission peuvent à leur tour vivre la même alternance de fusion avec d’autres blocs et de division interne. Cette **récurrence** d’un schéma de “fusion si cohérent, division si incohérent” constitue une base favorable à l’**invariance d’échelle**.

Pour que cette récurrence se traduise en **patterns fractals**, il faut que les **paramètres** η , τ , ou le critère \mathcal{H} de division ne fixent pas un unique “seuil absolu” de taille, mais soient paramétrés **relativement** à la structure courante. Autrement dit, si, à chaque échelle, la règle DSL et la procédure de scission conservent la *même forme*, on ne rompt pas la *loi de croissance* ni la *loi de découpage*. Dès lors, le système peut afficher des **lois de puissance** en termes de distribution de tailles de clusters, signatures typiques des structures fractales.

On peut imaginer un bloc \mathcal{C} de taille N . À chaque cycle, un sous-ensemble de αN entités fusionnent avec βN entités voisines si leur synergie dépasse un **seuil** relatif, puis on scinde γN entités si leur cohésion chute en deçà d’un certain ratio. Une telle règle, conservée à chaque palier, entretient un “même flux” à chaque échelle, propice à la fractalité.

B. Mécanisme de Stabilisation Fractale

Le **DSL** (Deep Synergy Learning) met à jour les pondérations $\omega_{i,j}$ sous une forme générale :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)] + \Delta_{\text{feedback}}(i,j),$$

où $\eta > 0$ et $\tau > 0$ paramètrent le renforcement/décroissance, tandis que Δ_{feedback} reflète les influences descendantes (scission sélective) ou latérales (coordination). Si Δ_{feedback} applique la **même** forme d'inhibition (ou de récompense) à chaque “palier” avec un rescaling adapté à la taille courante, la dynamique DSL aboutit parfois à un **cycle récurrent** ; des clusters se forment par renforcement, s'agrandissent, puis reçoivent un retour descendant provoquant une scission partielle lorsque leur hétérogénéité interne augmente.

Un **renforcement local** se produit lorsque les entités ou *clusters* présentant une synergie élevée $S(i, j)$ consolident leurs liens et se rassemblent en blocs plus vastes, tandis qu'une **inhibition macro** s'exerce si un bloc \mathcal{C}_α présente une trop grande dispersion, dans ce cas, une procédure descendante réduit spécifiquement les pondérations $\omega_{i,j}$ responsables de la discorde, forçant ainsi la scission de \mathcal{C}_α en sous-blocs plus homogènes. Cette dynamique se **répète** à chaque niveau, suivant un principe fractal où la même logique s'applique quel que soit le nombre d'entités dans \mathcal{C}_α . Que ce groupe contienne 50, 500 ou 50 000 entités, on observe des **motifs auto-similaires**, caractérisés par une agrégation locale et une division globale à chaque échelle.

C. Observations Pratiques et Limites

Les **patterns fractals** déployés dans un réseau hiérarchique reposent sur l'idée que la **même** combinaison de renforcement et d'**inhibition** agit à toutes les échelles de la structure. Cette exigence d'**auto-similarité** peut cependant être mise en péril si l'on introduit une rupture de symétrie d'échelle. Il se produit alors une “cassure” dans la cohérence fractale si, par exemple, les seuils de scission θ_k diffèrent trop fortement d'un palier k à l'autre, ou si le taux η est multiplié par un facteur grandissant à mesure qu'on progresse dans la hiérarchie. Il en résulte qu'une échelle caractéristique apparaît et détruit le principe d'invariance.

Un second écueil consiste en un **risque d'oscillation** voire de quasi-chaos si la logique macro descendante intervient de manière trop fréquente. Il se peut qu'un cluster émerge localement, puis se retrouve scindé au niveau supérieur, puis se reforme, engendrant une dynamique d'allers et retours incessants. Pour qu'une **structure fractale** stable (ou un régime quasi-périodique) puisse s'établir, il est souvent nécessaire d'introduire un mécanisme d'**hystérésis** ou un délai temporel dans le déclenchement des fusions et divisions, afin de prévenir les réorganisations à trop haute fréquence.

Un troisième point d'intérêt réside dans les **exemples de lois de puissance** qui peuvent s'ensuivre. On peut observer, dans certains scénarios, un spectre des tailles de clusters satisfaisant une **distribution** $P(N) \propto N^{-\alpha}$. Dans ce cas, on se situe face à un **comportement fractal** (ou un état critique) qui se manifeste lorsqu'il n'existe pas d'échelle caractéristique imposée au réseau. Pour vérifier cette structure “scale-free”, on recourt fréquemment à des analyses en représentation log-log ou à des mesures de **box-counting** permettant d'évaluer la dimension fractale du système, que ce soit dans la topologie spatiale ou dans la distribution des connexions. Il convient de souligner que cette propriété n'est pas garantie, un suivi rigoureux du réseau est nécessaire pour vérifier que l'**auto-organisation** aboutit bien à une loi de puissance significative. Le moindre paramètre variant d'un palier à l'autre peut rompre cette loi et faire disparaître la fractalité attendue.

6.5.4. Algorithmes de Mise en Œuvre

Après avoir décrit les principes (agrégation vs. division) et l'idée d'une approche **mixte** (6.5.3), il est temps d'aborder plus la dimension algorithmique. La section 6.5.4.1 propose un **pseudo-code** illustratif qui montre comment, dans un **SCN** (Synergistic Connection Network), on peut :

- **Détecter les micro-clusters**,
- **Fusionner** ces micro-clusters en “super-nœuds”,
- Gérer un **contrôle macro** (top-down) pour scinder ou valider ces super-nœuds.

6.5.4.1. Pseudo-Codes : (1) Détection Micro-Cluster, (2) Fusion en Super-Nœud, (3) Contrôle des Macro-Niveaux

Cadre Global. L'objectif est de décrire, sous forme de pseudo-codes, l'articulation pratique de la **dynamique locale** (détection de micro-clusters via DSL), de la **fusion** en super-nœuds (agrégation), et du **contrôle** macro (avec possibilité de division). On suppose qu'au **niveau micro** (ou $\ell = 0$), on gère un ensemble d'entités $\{\mathcal{E}_i\}_{i=1\dots n}$ reliées par des pondérations $\omega_{i,j}^{(0)}$. L'itération de cette procédure peut être prolongée vers des **niveaux** 1,2, ... jusqu'à un palier macro ou intermédiaire.

La structure principale se décompose en trois volets :

- Une **dynamique locale** qui met à jour $\omega_{i,j}^{(0)}$ selon la règle DSL et détecte les micro-clusters.
- Une **fusion** (agrégation) pour construire des super-nœuds à partir des clusters détectés.
- Un **contrôle** macro, qui valide ou scinde (division) les super-nœuds jugés trop hétérogènes.

La présentation qui suit reste schématique. Les détails algorithmiques (complexité, choix des seuils, etc.) sont modulables.

A. Détection Micro-Cluster : la Dynamique Locale

Ce premier pseudo-code réalise la **mise à jour** DSL au **niveau micro** (ou palier $\ell = 0$), puis détecte des **clusters** localement, qualifiés de *micro-clusters*.

```
1: Initialize  $\omega(i,j)^{(0)}$  for all  $(i,j)$  with small or random values
2: for  $t = 1$  to  $T_{\text{local}}$  do
3:   for each pair  $(i,j)$ :
4:      $\omega(i,j)^{(0)} \leftarrow \omega(i,j)^{(0)} + \eta_0 * [So(i,j) - \tau_0 * \omega(i,j)^{(0)}]$ 
5:   # Optionally clamp or saturate  $\omega(i,j)^{(0)}$  to  $[0,1]$  or a chosen range
6: end for

7:  $\text{clusters\_level0} \leftarrow \text{find\_clusters}(\omega(\cdot,\cdot)^{(0)}, \text{threshold\_local})$ 
   # e.g., BFS or community detection where  $\omega^{(0)}(i,j) > \text{threshold\_local}$ 
```

Interprétation Mathématique.

Le *taux* $\eta_0 > 0$ et le *terme* $\tau_0 > 0$ régulent la vitesse d'adaptation, et $S_0(i, j)$ indique la **synergie** ou **similarité** entre entités \mathcal{E}_i et \mathcal{E}_j . La fonction *find_clusters* détecte des composantes connexes si $\omega_{i,j}^{(0)}$ dépasse un certain θ_{loc} , ou exploite un algorithme de partition (ex. Louvain, *community detection* légère) adapté à la densité.

La ligne 5 illustre la possibilité de forcer $\omega_{i,j}^{(0)}$ à rester dans $[0,1]$, une pratique courante pour éviter la divergence ou les valeurs négatives.

B. Fusion en Super-Nœuds (Agrégation)

Une fois identifiés, les micro-clusters de **niveau 0** sont transformés en **super-nœuds** au **niveau 1**. On reconstruit alors les pondérations $\omega^{(1)}$ entre ces super-nœuds via une fonction Ψ .

```

9: super_nodes_level1 = {}
10: for each cluster C $\alpha$  in clusters_level0:
11:   create super-node N $\alpha^{(1)}$ 
12:   N $\alpha^{(1)}$ .members = C $\alpha$ 
13:   super_nodes_level1.add( N $\alpha^{(1)}$  )

14: for each pair (N $\alpha^{(1)}$ , N $\beta^{(1)}$ ):
15:    $\omega(\alpha, \beta)^{(1)} = \Psi( \{ \omega(i, j)^{(0)} \mid i \in C_\alpha, j \in C_\beta \} )$ 

```

Description.

Les lignes 10–13 établissent une correspondance directe entre un cluster $\mathcal{C}_\alpha^{(0)}$ détecté et un super-nœud $\mathcal{N}_\alpha^{(1)}$. La **fonction** Ψ agrège les valeurs $\omega_{i,j}^{(0)}$ entre les entités internes à \mathcal{C}_α et \mathcal{C}_β . Typiquement, Ψ peut être :

$$\omega_{\alpha, \beta}^{(1)} = \Psi\left(\{\omega_{i,j}^{(0)} : i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta\}\right),$$

où Ψ est une somme, une moyenne, un maximum, etc. Ce calcul produit une **nouvelle** matrice $\omega^{(1)}$ représentant les liens entre super-nœuds $\{\mathcal{N}_\alpha^{(1)}\}$.

C. Contrôle des Macro-Niveaux : Division ou Validation

Au **niveau 1**, chaque super-nœud $\mathcal{N}_\alpha^{(1)}$ peut être jugé **hétérogène** ou **cohésif**. En cas d'hétérogénéité, une **scission** (top-down) est possible. Le pseudo-code ci-après montre une version simplifiée de cette opération :

```

18: for each super-node N $\alpha^{(1)}$  in super_nodes_level1:
19:   H $\alpha$  = heterogeneity( N $\alpha^{(1)}$ ,  $\omega^{(1)}$  )
20:   if H $\alpha$  > threshold_macro:
21:     sub_clusts = subdivide( N $\alpha^{(1)}$ .members,  $\omega^{(0)}$  )
22:     # Possibly do BFS or community detection again at level 0
23:     remove N $\alpha^{(1)}$  from super_nodes_level1
24:     for each subc in sub_clusts:
25:       Nx = create new super-node
26:       Nx.members = subc

```

26: `super_nodes_level1.add(Nx)`
 27: `re_compute $\omega^{(1)}$ using function $\Psi(\cdot)$ # or partial update`

Explication.

La fonction *heterogeneity* (ligne 19) calcule un indice \mathcal{H} de cohésion interne. Si $\mathcal{H}(\mathcal{N}_\alpha^{(1)}) > \theta_{\text{macro}}$, la procédure *subdivide* revisite les entités $\mathcal{E}_i \in \mathcal{N}_\alpha^{(1)}.members$ et cherche une partition interne plus cohérente (lignes 21–25). Les sous-blocs substituent le super-nœud $\mathcal{N}_\alpha^{(1)}$ d’origine dans *super_nodes_level1*. On met ensuite à jour la matrice $\omega^{(1)}$, voire on itère une dynamique DSL au niveau 1.

D. Intégration en Boucle Itérative

Pour former une **architecture hiérarchique** (et pas seulement un niveau 0 puis un niveau 1), on répète le principe suivant :

- **Mise à jour DSL** au palier $\ell = 0$ et **détection** micro-clusters,
- **Fusion** de ces micro-clusters en super-nœuds $\mathcal{N}_\alpha^{(1)}$,
- **Contrôle** macro (ligne 18+) : on valide ou on subdivise si besoin,
- Si la taille du réseau l’exige, on réapplique un **processus** DSL au niveau 1 avec η_1, τ_1 , on détecte des clusters de niveau 1, on crée des super-nœuds $\mathcal{N}_\alpha^{(2)}$, etc.

La hiérarchie s’échelonne en paliers $k = 0, 1, 2, \dots$ jusqu’à atteindre un niveau **macro** où il ne reste plus qu’un petit nombre de super-nœuds ou un unique bloc. On peut inclure un **flux descendant** correctif pour forcer une scission sélective, notamment si un super-nœud est repéré comme trop hétérogène.

E. Commentaires Mathématiques et Convergence

Lien avec les Algorithmes de Community Detection. Le schéma ci-dessus s’apparente à une **détection de communautés** multi-niveau. Toutefois, la spécificité **DSL** réside dans la **mise à jour** $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S - \tau\omega_{i,j}(t)]$ qui fait *émerger* les clusters plutôt que de simplement les “découper”.

Complexité et Approches Heuristiques. Chaque niveau requiert un **find_clusters** (ligne 8) et un **subdivide** (ligne 21) potentiellement complexes. On recourt souvent à des heuristiques (BFS sur les liens $> \theta$, modularité, etc.) pour demeurer en $O(n \log n)$ ou $O(n)$ si le graphe est éparé. Les algorithmes exacts de partition restent NP-difficiles pour n grand.

Convergence ou Évolution Continue.

Lorsque l’environnement est **statique**, un état stable peut émerger où aucun super-nœud n’est scindé et aucun micro-cluster ne se modifie. Dans des **contextes dynamiques**, avec des flux de données et des changements de liens, ces boucles s’exécutent en continu pour adapter la hiérarchie en **temps réel** (chap. 9).

6.5.4.2. Paramètres Clés (Seuil de Synergie, Ratio d'Homogénéité, etc.)

Contexte et Problématique. Les algorithmes multi-niveaux décrits précédemment (section 6.5.4.1) reposent sur une alternance entre l'**agrégation** (bottom-up) et la **division** (top-down). Cette dynamique implique divers **paramètres** cruciaux qui déterminent les conditions de fusion et de scission, et donc la forme finale de la hiérarchie. Parmi ces paramètres, on trouve notamment le **seuil de synergie** dictant la fusion locale et le **ratio d'homogénéité** influençant la division macro. D'autres, comme les taux η et les facteurs τ de la mise à jour DSL, ou la fonction Ψ d'agrégation, ont également un impact décisif. Les développements ci-après décrivent comment ces paramètres s'imbriquent et comment ils orientent l'équilibre entre la formation de super-nœuds et leur éventuelle scission.

A. Seuil de Synergie $\theta_{synergy}$

Dans la démarche **bottom-up**, un **seuil** de synergie $\theta_{synergy}$ sert à déterminer quand deux entités (ou deux clusters) méritent d'être **fusionnés**. Mathématiquement, on peut considérer qu'on ne fusionne un couple $\{i, j\}$ que si $\omega_{i,j} \geq \theta_{synergy}$. De même, dans l'algorithme "find_clusters", on ne retient que les arêtes du graphe dont la pondération dépasse ce seuil. Cela aboutit à des composantes connexes que l'on assimile à des **micro-clusters**.

Un $\theta_{synergy}$ trop élevé force une exigence stricte de cohérence, ce qui restreint la fusion à des liens très forts. Cela tend à créer un nombre important de **petits** clusters ou à empêcher la formation de sous-groupes plus volumineux. Au contraire, un $\theta_{synergy}$ trop bas mène à des blocs de grande taille peu homogènes, qu'il faudra ensuite probablement **scinder** (top-down) en raison d'une hétérogénéité interne excessive.

Il est possible de rendre ce seuil **adaptatif** en définissant $\theta_{synergy}$ comme un quantile de la distribution des $\omega_{i,j}$ plutôt que de le fixer a priori. Par exemple, en retenant les 20 % des liens les plus forts, on garantit un pourcentage prévisible de liaisons considérées comme "actives" pour la fusion, assurant ainsi une stabilité du nombre de clusters locaux.

B. Ratio d'Homogénéité / Hétérogénéité θ_{macro}

L'**hétérogénéité** $\mathcal{H}(\mathcal{C})$ est un indicateur crucial dans la phase **top-down**, au moment de décider la scission d'un super-nœud \mathcal{C} . Cette hétérogénéité peut être quantifiée de plusieurs manières, notamment par la variance des pondérations $\omega_{i,j}$, l'écart absolu moyen ou le ratio entre liens forts et liens faibles. Une approche courante repose sur la somme des écarts entre chaque poids $\omega_{i,j}$ et la moyenne des poids internes, ce qui permet d'évaluer la dispersion des synergies au sein d'un cluster. Plus précisément, on introduit

$$\mathcal{H}(\mathcal{C}) = \frac{1}{|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} |\omega_{i,j} - \bar{\omega}_{\mathcal{C}}|$$

où $\bar{\omega}_{\mathcal{C}}$ désigne la moyenne des $\omega_{i,j}$ pour les entités internes à \mathcal{C} . Si la somme des écarts $|\omega_{i,j} - \bar{\omega}_{\mathcal{C}}|$ se révèle élevée, alors \mathcal{C} présente un fort degré d'hétérogénéité.

