The Extended Systems Theory: The Semantic Transition Space

Author: Metehan Yurtseven
Version: 0.1
Date: May 12, 2025

Abstract

The Semantic Transition Space is a conceptual dynamic framework for representing and evolving semantic structures. It combines elements from semantics, category theory, relativity theory, vector space logic, and AI to systematically and multidimensionally map the meaning and relationships of concepts. This whitepaper presents the theoretical foundation, describes central axis types, meta-levels, and the underlying system dynamics, and provides initial insights into possible applications in fields such as artificial intelligence, cognitive semantics, and systems theory.

The underlying model is conceived as an extension of classical systems theory approaches – as an **extended systems theory** that is mathematically precise and treats meaning as a dynamically emergent field.

Glossary of Central Terms

Term	Short Definition
Concept	Elementary point in the semantic space, defined by values on axes
Axis	Semantic dimension along which concepts are distinguished
Meta-Axis	Control variable that influences properties of axes (e.g., universality)
Meta-Rule	Rule acting on the meta-level, e.g., governing emergence or reduction
Transition Space	The entire dynamic semantic space with all concepts and axes
Tick	A development step in the system in which rules are applied

1. Motivation & Context

In a world of growing semantic complexity—where modern information systems, artificial intelligence, and human language increasingly operate in meaning spaces whose structure is context-dependent, dynamic, and emergent—there is a lack of models that can systematically represent these conditions. Classical semantic models are mostly static, reductionist, or insensitive to context, and fail where meaning changes continuously.

We propose a theoretical model that extends classical semantic structures and complements them with a dynamically transformable axis architecture—the *Semantic Transition Space*. It combines structural depth, emergence, and meta-dynamics—comparable to a curved spacetime in which semantic forces act. The core idea: meaning is not a point in a rigid vector space, but a state in the field of tension of several semantic axes, which themselves can be changed by meta-axes.

Inspired by physical spacetime models (e.g., relativity theory), categorical logics, and systemic emergence processes, the Semantic Transition Space models concepts as points in a dynamic meaning space. Meta-axes act like forces on axes—they curve, reduce, intensify, or couple them. Meta-axes modulate properties of semantic axes—such as their universality, i.e., how many concepts can be arranged along a given dimension. Another property is carrying capacity, i.e., how strongly an axis differentiates meanings. Both act analogously to spacetime curvature: they influence how concepts are distributed in the space, how quickly they move along an axis, and which structures can emerge. This results in a system that not only describes meaning, but can itself generate new semantic dimensions.

This structure opens up new theoretical and practical spaces that enable radical shifts in perspective across different disciplines. The potential fields of application are far-reaching: In **computer science**, self-organizing vector databases could emerge, whose semantic structure dynamically adapts to content changes. In **cognitive semantics**, a system could be developed that not only maps language, but also makes **feelings, motives, and meaning shifts** traceable. In **physics**, new approaches to modeling singularities—such as black holes or the Big Bang—could be developed by transferring the meta-space structure to the gravitational level. And in **simulation technology**, semantically controlled agent worlds could arise that do not operate by fixed rules, but by dynamically emergent meaning.

The Semantic Transition Space is thus not only a new framework for meaning modeling, but can also be seen as an **extended systems theory**: it takes up basic principles of classical systems theory—such as operational closure, level formation, and context dependence—and extends them with formalized meta-levels, dynamic curvature forces, and a recursive axis system that makes semantic evolution explicitly describable.

The aim of this whitepaper is to formulate the theoretical foundation of the Transition Space, to sketch an axiomatic structure, and to provide initial insights into possible applications—in AI architecture, cognitive semantics, and epistemological model building.

2. System Overview & Terminology Architecture

The Semantic Transition Space is based on a hierarchically structured architecture in which concepts, semantic axes, and meta-axes interact within a dynamic space. The system is designed not only to represent meaning, but to actively generate, shift, and differentiate it. The following core components define the fundamental structure:

2.0 Meaning as Relation

The Semantic Transition Space follows a central principle: **Meaning arises solely through relation.** A single concept in an empty space is semantically meaningless, as there is no comparison, no difference, no interaction. Only when a second concept is added can a distinction arise—for example, in color, shape, function, or abstract category.

