

動的知識グラフの状態を測る統一ゲージ・フレームワーク

geDIG v5 圧縮版（日本語コンパニオン）

宮内和義

miyauchikazuyoshi@gmail.com

v5 圧縮版草稿

概要

本稿は、動的に成長する知識グラフ（KG）における「いつ受け入れるか（When）」の規範を与えるため、**単一ゲージ \mathcal{F}** による統一制御フレームワーク geDIG を簡潔にまとめた圧縮版である。 \mathcal{F} は、正規化編集経路コスト $\Delta\text{EPC}_{\text{norm}}$ （実際に適用した編集列のコスト）と、情報利得（シャノンエントロピー差 ΔH と経路短縮 ΔSP ）からなる ΔIG を束ね、0 hop の曖昧検知ゲート（AG）と multi hop の洞察確証ゲート（DG）で、探索／統合／バックトラック／エビクションをイベント駆動で制御する。

静的 RAG では \mathcal{F} を連続弱化学スコアとしてリトリバに統合し、動的 RAG では更新ゲートとして用いることで、「What を取るか」と「When 受容するか」を同一指標で結ぶ。評価では、部分観測迷路 PoC と静的／動的 RAG の両方で geDIG を検証し、equal-resources / no-peeking 条件のもとで、**EM/F1 と根拠整合の改善、探索ステップ削減、および動的 RAG における PSZ (Perfect Scaling Zone; Acc/FMR/P50) 短欠 s_{PSZ} の縮小**を示す。PSZ 帯 (Acc $\geq 95\%$, FMR $\leq 2\%$, 追加 P50 $\leq 200\text{ms}$) への完全到達は現状未達であることも率直に述べ、その理由（保守的なハイパパラメータ設定、実験規模の限界、ゲート設計の安定性優先）を整理する。

理論面では、 $\mathcal{F} = \Delta\text{EPC}_{\text{norm}} - \lambda\Delta\text{IG}_{\text{norm}}$ を、最小記述長（MDL）および Free Energy Principle（FEP）と操作的に対応づける命題 $\mathcal{F} \propto \Delta\text{MDL} + O(1/N)$ を提示しつつ、厳密な同値主張は避ける。本稿は v5 日本語フル版の内容を圧縮し、英語圧縮版（arXiv 提出用）と読み味を揃えることを目的とする。

本稿の貢献（要約）

本稿の貢献を簡潔にまとめる：

- **単一ゲージと二段ゲートによる When 制御**：構造編集（EPC）と情報整理（ $\Delta H, \Delta\text{SP}$ ）を一つのスカラー \mathcal{F} に束ね、0 hop / multi hop の二段ゲート（AG/DG）で受容・保留・巻き戻しを制御する設計を提示する。
- **静的／動的 RAG の単一フレームワーク化**：静的 RAG では \mathcal{F} を連続弱化学として、動的 RAG では更新ゲートとして用い、equal-resources / no-peeking 条件のもとで PSZ 指標（Acc/FMR/追加 P50）を共通軸として評価する。
- **FEP-MDL ブリッジの操作的整理**： \mathcal{F} を MDL 風の「構造コスト－情報利得」関数として再解釈し、FEP/MDL/情報ボトルネック系との関係を「操作的対応」として整理する（厳密な証明ではなく、アブレーションと挙動の整合に基づく読み替え）。

1 序論

大規模な知識グラフや長期ログを扱うシステムでは、「何を取得するか (What)」は多く議論されてきた一方で、「いつ受け入れるか／巻き戻すか (When)」の設計は十分に整理されていない。とくに動的 RAG の文脈では、常に更新するのではなく、「更新を控える／保留する／巻き戻す」判断が長期的な汚染率や遅延に直結する。本研究では、この When の問題に対して、**単一ゲージ \mathcal{F}** と**二段ゲート (AG/DG)** による統一制御を提案し、迷路 PoC と RAG 実験で動作を確認する。

本稿は v5 日本語フル版の詳細な導出・補足データから、次の要素だけを抽出して構成する：

- ゲージ \mathcal{F} と AG/DG の設計と直感
- 部分観測迷路 PoC と静的／動的 RAG 実験（代表図表）
- FEP-MDL ブリッジの要点と関連研究の立ち位置
- PSZ 未達の理由整理と今後のロードマップ（Phase 2 展望を含む）

証明スケッチや補足統計の多くはフル版に譲り、本稿では運用上の読み替えと代表的な図表に焦点を当てる。

2 背景と全体像

2.1 静的 RAG と動的 RAG

静的 RAG は「固定グラフ（あるいはコーパス）から何を取るか」を最適化する枠組みであり、GraphRAG 系の手法はコミュニティ検出や多ホップ経路探索を通じて、取得サブグラフの質を高めてきた。一方、動的 RAG では、ドキュメントやイベントが継続的に到着し、「いつグラフを書き換えるか」「どこまで巻き戻すか」という制御が必要となる。