Le **critère de scission** consiste à fixer un **seuil** θ_{macro} . Lorsque $\mathcal{H}(\mathcal{C})$ dépasse θ_{macro} , on considère que la cohésion interne du super-nœud est insuffisante et qu'il convient d'opérer une division.

Conformément au **mécanisme** top-down, un **feedback** descendant vient alors réduire spécifiquement certaines liaisons $\omega_{i,j}$ au sein de \mathcal{C} , ce qui “casse” la communauté en sous-groupes plus homogènes. Un θ_{macro} trop élevé entraîne une scission uniquement pour les blocs très hétéroclites, au risque de conserver parfois des clusters imposants. Inversement, un θ_{macro} trop bas suscite une **sur-segmentation** (cf. 6.5.2.3), dans laquelle même de modestes divergences déclenchent la division.

Le **paramétrage adaptatif** offre la possibilité de modifier θ_{macro} au fil de l’évolution du réseau ou selon la distribution de $\mathcal{H}(\cdot)$. Par exemple, on peut sélectionner un quantile donné pour définir la valeur du seuil, ou ajuster θ_{macro} en fonction de la phase d’itération. Cette démarche assure une maîtrise plus fine du rythme de scission au niveau macro, conciliant la nécessité d’homogénéité dans un super-nœud et la flexibilité requise pour l’**auto-organisation** hiérarchique.

C. Paramètres de Vitesse : η et τ dans la Dynamique DSL

La mise à jour $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$ dépend de manière linéaire du **taux d’apprentissage** η . Lorsque η est trop élevé, les variations de ω sont rapides, voire abruptes, pouvant générer des **oscillations** ou déstabiliser le réseau. À l’inverse, un η très petit étale la convergence sur un nombre important d’itérations, ralentissant la capacité du système à repérer et consolider les clusters. Dans un **SCN** hiérarchique, on adopte fréquemment un η_0 relativement grand au **niveau micro** pour identifier les clusters plus vite, tandis que, pour les niveaux macro ($\ell = 1, 2, \dots$), on utilise un η_k plus faible afin d’éviter que la structure globale ne se reconfigure trop brutalement au moindre changement.

Le **facteur** $\tau > 0$ représente la décroissance dans la même dynamique, par l’expression $\tau \omega_{i,j}$ à retrancher de la mise à jour. Cet élément évite la croissance sans limite des pondérations et limite la persistance de liaisons peu utiles. Un τ plus grand accentue l’**oubli**, c’est-à-dire que tout lien non soutenu par une synergie $S(i,j)$ élevée tendra rapidement vers zéro. À l’inverse, une valeur de τ trop faible maintient un réseau plus **dense**, en conservant de nombreux liens faibles. Cela augmente la complexité de la structure et réduit la sélectivité dans la formation des clusters, pouvant conduire à des agrégations moins cohérentes.

Du fait de l’existence de **niveaux multiples** (ℓ), il est envisageable d’assigner à chaque palier ℓ ses propres valeurs η_ℓ et τ_ℓ . De cette façon, la **dynamique DSL** se module en fonction de la granularité. Une approche classique consiste à rendre η_0 (et parfois τ_0) relativement importants, de sorte que le niveau micro réagisse vivement pour créer ou dissoudre rapidement des liaisons, et à choisir η_k, τ_k plus tempérés lorsqu’on s’élève dans la hiérarchie, de manière à ne pas ébranler trop fréquemment la configuration macro. En procédant ainsi, on préserve la **réactivité** souhaitée dans les couches inférieures, tout en garantissant une plus grande **stabilité** du réseau aux niveaux supérieurs.

D. Paramètres de Filtrage et de Sélection dans l’Agrégation

Lors de la création de **super-nœuds** au niveau $\ell + 1$ à partir de **clusters** de niveau ℓ , on introduit une **fonction** Ψ pour définir la pondération $\omega_{\alpha,\beta}^{(\ell+1)}$. De manière générale, on écrit

$$\omega_{\alpha,\beta}^{(\ell+1)} = \Psi\left(\{\omega_{i,j}^{(\ell)} \mid i \in \mathcal{C}_\alpha^{(\ell)}, j \in \mathcal{C}_\beta^{(\ell)}\}\right),$$

ce qui signifie que l'on agrège l'ensemble des liens $\omega_{i,j}^{(\ell)}$ reliant les entités du bloc $\mathcal{C}_\alpha^{(\ell)}$ à celles du bloc $\mathcal{C}_\beta^{(\ell)}$.

Le **choix** de Ψ dépend de la logique du réseau et peut prendre différentes formes comme la somme, la moyenne, le maximum, le minimum ou un quantile.

Un **maximum** favorise la propagation des liens les plus forts et peut surestimer la synergie globale, alors qu'une **moyenne** rend l'agrégation plus modérée. Dans certains contextes, un **minimum** ou un **quantile** s'avère judicieux pour refléter la robustesse d'un lien dominant ou, au contraire, l'homogénéité générale d'un ensemble. Une fois $\omega_{\alpha,\beta}^{(\ell+1)}$ calculé, un **filtrage** ou un **seuil de transmission** $\theta_{(\ell+1)}$ est souvent appliqué afin de ne conserver que les super-arêtes dont la pondération dépasse cette valeur, ce qui réduit la densité du graphe au palier macro ou méso.

Cette opération influence directement la forme hiérarchique résultante, car elle régit la connectivité entre super-nœuds : si le filtre est trop strict, on risque d'aboutir à une structure dispersée de blocs faiblement reliés ; s'il est trop permissif, le réseau demeure très dense, et les fusions tendent à se multiplier, parfois sans réelle cohésion. L'équilibre entre Ψ et le filtrage $\theta_{(\ell+1)}$ s'avère donc déterminant pour réguler la **logique DSL** à chaque palier et assurer une **auto-organisation** stable et adaptée au niveau d'échelle visé.

E. Combinaisons et Calibration Globale

L'**hystérésis** et l'introduction de **fenêtres de stabilisation** constituent l'une des clés pour limiter les oscillations brutales dans un système multi-niveau. Dans un **algorithme DSL** complexifié, on empêche qu'une fusion ou une scission déjà réalisée ne soit immédiatement inversée par la phase suivante. On se dote par exemple de deux seuils de fusion, $\theta_{\text{fusion}}^{\text{high}}$ et $\theta_{\text{fusion}}^{\text{low}}$, afin de fusionner deux blocs si $\omega_{i,j}$ dépasse $\theta_{\text{fusion}}^{\text{high}}$, tout en s'assurant de ne pas "dé-fusionner" tant que le poids reste au-dessus de $\theta_{\text{fusion}}^{\text{low}}$. Un principe analogue s'applique aux divisions. Un bloc fraîchement scindé ne sera pas réagrégré tant que sa pondération ou sa cohésion interne n'a pas atteint un seuil supérieur. Cette stratégie atténue le "va-et-vient" permanent, permettant une **stabilisation** avant que le mécanisme inverse ne prenne le relais.

La **gestion multi-échelle** impose souvent que chaque niveau ℓ du réseau hiérarchique puisse définir ses propres paramètres $\{\eta_\ell, \tau_\ell, \theta_{\text{synergy}}^{(\ell)}, \theta_{\text{macro}}^{(\ell)}\}$. Cette variabilité rend possible la **réactivité** à petite échelle (où η_0 et τ_0 sont assez importants pour permettre des ajustements rapides) et la **stabilité** à grande échelle (où η_k et τ_k décroissent ou se fixent à des valeurs plus modérées). Bien que cette souplesse des réglages soit précieuse, elle complique la configuration globale. Dans la pratique, on recourt à des essais ou à une adaptation automatique pour ajuster ces seuils et vitesses de mise à jour.

Les **boucles d'itération** impliquent trois étapes clés qui doivent être orchestrées avec précision. La première est la mise à jour **DSL locale**, où les paramètres η_0 et τ_0 ajustent les pondérations $\omega_{i,j}$ en fonction de la synergie locale, renforçant ou affaiblissant les liens selon leur pertinence. La seconde est la **détection et fusion**, qui repose sur un seuil θ_{synergy} permettant d'agréger les entités ou sous-blocs fortement cohérents dans une approche ascendante. Enfin, la **validation ou division macro** intervient lorsqu'un bloc présente une hétérogénéité trop élevée, mesurée par $\mathcal{H}(\mathcal{C})$ et

comparée à un seuil θ_{macro} , entraînant une scission descendante pour maintenir la cohérence du réseau.

L'ordre et la périodicité de ces actions doivent être calibrés avec soin afin d'assurer une convergence harmonieuse. Un feedback macro déclenché trop tôt risquerait de fragmenter des clusters encore en formation, tandis qu'un retard excessif dans la division maintiendrait des super-blocs incohérents. La coordination entre la dynamique locale, l'agrégation et la division représente ainsi un enjeu fondamental dans la **conception** d'un SCN multi-niveau.

6.5.4.3. Intégration avec l'Inhibition ou le Recuit (Chap. 7) pour Éviter la Stagnation

Introduction et Contexte. Les algorithmes multi-niveau (section 6.5.4.1) reposant sur l'**agrégation** (bottom-up) et la **division** (top-down) nécessitent un certain “dynamisme” pour évoluer vers des structures hiérarchiques suffisamment pertinentes. Or, il arrive qu'un **blocage** ou une **stagnation** survienne lorsque le réseau se fige dans une configuration localement stable sans parvenir à un équilibre global. Certaines liaisons $\omega_{i,j}$ peuvent rester à des valeurs intermédiaires, empêchant ainsi la différenciation de sous-groupes plus cohérents. Pour s'extraire de ces situations, on recourt à des mécanismes complémentaires comme l'**inhibition** ou le **recuit simulé**, décrits en détail au **Chap. 7**. Les développements ci-après résument l'idée et montrent comment ces procédés s'intègrent dans le cadre DSL multi-niveau.

A. Principe de l'Inhibition

Dans la dynamique DSL classique, la mise à jour des pondérations suit la formule $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$. Il est cependant possible d'ajouter un **terme d'inhibition** pour contraindre la somme des liens sortants $\sum_k \omega_{i,k}(t)$ et inciter chaque nœud \mathcal{E}_i à limiter la densité de ses connexions. Une forme représentative s'écrit

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t),$$

avec $\gamma > 0$ modulant la force inhibitrice. Ce mécanisme, qualifié d'**inhibition latérale ou globale**, a pour effet de “pousser” chaque entité i à sélectionner quelques liaisons prioritaires tout en relâchant les autres, ce qui favorise un **contraste** plus marqué entre liens forts et liens faibles et empêche la stagnation dans un état moyennement connecté.

Lorsque l'on souhaite éviter qu'un cluster se stabilise dans une configuration sous-optimale où de nombreux liens médians subsistent, l'inhibition pousse en effet $\omega_{i,j}$ vers zéro sauf pour les paires (i,j) réellement synergiques.

Sur un plan **multi-niveau**, le palier supérieur peut introduire une inhibition partielle pour réguler l'expansion des super-nœuds surconnectés, limitant ainsi leur extension artificielle. Cette approche rejoint la logique d'une division top-down où, lorsqu'un bloc devient trop vaste ou trop hétérogène, un **feedback** descendant réduit certains liens internes, facilitant ainsi une scission ultérieure et assurant une structure globale plus équilibrée.

B. Principe du Recuit Simulé (Injection de Bruit)

Le **recuit simulé** introduit un terme aléatoire dans la mise à jour des pondérations afin d'échapper aux états de blocage ou aux minima locaux. La **dynamique** se généralise alors en

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)] + \sigma(t) \xi_{i,j}(t),$$

où $\xi_{i,j}(t)$ correspond à un bruit (gaussien, uniforme) et $\sigma(t)$ définit une “température” positive qui décroît au fil des itérations ($\sigma(t) \rightarrow 0$ quand $t \rightarrow \infty$). Lorsque $\sigma(t)$ est élevée, les fluctuations dans $\omega_{i,j}$ demeurent importantes, ce qui encourage une **exploration** de nombreuses configurations. À mesure que la température diminue, le réseau “gèle” progressivement sa structure dans un état supposé plus stable ou plus optimal. L'absence de bruit dans un **DSL** peut conduire à une convergence précoce vers un **minimum local**, alors que l'injection contrôlée de $\sigma(t)$ autorise certaines liaisons $\omega_{i,j}$ à monter ou à descendre aléatoirement, même si la synergie $S(i,j)$ n'est pas très marquée, révélant ainsi des sous-groupes latents ou détruisant des connexions faiblement fondées. L'**agrégation** se trouve de ce fait renforcée lorsqu'un couple (i,j) bénéficie d'un surcroît de bruit positif, et la **division** top-down peut s'opérer plus facilement si le bruit négatif fragilise des liens internes déjà instables. Au final, le recuit simulé, par la variation $\sigma(t)$, offre une flexibilité complémentaire à la dynamique DSL, favorisant l'**exploration** quand la température est haute et la **consolidation** quand la température se rapproche de zéro.

C. Combinaison dans un Cycle Multi-Niveau

Les **phases** d'un **SCN** hiérarchique suivent un cycle dans lequel on exécute successivement la **mise à jour** DSL locale, l'agrégation ascendante (voir 6.5.3.2) et la potentielle division descendante (selon la logique top-down).

Le **recuit** peut être intégré au cours de la mise à jour en introduisant un terme $\sigma(t) \xi_{i,j}(t)$ afin de relancer la dynamique lorsqu'une stagnation est détectée. De même, l'on peut recourir à un **terme d'inhibition** si on estime qu'un nœud ou qu'un bloc agrège trop de liaisons, en réduisant spécifiquement la somme de ses pondérations $\sum_j \omega_{i,j}$.

Dans les faits, il arrive qu'on conditionne l'activation du bruit ou de l'inhibition à la détection d'un faible gradient $\|\Delta \omega(t)\|$ signalant une convergence prématurée. On peut alors, pour débloquer la situation, élever temporairement $\sigma(t)$ (pour le recuit) ou γ (pour l'inhibition). La structure hiérarchique profite ainsi d'un “**shake-up**” contrôlé qui modifie l'agencement local ou macro, libérant le réseau d'un éventuel état sous-optimal.

L'**équilibre** final suppose de faire décroître la température $\sigma(t)$ et de stabiliser la force inhibitrice, de sorte que le SCN se cristallise dans une configuration plus stable. Cette combinatoire d'agrégation, de scission, de recuit et d'inhibition, orchestrée dans un **cycle multi-niveau**, constitue l'une des stratégies les plus robustes pour organiser un SCN complexe, lui permettant de traverser les minima locaux, de réajuster ses blocs hiérarchiques et de maintenir une flexibilité face aux évolutions de la synergie ou aux aléas du réseau.

Exemple d'un Pseudo-Code.

Un modèle :

```

for iteration in 1..T:
  for each pair (i,j):
     $\omega(i,j) \leftarrow \omega(i,j) + \eta * [S(i,j) - \tau * \omega(i,j)]$ 
      +  $\sigma(t) * \text{random\_noise}(i,j)$  # recuit
      -  $\gamma_{\text{inhibit}}(i) * \text{sum}(\omega(i,*))$  # inhibition latérale
  if  $\text{norm}(\Delta\omega) < \text{epsilon}$ :
    raise  $\sigma(t)$  or  $\gamma_{\text{inhibit}}$  # or apply "pulse"
  # then do detection of clusters, do macro merges or splits, etc.

```

Cette structure assure qu'on ne s'enferme pas trop tôt dans des configurations rigides, et qu'on a la **possibilité** de se ré-agencer.

D. Avantages

Le principal avantage est de sortir d'une *configuration stable* localement en introduisant un aléa modéré (recuit) ou en contraignant la somme des liens (inhibition). On évite ainsi les configurations indécises où de nombreuses liaisons intermédiaires empêchent la formation de clusters clairs, ou où un super-nœud mal structuré demeure intact faute d'une scission appropriée. Sur le plan **mathématique**, l'approche recuit s'inspire de techniques globales de minimisation d'énergie. En embrassant un peu plus d'exploration, on peut accéder à un **minimum** plus global, dépassant le minimum local où le DSL se serait bloqué. L'inhibition, quant à elle, impose une **sélection** plus franche des connexions, clarifiant la hiérarchie.

Les mécanismes d'inhibition et de recuit peuvent être utilisés ensemble. On peut, par exemple, lancer le recuit au début (température élevée) pour explorer, puis introduire l'inhibition latérale en cours de route afin de forcer la **sparsité**. Cette séquence suit la dynamique d'**agrégation** et de **division** où la fusion devient plus fluide après une phase de réajustement du réseau, tandis que la division se réalise plus nettement lorsque l'inhibition a clarifié les liens.

6.6. Études de Cas et Illustrations

Les principes **multi-niveaux** du **DSL** (Deep Synergy Learning) et les mécanismes d'**auto-organisation** qu'il propose (agrégation, division, synchronisation, etc.) trouvent des champs d'application variés et concrets. Dans cette section 6.6, nous passons en revue quelques **études de cas** et **domaines** illustrant l'impact potentiel de la synergie multi-échelle. Qu'il s'agisse de **vision multi-modal**, d'**analyse contextuelle** de la parole, de **robotique sensorimotrice**, d'**agents conversationnels** ou de **simulation d'événements**, le fil conducteur demeure la capacité du réseau à gérer les **données** (ou entités) à différents **paliers**, créant un équilibre entre détails locaux et structures globales.

6.6.1. Systèmes de Vision Multi-Modal

Les systèmes de **vision** actuels doivent non seulement reconnaître des images, mais souvent intégrer plusieurs **modalités** (ex. image + annotation textuelle, ou image + métadonnées audio). Dans un contexte **multi-modal**, le **DSL** propose de structurer l'information en **entités** et **liens synergiques**, afin de repérer des clusters pertinents à plusieurs échelles.