At this moment, a **semantic axis** begins to form. It is not statically predetermined, but is implicitly defined by the difference between concepts. The more concepts can be located along such an axis, the clearer, more relevant, and "weightier" this dimension becomes in the semantic space.

This process can be described step by step:

1. **A single concept**: No relation, no meaning.
2. **Two concepts**: A first difference emerges—a potential axis is formed.
3. **Third concept**: If this one is also based on the same distinction, the axis stabilizes.
4. **Further concepts**: The more concepts can be meaningfully projected along this dimension, the more strongly the axis is established in the space.

Meaning is thus not absolute, but **relational**—a dynamic phenomenon that arises from the interplay of differences and similarities. This idea is closely related to concepts from **systems theory**: There, meaning does not arise from isolated elements, but from **distinctions in the system context** (cf. Niklas Luhmann). The Semantic Transition Space transfers this principle into a structural-mathematical form, in which axes and concepts mutually define and condition each other.

On this basis, the architecture of the system is explained below.

2.1 Concepts (SemanticConcept)

Concepts are the fundamental units in the Transition Space. They can be concrete objects, abstract ideas, states, or relations. Each concept has values along several semantic axes and is thus uniquely located in the space.

2.2 Semantic Axes (SemanticAxis)

A semantic axis describes a dimension of meaning. Axes have different types, including:

- **Scalar** (e.g., intensity, size, proximity)
- **Polar** (e.g., positive/negative, good/evil)
- **Ordinal** (e.g., rankings)
- **Binary** (e.g., yes/no)
- **Temporal** (temporal progression)
- **Spatial** (position, direction)

Concepts can be located on multiple axes simultaneously. The axis structure enables a semantically dense description even of complex terms.

2.3 Meta-Axes (MetaAxis)

Meta-axes do not act directly on concepts, but influence the properties of the semantic axes themselves. They form the meta-level of the system and allow for higher-level control of the semantic landscape. Important meta-axes are:

- **Universality**: Indicates how many concepts can be meaningfully arranged along an axis—a measure of its semantic range.
- **Carrying Capacity**: Determines how strongly an axis differentiates concepts—i.e., its semantic differentiation power.
- **Redundancy Density**: Measures how similar an axis is to other axes—high density indicates possible mergers.
- **Semantic Gravitation**: Describes the overall attractive force of a semantic zone—the higher the density and concentration of meaning, the stronger the semantic curvature in the space.

Meta-axes act dynamically on the entire system: they can strengthen, weaken, bend, or even eliminate axes, thereby transforming the semantic space itself.

2.4 Dynamic Object Space (SemanticSpace)

The Transition Space is not static, but a dynamically structured space in which concepts emerge, move, disappear, or reorganize. This space is subject to iterative cycles in which the following processes take place:

- Application of meta-axes
- Reduction of redundant or weak axes
- Emergence of new structures or categories
- Recalculation of axis profiles

This creates a self-reflexive, evolutionary field of meaning.

2.5 Physical Analogy

The Semantic Transition Space is structurally oriented toward physical models of spacetime. Just as mass in physical space generates gravitation and thus curves space and time, a high semantic density in the Transition Space generates so-called **semantic**

gravitation. This affects the structure and dynamics of the meaning space—especially the mobility of concepts and their positioning along relevant axes.

A central concept here is **semantic time velocity**. It describes how quickly concepts develop along semantic processes or transformations. In regions of high semantic gravitation—such as through many concepts, high universality, or strong differentiation—this development slows down, comparable to time dilation in relativity theory.

Time in the Transition Space is understood as an emergent, location-dependent phenomenon—not as a fixed axis, but as a dynamic structure arising from semantic relations. This perspective can also provide impulses for an extended view of physical time, in which time is not an absolute flow, but a local effect of relational density and structure—similar to hypotheses in modern physics on quantum gravity or relational concepts of time.

Furthermore, the concept of the meta-axis can be extended recursively: it is conceivable that there are **multiple meta-levels**—i.e., meta-axes that in turn influence other meta-axes. This creates a hierarchically layered, potentially infinite control system that enables an extremely fine-grained and adaptive semantic structure. This recursive structure allows not only the semantic structure itself, but also its rules of change to be modeled in a context-dependent way.