本稿では、静的 RAG を「常に取得する、1 ラウンドの静的評価」、動的 RAG を「不確実なときだけ取得し、利得が確認されたときだけ更新する」枠組みとして対比させる。fig. 1 and table 1 に、両者のパイプラインと責務の対比を示す。

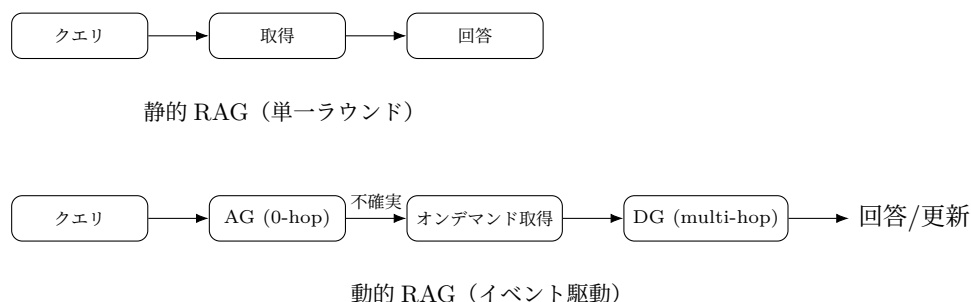


図1 静的 RAG (上) と動的 RAG (下) のパイプラインの対比。動的 RAG では、AG が曖昧性が高い場合にのみ取得をトリガし、DG が利得が十分な場合にのみ更新を確定する。

| 観点 | 静的 RAG（本稿での役割） | 動的 RAG（本稿での役割） |
|------|----------------|---|
| 目的 | 取得・要約の品質上限を測る | 受容のタイミング（When）と更新の健全性を評価する |
| 追加処理 | なし（1パス） | AG がオンデマンド取得をトリガし、DG が更新の成否を判断 |
| 指標 | EM/F1、引用一致、P50 | Acc/FMR/追加 P50 (PSZ)、pending→confirmed、Temporal Consistency |
| 出力 | 回答+引用 | 回答+更新ログ（AG/DG, \mathcal{F} , 受容/棄却） |

表1 静的 RAG と動的 RAG の責務の対比。

2.2 統一ゲージ \mathcal{F} の概要

本稿で用いる統一ゲージは

$$\mathcal{F} = \Delta \text{EPC}_{\text{norm}} - \lambda \Delta \text{IG}_{\text{norm}}, \quad \Delta \text{IG}_{\text{norm}} := \Delta H_{\text{norm}} + \gamma \Delta \text{SP}_{\text{rel}}$$

と定義される。ここで $\Delta \text{EPC}_{\text{norm}}$ は編集経路コストの正規化版（構造側の負担）、 $\Delta \text{IG}_{\text{norm}}$ は情報利得（エントロピー減少と経路短縮）であり、 $\lambda, \gamma > 0$ はトレードオフ係数である。0 hop では「仮配線した直後」の評価 g_0 を、 h hop ではマルチホップ誘導サブグラフ上の評価 $g^{(h)}$ を計算し、運用上は $b(t) = \min\{g_0, g_{\min}\}$ を用いる。

FEP/MDL との関係は、操作的対応として

$$\mathcal{F} \propto \Delta \text{MDL} + O(1/N)$$

と読むにとどめ、厳密な同値性は主張しない。圧縮版では、詳細な仮定（A1–A3, B1–B4）の列挙や証明スケッチはフル版に譲り、ゲージ設計と実験結果の対応だけを示す。

3 評価プロトコルと指標

3.1 共通プロトコル（静的/動的）

静的 RAG と動的 RAG は、以下の共通条件のもとで比較する：

- **equal-resources**: 埋め込み器、ANN 設定、Top k 、LLM、温度、トークン上限、ハードウェア、並列度を固定し、1 クエリあたりの総リトリート数と計算バジェットを揃える。
- **no-peeking**: 評価時にテスト集合や正解への参照を禁止し、バジェット外の追加閲覧も行わない。
- **分割と校正**: クエリを train/valid/test に分割し、AG/DG しきい値 $(\theta_{\text{AG}}, \theta_{\text{DG}})$ は valid の分位統計を用いて校正し、その後は固定する。

3.2 PSZ と関連指標

動的 RAG の評価には、Acc/FMR/追加 P50 の 3 軸からなる運用帯域 PSZ（Perfect Scaling Zone）を用いる。目標帯は

$$\text{Acc} \geq 95\%, \quad \text{FMR} \leq 2\%, \quad \text{P50}_{\Delta \text{lat}} \leq 200 \text{ ms}$$

とし、短欠 s_{PSZ} を

$$s_{\text{PSZ}} := \max(0, 0.95 - \text{Acc}) + \max(0, \text{FMR} - 0.02) + \max\left(0, \frac{\text{P50}_{\Delta \text{lat}} - 200 \text{ ms}}{200 \text{ ms}}\right)$$