6.6.1.1. Micro-Clusters = Images Localement Semblables, Macro = Classes Sémantiques

Dans un système de **vision** à grande échelle, la gestion d'un grand nombre d'images passe par une organisation hiérarchique. Au **niveau local**, les patches aux caractéristiques similaires, comme les textures, les couleurs et les formes, se regroupent en micro-clusters. Au **niveau global**, ces groupes s'agrègent pour former des **catégories sémantiques**, correspondant aux types d'objets ou aux classes d'usages. La **dynamique DSL**, avec sa mise à jour adaptative des pondérations ω et son agrégation multi-niveau, structure naturellement cette organisation. Ce processus permet la formation de micro-clusters de "mini-images" qui, par agrégation progressive, aboutissent à des classes plus abstraites.

A. Micro-Clustering Local : Groupement de Patches Visuels

Au niveau le plus **bas** (noté $\ell = 0$), on considère un ensemble massif de **mini-images** ou "patches" extraits des grandes images. Chaque patch \mathcal{E}_i est relié à d'autres patches \mathcal{E}_j par des pondérations $\omega_{i,j}$ mesurant leur **similarité** visuelle locale. Cette similarité (score $S(i,j)$) peut reposer sur des comparaisons de **couleurs**, de **descripteurs** SIFT ou HOG, ou encore de **textures**.

La mise à jour DSL local applique la formule :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta_0 [S_0(i,j) - \tau_0 \omega_{i,j}(t)].$$

Lorsque $S_0(i,j)$ est élevé (patches très semblables), $\omega_{i,j}$ tend à croître, **favorisant** la formation de groupes de patches proches. Après plusieurs itérations, on détecte ainsi, via un critère de sur-seuil ou un algorithme de **connected components**, des **micro-clusters** de patches. Ces micro-clusters

peuvent correspondre à des **motifs** récurrents (ciel bleu, feuillage vert, zones de textures identiques, etc.) ou à des éléments communs (fragments de visages, de lettres, etc.).

Le **rôle** de cette étape micro est d'extraire une **granularité** fine, en laissant la dynamique DSL créer des petits regroupements localement cohérents. Il n'est pas nécessaire d'avoir un label humain ou une connaissance externe, car l'**auto-organisation** repose uniquement sur le renforcement des liens $\omega_{i,j}$ les plus **synergiques**, reliant naturellement les patches quasi identiques ou très proches.

B. Passage Macro : Classes Sémantiques

Une fois les micro-clusters formés, on se place au **niveau** $\ell = 1$, où chaque cluster local de patches se voit représenté par un **super-nœud** $\mathcal{N}_\alpha^{(1)}$. Les pondérations $\omega_{\alpha,\beta}^{(1)}$ entre ces super-nœuds se calculent via une fonction Ψ (moyenne, somme, maximum, etc.) appliquée aux liens $\{\omega_{i,j}^{(0)}\}$ entre les patches membres de α et β .

Sur ce graphe de **super-nœuds**, on applique de nouveau la démarche DSL ou un critère d'**homogénéité** pour regrouper en classes plus vastes. On peut ainsi former un macro-nœud $\mathcal{N}_p^{(2)}$ englobant plusieurs super-nœuds $\{\mathcal{N}_\alpha^{(1)}, \dots\}$. Ces macro-nœuds, ou **catégories** plus globales, peuvent correspondre à de véritables **concepts** sémantiques comme les animaux, les véhicules, les paysages ou les visages.

La **logique** d'agrégation hiérarchique permet donc, d'une part, de détecter localement les similitudes visuelles (couches micro) et, d'autre part, d'**unifier** ces similarités en classes d'objets pertinents (couche macro). Le DSL répercute la force des liens $\omega_{i,j}$ depuis le niveau patch (textures) jusqu'à la reconnaissance de **familles** d'images plus abstraites.

C. Retombées Pratiques et Architecture Finalisée

À l'**échelle micro**, on dispose d'informations extrêmement **locales**, comme un patch de taille 8×8 représentant une petite zone homogène. La **dynamique** DSL fait émerger des micro-clusters homogènes, utiles pour un pré-classement ou une indexation fine. Au **niveau macro**, l'agrégation délivre des **classes** globales qui reflètent des **catégories** ou des **objets**. Ainsi, dans une base de plusieurs millions d'images, on aboutit à une structure hiérarchique où chaque image se décompose en divers patches, lesquels appartiennent à des micro-clusters, et ces micro-clusters se rattachent à des catégories plus vastes.

Cette **organisation** hiérarchique facilite la **navigation** multi-résolution en permettant une recherche par similitude locale, pour identifier des patches proches d'un motif donné, ou une exploration par **grande classe**, afin de retrouver des images correspondant à des catégories générales comme un chat, une voiture ou un paysage enneigé. Le **DSL** assure la liaison entre ces deux échelles, en propageant la **cohérence** visuelle locale vers des regroupements plus sémantiques.

6.6.1.2. Approche Fractale : Répétition de Motifs “Objets” à Différentes Granularités (ex. Sous-Catégories)

De nombreux systèmes de **vision multi-modal** illustrent la présence d'**objets** ou de *sous-objets* dont la **forme** ou la **structure** se retrouve à diverses **échelles**. Un objet complexe (tel qu'un bâtiment, un végétal) peut receler des éléments répétés (fenêtres, feuilles) présentant des caractéristiques quasi identiques, simplement redimensionnées ou agencées différemment. Cette **répétition** et **auto-similarité** évoque le **concept fractal** (cf. chap. 6.3 sur la fractalité). Du point de vue **Deep Synergy Learning (DSL)**, la possibilité de décrire un motif à l'échelle micro et de le retrouver dans une **catégorie** plus large (macro) s'appuie sur le principe d'**auto-organisation** hiérarchique. Les développements ci-après explorent en quoi cette logique fractale se matérialise et quels bénéfices elle apporte à la reconnaissance d'images (ou de sous-catégories d'objets) via le **DSL**.

A. Répétition de Motifs et Multi-Échelle dans les Images

La notion de **fractal** en vision repose sur une **invariance** lorsqu'on change d'échelle. Les mêmes *motifs* réapparaissent, que l'on observe une zone microscopique ou l'image entière. Dans un système DSL, cette propriété se manifeste par deux phénomènes.

Les **motifs locaux répétés** apparaissent dans de nombreuses images naturelles, où des *sous-formes* identiques sont distribuées au sein d'un *grand* objet. Par exemple, des fenêtres similaires sont réparties sur une façade ou plusieurs roues identiques composent un convoi de wagons. Au **niveau micro**, le DSL repère ces similitudes entre patches partageant la même texture ou géométrie et les agrège progressivement. Le résultat est un **cluster** local reflétant un motif récurrent, comme une forme de fenêtre ou un détail architectural.

Les **sous-catégories issues d'un même patron** émergent lorsqu'on passe du niveau des patches micro à un niveau de classes plus larges. Dans la classification d'espèces végétales, des arbres présentant un motif foliaire similaire forment une sous-catégorie spécialisée, bien que leur structure générale suive un modèle récurrent. Le **DSL** hiérarchique regroupe ces clusters locaux en *sub-classes* macro, transmettant la cohérence de la structure locale à une catégorie englobante. Cette dynamique reflète l'**auto-similarité** propre aux fractals, où une même organisation se répète à différentes échelles, maintenant une structure cohérente tout en s'adaptant au niveau de détail.

B. Notion de Fractalité dans la Vision

L'**auto-similarité** fractale est présente dans de nombreuses images naturelles comme les motifs de nuages ou de feuillages. Dans un **DSL**, qui permet à la fois une mise à jour locale par renforcement de $\omega_{i,j}$ et une agrégation ou division multi-niveau, cette structure peut être mise en évidence de plusieurs manières.

Le **box-counting** et les **lois de puissance** permettent d'analyser la répartition des groupes en fonction de leur taille ou d'évaluer l'intensité moyenne de similarité à différentes échelles. Si le graphe des $\omega_{i,j}$ affiche une **dimension fractale** lors du *box-counting*, cela révèle une **récence structurelle**, prouvant que la hiérarchie conserve une **auto-similarité** sur plusieurs niveaux.

Un **exemple de sous-catégories** illustre cette idée. À l'échelle **macro**, la catégorie “bâtiments” apparaît. En descendant d'un niveau, on distingue des sous-groupes comme “bâtiments à fenêtres rectangulaires” et “bâtiments à arches rondes”. En explorant plus en détail, chaque fenêtre ou arche

se divise en éléments plus petits selon une même **logique d'agencement**. Le **DSL** met à jour ω en suivant cette cohérence : un cluster trop hétérogène se divise, un micro-groupe homogène se fusionne, et cette *répétition d'organisation* à différents niveaux traduit une **structure fractale**.

C. Avantages d'une Approche Fractale en Vision

Robustesse à l'Échelle. Les systèmes de vision doivent gérer des variations d'échelle (un objet vu de près ou de loin) et de perspective. Dans une **structure fractale**, la répétition de motifs à différentes tailles autorise l'identification d'un *même type d'entité* qu'il soit grand ou petit dans l'image. Le **DSL** hiérarchique renforce cette capacité de reconnaître des patches similaires même s'ils sont "scalés" ou partiels, aboutissant à la **convergence** d'indices vers une même catégorie.

Multi-Résolution. Une scène peut être observée à différentes échelles. De loin, un objet apparaît comme une forme globale sans détail, tandis qu'en zoomant, des micro-structures émergent et révèlent des patterns locaux. Le **DSL**, structuré de manière fractale, exploite cette double perspective en regroupant les entités à grande échelle tout en préservant des micro-clusters cohérents. Cette organisation garantit une continuité auto-similaire entre les différents niveaux d'analyse.

Découverte de Structures. Sans supervision externe, un réseau DSL auto-similaire découvre des **patterns récurrents** (fenêtres, roues, feuilles) au niveau micro, les agrège en "sous-ensembles d'objets" (ex. sous-catégories de voitures selon le type de roues), puis les regroupe dans des super-catégories (voitures, camions, chars) au niveau macro. Cette **imbrication** reflète une structure fractale où chaque entité complexe est constituée d'entités plus petites suivant le même schéma. Cette répétition à différentes échelles crée un effet de **mise en abyme**, garantissant une organisation cohérente et homogène à tous les niveaux du réseau.

6.6.1.3. Gains en Robustesse : Le Système s'Adapte si de Nouvelles Images Arrivent (Chap. 9)

Dans un **système de vision** reposant sur l'**architecture multi-niveau** d'un **Deep Synergy Learning (DSL)**, il est fréquent d'ajouter de nouvelles images (ou de nouveaux flux multimédias) au fil du temps. La capacité d'intégrer ces données additionnelles sans redémarrer l'entraînement à zéro, ni déstabiliser la structure hiérarchique, constitue un atout majeur en matière de **robustesse**. Les développements qui suivent clarifient la manière dont, grâce à la logique DSL, l'arrivée d'images inédite se traduit par une **mise à jour** locale des pondérations, qui se propage aux niveaux supérieurs (macro) de manière continue et adaptative.

A. Réactivité des Liens Synergiques face à l'Arrivée de Nouvelles Images

Le **DSL** procède par mise à jour itérative des pondérations $\omega_{i,j}(t)$ à l'aide d'une formule du type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ désigne un taux d'apprentissage, $\tau > 0$ un facteur de décroissance, et $S(\mathcal{E}_i, \mathcal{E}_j)$ la **synergie** (ou similarité) entre deux entités $\mathcal{E}_i, \mathcal{E}_j$. Lorsqu'une image inédite \mathcal{E}_{new} survient, on l'introduit comme un nouveau nœud dans le réseau, initialisant $\omega_{\text{new},j}$ à de petites valeurs (ou zéro)

et évaluant $S(\mathcal{E}_{\text{new}}, \mathcal{E}_j)$ pour les entités $\{\mathcal{E}_j\}$ préexistantes. Les mises à jour successives renforcent ou réduisent $\omega_{\text{new},j}$ en fonction de la cohérence (ou non) entre \mathcal{E}_{new} et \mathcal{E}_j . Si la synergie est élevée, la nouvelle image consolide un lien fort et se rattache à un cluster existant ; si la synergie reste faible, elle demeure isolée ou forme un micro-cluster indépendant, jusqu'à ce que d'autres images similaires surviennent.

Après un certain temps, les **informations** liées à \mathcal{E}_{new} (ses liens forts et faibles) s'intègrent naturellement dans la structure multi-niveau. Le super-nœud correspondant au cluster local l'incorpore progressivement, tandis que les pondérations $\omega_{\alpha,\beta}^{(\ell+1)}$ sont ajustées via la fonction Ψ . Cette mise à jour assure une adaptation fluide des liaisons au **niveau macro** sans imposer de restructuration globale.

B. Robustesse et Adaptation

Ce mécanisme permet au réseau de **continuer** son évolution en absorbant de la **nouvelle** information sans ré-apprendre globalement.

Dans des algorithmes classiques (par ex. entraînement supervisé avec un réseau de neurones convolutionnel), l'arrivée d'un nouveau lot d'images out-of-distribution impose souvent de **réentraîner** (ou au moins de fine-tuner) le modèle sur l'ensemble des données. Avec le **DSL**, on ne reconstruit pas l'architecture, on **insère** \mathcal{E}_{new} au sein de la matrice des pondérations, et on laisse les règles de synergie influencer sur la consolidation ou la suppression de liens.

Si la nouvelle image se rapproche d'un cluster existant, comme un groupe d'images de chats, elle renforce naturellement le micro-cluster correspondant. Si, en revanche, elle introduit une hétérogénéité inattendue et qu'un micro-cluster devient trop vaste et disparate, une **division** peut être appliquée au niveau macro (section 6.5.2). Un bloc d'images se scinde alors pour réajuster la structure globale. Ce mécanisme garantit une **flexibilité** du système, lui permettant de reconfigurer partiellement la hiérarchie en fonction des nouvelles données.

Lorsque la nouvelle image est anomalie ou bruit, ses liaisons demeurent faibles (faible $S(\mathcal{E}_{\text{new}}, \mathcal{E}_j)$), donc $\omega_{\text{new},j}$ ne grandit pas. Elle peut être isolée ou former un cluster marginal avec d'autres images bruitées. Cette structure peu connectée ne perturbe pas la classification globale, ce qui renforce la **robustesse**.

C. Dimensions Mathématiques de l'Adaptation

Dans un réseau déjà stabilisé, l'ajout d'une nouvelle image peut rester sans effet si aucune pondération $\omega_{\text{new},j}$ ne réussit à s'élever au-delà d'un seuil. Pour **assurer** qu'on réexamine les liens, on peut injecter un **bruit** contrôlé (recuit simulé, cf. chap. 7) ou un **terme** d'inhibition forçant un rééquilibrage local. Mathématiquement :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)] + \sigma(t) \xi_{i,j}(t) - \gamma \sum_k \omega_{i,k}(t).$$

Cette version plus complète mixe bruit ($\sigma(t) \xi$) et inhibition latérale ($\gamma \sum_k \omega_{i,k}$), qui aident à **sortir** d'un équilibre trop figé.

La logique **DSL** reste inchangée. Localement, $\omega_{i,j}$ s’ajuste, entraînant une mise à jour du **cluster** concerné, qui se propage ensuite aux paliers supérieurs via la fonction Ψ . Après un certain nombre d’itérations, la nouvelle image est pleinement **intégrée**. Si son introduction a provoqué une incohérence, le bloc concerné se sera soit **subdivisé**, soit **re-fusionné partiellement** pour retrouver un état stable.

D. Liens avec Chap. 9 (Évolutions Temps Réel)

Le **chapitre 9** approfondit cette idée d’**apprentissage continu** ou **streaming**, où les données ne sont pas figées mais arrivent en flux constant. Le **DSL**, en assurant un algorithme **auto-organisé** et **distribué**, traite cette arrivée progressive d’images, guidé par la notion de synergie $\{\omega_{i,j}\}$. La **robustesse** ainsi conférée se traduit par :

- Un **coût** de mise à jour local (pas de recalcul global).
- Une possibilité de **réorganisation** si nécessaire (division top-down, recuit pour sortir d’un plateau).
- Une **navigation hiérarchique** qui reste fonctionnelle au fur et à mesure que de nouveaux objets ou de nouvelles catégories émergent.

6.6.2. Analyse Contextuelle de la Parole et du Langage

Les principes du **DSL** (Deep Synergy Learning), initialement illustrés en vision multi-modal (6.6.1), s’appliquent tout aussi bien à l’**analyse** de la parole ou du texte. Le langage humain se compose en effet de **niveaux multiples** — des sons élémentaires (phonèmes) jusqu’aux concepts abstraits — offrant ainsi un cadre naturel pour la synergie multi-échelle.

6.6.2.1. Micro : phonèmes ou chunks de phrase, macro : concepts sémantiques

Au plus bas niveau (micro), la parole ou le texte se décompose en **unités élémentaires**. On retrouve les **phonèmes** pour l’oral et les **morphèmes** ou “chunks” pour l’écrit. Par exemple, un flux sonore est découpé en courtes séquences (ex. 20–50 millisecondes) dont on extrait des **features** (MFCC, spectrogrammes locaux...), alors qu’un texte peut se scinder en tokens ou petits segments (une partie d’un mot, un groupement de mots, etc.).

Le **DSL**, à ce niveau, gère un ensemble d’entités $\{\mathcal{E}_i\}$ représentant ces briques linguistiques. Les pondérations $\omega_{i,j}$ quantifient la **synergie** ou la similarité entre fragments. Un $\omega_{i,j}$ élevé signifie que deux phonèmes (ou deux tokens) **coïncident** souvent ou s’articulent fréquemment dans un même contexte local.