3. Mathematical Framework

The mathematical foundation of the Semantic Transition Space is deliberately kept minimalistic and abstract. The goal is to formulate a universally applicable model of meaning that remains independent of concrete implementations, yet is still precisely formalizable.

3.1 Legend of the Most Important Symbols

Symbol	Meaning
t	Time (discrete time steps)
R	Space, the base set of all structures
$E^{(t)}$	Set of all levels in the space at time t
$E_0^{(t)}$	Base level (level 0) at time t
$E_1^{(t)}$	Meta-level (level 1) at time t
$A^{(t)}$	Set of all axes in the space at time t
$A_{E_0}^{(t)}$	Axes of the base level at time t

$A_{E_1}^{(t)}$	Axes of the meta-level at time t
$W_a^{(t)}$	Value range of axis a at time t
$C_{E_i}^{(t)}$	Set of concepts on level i at time t
c	Single concept
x_a	Value of concept c on axis a
$\pi_a(c)$	Projection of a concept c onto axis a
F, F_i	Meta-rule operators (update functions for levels)

3.2 Fundamental Sets, Time, and Mappings

$$t \in \mathbb{N} : \text{Time (discrete time steps)}$$

Time t describes the evolution of the system in discrete steps.

$$R = \text{Space}$$

R is the base set of all structures under consideration.

$$E^{(t)} \subseteq R$$

$E^{(t)}$ is the set of all levels in the space at time t.

$$A^{(t)} = \text{Set of all axes in the space at time } t$$

$A^{(t)}$ is the set of all axes that can occur in any level at time t.

$$\forall a \in A^{(t)} : \quad W_a^{(t)} = \text{Value range of } a \text{ at time } t$$

Each axis a has its own value range $W_a^{(t)}$ at each time t, which can change over time.

3.3 Levels and Meta-Levels (time-dependent)

$$E_0^{(t)} \in E^{(t)} : \text{First level at time } t$$

$$E_1^{(t)} \in E^{(t)} : \text{Second level (meta-level) at time } t$$

$E_0^{(t)}$ is the base level, $E_1^{(t)}$ is a meta-level above it, each at time t.

$$A_{E_0}^{(t)} \subseteq A^{(t)} : \text{Axes on level } E_0 \text{ at time } t$$

$$A_{E_1}^{(t)} \subseteq A^{(t)} : \text{Axes on level } E_1 \text{ at time } t$$

Each level has, at each time, a subset of the axes as its own axes.

3.4 Recursive Structure of Levels (time-dependent)

Let $C_{E_i}^{(t)}$ be the set of all concepts on level E_i at time t.

$$\forall i \geq 1 : \quad A_{E_{i-1}}^{(t)} \subseteq C_{E_i}^{(t)}$$

The axes of a level are, at each time, concepts on the next higher meta-level. This is a recursive definition.

3.5 Dynamics: Meta-Rules as Operators Over Time

The meta-level E_1 defines rules that are applied to E_0 at each time step:

$$E_0^{(t+1)} = F(E_0^{(t)}, \text{Meta-rules on } E_1^{(t)})$$

The state of the base level at the next time step results from its current state and the meta-rules of the meta-level.

Analogously, this applies to all levels:

$$E_i^{(t+1)} = F_i(E_i^{(t)}, \text{Meta-rules on } E_{i+1}^{(t)})$$

Each level can be controlled by the meta-rules of the next higher level.

3.6 Concepts as Tuples on Axes (time-dependent)

$$C_{E_0}^{(t)} : \text{Concepts on level } E_0 \text{ at time } t$$

The concepts on the base level at time t.

$$\forall c \in C_{E_0}^{(t)} : \quad c \in \prod_{a \in A_{E_0}^{(t)}} W_a^{(t)}$$

Each concept c is a tuple of values, where each value belongs to an axis from $A_{E_0}^{(t)}$.

$$c = (x_a)_{a \in A_{E_0}^{(t)}}, \quad x_a \in W_a^{(t)}$$

The tuple c consists of the values x_a for each axis a of the level at time t.

The notation applies analogously for concepts and axes on any level E_i and at any time t.