Lorsque ces fragments s’**agrègent**, ils forment des unités plus larges comme les **mots**, les **groupes de mots**, puis des **concepts** sémantiques regroupant des familles de termes. Le **DSL** réévalue alors la synergie $\omega_{\alpha,\beta}$ à ce palier, permettant d’associer des notions proches comme “ordinateur”, “programme” et “machine”.

Mathématiquement, on définit une fonction d'agrégation Ψ qui combine les pondérations $\omega_{i,j}^{(0)}$ en pondérations $\omega_{\alpha,\beta}^{(1)}$ au niveau conceptuel. On obtient alors un graphe sémantique plus abstrait, où les nœuds sont des **concepts** détectés (macro-niveau).

Dans un **DSL**, la distinction micro–macro s'opère **dynamiquement**, sans imposer un modèle figé comme un “parse tree” unique. On peut former plusieurs **clusters** de tokens/phonèmes selon le contexte. L'architecture multi-niveau (chap. 6.2) laisse place à une organisation hiérarchique plus flexible que le schéma strict “mot \rightarrow syntagme \rightarrow phrase”.

Cette **souplesse** permet au système de repérer de nouveaux concepts, d'en fusionner certains, ou d'en séparer d'autres, en fonction de la synergie calculée lors de l'analyse sémantique ou phonétique.

A. Exemples de Mise en Pratique

Reconnaissance de la Parole (ASR)

Au niveau **micro**, des segments audio (frames) sont reliés en fonction de leurs similarités spectrales, formant des **clusters** phonétiques. Puis, au niveau plus haut, on agrège ces phonèmes en **mots**. Enfin, on peut détecter des concepts / intentions (macro).

Le **DSL** facilite la prise en compte simultanée de différents signaux acoustiques (voix), et de la synergie entre phonèmes courants pour **valider** ou **corriger** un mot ambigu.

Analyse Sémantique de Texte

Au niveau **micro**, un texte se segmente en tokens (mots ou sous-mots). Les liens $\omega_{i,j}$ se renforcent si deux tokens apparaissent souvent ensemble ou dans des contextes semblables. Des **clusters** de mots synonymes ou fortement associés émergent localement.

Au **niveau macro**, les **concepts** s'organisent en thèmes regroupant un champ lexical complet, comme un cluster “sport” ou un cluster “finance”. Ces concepts structurent alors un graphe sémantique global, offrant une vision d'ensemble des relations entre les différentes notions.

B. Intérêt Mathématique et Pratique

Mathématiquement, la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ peut se définir comme une **similarité** sémantique (ex. co-occurrence, vecteurs embedding) ou une **compatibilité** phonétique. Le DSL actualise les pondérations $\omega_{i,j}$ par la formule :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t)\eta[S(i,j)\tau\omega_{i,j}(t)].$$

Plus les entités (tokens, phonèmes) apparaissent conjointement, *plus* le lien se renforce. Au contraire, si elles se dissocient, $\omega_{i,j}$ tend vers 0 (ou un niveau faible).

Au fur et à mesure qu'on monte en échelle, on combine $\{\omega_{i,j}^{(k)}\}$ via une fonction Ψ . Les super-nœuds (macro) reflètent des concepts plus abstraits.

Cette hiérarchie dynamique est **robuste** puisque l'apparition fréquente d'un nouveau token lui permet soit d'intégrer un cluster existant, soit d'en former un nouveau. Grâce aux mécanismes adaptatifs (voir 6.5.4.2 sur les paramètres), aucune **réinitialisation** ou **réapprentissage** global n'est nécessaire.

Applications

Compréhension du langage : on peut imaginer un DSL gérant la **cohérence contextuelle** (ex. “ce segment phonétique se connecte fortement à tel concept s'il apparaît dans un champ lexical précis”).

Correction automatique : si un token se retrouve isolé (faible synergie), on peut le **réassigner** à un cluster sémantique proche, jouant un rôle analogue aux algorithmes de correction contextuelle.

6.6.2.2. Multi-Échelle : Phrase \rightarrow Paragraphe \rightarrow Document, Fractalité Possible si la Structure se Répète (Thèmes, Sous-Thèmes)

Introduction et Contexte Général. Le **langage** (oral ou écrit) s'organise naturellement en **paliers** imbriqués. Le palier **micro** regroupe les mots ou tokens, qui s'assemblent en **phrases**, puis en **paragraphes** formant des sous-thèmes, jusqu'au **document** complet qui véhicule un thème global. Cette hiérarchie est omniprésente dans les textes tels que les articles, essais ou romans. Dans un **Deep Synergy Learning (DSL)** multi-niveau, la **synergie** des pondérations $\omega_{i,j}$ et la **dynamique** d'agrégation et de division façonnent cette organisation. Lorsqu'un même thème ou sous-thème se manifeste à plusieurs échelles, des **patterns** auto-similaires peuvent émerger, rappelant une structure fractale.

A. Paliers du Texte : De la Phrase au Document

Niveau Micro : la Phrase.

Le premier palier (micro) regroupe les **mots** (tokens, segments). On définit des pondérations $\omega_{i,j}$ évaluant la similarité ou la synergie entre deux mots (par exemple, affinité sémantique, co-occurrence, relations syntaxiques). Après application de la mise à jour DSL, on peut détecter des *clusters* de mots qui forment un **énoncé** cohérent. Sur le plan mathématique, cette phrase $\mathcal{C}_\alpha^{(0)}$ se caractérise par une somme de pondérations $\sum_{i,j \in \mathcal{C}_\alpha} \omega_{i,j}$ relativement importante, traduisant la focalisation contextuelle.

Niveau Mésoscopique : le Paragraphe.

Le paragraphe regroupe plusieurs **phrases** reliées par un sous-thème commun. Dans un **DSL** multi-niveau, chaque phrase \mathcal{C}_α devient un super-nœud $\mathcal{N}_\alpha^{(1)}$. Les liens $\omega_{\alpha,\beta}^{(1)}$ entre phrases se calculent en agrégeant les poids $\{\omega_{i,j}^{(0)}\}$. Un paragraphe cohésif présente donc un bloc de phrases fortement interconnectées au niveau 1, selon la fonction Ψ (moyenne, somme, etc.). Un paragraphe plus hétérogène peut inciter une **division** (top-down) si la cohésion est jugée insuffisante.

Niveau Macro : le Document.

Le document entier, ou un chapitre de livre, se construit en **fusionnant** les paragraphes partageant un thème central. La **dynamique DSL** reste active : si un document trop hétérogène émerge, il se scinde en chapitres distincts ; s’il conserve une forte cohérence, il devient un macro-nœud stable. Cette logique peut s’étendre à la connexion de plusieurs documents, formant ainsi un **réseau inter-articles**, facilitant l’analyse de corpus et la structuration de connaissances à grande échelle.

B. Fractalité et Répétition de Thèmes

Dans un *texte* ou *discours* structuré, un **même** concept (thème principal) se décline souvent en sous-thèmes, puis en sous-sous-thèmes, à différents paliers (chapitre, section, paragraphe, phrase). Cet agencement **auto-similaire** peut être vu comme fractal ; on retrouve le *même* motif d’organisation à des granularités distinctes, par exemple la reprise d’un motif argumentatif ou la récurrence d’un champ lexical sous des variations minimales.

Les textes et leurs distributions lexicales suivent souvent des **lois de puissance**, comme la loi de Zipf pour la fréquence des mots ou la répartition des occurrences de thèmes. Un **DSL** appliquant un cycle d’**agrégation–division** exploite cette **récurrence** en divisant un bloc lorsque son hétérogénéité interne dépasse un seuil et en fusionnant localement des entités très similaires. Cette dynamique génère une **autosimilarité**, où les mêmes règles de regroupement et de scission s’appliquent aux paragraphes, aux chapitres et aux structures plus vastes, assurant une organisation cohérente à différentes échelles du texte.

Si l’on trace les super-nœuds (chapitres, sections, paragraphes, phrases) dans un graphe DSL multi-niveau, on voit se dessiner un **arbre** (ou un “graphe fractal”) révélant l’invariance d’échelle. Un concept macroscopique (ex. “analyse fractale du texte”) se reflète en multiples sous-sections, qui elles-mêmes se divisent en paragraphes, etc., chacun reprenant le même *patron* argumentatif.

C. Synergie Multi-Échelle et Valeur Pratique

Grâce à cette **structure** hiérarchique, on navigue dans le texte depuis un **niveau** macro (thème principal) jusqu’à un **niveau** micro (détails dans une phrase). Sur le plan mathématique, on utilise $\omega_{\alpha,\beta}^{(k)}$ pour passer d’un paragraphe α à un autre paragraphe β si leur synergie est forte, ou pour descendre d’un paragraphe α vers ses **phrases** internes. Cette multi-résolution rend l’exploration plus souple.

Dans un contexte évolutif (un document collaboratif, un flux textuel), l’ajout d’un **nouveau paragraphe** ou d’une phrase modifiée se répercute localement (mise à jour des liaisons ω), pouvant mener à la **fusion** avec un sous-thème existant ou la scission d’un paragraphe devenu hétéroclite. Ce fonctionnement limite la nécessité de reconstruire la totalité de la structure, assurant une **adaptation** progressive.

Chaque *phrase* $\mathcal{C}_{\alpha}^{(0)}$ est un cluster de mots, agrégé en un super-nœud $\mathcal{N}_{\alpha}^{(1)}$ pour le *paragraphe*. Les paragraphes $\mathcal{N}_{\beta}^{(1)}$ se regroupent en $\mathcal{N}_{\beta}^{(2)}$ (chapitre), etc. À chaque niveau, la *même* mise à jour DSL ou la *même* condition d’homogénéité s’exerce, de sorte que l’on obtient, si les thèmes se répètent, une forme de *pattern fractal* (la distribution des liens ω suit une structure similaire d’un niveau à l’autre).

6.6.2.3. Rétroaction Top-Down si un Sujet Global est Identifié, Influant l'Interprétation Locale

Dans un **DSL** multi-niveau appliqué à l'**analyse du langage**, l'organisation des entités débute au **niveau micro** avec le regroupement des mots et fragments de phrases en clusters locaux. Ces ensembles évoluent ensuite vers des **sous-thèmes** ou des **paragraphes** au **niveau intermédiaire**, avant de constituer un **sujet global** au **niveau macro**. L'un des aspects fondamentaux du modèle DSL repose sur la **rétroaction descendante** où le **thème principal** identifié au niveau macro influence les niveaux inférieurs en ajustant certains liens ou en clarifiant des ambiguïtés locales. Ce processus assure une **cohérence** à travers les différents niveaux en adaptant l'interprétation des segments microscopiques selon la structure globale.

A. Identification d'un Sujet Global et Flux Descendant

Au fur et à mesure que le **DSL** agrège les liens $\omega_{i,j}$ entre entités (mots, expressions, etc.), on peut parvenir, par un processus d'**agrégation** (bottom-up), à un macro-nœud représentant un **sujet** majeur, tel qu'« économie », « biologie », « intelligence artificielle », ou tout autre concept principal. Cette identification s'effectue souvent via un **cluster** de grande taille, rendu cohérent par la somme ou la moyenne de ses liens internes ($\omega_{\alpha,\beta}$).

Une fois ce thème principal \mathcal{T} établi, un **flux** top-down est possible. Sur le plan **formel**, on introduit un terme $\Delta_{\text{down}}^{(\text{macro})}(i, j)$ dans la mise à jour des pondérations :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i, j) - \tau \omega_{i,j}(t)] + \Delta_{\text{down}}^{(\text{macro})}(i, j).$$

Ce terme $\Delta_{\text{down}}^{(\text{macro})}(i, j)$ peut être **positif** ou **négatif** selon que le macro-nœud \mathcal{T} estime utile de renforcer la liaison (i, j) (car il est jugé pertinent pour le sujet) ou de l'inhiber (car il est hors sujet).

B. Influence sur l'Interprétation Locale (Mots Ambigus, Chunks Polyvalents)

En **linguistique**, de nombreux mots (ex. « bank », « seal », « network ») présentent plusieurs sens possibles. Le choix d'une **interprétation** dépend fortement du **contexte**. Si le macro-nœud a identifié le sujet global $\mathcal{T} = \text{« finance »}$, alors un mot comme « bank » est lu dans son acception financière plutôt que géographique (rive d'un fleuve). Mathématiquement, le flux $\Delta_{\text{down}}^{(\text{macro})}$ agit sur les liens $\omega_{\text{bank,loan}}$ (renforcement) vs. $\omega_{\text{bank,river}}$ (diminution).

De la même manière, si le macro-niveau identifie un **champ lexical** comme « biologie », les tokens ambigus au niveau micro sont orientés vers des clusters locaux en accord avec ce thème tandis que leurs liens avec d'autres domaines sont atténués. Cela permet de **stabiliser** la structure micro en renforçant la cohérence thématique. Le mot « cellule » se connecte plus fortement à « ADN » et « protéine » qu'à « prison » ou « téléphone » dès lors que le macro-nœud « biologie » est établi.

Ce flux descendant ne supprime pas la logique micro puisque si un mot ou une expression n'a **aucun** lien réel avec le sujet macro, le score de synergie S reste faible et la rétroaction top-down ne peut imposer un lien artificiel. Un **équilibre** émerge alors de la confrontation entre le flux ascendant, qui reflète les représentations locales factuelles, et le flux descendant, qui apporte un contexte global, assurant ainsi une **interprétation plus riche** et moins ambiguë.

Mécanisme de contrôle.

Pour éviter des oscillations extrêmes, on peut régler l'amplitude de la rétroaction $\Delta_{\text{down}}^{(\text{macro})}$ par un paramètre $\gamma_{\text{topdown}} > 0$. On définit, par exemple, une fonction $\Phi(\mathcal{T}, i, j)$ qui indique la pertinence du lien (i, j) au vu du sujet \mathcal{T} , puis

$$\Delta_{\text{down}}^{(\text{macro})}(i, j) = \gamma_{\text{topdown}} \Phi(\mathcal{T}, i, j).$$

Si la valeur de Φ est positive, le lien (i, j) est encouragé ; si elle est négative, il est affaibli. Par cette modération, le DSL ne bascule pas trop rapidement.

Analogie cognitive.

Le fait qu'un locuteur ou un lecteur « comprenne » le thème d'un texte et revoie son interprétation des mots locaux renvoie à la **mise en contexte**. Au niveau micro, on s'appuie sur des associations directes (synergie sémantique brute) ; au niveau macro, on oriente ou on “désambiguïse” ces associations pour coller au sujet principal.

Exemple pratique.

Si un article est globalement identifié comme un texte sur l'« IA » (intelligence artificielle), des mots comme « réseau », « neurone » ou « entraînement » seront alignés sur le champ “machine learning” plutôt que sur des sens moins pertinents (réseau social, neurone biologique, entraînement sportif). Les liens $\{\omega_{\text{neurone}, \text{entraînement}}\}$ se verront renforcés sous l'impulsion top-down, consolidant la cohérence IA.

6.6.3. Robotique Synergique : Intégration Sensorimotrice

Dans le domaine de la **robotique**, les principes d'**auto-organisation** et de **synergie** (au cœur du **DSL**) s'appliquent tout particulièrement à la gestion **multi-niveau** des informations sensorielles et des actions motrices. Le robot, équipé de multiples **capteurs** (caméras, gyroscopes, LIDAR, capteurs de force, etc.), doit non seulement **fusionner** ces données pour percevoir son environnement, mais également **coordonner** diverses actions (rouler, saisir, se déplacer en essaim avec d'autres robots...) pour mener à bien un **comportement global** cohérent.

6.6.3.1. Micro-Niveau : Capteurs Basiques (Caméras, Gyros...), Macro : Comportement Global (Navigation, Prise d'Objets...)

Les systèmes robotiques s'appuient sur une **pluralité** de capteurs incluant caméras, gyroscopes, LIDAR et capteurs de force. Pour exécuter des **tâches** complexes comme la navigation, l'interaction ou la manipulation d'objets, il est nécessaire d'intégrer ces flux sensoriels de manière cohérente. Dans une **approche** de **Deep Synergy Learning (DSL)** multi-niveau, le niveau **micro** gère les synergies entre capteurs individuels, tandis que le niveau **macro** permet d'émerger des comportements globaux. Grâce à la mise à jour adaptative des pondérations $\omega_{i,j}$, le DSL forme

progressivement des **clusters** sensoriels cohérents qui, par agrégation, conduisent à des modules comportementaux plus vastes.

A. Niveau Micro : Le Monde des Capteurs

Au niveau le plus bas (ou $\ell = 0$), on considère chaque capteur \mathcal{C}_i comme un **nœud** dans un **SCN**. Il produit un flux de mesures (images, angles, distances) associé à une **entité** \mathcal{E}_i . On relie les paires $\{\mathcal{E}_i, \mathcal{E}_j\}$ par des pondérations $\omega_{i,j}$, qui traduisent la **synergie** ou la **corrélation** entre deux flux sensoriels. Ce mécanisme s’inspire de la mise à jour DSL :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Si deux capteurs (par exemple, une caméra frontale et un LIDAR) “voient” la même forme d’obstacle, leur **similarité** $S(\mathcal{E}_i, \mathcal{E}_j)$ sera forte, entraînant une hausse de $\omega_{i,j}$. Inversement, des capteurs faiblement corrélés (vision frontale et capteur de force sur la pince) n’auront pas de liaison importante. La **dynamique** DSL, en renforçant ou en diminuant les pondérations, fait émerger naturellement des **clusters** de capteurs coopératifs (ex. cluster “détection d’obstacle” regroupant les capteurs directionnels, cluster “calibration d’équilibre” regroupant gyroscopes + accéléromètres).