3.7 Projection onto an Axis (time-dependent)

$$\pi_a(c) = x_a$$

The projection of a concept c onto an axis a is simply the value x_a of c on this axis (at the current time t).

3.8 Concrete Example of Temporal Development

Assume:

- $A_{E_0}^{(0)} = \{a_1, a_2\}$ with $a_1 = x$, $a_2 = y$
- $W_{a_1}^{(0)} = W_{a_2}^{(0)} = [0, 1]$
- $c^{(0)} = (0.2, 0.8)$

A meta-rule on E_1 could be: "If many concepts are close to $x = 1$, extend W_{a_1} upwards."

After applying the rule in the next time step:

- $W_{a_1}^{(1)} = [0, 1.5]$
- $c^{(1)} = (0.2, 0.8)$ (or new concepts are added)

In this way, the development of the space and the concepts over time can be modeled. Each axis of the base level is a concept on the meta-level.

4. Example Application & Dynamics

4.1 Purpose of This Section

This chapter provides an exemplary illustration of how the Semantic Transition Space behaves in operation—without being tied to a specific application domain. The aim is to make the dynamic character of the system tangible and to clearly show the interplay between concepts, axes, meta-axes, and temporal development.

Instead of working through a fixed use case in full, this section focuses on the **systemic perspective**: How do the components interact? What happens between two states of the space? How do new axes emerge? How do meta-rules act?

Three levels of representation are used:

1. **Visual sketches and tabular mini-examples** to illustrate emergence processes.
2. **Pseudocode in Python-like syntax** to formally and implementation-near describe the dynamics.
3. **Object-oriented model structure** to show how such a system can be modularly

constructed.

The presented content is based on a real prototype developed in Python. However, the implementation is not intended as proof, but for **understanding and strategic connectivity** of the theoretical model.

4.2 Visual Mini-Example: Axis Formation Through Semantic Difference

To illustrate the emergence mechanism of the Transition Space, we consider a highly simplified scenario with initially two concepts in a space without axes:

Initial State (t_0):

Concept	Property A	Property B
C_1	"red"	"round"
C_2	"blue"	"round"

Since "round" is the same for both concepts, but "red" ≠ "blue", a semantic difference arises along a potential color axis.

Conclusion: → Automatic emergence of a new semantic axis `A₁ = color`
→ Value range e.g. `{red, blue}` or as an abstract vector `[c₁.color, c₂.color]`

Structure after Emergence (t_1):

Axis	Type	Concepts with Position
A_1	categorical	C_1 (red), C_2 (blue)

Further concepts could extend or help differentiate the axis.

Note:
In the implementation, a meta-rule could recognize this distinction, for example, if:

- many concepts have the same value for a property (=> no axis needed),
- but differ significantly in exactly one other property.

→ This would classify the axis as **universal and carrying**.
In later ticks, the axis can be stabilized, transformed, or possibly removed again—depending on semantic gravitation and axis density.

Important: Axis formation is an iterative process. With each new concept introduced into the space, existing axes can further differentiate, new axes can emerge, or superfluous axes can disappear. Thus, the structure of the Transition Space remains dynamic and adaptable at all times.

4.3 Dynamics as Pseudocode (Tick Cycle)

To make the systemic dynamics of the Transition Space understandable in its iterative logic, the following pseudocode serves as an abstraction of the `tick()` cycle. The focus is on semantic comprehensibility, not implementation efficiency.

```
def tick(concepts, axes, meta_axes):
    # 1. Meta-axes act on axes (e.g., curvature, weighting)
    for meta in meta_axes:
        axes = meta.apply(axes)

    # 2. Reduction: Remove unnecessary or redundant axes
    axes = remove_redundant_axes(axes)

    # 3. Emergence: Derive new axes from concept differences
    new_axes = detect_new_axes_from(concepts)
    axes.extend(new_axes)

    # 4. Reprojection phase: Map concepts onto updated axes
    for concept in concepts:
        concept.update_projection(axes)

    # 5. Update axis profiles (universality, carrying capacity, etc
    for axis in axes:
        axis.update_metrics(concepts)

    return concepts, axes
```

This structure illustrates the control loop of observation, transformation, reduction, and emergence. With each `tick()`, the semantic space evolves—either through internal reconfiguration or by the addition of new concepts.