La **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ peut être exprimée de différentes manières selon le contexte. Elle peut correspondre à un indicateur de **corrélation statistique** lorsque deux capteurs produisent des mesures parallèles, ou à un **indice de concordance** si la vision confirme les distances mesurées par le LIDAR. Le **DSL** exploite ces indices pour structurer, au niveau **micro**, des sous-ensembles de capteurs spécialisés qui interagissent de manière cohérente.

B. Niveau Macro : Le Comportement Global

Au palier supérieur, on **agrège** (via une fonction Ψ) les liaisons $\omega_{i,j}$ internes à chaque cluster sensoriel afin d’obtenir des super-nœuds décrivant des **modules** plus larges. Le robot peut disposer d’un super-nœud “évitement d’obstacle” englobant les capteurs frontaux cohérents, d’un super-nœud “navigation globale” englobant GPS + boussole + gyroscopes, etc. Chacun de ces super-nœuds fournit une **fonctionnalité** ou un **sous-comportement**.

La **tâche** globale du robot, comme se déplacer d’un point A à un point B en évitant les obstacles, repose sur une **composition** de super-nœuds sensoriels et d’algorithmes de contrôle. Le **DSL** établit les connexions entre ces super-nœuds et définit un **module** de navigation intégrant la synergie entre vision, LIDAR et odométrie. Pour la **manipulation d’objets**, un autre macro-module associe les caméras de détection d’objets et les capteurs de force de la pince. Sur le plan **formel**, ces super-nœuds sont représentés par des **nœuds** $\mathcal{N}_\alpha^{(k)}$ au palier k , et un **macro-nœud** fusionnant ces sous-comportements apparaît au niveau supérieur.

C. Exemples Concrets

Robot mobile.

Un robot à roue peut comporter :

- Un cluster sensoriel “orientation” (gyroscope + accéléromètre) pour stabiliser la trajectoire,

- Un cluster “vision frontale + LIDAR” pour évaluer les obstacles,
- Un cluster “GPS + boussole” pour la localisation générale.

Ces ensembles, identifiés localement par la logique DSL, se combinent au niveau macro pour orchestrer la **navigation** (choisir la route, freiner si un obstacle est trop proche, etc.).

Bras manipulateur.

Dans une tâche de **saisie d’objets**, on identifie un cluster “capteur force + caméra pince” pour le contrôle de la préhension, éventuellement un cluster “vision externe + localisation d’objet” pour pointer l’objet, et un cluster “contrôle bras + base” assurant la posture globale. La hiérarchie de super-nœuds (micro→macro) donne une vue modulaire de la robotique, où chaque sous-module sensoriel s’intègre dans un comportement de plus haut niveau (saisir, déplacer, poser).

D. Avantages Mathématiques et Opérationnels

Plutôt que de traiter chaque capteur individuellement dans un *unique* bloc de décision central, le DSL multi-niveau crée des **clusters** sensoriels naturellement cohérents. D’un point de vue mathématique, on aboutit à un **réseau** $\{\omega_{i,j}\}$ plus épars, où seuls les liens justifiés par la **synergie** subsistent. Cela allège la complexité, en substituant un trop-plein de connexions par une structure plus sélective.

Si un capteur se dégrade (ex. caméra obstruée), son lien ω avec les autres capteurs chute (faible S), incitant le système à s’appuyer sur d’autres sources. Au niveau macro, le **comportement global** (ex. “éviter d’obstacles”) s’ajuste en s’alimentant davantage d’un LIDAR intact ou d’une caméra latérale encore fiable.

Les clusters sensoriels évoluent **localement** en comparant les mesures de chaque capteur à celles d’un sous-ensemble de capteurs apparentés. L’agrégation en super-nœuds de plus haut niveau s’effectue de manière asynchrone et incrémentale, ce qui favorise la **réactivité**. Il n’est pas nécessaire de réapprendre entièrement ou de centraliser les données, un atout essentiel pour un robot en mouvement.

6.6.3.2. Logiciel Multi-Threads où le DSL Gère la Synergie Localement, puis un “Super-Nœud” Pilote l’Action Globale

Dans la robotique à grande échelle, l’**architecture logicielle** repose souvent sur un **multi-threading** ou un **multi-processus léger** pour traiter en parallèle les données issues des capteurs tels que les caméras, les gyroscopes et le LIDAR. Chaque fil d’exécution gère un sous-ensemble de données ou une tâche spécifique, comme le traitement d’images ou l’acquisition lidar, tandis qu’un **module central** assure la coordination du **comportement global** du robot, notamment pour la navigation, la saisie d’objets ou la collaboration avec d’autres agents.

Dans le **Deep Synergy Learning (DSL)**, cette répartition logicielle est exploitée de manière à ce que chaque thread applique localement la **règle** de mise à jour des pondérations $\omega_{i,j}$, tandis que le module central agrège ces informations et guide la prise de décision. Cette approche permet une

auto-organisation distribuée au niveau des capteurs tout en maintenant une **coordination hiérarchique** efficace pour l'action finale du robot.

A. Répartition Multi-Threads pour la Gestion Locale de la Synergie

Au **niveau micro**, on considère qu'un ensemble de capteurs $\{\mathcal{E}_i\}$ sont distribués dans différents fils d'exécution (threads). Chaque thread reçoit les flux de quelques capteurs (caméra frontale, caméra latérale, LIDAR, etc.) et traite localement la **synergie** $\omega_{i,j}$. Par exemple, une **mise à jour** DSL de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

s'exécute au sein d'un thread si \mathcal{E}_i et \mathcal{E}_j relèvent d'un même groupe de capteurs ou d'un domaine d'opération connexe. L'information $S(\mathcal{E}_i, \mathcal{E}_j)$ peut être un indice de **similarité** (corrélation d'images, cohérence de mesures lidar) ou de **complémentarité** (caméra + infrarouge).

Dans un système multi-threads, chaque fil d'exécution fonctionne à sa propre fréquence et met à jour certaines parties de la matrice ω . Pour garantir la **cohérence**, on utilise des verrous mutex ou des algorithmes lock-free afin d'éviter des modifications incohérentes de $\omega_{i,j}$ par plusieurs processus simultanés. Chaque thread reçoit périodiquement les états des capteurs et calcule la variation $\Delta\omega$ pour les liens concernés avant de valider ces changements. Cette approche permet de répartir la mise à jour du **Deep Synergy Learning (DSL)** et favorise une auto-organisation progressive, où les clusters sensoriels émergent naturellement par renforcement ou décroissance des liaisons en temps réel.

Mathématiquement, la matrice ω est divisée en plusieurs **blocs**, chacun étant géré par un thread distinct. Chaque bloc regroupe les liens (i,j) entre capteurs dont la synergie doit être évaluée ensemble. Par exemple, la caméra frontale et le LIDAR avant peuvent être intégrés dans un **bloc** dédié à la détection d'obstacles, tandis qu'un autre bloc regroupe les capteurs inertiels comme les gyroscopes et accéléromètres pour la stabilisation. Les **clusters** de capteurs émergent naturellement lorsque les pondérations $\omega_{i,j}$ se renforcent, favorisant une coopération locale entre entités fortement corrélées.

B. Super-Nœud pour le Pilotage Global

Une fois les synergies locales établies au niveau micro, le robot doit agir en fonction de la tâche à accomplir, qu'il s'agisse de se déplacer, de saisir un objet ou d'éviter un obstacle. Dans un **DSL** multi-niveau, un **super-nœud** regroupe les **clusters sensoriels** issus de l'agrégation pour fournir une vision globale. Ce super-nœud intègre les informations des différents ensembles, comme le cluster "vision-frontale-lidar" ou "accéléromètre-gyroscope", afin de coordonner la prise de décision. Si la synergie indiquant un danger est forte, il enclenche la commande de freinage. Si la synergie dans le cluster "saisie d'objet" domine, il oriente le bras robotique vers l'objectif.

Périodiquement, chaque thread local envoie au super-nœud un **résumé** des synergies calculées, sous forme de pondérations $\omega_{\alpha,\beta}$ entre segments sensoriels ou d'un **score** de fiabilité. Le super-nœud agrège ces données à l'aide de Ψ et ajuste la structure globale en fonction de l'hétérogénéité détectée (voir 6.5.2). Cela se formalise par :

$$\omega_{\alpha,\beta}^{(\text{macro})} = \Psi\left(\{\omega_{i,j}^{(\text{local})} : i \in \alpha, j \in \beta\}\right),$$

puis le super-nœud évalue l'opportunité de moduler (inhiber ou renforcer) des liens via un flux descendant (chap. 6.4).

Dans la pratique, un **robot mobile** gère la synergie locale d'“obstacle detection” par un thread “lidar+caméra frontale” ; un autre thread combine inertiels + GPS pour la navigation large. Le super-nœud global reçoit leurs états, voit s'il doit prioriser l'évitement d'obstacles ou continuer la route, et envoie des **commandes** au système moteur. En cas de conflit (capteur en panne), il peut forcer une reconfiguration en poussant un *terme descendant* défavorable aux liaisons du capteur défectueux.

C. Avantages en Robotique Synergique

Une approche **multi-threads** répartit la gestion de la synergie $\omega_{i,j}$; chaque fil d'exécution prend en charge un sous-ensemble de capteurs, allégeant la complexité globale. Le super-nœud ne fait qu'**agréger** et **décider**, évitant de gérer $O(n^2)$ liaisons en un seul point central.

Un événement local (caméra détectant un obstacle) modifie $\omega_{\text{cam,lidar}}$ presque instantanément. Le super-nœud, en lisant ces informations, enclenche l'action appropriée. Cette asynchronie évite d'attendre un cycle complet d'un pipeline monolithique.

La défaillance d'un thread, comme une caméra latérale obstruée, ne bloque pas l'ensemble du système. Les autres threads poursuivent leur mise à jour DSL tandis que le super-nœud diminue la confiance ou la synergie associée au module défaillant. Si nécessaire, un nouveau capteur peut être ajouté sans reprogrammer toute la structure. Ses liens ω émergent progressivement dans la dynamique DSL et le super-nœud l'intègre dès qu'un cluster cohérent se forme autour de lui.

6.6.3.3. Notion Fractale si on Observe la Même Logique de Commande à Divers Sous-Systèmes Moteurs

L'architecture **DSL** (Deep Synergy Learning) appliquée à la robotique ne se borne pas au traitement des capteurs ; elle peut également s'étendre aux **ensembles d'actionneurs** et de commandes motrices. À l'échelle la plus locale, des groupements de joints ou de moteurs se coordonnent pour des mouvements élémentaires (par exemple, les doigts d'une pince ou les roues antérieures d'un véhicule). À un niveau intermédiaire, plusieurs de ces sous-ensembles coopèrent dans un module plus vaste (un bras complet, un chariot de locomotion), tandis qu'au niveau le plus global (macro-nœud), le robot exécute un **comportement** ou une **tâche** englobante. Lorsque la même logique DSL (mise à jour des pondérations ω , agrégation, division, rétroaction descendante, etc.) est reproduite à chaque échelle d'organisation, il se crée une forme de **structure fractale**, car la même règle d'auto-organisation se réplique du plus petit sous-système moteur à l'unité robotique intégrale.

Il est d'abord essentiel de comprendre le rôle des **sous-systèmes moteurs** à l'échelle locale. Chaque actionneur ou groupe de joints est traité comme une entité \mathcal{E}_i . Les pondérations $\omega_{i,j}$ traduisent la synergie ou la synchronisation entre deux actionneurs i et j . La règle DSL

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

peut conduire à l'émergence de **clusters** moteurs dès lors que plusieurs joints se trouvent souvent co-activés ou dont les positions se coordonnent. Les doigts d'une pince se coordonnent dans la "manipulation fine", les roues droites d'un châssis forment un mini-groupe fonctionnel pour déplacer le côté droit, etc. La mise à jour adaptative des pondérations ω autorise ces regroupements à évoluer si le robot entreprend des tâches différentes (de la locomotion à la saisie d'objets), ce qui modifie les fréquences et les patrons de co-activation.

Une fois ces **clusters** moteurs identifiés au niveau micro, un **palier intermédiaire** agrège leurs synergies et forme des super-nœuds correspondant à des modules plus vastes. Un bras gauche ou une base roulante où se concentrent les synergies internes. Le même mécanisme que pour les capteurs s'applique ici, simplement transposé aux actionneurs. Une fonction Ψ agrège les pondérations $\omega_{i,j}$ au sein d'un sous-système donné et génère les pondérations $\omega_{\alpha,\beta}$ entre super-nœuds α et β . La dynamique DSL suit alors la même logique. Si un sous-système devient trop hétérogène, par exemple un regroupement d'actionneurs qui n'ont pas un besoin réel de coopération, on le scinde. Si au contraire deux sous-systèmes collaborent fréquemment, leurs liens ω se renforcent et favorisent une fusion.

Le **palier macro** couronne le processus, lorsque le robot exécute un **comportement intégral**. Le super-nœud global, englobant l'ensemble des sous-nœuds moteurs, établit quelles combinaisons d'ensembles d'actionneurs déclencher pour chaque tâche. Lorsque le robot alterne entre locomotion et manipulation, il active ou module les liens $\omega_{\alpha,\beta}$ entre la base roulante et le bras, reflétant la co-activation ou la "pertinence conjointe" de ces modules. La même dynamique DSL assure qu'un comportement très souvent sollicité se traduise par des pondérations ω plus élevées entre les sous-systèmes pertinents, tandis qu'un comportement moins usuel demeure moins intégré ou subit la décroissance paramétrée par τ .

La **notion fractale** émerge du fait que la *même* logique se retrouve à chaque échelle. Le niveau micro, qui comprend quelques joints, applique la règle DSL pour gérer les liens $\omega_{i,j}$. Le niveau intermédiaire, qui correspond à un sous-système comme un bras complet, applique la même équation pour agréger ou diviser les sous-modules. Le niveau macro, qui englobe l'ensemble du robot, applique encore cette formule pour orchestrer les super-nœuds. En pratique, on observe des distributions de pondérations ω qui conservent une **auto-similarité** lorsqu'on change d'échelle. On constate également que la fonction de clusterisation suit une loi de puissance concernant la taille des modules moteurs. Cela met en évidence une **structure fractale**, car la *même forme* d'auto-organisation DSL se reproduit du plus petit sous-système jusqu'à la commande intégrale de l'agent.

Un bénéfice évident réside dans la **flexibilité**. À l'échelle micro, un cluster local peut gérer la coordination fine de quelques doigts. Si le robot doit synchroniser deux bras pour soulever un objet, un palier intermédiaire détecte cette synergie par la co-activation récurrente et forme un super-nœud "opération bimanipulative". Au niveau macro, ce super-nœud peut se connecter à la base roulante si le robot doit transporter l'objet. Cette application de la *même règle* DSL sur plusieurs niveaux permet à la structure de se reconfigurer si un module d'actionneur n'a plus besoin de coopérer. D'un point de vue mathématique, cela revient à un graphe ω multi-niveau où les principes de renforcement et d'inhibition font émerger un ensemble de super-nœuds cohérents.

6.6.4. Agents Conversationnels Riches et Contextuels

Le domaine des **agents conversationnels** (chatbots, systèmes de dialogue) repose sur une organisation **multi-échelle**. Une conversation ne se réduit pas à une simple suite de répliques, mais se structure en **segments**, **sous-thèmes** et **intentions** globales. Le **DSL** (Deep Synergy Learning), avec son mécanisme d'**auto-organisation** basé sur des pondérations adaptatives ω et une agrégation ou division multi-niveau, permet une structuration dynamique du dialogue.

Les **segments** de conversation au niveau micro se regroupent selon leur cohérence et leur synergie sémantique. À un niveau plus global, les **intentions** ou **topics** émergent de l'agrégation de ces segments. Le réseau conversationnel s'adapte ainsi en temps réel, assurant une gestion dynamique du contexte au fil du dialogue.

Dans ce qui suit (6.6.4.1), nous détaillons comment, au *niveau micro*, on manipule des **segments** de conversation, tandis qu'au *niveau macro*, on repère des **intentions** ou **sujets** d'échange.

6.6.4.1. Micro : Segments de Conversation, Macro : Intentions ou Topics

Dans l'analyse conversationnelle, la discussion s'organise naturellement en plusieurs paliers. Au niveau **micro**, les segments ou “turns” correspondent à une question, une réponse ou un bloc de discours. Au niveau **macro**, ces segments s'agrègent pour former des intentions ou “topics” plus larges, comme un échange autour de la “météo” ou des “horaires de train”.

Le **Deep Synergy Learning (DSL)** permet de faire émerger cette organisation multi-niveau de manière automatique. Le niveau micro regroupe les segments proches en fonction de leur synergie, tandis que le niveau macro identifie les intentions sous-jacentes, structurant ainsi dynamiquement le dialogue.

Il est d'abord utile de décrire comment se traite le **niveau micro** des segments de conversation. Chaque segment \mathcal{E}_i représente un bloc de discours (par exemple, une phrase d'un utilisateur, une réponse du chatbot) que l'on encode sous la forme d'un vecteur sémantique \mathbf{v}_i (embedding). On introduit une **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ mesurant la similarité sémantique ou contextuelle entre deux segments. Une fonction usuelle est la similarité cosinus :

$$S(\mathcal{E}_i, \mathcal{E}_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}.$$

Dans le DSL, on maintient pour chaque couple (i, j) une pondération $\omega_{i,j}$, mise à jour à l'aide de l'équation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est un taux d'apprentissage et $\tau > 0$ un facteur de décroissance. Lorsque deux segments de conversation \mathcal{E}_i et \mathcal{E}_j abordent un même sujet, leurs embeddings sont proches, ce qui donne une synergie $S(i, j)$ élevée et fait croître $\omega_{i,j}$. Un ensemble de segments fortement interconnectés peut

dès lors constituer un **cluster** local, reflétant la poursuite d'un sujet précis (questions et réponses cohérentes, réactions enchaînées, etc.).