Note: The order of steps in the `tick()` cycle is deliberately chosen: First, meta-axes act on the axis structure, then redundant or weak axes are removed, before new axes can emerge from concept differences. This sequence ensures that the system dynamics remain controlled and adaptive.

The function `detect_new_axes_from()` can, for example, be implemented as a heuristic rule based on maximal semantic difference between concepts. Meta-axes such as "semantic gravitation" could also have a direct influence here, for example by locally dampening or amplifying the relevance of new axes.

Important: This representation is deliberately abstracted. In a real simulation (see section 4.5), additional hierarchies, levels, and storage concepts are used.

4.4 Object-Oriented Model Structure

4.4.1 Purpose of the Architecture

The object-oriented structure of the Transition Space follows systemic principles such as modularity, extensibility, and hierarchical recursion. The goal is to enable a dynamic, multi-layered model of meaning that remains manageable both algorithmically and theoretically.

The class structure reflects the functional roles of the framework and enables a clear separation between content representation (concepts, axes) and controlling structure (meta-rules, space logic).

4.4.2 Core Classes

Class	Function
`Concept`	Single concept with attributes and projections
`Axis`	Semantic dimension with type, weighting, universality
`MetaRule`	Implementation of meta-axes through rules (e.g., emergence logic)
`Level`	Encapsulation of concepts and axes on a system level
`Space`	The entire transition space including ticks and meta-levels
`Simulation` (optional)	Control, initialization, and timing

4.4.3 Understanding Meta-Axes in Practice

Meta-axes are structural control variables that act on other axes or axis systems. They are not axes in the classical sense, but higher-level operators on the meta-level of the system. Their function is comparable to control forces in physical space or metacognitive processes in self-referential models.

Typical tasks of meta-axes:

- **Emergence control**: Triggering new axes when semantic differences consistently occur
- **Reduction logic**: Removing redundant or weak axes
- **Axis coupling**: Merging strongly correlated axes
- **Gravitation control**: Local change of time velocity or axis density

In code, meta-axes are realized by the `MetaRule` class. A `MetaRule` contains:

- a name,
- a target structure (e.g., level or axis),
- and a transformation function `apply()`.

Example of a concrete MetaRule: A MetaRule could be: *"Eliminate all axes whose carrying capacity falls below a certain threshold."* In the implementation, this rule would regularly check how strongly each axis contributes to the differentiation of concepts. If an axis is found to be weakly differentiating, it is removed—the space thus becomes more efficient and semantically focused.

4.4.4 Pseudocode for Each Class (optional)

In the following examples, each of the above classes can be shown in pseudocode to better illustrate the interplay. The intention is **didactic**, not syntactically complete:

Example Pseudocode Implementation of the Core Classes

```
class Concept:
    """
    A concept is a point in the meaning space, defined by values on
    """
    def __init__(self, axes, values, **attributes):
        # axes: list of axis objects
        # values: dict {axis_name: value}
        self.axes = {axis.name: axis for axis in axes}
        self.values = values
        self.attributes = attributes  # e.g., {'importance': 5}

    def get_value(self, axis_name):
        return self.values.get(axis_name)

    def set_value(self, axis_name, value):
        self.values[axis_name] = value

    def projection(self, axis_name):
        return self.get_value(axis_name)


class Axis:
    """
    An axis describes a semantic dimension (e.g., color, size, pola
    """
    def __init__(self, name, value_range, **attributes):
        self.name = name
        self.value_range = value_range  # e.g., (min, max) or list
        self.attributes = attributes

    def get_value_range(self):
        return self.value_range

    def set_value_range(self, new_range):
        self.value_range = new_range
```

```python
class MetaRule:
    """
    Meta-rules act on levels and control their development (e.g., e
    """
    def __init__(self, name, target_level, description=None, **para
        self.name = name
        self.target_level = target_level
        self.description = description or ""
        self.parameters = parameters

    def apply(self, t):
        # Transformation of the target level at time step t
        pass  # To be implemented in subclasses


class Level:
    """
    A level encapsulates axes, concepts, and meta-rules on a system
    """
    def __init__(self, name, axes=None, concepts=None, meta_rules=N
        self.name = name
        self.axes = axes if axes is not None else []
        self.concepts = concepts if concepts is not None else []
        self.meta_rules = meta_rules if meta_rules is not None else
        self.attributes = attributes

    def add_axis(self, axis):
        self.axes.append(axis)

    def add_concept(self, concept):
        self.concepts.append(concept)

    def add_meta_rule(self, rule):
        self.meta_rules.append(rule)


class Space:
    """
    The entire transition space: contains all levels and manages ti
    """
    def __init__(self, levels=None, t=0, **attributes):
        self.levels = levels if levels is not None else []
        self.t = t
        self.attributes = attributes

    def add_level(self, level):
        self.levels.append(level)

    def step(self):
        self.t += 1  # Increase time step
```

```python
class Simulation:
    """
    Orchestrates the application of meta-rules and controls the sim
    """
    def __init__(self, space, max_steps=100, verbose=False):
        self.space = space
        self.max_steps = max_steps
        self.verbose = verbose
        self.running = False

    def run(self):
        self.running = True
        for step in range(self.max_steps):
            if not self.running:
                break
            for level in self.space.levels:
                for rule in level.meta_rules:
                    rule.apply(self.space.t)
            self.space.step()
```