Une fois qu'au palier micro on observe des groupes de segments fortement reliés, on peut passer au **niveau macro**. L'agrégation de ces segments permet de définir des super-nœuds, c'est-à-dire des **intentions** ou **topics** plus vastes. On applique généralement une fonction Ψ pour agréger les pondérations micro en $\omega_{\alpha,\beta}^{(\text{macro})}$, selon

$$\omega_{\alpha,\beta}^{(\text{macro})} = \Psi(\{\omega_{i,j} \mid i \in \alpha, j \in \beta\}),$$

où α, β sont des ensembles de segments (clusters) déjà identifiés. On peut choisir Ψ comme une moyenne, une somme, ou un maximum. On obtient alors un graphe macro où chaque super-nœud représente un **topic** (par exemple “discussion sur la météo”, “demandes d'information sur les horaires de train”), et où les arcs $\omega_{\alpha,\beta}^{(\text{macro})}$ traduisent la proximité ou la cohérence entre ces topics (si la conversation a fusionné deux thématiques).

Ce fonctionnement peut être comparé à celui d'un chatbot classique où l'identification des intentions repose sur un classifieur supervisé qui assigne chaque phrase à un label prédéfini comme une salutation, une question sur la météo ou une demande d'horaires.

Dans un **DSL**, l'intention n'est pas définie à l'avance mais **émerge** de la synergie entre segments. Il n'est donc pas nécessaire de spécifier une liste fermée d'intentions. Si un nouveau sujet apparaît, le système l'identifie spontanément en regroupant les segments corrélés dans un nouveau cluster. Cette plasticité est particulièrement précieuse pour gérer plusieurs sujets en parallèle sans risquer d'entremêler leurs segments de manière incohérente.

On peut illustrer ce mécanisme par un petit exemple numérique. Considérons des segments $\{E_1, E_2, E_3, \dots\}$. Les similarités $S(E_1, E_2) = 0.8$, $S(E_2, E_3) = 0.7$, etc., alimentent la mise à jour DSL. Les pondérations $\omega_{1,2}$ et $\omega_{2,3}$ croissent vers $0.8/\tau$ et $0.7/\tau$. En conséquence, on regroupe $\{E_1, E_2, E_3\}$ en un cluster (la même discussion autour d'un thème local). De leur côté, $\{E_4, E_5\}$ peuvent constituer un autre cluster s'ils abordent un sujet distinct (par exemple, un bavardage sur la cuisine). On obtient alors deux **topics**.

Sur le plan pratique, ce schéma confère des **gains** en analyse conversationnelle. Les segments qu'un agent conversationnel reçoit sont organisés en clusters, ce qui permet à l'agent de “savoir” quel sous-ensemble discute d'un sujet donné. Lorsqu'un nouveau segment E_{new} arrive, il calcule la synergie $S(E_{\text{new}}, E_i)$ pour les segments existants $\{E_i\}$. Si cette synergie est forte avec ceux du topic “météo”, alors $\omega_{\text{new},i}$ se renforce, et E_{new} rejoint ce topic, assurant une **contextualisation** immédiate. S'il n'existe pas de topic correspondant (faible synergie avec tout l'existant), le DSL alloue un **nouveau** cluster pour cette thématique émergente.

Le **DSL** multi-niveau permet la **coexistence** de plusieurs topics en parallèle, comme un fil sur la météo, un autre sur la cuisine et éventuellement un sur l'économie. Contrairement à un classifieur d'intentions qui impose une assignation unique à chaque phrase, le système ne limite pas la conversation à un mode monothématique.

Les topics émergent sous forme de **super-nœuds** issus de l'agrégation des pondérations $\omega_{i,j}$ entre segments. Ils peuvent entretenir des connexions plus ou moins fortes entre eux, ce qui reflète la dynamique naturelle des transitions thématiques dans une conversation fluide et évolutive.

6.6.4.2. Multi-Échelle : La Conversation se Hiérarchise en Sous-Thèmes, Thèmes plus Généraux

Lors d'un échange conversationnel, qu'il s'agisse d'un dialogue humain-humain ou d'un système de chatbot interagissant avec plusieurs utilisateurs, on observe fréquemment plusieurs **sous-thèmes** parallèles qui s'insèrent dans un **thème** plus global ou un **contexte** englobant. Le **Deep Synergy Learning (DSL)**, par sa faculté de créer des clusters localement (niveau micro) puis de les **agréger** ou de les **diviser** au fil du temps (niveau macro), permet une **hiérarchisation** organique de la conversation. Les paragraphes qui suivent décrivent comment, au sein d'un DSL, des segments localement liés par la synergie forment des sous-thèmes, et comment ces sous-thèmes à leur tour se rejoignent (ou se distinguent) pour générer des thèmes d'échelle supérieure, parfois avec un caractère **fractal** lorsque la même structure se répète à divers niveaux.

A. Du Sous-Thème Local au Thème Général : Construction Progressive

Tout d'abord, on considère la conversation découpée en **segments** $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. Chaque segment (phrase, réplique ou bloc de texte) est relié aux autres par des pondérations $\omega_{i,j}$. Dans la **logique** DSL, la mise à jour itérative

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

se fonde sur une **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ (similarité sémantique, continuité thématique, etc.). Au fil des itérations, on voit émerger des **clusters** regroupant des segments qui traitent le même micro-sujet, comme “recherche de billets de train” ou “discussion anecdotique sur la météo”. Chacun de ces clusters peut être assimilé à un **sous-thème** local.

Une fois ces sous-thèmes définis, le système agrège les pondérations internes pour former des **super-nœuds** $\{\mathcal{N}_\alpha\}$. Si plusieurs sous-thèmes se trouvent eux-mêmes synergiques (par exemple, tous traitent de voyages, l'un portant sur le train, l'autre sur l'avion, un troisième sur les hôtels), ils se **fusionnent** en un **thème** plus large. Pour ce faire, on utilise une fonction d'agrégation Ψ :

$$\omega_{\alpha,\beta}^{(\text{macro})} = \Psi(\{\omega_{i,j}\}_{i \in \alpha, j \in \beta}),$$

puis on applique la même règle DSL ou un critère \mathcal{H} (hétérogénéité) pour décider de la **fusion** ou de la **division** de thèmes. Ainsi, un grand thème “voyage” peut s'auto-organiser en sous-thèmes “train”, “avion” et “réservation d'hôtel”.

B. Logique Fractale dans la Hiérarchisation de la Conversation

Il est fréquent qu'un même **concept**, comme les déplacements, apparaisse à plusieurs niveaux. D'abord, il est mentionné localement dans quelques segments, par exemple pour discuter des détails de billets. Ensuite, il se regroupe en un sous-thème structurant ces échanges. Enfin, il réapparaît à un niveau plus global lorsque la conversation aborde un projet de voyage plus large.

Cette récurrence multi-niveau illustre une **organisation fractale** décrite au chapitre 6.3. La *même structure* se reproduit à différentes échelles, garantissant une continuité et une cohérence dans l'évolution du dialogue.

Dans un DSL, la *même* règle de mise à jour $\omega(t + 1) = \omega(t) + \dots$ se rejoue entre :

- (i) les entités segments (niveau micro),
- (ii) les super-nœuds sous-thèmes (niveau intermédiaire),
- (iii) les super-nœuds plus englobants (niveau macro).

Le schéma d'agrégation–division (chap. 6.5) se reproduit donc à plusieurs paliers, générant une structure de clusters (ou super-nœuds) “auto-similaire” d'un niveau à l'autre. On peut donc, en certain sens, qualifier de “fractale” cette hiérarchie conversationnelle, puisqu'elle reproduit un *motif* identique d'**auto-organisation**.

C. Avantages Concrets pour l'Agent Conversationnel

Quand un **agent** conversationnel identifie un sous-thème α , comme “le train de demain matin”, il dispose de la liste des segments \mathcal{E}_i qui y sont rattachés. Si ce sous-thème se connecte à un autre β , comme “réservation d'hôtel”, sous un thème plus large “voyage”, il ajuste son contexte en conséquence.

L'agent comprend alors que tous ces segments relèvent d'un même projet de voyage, ce qui renforce la **cohérence** des réponses. Cette structuration permet aussi de naviguer entre des détails précis, comme le “prix du billet de train”, et des considérations plus globales, comme un “voyage en Europe la semaine prochaine”.

Dans une conversation chaotique où plusieurs sujets se déploient, le DSL autorise la **coexistence** de clusters (sous-thèmes distincts), lesquels s'agrègent ou se divisent au fil des nouveaux segments. On n'a pas besoin d'une structure préétablie. Chaque **nouveau** segment \mathcal{E}_{new} calcule sa similarité $S(\mathcal{E}_{\text{new}}, \mathcal{E}_i)$ avec les segments existants, renforce ou diminue $\omega_{\text{new},i}$, et finit par rallier un sous-thème ou en créer un nouveau. L'agent peut traiter simultanément plusieurs fils de discussion.

Si un grand sous-thème devient trop large, la procédure DSL décrite au **chapitre 6.5.2** peut déclencher une scission top-down en divisant un “macro-sous-thème” en deux entités plus précises. À l'inverse, si deux sous-thèmes apparaissent trop similaires, ils peuvent fusionner pour former une unité plus cohérente.

Ces ajustements s'opèrent directement dans la matrice ω , sans nécessiter un recalcul global. Cette **adaptation dynamique** permet ainsi de gérer efficacement les dialogues longs ou sujets à des changements fréquents.

6.6.4.3. Avantage : Le DSL Repère des “Patrons Fractals” de Dialogue ou d'Échanges

Dans un **échange** ou un **dialogue**, il n'est pas rare d'observer la récurrence d'un même *modèle d'interaction* à différents niveaux de granularité. Par exemple, on peut avoir un motif local (question–réponse) reproduit à l'échelle plus large (ensemble de sous-thèmes, chacun contenant

des séquences de questions–réponses) et enfin à l’échelle globale (plusieurs blocs de discussion qui suivent la même logique). Lorsqu’un **Deep Synergy Learning (DSL)** multi-niveau est appliqué à l’analyse et la structuration du dialogue, cette **auto-similarité** récurrente s’assimile à une **fractalité**. Les développements qui suivent décrivent comment le DSL, par sa démarche d’agrégation et de division hiérarchique, parvient à **repérer** des “patrons fractals” dans l’agencement des segments, et en quoi cela **profite** à l’agent conversationnel.

A. Patrons Fractals : Définitions et Parallèles

Une structure fractale repose sur l’**auto-similarité**, où une même organisation se répète à différentes échelles. Dans un dialogue, un motif local comme “question–réponse–clarification” peut se retrouver à un niveau plus large sous forme de plusieurs échanges construisant une même logique, puis réapparaître au niveau d’un macro-sujet où une suite de sous-thèmes suit ce schéma.

Le **DSL** capture cette **auto-similarité** en appliquant la même règle de mise à jour des liaisons ω et d’agrégation à chaque palier, permettant ainsi une structuration cohérente du dialogue sur plusieurs niveaux.

Exemple de motif fractal.

Un motif “A pose une question, B répond, A clarifie” peut se répéter dans une conversation courte (quelques segments) ou plus longue (série de sous-thèmes). Chaque *patern* (A, B, C) se traduit par des pondérations $\omega_{i,j}$ élevées lorsque le segment “B” se lie fortement à “A” (réponse cohérente) et “C” se lie à “B” (clarification). Au niveau macro, on retrouve un agrégat de telles triades, mettant en évidence la même structure conversationnelle répétée.

B. Mécanisme DSL de Détection Fractale

Hiérarchie ascendante (agrégation).

Dans le DSL, on commence par **analyser** les segments individuels $\{\mathcal{E}_i\}$. Si deux segments $\mathcal{E}_i, \mathcal{E}_j$ se montrent récurrents dans la même logique (ex. question–réponse), leur pondération $\omega_{i,j}$ grimpe. Si, en outre, un troisième segment \mathcal{E}_k s’active avec $\{i, j\}$ (par ex. clarification), on peut aboutir à un cluster $\{\mathcal{E}_i, \mathcal{E}_j, \mathcal{E}_k\}$ correspondant à un schéma local. Au palier supérieur, si plusieurs de ces triades suivent une *même* structure, le DSL agrège les clusters en un **super-nœud** “pattern conversationnel”.

Pour formaliser cela, on définit une *énergie* ou un *score* inter-segments. Si un motif $\mathcal{M} = \{\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c\}$ reproduit la séquence question–réponse–clarif, alors la somme

$$\Omega(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M} \times \mathcal{M}} \omega_{x,y}$$

peut devenir élevée (les segments $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c$ se relient fortement). Lorsque cette somme $\Omega(\mathcal{M})$ dépasse un certain θ_{motif} , on identifie \mathcal{M} comme un “pattern” stable. Plusieurs motifs analogues $\mathcal{M}', \mathcal{M}''$ peuvent avoir la même structure, révélant la *récurrence* fractale.

Au niveau macro, la même **logique** DSL agit sur ces motifs (clusters) entre eux. Si l’on retrouve \mathcal{M} ou un motif similaire \mathcal{M}' dans un autre sous-thème, on observe une **auto-similarité** dans la distribution des liens. On peut donc parler de fractalité, car l’organisation question–réponse–clarif

se répercute à une échelle plus large, comme un “schéma” s’appliquant à plusieurs groupes de segments.

C. Avantages pour l’Agent Conversationnel

Un utilisateur peut aborder plusieurs sous-thèmes en suivant un même schéma avec une séquence de question, réponse et clarification. Le **DSL**, en reconnaissant ce **pattern**, permet à l’agent de mieux **anticiper** la structure attendue. Il peut par exemple plus rapidement fournir des réponses plus ciblées si la séquence se répète.

Le motif fractal n’est pas rigide. On repère la forme d’échange (ex. question–réponse–clarif) peu importe le *contenant* lexical. Le DSL, via la similarité $\omega_{i,j}$, autorise une grande variété de formulations sans casser le *pattern*. Ainsi, l’agent conversationnel peut exploiter ce schéma structurel pour organiser ses réactions, même si le contenu des segments varie beaucoup.

Dans des dialogues massifs impliquant plusieurs participants et de multiples bifurcations, repérer des **motifs fractals** permet une **compression efficace**. Plutôt que de traiter un graphe monolithique de segments, on exploite la répétition de certains motifs locaux sous différentes occurrences. En regroupant ces occurrences, on simplifie la navigation et l’organisation de la structure du dialogue.

6.6.5. Applications dans la Simulation et la Prédiction d’Événements

Les principes **multi-échelle** du DSL (Deep Synergy Learning) trouvent un écho tout particulier dans la **simulation** et la **prédiction** d’événements. En effet, de nombreux phénomènes (économiques, climatiques, épidémiologiques, boursiers, etc.) s’expriment à la fois sous forme d’**événements ponctuels** (micro) et de **tendances** (macro). Le **DSL** permet alors d’**organiser** ces événements de manière adaptative, dévoilant des **mégapatterns** ou tendances globales tout en gardant la granularité des points individuels.

6.6.5.1. Micro : Événements Ponctuels, Macro : Tendances ou Mégapatterns (ex. Bourse, Climat)

Dans de nombreux flux temporels, comme les données boursières, les variables climatiques ou les événements sporadiques, chaque point observé correspond à un **événement** local.

Pour en extraire des **tendances** plus globales, une analyse multi-niveau s’avère nécessaire. Un **niveau micro** identifie les interactions et similarités entre événements ponctuels, tandis qu’un **niveau macro** agrège ces événements pour révéler des **mégapatterns** plus larges, tels qu’un cycle boursier haussier ou un changement climatique global. Le **Deep Synergy Learning (DSL)** permet de structurer et d’adapter ces niveaux, tout en intégrant l’apparition progressive de nouveaux événements au fil du temps.

A. Niveau Micro : Événements Ponctuels

Il est courant, dans un flux temporel (bourse, climat, logs...), de définir chaque **événement** \mathcal{E}_i comme une entité $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. Un événement peut consister, par exemple, en une transaction (bourse) ou une perturbation locale (climat).

Pour deux événements \mathcal{E}_i et \mathcal{E}_j , on introduit une **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ reflétant leur **corrélation** ou leur **similarité** dans l'espace de variables considérées. On peut écrire :

$$S(\mathcal{E}_i, \mathcal{E}_j) = \rho(\mathbf{x}_i, \mathbf{x}_j),$$

où ρ est un coefficient de corrélation (Pearson, Spearman, etc.) ou tout autre indicateur de concomitance. Cette approche s'adapte aisément aux données boursières (corrélation entre deux transactions) ou aux mesures météorologiques (similitude spatio-temporelle de deux perturbations).

Dans la logique DSL, on entretient pour toute paire (i, j) une pondération $\omega_{i,j}$ modifiée selon :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Lorsque deux événements s'avèrent fortement corrélés ou se produisent de manière répétée à des moments similaires, leur synergie S est élevée et $\omega_{i,j}$ augmente. Cela aboutit à la **clusterisation** d'événements ponctuels fortement liés.