5. Emergence, Self-Structuring & Reflection

This chapter summarizes the central system principles of the Semantic Transition Space, which have so far been applied implicitly in examples, code, and architecture—but are now to be described explicitly and distinguished from one another. They do not concern the external structure of the model, but its internal logic: How does order arise, how is it maintained, and how does it change through itself?

5.1 Emergence

Emergence refers to the spontaneous appearance of new structures that are not explicitly predefined. In the Transition Space, this means:

- new axes do not arise through external definitions,
- but from **semantic differences** between concepts,
- governed by **meta-rules** that observe similarity density and distinguishability.

An emergent element is more than the sum of its parts. The "color" axis from section 4.2 does not arise because "red" or "blue" are important individually—but because their **juxtaposition** justifies a new semantic dimension.

5.2 Self-Structuring

A system is self-structuring when it actively modifies its own organizational logic—without external intervention.
In the Transition Space, self-structuring occurs through:

- the effect of meta-axes on the structure of the axis level,
- recursive application of the `tick()` function,
- automatic reduction, emergence, and strengthening of axes.

Meta-axes (see section 4.4.3) are the operative agents of self-structuring: They act on the semantic framework, not its contents, and define the system's internal development logic.

Importantly: **There is no overarching designer** who specifies the structure. Everything arises from rules encoded within the system itself—especially in the meta-levels. Thus, even the **logic of the rules themselves** becomes changeable.

5.3 Reflection

Reflection in the Semantic Transition Space means:

- one level **observes** another,
- and **acts back on it** through rule-based feedback,
- taking into account **its own semantic criteria**.

Technically, reflection is carried out by the **meta-levels**, which can observe, evaluate, and transform concepts, axes, and even other meta-rules.

Example:

- If meta-rule A detects "strong redundancy" and meta-rule B detects "high carrying capacity," a third meta-rule C can decide whether an axis should be kept, modified, or removed.
- This creates a **rule-based dialogue** between levels—a structured reflection process.

5.4 Comparison Table: Dynamics Concepts in the Transition Space

Principle	Description	Example from the Model
Emergence	New arises from interaction, not addition	Axis arises from difference "red/blue"
Self-Structuring	System changes without external control	`tick()` performs axis reduction
Reflection	Higher levels observe and regulate lower levels	Meta-rules analyze axis density

These terms mark the boundary between a **reactive data model** and a **reflexive meaning system**.
"The Transition Space is not a configurable schema—it is a self-organizing field of meaning."

6. Outlook & Application Fields

The Semantic Transition Space is designed as a universal model that goes far beyond semantic categorization or formal ontologies. Its recursive, self-structuring architecture makes it adaptable to a wide range of fields—from theoretical physics to the modeling of emotions, markets, or learning agent systems. The following overview presents exemplary application fields and further-reaching visions:

6.1 Physics & Spacetime Simulation

The Transition Space enables a structured simulation of meaning, curvature, time dynamics, and gravitation. In connection with physical theories—especially string theory, quantum gravity, or information geometry—novel concepts for modeling singularities, field relationships, or emergent dimensions could be developed. A true "semantic spacetime" would thus be realizable not only as an analogy, but as its own conceptual field.