B. Niveau Macro : Mégapatterns et Tendances

Une fois que, au niveau micro, les événements $\{\mathcal{E}_i\}$ s'organisent en **clusters** ou sous-ensembles homogènes (transactions identiques, perturbations climatiques similaires), on peut agréger ces ensembles pour former des **super-nœuds** $\{\mathcal{N}_\alpha\}$. Dans le cadre du DSL, on emploie une fonction Ψ :

$$\omega_{\alpha,\beta}^{(\text{macro})} = \Psi(\{\omega_{i,j}^{(\text{micro})}\} : i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta).$$

On obtient alors un **graphe** macro où chaque super-nœud \mathcal{N}_α correspond à un **ensemble** d'événements corrélés (e.g. transactions à un moment précis ou perturbations climatiques simultanées). Ce graphe macro, lui aussi soumis à la dynamique DSL, permet de repérer quand plusieurs super-nœuds $\mathcal{N}_\alpha, \mathcal{N}_\beta$ partagent une synergie élevée, suggérant qu'ils forment un **mégapattern** (ex. phase de marché haussière, vague de chaleur régionale).

Dans ce macro-niveau, un mégapattern correspond à un **regroupement** de super-nœuds $\mathcal{N}_\alpha, \mathcal{N}_\beta, \dots$ qui coopèrent régulièrement, signant une tendance persistante. En bourse, on reconnaîtra un **Bull Market** si un large bloc de transactions entretient des liens ω élevés (forte concordance de prix en hausse). En climat, on rassemblera plusieurs perturbations locales (orages, vents forts) en un **bloc** cohérent, trahissant une dépression qui s'étend sur une vaste zone ou un effet de "vague de chaleur".

C. Exemples Concrets

Bourse et événements.

Dans un flux boursier, chaque **transaction** forme un événement ponctuel. On définit la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ par la ressemblance de leur prix, heure, ou "carnet d'ordre". Les transactions extrêmement proches (mêmes heures, mêmes montants, mêmes actifs financiers) agrègent localement $\omega_{i,j}$ en un **cluster** de micro-événements. Au **niveau macro**, on assemble ces clusters

en **phases** (Bull, Bear, neutral), aboutissant à un mégapattern (par ex., un marché systématiquement acheteur pendant deux semaines).

Climat et perturbations.

On conçoit chaque **orage**, chaque **tempête locale**, ou chaque **front froid** comme un événement \mathcal{E}_i ; la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ tient compte de la distance géographique, de l'intervalle temporel, de la similarité d'intensité. Les orages rapprochés s'agrègent localement. Au niveau macro, des ensembles d'orages répétitifs sur une zone conduisent à un mégapattern "vague de tempêtes" ou "saison cyclonique".

D. Bénéfices Mathématiques et Opérationnels

L'approche DSL se prête particulièrement bien à un flux en **temps réel** ou incrémental. Dès qu'un nouvel événement \mathcal{E}_{new} survient (transaction, perturbation), on l'associe à la matrice ω , on calcule $\omega_{\text{new},j}$ pour les $\{\mathcal{E}_j\}$ déjà existantes, et l'on actualise la **synergie** correspondante. S'il y a suffisamment de liens forts, \mathcal{E}_{new} rejoint un cluster ou, sinon, crée un nouvel ensemble.

Le DSL crée en parallèle un **clustering** de niveau micro (évolution locale) et un de niveau macro (tendances). Cette hiérarchie autorise un compromis entre la **finesse** (événements ponctuels) et la **vision** d'ensemble (cycles, vagues de température). En $O(n)$ ou $O(n^2)$ selon l'algorithme, on peut rafraîchir la structure après chaque événement, voire lot d'événements.

Une fois un mégapattern stabilisé (une synergie $\omega_{\alpha,\beta}^{(\text{macro})}$ élevée entre plusieurs super-nœuds $\{\alpha, \beta\}$), on peut en **déduire** qu'il y a de fortes chances pour que les prochains événements prolongeant cette cohérence se rattachent au même pattern. Sur le plan boursier, on anticipe la poursuite d'une tendance haussière ; sur le plan climatique, on anticipe la prolongation d'une vague de chaleur. Le **DSL** fournit ainsi un mécanisme de repérage de phénomènes étendus sans recourir à une modélisation paramétrique rigide.

6.6.5.2. Éventuels Invariants d'Échelle (Lois de Puissance) dans les Distributions d'Événements

De nombreux phénomènes, qu'ils relèvent de la bourse, de la climatologie, de l'épidémiologie ou de la sismologie, présentent une répartition des **événements** observés qui suit des **lois de puissance** plutôt que des lois gaussiennes. Les transactions financières, les orages ou les éruptions volcaniques illustrent cette dynamique, caractérisée par une structure *scale-free* ou fractale. On y observe une **auto-similarité**, où la même distribution se manifeste à différentes **échelles** d'observation. Le **Deep Synergy Learning (DSL)** est particulièrement adapté pour analyser et exploiter ces invariants d'échelle, en s'appuyant sur une **auto-organisation** multi-niveau des événements.

A. Lois de Puissance et Distributions "Scale-Free"

Une **loi de puissance** (ou loi Pareto) se manifeste lorsque la probabilité $P(X > x)$ d'observer un événement d'**intensité** ou de **taille** supérieure à x se comporte comme $x^{-\alpha}$ pour $\alpha > 0$. Concrètement, cela signifie qu'il existe beaucoup d'événements de taille modeste, mais que des

événements extrêmement grands (krach boursier, cyclone majeur, séisme de haute magnitude) restent non négligeables, beaucoup plus fréquents que ne le prédirait une loi normale.

Dans les **marchés financiers**, la queue de la distribution des variations de prix s’approche souvent d’une loi de puissance, expliquant les “fat tails” (événements extrêmes plus fréquents qu’en gaussien). En **climat**, l’intensité des tempêtes ou des inondations peut s’inscrire dans une dynamique *heavy-tailed*. En épidémiologie, la survenue d’un foyer extrêmement infectieux correspond à un comportement hors de la zone médiane.

On parle de *scale-free* lorsque la distribution reste stable malgré un changement d’unité ou un ajustement d’échelle. Dans un cadre fractal, cette invariance d’échelle traduit une **auto-similarité**, où les motifs observés localement, comme quelques événements isolés, se retrouvent sous forme de structures plus vastes à un niveau global. Le **Deep Synergy Learning (DSL)**, grâce à son mécanisme d’agrégation hiérarchique, met en évidence cette propriété en structurant la distribution des clusters selon des lois d’échelle.

B. DSL et Détection des Invariants d’Échelle

Dans un **DSL**, on définit un **graphe** de pondérations $\omega_{i,j}$ reliant des **événements** $\{\mathcal{E}_i\}$. La mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

permette de renforcer ou d’affaiblir les liens en fonction de la synergie S , qui peut refléter une proximité temporelle, géographique, ou d’intensité. Lorsque le processus est **multi-niveau**, on construit des super-nœuds (macro) par agrégation Ψ . La distribution des degrés (ou des tailles de cluster) peut, dans un phénomène scale-free, présenter une loi de puissance $P(k) \sim k^{-\alpha}$. Le **DSL** ne postule pas a priori de gaussien ou non, mais organise les événements en clusters, possiblement **dominés** par quelques “hubs” et un grand nombre de petits nœuds, reproduisant la signature d’un réseau scale-free.

Dans cette structure, le niveau **micro** gère individuellement chaque événement \mathcal{E}_i . Lorsque plusieurs événements sont fortement liés, ils forment un **cluster** local. Certains **clusters** deviennent très volumineux, englobant plusieurs macro-événements, tandis qu’une multitude de petits clusters demeurent plus isolés. Ce **déséquilibre**, où quelques **hubs massifs** coexistent avec une grande quantité de petits nœuds, indique une possible *power-law*. Au niveau **macro**, la même organisation se répète, donnant naissance à quelques super-nœuds dominants (mégapatterns) et à de nombreux plus petits, conservant ainsi une structure fractale à différentes échelles.

En théorie des réseaux complexes, un graphe **scale-free** est défini par une distribution en loi de puissance de ses degrés $\{k_i\}$. Si on associe chaque événement \mathcal{E}_i à un nœud et on calcule le degré en fonction de $\{\omega_{i,j}\}$ (ou de la somme de ces poids), on peut retrouver un $P(k) \sim k^{-\gamma}$. Le DSL, par son mécanisme de croissance (renforcer les liens les plus utiles, laisser s’éteindre les autres), tend à engendrer ou consolider ce genre de structure, avec quelques nœuds “majoritairement connectés”, correspondant aux événements les plus extrêmes ou les plus récurrents.

C. Conséquences et Intérêts Pratiques

Une loi de puissance implique que des événements extrêmes, bien que rares, conservent une influence significative, contrairement à une distribution gaussienne où la décroissance est plus

rapide. L'**auto-organisation** DSL identifie naturellement des **macro-nœuds** de grande taille, représentant des regroupements d'événements fortement corrélés. Ces structures signalent l'émergence d'un “hub” ou d'un “mégapattern” dominant. Lorsque de nouveaux événements \mathcal{E}_{new} apparaissent, ils ont une forte probabilité de se rattacher à ces hubs si la dynamique sous-jacente se maintient, comme dans un marché boursier haussier ou une période prolongée d'instabilité climatique.

De nombreux modèles statistiques rencontrent des difficultés face aux distributions à queue lourde. Le **DSL** ne contraint pas la distribution à une forme paramétrique spécifique, mais ajuste dynamiquement les pondérations ω selon les structures émergentes. Cela permet de laisser apparaître naturellement une queue lourde avec quelques clusters dominants ou, au contraire, une absence de seuil marqué, sans imposer un modèle gaussien qui pourrait fausser l'interprétation.

L'**invariance d'échelle** peut être mise en évidence en analysant la distribution des tailles de clusters à différents niveaux. Une pente stable sur un graphique **log-log** suggère une fractalité sous-jacente. Le **DSL**, en mettant en lumière les macro-clusters dominants ou en révélant une multitude de petits groupes, confirme cette structure. Il permet ainsi de vérifier si l'on retrouve un schéma de type **hub and spoke** dans le graphe ω , caractéristique des *scale-free networks*.

6.6.5.3. Gains : Meilleure Anticipation, Clusterisation Plus Lisible

Les sections précédentes (6.6.5.1–6.6.5.2) ont abordé l'application du **Deep Synergy Learning (DSL)** à des scénarios de simulation ou de prédiction d'événements (transactions boursières, données climatiques, épidémies, etc.). Dans ce contexte, chaque événement ponctuel forme une entité \mathcal{E}_i au niveau micro, tandis que le **DSL** agrège ces entités en super-nœuds macro pour déceler des “mégapatterns” ou tendances globales. Cette structuration multi-échelle engendre deux avantages concrets : une **meilleure anticipation** d'évolutions (ou crises) et une **clusterisation** beaucoup plus lisible, facilitant l'analyse et la prise de décision.

A. Meilleure Anticipation des Évolutions ou Crises

Renforcement local et indicateurs d'alarme.

Dans la dynamique DSL, on maintient pour chaque paire (i, j) un poids $\omega_{i,j}$. Quand un ensemble d'événements $\{\mathcal{E}_k\}$ se révèle fortement corrélé — par exemple, une série de variations boursières liées ou un ensemble de perturbations climatiques proches — leurs pondérations $\omega_{k,\ell}$ s'élèvent, formant un **cluster** local. Lorsque la somme

$$\Omega(\mathcal{C}_\alpha) = \sum_{k,\ell \in \mathcal{C}_\alpha} \omega_{k,\ell}$$

d'un cluster \mathcal{C}_α dépasse un certain seuil θ , on en déduit qu'un **mégapattern** naît (au palier macro). Il se peut qu'il s'agisse d'un début de bulle spéculative en bourse ou d'une instabilité météorologique se généralisant. Le fait de **surveiller** $\Omega(\mathcal{C}_\alpha)$ ou la taille de ce cluster peut fournir un **indicateur** d'émergence potentielle de crise ou de phénomène extrême.

Anticipation de phénomènes extrêmes.

La logique du DSL **prévient** l’analyste ou l’automate d’une dérive importante dès que les pondérations ω se synchronisent autour d’un **sous-groupe** d’événements. Contrairement à un modèle statistique statique (par ex. Gaussien), le DSL s’adapte localement, non seulement aux moyennes, mais aussi aux corrélations actives. Dès la mise en place d’une concentration d’événements inhabituels, le **cluster** associé grandit — signe avant-coureur. Cela facilite des **alertes** précoces, car la synergie locale s’affirme avant même que le mégapattern ne devienne flagrant à la simple observation des moyennes.

B. Clusterisation Plus Lisible et Hiérarchie Compréhensible

Dans des flux de données complexes comme la bourse, le climat ou les logs de grande dimension, une simple liste d’événements peut rapidement devenir incohérente ou surchargée. Le **DSL** introduit une **hiérarchie** en plusieurs étapes. D’abord, la mise à jour des pondérations ω détecte les micro-regroupements locaux. Ensuite, l’**agrégation** à l’aide de la fonction Ψ construit un graphe macro où chaque super-nœud représente un **mégapattern** significatif. Cette structuration permet une **réduction** de la complexité. Quelques super-nœuds dominants traduisent la structure globale, tandis que les **petits** clusters conservent une granularité fine pour ceux qui souhaitent approfondir les détails.

Au niveau **macro**, les clusters identifiés traduisent des phases ou des régimes comme un *bull market* ou une *saison cyclonique*, qui sont directement **significatifs** pour la prise de décision. L’analyste, le trader ou le climatologue n’a plus besoin de traiter individuellement des milliers d’événements. Ils peuvent s’appuyer sur deux ou trois **mégapatterns** dominants. Sur le plan **mathématique**, cette clarification découle de la **dynamique** DSL qui accentue certains liens tout en en atténuant d’autres, aboutissant à un graphe plus **lisible**, où quelques **hubs majeurs** émergent parmi une multitude de petits nœuds.

Une entité (gestionnaire de portefeuille, centre de prévision météo) peut prendre des décisions plus **ciblées** en considérant seulement la configuration macro des mégapatterns. Si un super-nœud “hausse” grossit dangereusement, un opérateur en bourse peut se méfier d’un krach à venir, ou inversement profiter d’une flambée haussière. Si un super-nœud “vague de chaleur” s’impose, les pouvoirs publics peuvent anticiper des plans d’urgence (restriction d’eau, alertes canicule). Le DSL évite donc la trop grande granularité d’un clustering purement local et dévoile une **vision** multi-niveau.

C. Exemple Numérique Illustratif

Considérons un flux de 1 000 événements $\{\mathcal{E}_1, \dots, \mathcal{E}_{1000}\}$. Chaque nouvel événement \mathcal{E}_{new} met à jour $\omega_{\text{new},j}$ par la formule

$$\omega_{\text{new},j}(t+1) = \omega_{\text{new},j}(t) + \eta[S(\mathcal{E}_{\text{new}}, \mathcal{E}_j) - \tau \omega_{\text{new},j}(t)].$$

Si \mathcal{E}_{new} se révèle très cohérente avec plusieurs $\{\mathcal{E}_j\}$ déjà existants, elle renforce $\omega_{\text{new},j}$ et rejoint un cluster local \mathcal{C}_α . Au **niveau macro**, lorsque la somme des pondérations internes $\Omega(\mathcal{C}_\alpha) = \sum_{i \in \mathcal{C}_\alpha} \sum_{j \in \mathcal{C}_\alpha} \omega_{i,j}$ dépasse un seuil θ ou lorsqu’une corrélation transversale avec un autre cluster \mathcal{C}_β est détectée, un **super-nœud** $\mathcal{N}_{\alpha,\beta}$ plus vaste est formé. Cette fusion permet d’**alléger** la représentation en réduisant le nombre d’éléments à manipuler. Plutôt que de traiter un millier d’événements distincts, l’analyste peut se concentrer sur quelques **super-nœuds** significatifs. L’**anticipation**

s'améliore, car l'expansion d'un macro-nœud traduit une **dynamique** forte pouvant signaler l'émergence d'une crise ou d'un pic extrême.

6.7. Conclusion

Au fil de ce chapitre 6, nous avons mis en avant l'importance de l'**apprentissage multi-échelle** dans le **DSL** (Deep Synergy Learning) et la **possibilité** d'observer une **fractalité** ou une forme d'auto-similarité structurelle à divers paliers. Nous avons également détaillé la façon dont les **flux ascendants** (bottom-up) et **descendants** (top-down) interagissent, soutenant une **dynamique** qui relie intimement les micro-clusters (niveau local) et les macro-structures (niveau global).

6.7.1. Synthèse des Contributions du Chapitre

Apprentissage Multi-Échelle.

L'ensemble des sections étudiées met en évidence la **structure** multi-niveau du **Deep Synergy Learning (DSL)**. Contrairement à de nombreux modèles neuronaux ou algorithmes de clustering qui se limitent à un seul palier, le DSL organise les **entités** observées en plusieurs **niveaux** distincts. Les **éléments** au niveau **micro** (patchs visuels, tokens textuels, capteurs unitaires, transactions ponctuelles, etc.) sont d'abord reliés par des pondérations $\{\omega_{i,j}\}$ mises à jour via l'équation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)].$$

Cette étape aboutit à la formation de **clusters** ou regroupements locaux. Une **fonction d'agrégation** Ψ (voir la section 6.2) permet ensuite de **monter** à un niveau **macro** ou **méso**, créant ainsi des **super-nœuds** $\mathcal{N} * \alpha$ qui regroupent des ensembles de forte cohérence. Les pondérations $\omega * \alpha, \beta$ entre ces super-nœuds sont mises à jour en appliquant la même règle DSL. Cette **hiérarchisation** repose sur un **flux ascendant** (bottom-up), où les clusters micro se combinent pour former des structures plus vastes, et un **flux descendant** (top-down), où le niveau macro exerce un *feedback* pour clarifier, scinder ou réassigner les liens micro lorsqu'un sous-ensemble manque de cohérence.

Fractalité et Auto-Similarité.

Il est apparu que dans plusieurs domaines tels que la **vision multimodale**, la **conversation**, la **robotique** et les **événements temporels** (bourse, climat), une **répétition** de motifs se manifeste à différentes échelles. Une même **trame structurelle** ou un même **pattern** se retrouve aussi bien au niveau local, avec quelques entités, qu'au niveau global, à travers des macro-nœuds. Cette **auto-similarité** constitue l'essence de la **fractalité**, observable dans les **lois de puissance** (queues lourdes, distributions scale-free) et dans la répétition de motifs d'interactions comme les séquences de dialogue ou les organisations sensorimotrices. Le **DSL**, par son **auto-organisation** basée sur la mise à jour $\omega_{i,j}(t+1) = \dots$, met en évidence ces **invariants d'échelle** en renforçant les liens dans les zones denses, favorisant la **clusterisation locale**, puis en agrégeant ces clusters en **super-nœuds** macros, révélant ainsi des **hubs** dominants ou des **structures globales** à la topologie fractale.

Organisation des Flux.

Un point central est la **gestion** conjointe de deux types de flux. D'un côté, un **flux ascendant** (bottom-up) produit les agrégations successives – chaque palier transforme ses entités en super-

nœuds pour le palier supérieur. De l'autre côté, un **flux descendant** (top-down) s'exprime en imposant un **feedback** sur les pondérations micro en fonction du contexte macro. Cette rétroaction, dans la formulation DSL, prend la forme d'un **terme** Δ_{down} ou d'une modulation de $\omega_{i,j}$. Elle agit comme un mécanisme de validation, de scission ou d'inhibition quand le niveau macro perçoit une incohérence. Cette **cohérence** entre flux ascendants (consolidation locale) et flux descendants (guidage contextuel) garantit la **stabilité** de la structure globale, évitant les errances ou les convergences trop partielles.

Ainsi, ce chapitre a mis en évidence la puissance du DSL pour traiter la multi-échelle. Qu'il s'agisse de patches d'image s'assemblant en classes sémantiques, de segments de conversation formant des intentions, de capteurs robotiques convergeant vers un comportement global, ou d'événements ponctuels s'agrégeant en mégapatterns, la même logique d'auto-organisation permet de capturer à la fois les détails locaux et la vision macro. Les phénomènes fractals ou *scale-free* qui émergent sous forme de patterns répétitifs et de lois de puissance ne nécessitent pas de modèle statistique rigide. Le DSL s'adapte localement, détecte et exploite les récurrences et corrélations à différents niveaux, révélant ainsi des structures hiérarchiques complexes dans des systèmes de grande échelle.

6.7.2. Rôle de la Fractalité

La **fractalité** repose sur l'idée qu'un *même* schéma, observé à une échelle donnée, peut se **répliquer** ou se **réinventer** à d'autres échelles, traduisant ainsi une **auto-similarité** potentielle. Dans le cadre du **Deep Synergy Learning (DSL)**, cette propriété s'exprime par la récurrence des mécanismes d'agrégation et de division à tous les paliers de l'organisation (micro, méso, macro). Les configurations locales, qu'elles correspondent à des clusters de segments de conversation, de capteurs, de patches visuels ou de transactions, se retrouvent avec une forme analogue au niveau plus global, où l'on regroupe ces mêmes clusters en super-nœuds.

Sur un plan **mathématique**, un système fractal s'identifie souvent par l'analyse de la **dimension fractale** d'un graphe ou d'une distribution. Si la répartition des **degrés des nœuds**, des **énergies** ou des **intensités** suit une **loi de puissance**, on observe une **invariance d'échelle**, également appelée *scale-freeness*. Dans un **DSL**, la construction du graphe $\{\omega_{i,j}\}$ par la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

peut alors produire un réseau dont la topologie répond à cette caractéristique “hub-and-spoke”, où de gros clusters apparaissent et où la distribution des tailles (ou des degrés) se signale par un comportement de type Pareto. Cette observation reflète l'**auto-similarité**. Si, à l'échelle micro, de petits regroupements émergent selon la même règle d'auto-organisation, alors à l'échelle intermédiaire, ces regroupements se combinent. À l'échelle macro, on retrouve encore le même profil d'agencement, illustrant une cohérence structurelle à travers les niveaux.

D'un point de vue **conceptuel**, la fractalité présente un double intérêt. Elle facilite d'abord la **compréhension** de la hiérarchie formée en montrant que chaque niveau applique la même dynamique $\omega_{i,j}$ de renforcement et d'inhibition, d'agrégation et de division. Cette itération cyclique de la même loi donne une vision globale unifiée, car la même “grammaire” d'auto-organisation est utilisée à tous les niveaux. Elle simplifie ensuite la **modélisation**, car si les distributions suivent des lois de puissance, on peut s'attendre à observer des *fat tails* avec quelques

énormes clusters au milieu d’une myriade de petits, sans nécessiter un nouveau cadre statistique. La répétition d’une seule règle sur plusieurs paliers explique le caractère fractal et clarifie la lecture d’un ensemble complexe. La fractalité n’est donc pas un simple *concept*, mais un *levier* pour structurer le DSL de manière homogène du local au global et une *fenêtre* pour appréhender la distribution multi-niveau comme un tout, où la répétition des formes révèle les lois profondes d’organisation.

6.7.3. Limites et Perspectives

Conditions d’Apparition de la Fractalité.

Bien que l’approche **multi-échelle** du Deep Synergy Learning (DSL) s’applique dans de nombreux domaines (vision multimodale, robotique sensorimotrice, analyse conversationnelle, etc.), la **fractalité** n’est pas systématiquement garantie. Il s’agit d’un cas particulier dans lequel la structure ou la dynamique du **Synergistic Connection Network (SCN)** reproduit à plusieurs échelles une forme d’**auto-similarité**, souvent associée à des distributions en lois de puissance ou à un “effet hub” dans le graphe $\{\omega_{i,j}\}$. Un **SCN** qui ne verrait jamais se former de gros clusters dominants, ou dont la répartition des degrés se rapprocherait davantage d’une loi gaussienne, présenterait peu ou pas de **fractalité** mesurable.

D’un point de vue plus formel, la **fractalité** exige que la *même* loi d’**organisation** (agrégation, renforcement, division) produise, à chaque palier, une structure semblable à celle observée aux autres niveaux. Cela signifie que la répartition des liens $\omega_{i,j}$ ou la distribution des tailles de clusters suit une loi **autosimilaire** lorsqu’on passe du niveau micro au niveau macro. Toutefois, dans certaines applications, les **conditions** nécessaires à l’apparition de lois de puissance ou de distributions *scale-free* ne sont pas réunies. Un phénomène peut être plus linéaire, plus symétrique ou dominé par d’autres effets, comme une saturation rapide des ω ou une homogénéité rigide. Dans ces cas, la fractalité est soit marginale, soit inexistante.

Il importe donc de distinguer l’architecture multi-niveau du DSL (toujours présente) de la **fractalité** (cas où la structure s’avère *auto-similaire* à divers paliers). Il est parfaitement concevable qu’un **DSL** multi-niveau ne révèle aucune auto-similarité notable si les données ou la dynamique ne contiennent pas de régularité fractale. La fractalité n’est donc pas une conséquence obligatoire de l’algorithme, mais une propriété émergente dans certaines configurations de $\{\omega_{i,j}\}$ ou dans certains flux d’événements (voir 6.6.5.2) qui induisent, par leur “queue lourde” ou leur récurrence, une structure fractale.

Possibilités de Recherche et Perspectives

Étude de la dimension fractale d’un SCN.

Une piste intéressante consiste à étudier la **dimension fractale** du **Synergistic Connection Network**. On peut imaginer calculer une “dimension de boîte” du graphe $\{\omega_{i,j}\}$ à chaque itération ou pour chaque palier (micro, macro). La **stabilité** de cette dimension fractale, ou sa croissance/diminution, indiquerait dans quelle mesure le réseau tend à s’**auto-similariser**. Un **SCN** dont la **dimension** fractale s’avère stable (par ex. ≈ 1.6) pourrait être interprété comme

“parfaitement” organisé en lois de puissance. À l’inverse, un SCN dont la dimension fractale fluctue beaucoup ou tend vers un régime gaussien montrerait l’absence d’un patron fractal durable.

Indicateur de cohérence ou de maturité.

Sur un plan **conceptuel**, la fractalité peut servir d’indicateur de **maturation** du réseau. Un **score fractal** $\mathcal{F} \in [0,1]$ permettrait d’évaluer la répétition du même schéma à différentes échelles. Plus ce score est élevé, plus l’**auto-organisation** est stabilisée, avec une structure multi-niveau qui se reproduit à chaque palier. D’un point de vue pratique, cet indicateur permettrait de diagnostiquer si le SCN est encore en phase transitoire (score fractal bas) ou s’il a atteint un état structuré et abouti (score fractal élevé).

Heuristiques favorisant la fractalité.

Certains algorithmes DSL pourraient être **modulés** pour favoriser l’émergence de structures fractales en ajustant les paramètres η , τ , γ liés à l’inhibition et à la synergie. Ce réglage encouragerait la formation de quelques hubs majeurs entourés de nombreux petits clusters, typiques des graphes scale-free. Une telle approche renforcerait la **robustesse**, car un réseau fractal résiste mieux aux perturbations ou aux pannes locales, la structure globale restant stable grâce à sa répétition auto-similaire. Dans le cas des flux événementiels comme la bourse ou le climat, détecter et exploiter la fractalité permettrait une meilleure gestion des extrêmes, améliorant ainsi la **résilience** face aux fluctuations imprévisibles.

6.7.4. Liens avec les Chapitres Suivants

Chapitre 7 (Optimisations).

Le présent chapitre a mis en avant la puissance du **DSL** (Deep Synergy Learning) dans la gestion multi-niveau, allant du micro au macro, et la possibilité de **fractaliser** la structure d’auto-organisation. Pour stabiliser et améliorer cette hiérarchie, on utilise des mécanismes d’**optimisation** comme le **recuit simulé**, qui injecte du bruit pour éviter les minima locaux, ou l’**inhibition**, qui limite la croissance ubiquitaire des pondérations ω . Dans une configuration **multi-niveau**, ces techniques peuvent s’appliquer à différents paliers, avec un recuit “local” au niveau micro, un recuit “global” au palier macro, et des stratégies d’inhibition hiérarchique. Le **chapitre 7** approfondira cette approche et expliquera comment, en exploitant la répétition fractale d’une structure d’un palier à l’autre, on peut optimiser les niveaux inférieurs puis répliquer ces ajustements à un niveau macro, permettant ainsi une **réduction** significative de la complexité.

D’un point de vue **mathématique**, on adaptera les paramètres η , τ , γ (taux d’apprentissage, facteur de décroissance, amplitude d’inhibition) en jouant sur la récurrence fractale pour accélérer la convergence ou améliorer la robustesse à la sur-segmentation (trop de clusters) ou à la sous-segmentation (regroupement trop vaste).

Chapitre 8 (Multimodal).

La fractalité prend encore plus de sens lorsque l’on gère des **flux très hétérogènes** (vision, audio, texte, signaux divers) dans un cadre véritablement **multimodal**. Le **chapitre 8** décrira comment le **DSL** s’applique pour **fusionner** des sources de données de nature différente, tout en exploitant des

patterns récurrents à diverses résolutions (par exemple, motifs d’objets en vision, motifs de forme sonore en audio, motifs lexicaux en texte). Un système multimodal peut observer des invariants d’échelle si la même logique d’auto-organisation apparaît dans chaque modalité, puis se recopie lors de la fusion inter-modale. Le **DSL** fractal détectera et tirera parti de ces symétries à différents niveaux (micro = détail sensoriel, macro = concept global) avec un gain considérable dans l’interprétation d’environnements complexes.

Chapitre 9 (Temps réel).

Dans la plupart des applications réelles, les données (événements, segments, mesures) arrivent **en flux continu**. La fractalité, déjà introduite dans ce chapitre (6.7.3), suppose une auto-similarité potentielle. En **temps réel**, cela signifie que la **dimension fractale** ou la “signature scale-free” du réseau $\{\omega_{i,j}\}$ peut évoluer au fur et à mesure que de nouvelles entités sont incorporées. Le **chapitre 9** décrira précisément comment adapter la **clusterisation** micro et macro dans un **cadre** en perpétuel changement. Si une structure fractale était stabilisée, l’arrivée de nouveaux segments ou événements peut la perturber, forçant le **DSL** à réorganiser ses liens ω . On peut ainsi mesurer la fractalité au fil du temps (log–log plots, dimension fractale dynamique) ou se fixer un **objectif** de maintenir un certain degré d’auto-similarité, marque de l’équilibre ou de la maturité du SCN.

6.7.5. Vision Globale

La **démarche du Deep Synergy Learning (DSL)** ne se limite pas à un **traitement** local des paires d’entités. Elle repose sur un **apprentissage multi-échelle**, structurant la **complexité** de manière hiérarchique, du niveau **micro** où s’effectuent les regroupements locaux, au niveau **macro** où s’agrègent des clusters plus vastes. Cette organisation permet une répartition cohérente de la **synergie** $\omega_{i,j}$, chaque palier s’appuyant sur le précédent plutôt que de traiter toutes les entités dans un **graphe monolithique**. Cette **stratification** favorise une meilleure lisibilité, une adaptation plus fine et l’exploitation des propriétés fractales, rendant le modèle plus flexible et efficace.

La notion de **fractalité** joue un rôle clé dans la **vision globale** du DSL. Elle consiste à **observer** et **exploiter** l’**auto-similarité** lorsque des patterns ou lois d’organisation émergent de manière récurrente à différents niveaux. Si un phénomène apparaît à l’échelle micro, avec des entités renforçant leurs liens, et que la même structure se retrouve à un niveau supérieur sous forme de super-nœuds interconnectés de façon similaire, on parle alors d’**autosimilarité** d’échelle. Cette fractalité se traduit par une **répétition** du même schéma d’auto-organisation, pouvant conduire à des distributions en lois de puissance ou à l’apparition de “hubs” dominants dans le graphe.

En pratique, l’**apprentissage multi-échelle** proposé par le DSL s’avère particulièrement puissant pour plusieurs raisons :

1. Lecture améliorée de la dynamique interne (micro) et des schémas (macro) :
Le modèle permet de **circonscrire** des petits groupements de données (patches visuels, tokens textuels, capteurs locaux, transactions ponctuelles) afin d’en extraire des **clusters** cohérents. Simultanément, les **super-nœuds** obtenus aux niveaux supérieurs (macro) font ressortir des “grands schémas” (catégories visuelles, intentions conversationnelles, comportements robotiques, tendances de marché) qui, autrement, seraient noyés dans la masse. Cette complémentarité rend

possible une *navigation* entre le micro (analyses très localisées) et le macro (vision plus globale ou sémantique), sans se perdre dans la complexité.

2. Ordonnement de la synergie $\{\omega\}$:

La **hiérarchie** rend plus robuste et plus interprétable la mise à jour des pondérations $\omega_{i,j}$. Au lieu de maintenir un énorme graphe plat où les connexions foisonnent de manière incontrôlée, on regroupe, on agrège, on divise si besoin, et on définit des ponts clairs entre les niveaux. Cela **stabilise** souvent le processus, réduit les illusions de sur-connexion et permet d'**isoler** des incohérences au bon palier (micro ou macro).

3. Résilience et plasticité :

Le fait de disposer d'une **organisation** à plusieurs échelles permet au réseau de s'**adapter** localement aux bouleversements tels que l'ajout de nouveaux segments, la panne d'un capteur ou une brusque variation d'une action, tout en maintenant une **cohésion** d'ensemble au niveau **macro**. Les flux ascendants (bottom-up) et descendants (top-down) s'équilibrent, garantissant que les modifications locales restent contenues et que la reconfiguration d'un macro-nœud ajuste la structure globale sans perturber l'organisation micro.

La **fractalité** s'avère être un **levier** précieux dans cette vision globale. Lorsqu'on constate un patron fractal, cela signifie que les *mêmes* lois d'agrégation ou de distribution se retrouvent d'un niveau (micro) à l'autre (macro). On en tire deux conséquences principales :

- **Exploitation** des invariants d'échelle : on peut analyser un phénomène (visuel, conversationnel, boursier, etc.) à petite échelle et en extrapoler des **lois** semblables à grande échelle. Le DSL se charge, par la règle $\omega(t+1) = \omega(t) + \eta[S - \tau\omega(t)]$, de *conserver* cette structure lorsqu'on assemble les clusters.
- **Compréhension** plus profonde du système : une fractalité marquée indique une **auto-similarité** persistante, susceptible d'être utilisée pour la **prédiction** (p. ex. distribution heavy-tailed) ou pour l'**optimisation** (cibler seulement quelques "hubs" dominants). Si la structure fractale est souhaitable, on peut d'ailleurs concevoir des paramètres ou heuristiques (chap. 7) favorisant sa stabilisation.

En définitive, l'**apprentissage multi-échelle** propre au **DSL** et l'**auto-similarité** fractale à plusieurs paliers renforcent la structure du réseau. Cette organisation permet une meilleure **visualisation** de la dynamique interne, facilite l'**identification** de grands patterns et assure une **résilience** face aux nouvelles entités ou aux fluctuations extrêmes. Les chapitres suivants approfondiront divers **cas pratiques** tels que l'optimisation, la multimodalité et le temps réel, montrant que l'**apprentissage multi-niveau** et la **fractalité** ne sont pas de simples caractéristiques annexes, mais bien des **piliers fondamentaux** de l'adaptabilité et de l'auto-organisation du **DSL** dans des systèmes complexes ou *scale-free*.