6.2 Emotional Systems & Feeling Databases

A Transition Space on the affective level could model feelings, motives, and tensions between psychological states as concepts with axis structure. This would make not only emotional states, but also their transformations, blockages, or developmental trajectories visible. The construction of a "feeling database" would not be based on symbolic representation, but on relational meaning structure.

6.3 Language Models & Semantic AI

The Transition Space can serve as a semantic basis for language processing: meanings are not mapped via token statistics, but via projective structures. This opens up the possibility of developing **advanced language models with conceptual depth**—for example, by combining GPT-like systems with dynamic, self-reflective meaning levels. The boundary between LLM and semantic memory would thus become permeable.

6.4 Economics, Financial Markets & System Dynamics

Economic systems can also be modeled as dynamic transition spaces: Concepts = market participants or strategies, Axes = evaluation dimensions (risk, liquidity, behavior). Meta-axes can be used to explicitly describe regulatory principles or market mechanisms and to simulate their effects. This would make it possible, for example, not only to analyze stock market mechanisms, but also to make them **more transparently traceable** and controllable.

6.5 Autonomous Knowledge Systems & AGI Vision

When the Transition Space is combined with generative AI, the possibility arises that **the**

system itself generates semantic hypotheses—through emergence, recombination, and reflection. This creates a framework in which not only new categories are recognized, but also new axes of thought can be generated. This could be a building block on the way to **autonomous, semantically grounded artificial intelligence**—a possible next step toward AGI.

6.6 Outlook: Semantic Science

In the long term, the Transition Space could serve as the basis for a **semantic science**—a discipline that does not merely connect individual disciplines, but reveals their structural commonalities: emergence, meaning, relation, dynamics, reflection. This would create a framework for thinking that not only integrates knowledge, but also makes its **inner semantics systematically mappable**.

7. Literature / Sources of Inspiration

This whitepaper does not stand in a direct disciplinary line, but is inspired interdisciplinarily by physics, mathematics, systems theory, cognitive science, and AI research. The following works, questions, and impulses have significantly contributed to the theoretical foundation or conceptual orientation:

Systems Theory & Emergence

- Ludwig von Bertalanffy – *General System Theory* (1968)
- Niklas Luhmann – *Soziale Systeme* (1984)
- Heinz von Foerster – *Understanding Understanding* (2003)
- Hermann Haken – *Synergetik* (1977)

Mathematics & Structural Theory

- Eilenberg & Mac Lane – *General Theory of Natural Equivalences* (1945) [Category Theory]
- David Spivak – *Category Theory for the Sciences* (2014)
- Rudy Rucker – *Infinity and the Mind* (1982)

Physics & Spacetime Models

- Albert Einstein – *Relativity Theory* (1916)
- Carlo Rovelli – *The Order of Time* (2018)
- Brian Greene – *The Elegant Universe* (1999)
- Lee Smolin – *Three Roads to Quantum Gravity* (2001)

Language, Semantics & Cognition

- George Lakoff – *Women, Fire, and Dangerous Things* (1987)

- Gilles Fauconnier & Mark Turner – *The Way We Think* (2002)
- Douglas Hofstadter – *Gödel, Escher, Bach* (1979)
- Own reflections on the question: *What is semantics really—and how does it arise in the brain?*
- Meta-reflection on proximity, structure, and semantic pattern formation in thinking
- Dialogues with GPT-4 as an interactive reflection space (2023–2025)

AI, Agent Systems & AGI

- Norbert Wiener – *Cybernetics* (1948)
- Judea Pearl – *Causality* (2000)
- Yann LeCun – *A Path Towards Autonomous Machine Intelligence* (2022)
- DeepMind – *Open-Endedness in Intelligence* (2021, Research paper)

Own Experiments & Prototypes

- Python-based architecture (2024–2025)
- Dynamic meaning space with meta-rule engine
- Mini-simulation: Emergence and axis formation

This list is deliberately kept open and is not intended as an academic citation directory, but as a **context map of influences** that contributed to the development of the Transition Space.

8. License & Usage Guidelines