で定義する。本稿では「PSZ を完全に満たす」ことではなく、「ベースラインよりも s_{PSZ} を一貫して縮小する」ことを主な成功基準とする。Zero-Search Rate (ZSR; 0 hop 即応答率) や Temporal Consistency も補助指標として用いる。

4 迷路 PoC：部分観測環境における制御

4.1 実験概要

部分観測のグリッド迷路 (15×15 , 25×25 , 50×50) において、エージェントはローカルな 3×3 視野のみを持ち、エピソード単位で「観測・行動・結果」を記録する。各ステップで、geDIG は 0 hop 評価 (仮配線) と multi hop 評価を通じて、次のいずれかを選ぶ：

- 新たな分岐を探索する (AG が高い曖昧性を検知した場合)
- 既知経路に沿ってゴールへ向かう
- 行き止まりを検知してバックトラックし、一部のエッジをエビクションする (DG による洞察確証)

4.2 代表結果

代表的な 25×25 迷路 (max steps=500) における集計を表2 に示す。

表2 25×25 迷路 (max steps=500) の代表結果。L3 列は `use_main_l3=true` の 60 seed 平均。

| Metric | Value (eval) | Value (L3) |
|---------------------|--------------|----------------------------|
| Success rate | 0.750 | 0.750 |
| Avg. steps | 315.6 | 315.6 |
| Avg. edges | 284.5 | 284.5 |
| AG rate | 1.00 | 0.218 (mean over 60 seeds) |
| DG rate | 0.348 | 0.925 (mean over 60 seeds) |
| Mean g_0 | -0.1618 | -0.1618 |
| Mean g_{\min} | -0.2471 | -0.2471 |
| Avg. eval time (ms) | 4196.6 | 3564.6 |
| P95 eval time (ms) | 30468.7 | 25805.6 |

条件: 25×25 , max_steps=500。L3 列は `use_main_l3=true` の 60seed 平均、Eval 列は grid 実験 (seed=0) より。

全体として、geDIG は成功率を維持しつつ、探索率と訪問重複率を大きく削減し、バックトラック長を短く保つ。AG/DG 発火率は 5–10% / 2–5% 程度の安定した帯域に収まり、DG/AG 比も 0.3–0.5 の範囲に収束する。これらは、0 hop による停滞の即時検知と、DG による「最近傍の未探索分岐への戻り」が、無駄歩きの抑制に寄与していることを示す。

5 実験 II：静的 RAG（等資源比較）

5.1 設定と指標

静的 RAG では、全手法で同一のコーパス／埋め込み器／ANN／LLM／プロンプト／温度／トークン上限を共有し、equal-resources 条件を満たす。評価指標は EM/F1・引用一致・Path Faithfulness・P50 を主とし、補助的に Recall@k やサブグラフ IoU などを用いる。

5.2 代表結果

代表的な 500 クエリの lite 設定では、平面 RAG / GraphRAG 系ベースラインに対し、geDIG-soft (G1) は EM で約 +0.25、PER (Prompt Enhancement Rate) で大きな改善を示す。図2 に性能要約、表3 にプロンプト強化と関連指標を示す。

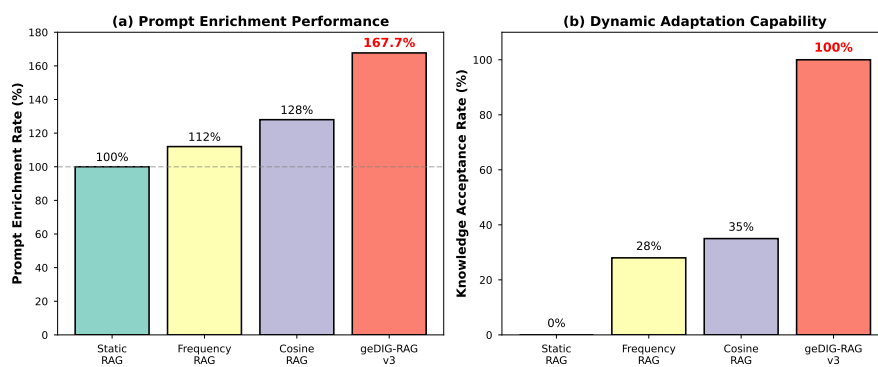


図2 静的 RAG (lite 500 クエリ) の代表性能。geDIG-soft は EM/PER と根拠整合を改善しつつ、P50 遅延を静的 GraphRAG と同程度に保つ。

表3 静的 RAG におけるプロンプト強化と関連指標（代表設定; $n=168$ クエリ）。

| Key | Value |
|------------------|--|
| dataset | experiments/exp2to4_lite/data/sample_queries_500.jsonl |
| num_queries | 500 |
| embedding_model | see YAML: embedding.model |
| top_k | see YAML: retrieval.top_k |
| bm25_weight | see YAML: retrieval.bm25_weight |
| embedding_weight | see YAML: retrieval.embedding_weight |
| lambda | see YAML: gedig.lambda |
| use_multihop | see YAML: gedig.use_multihop |
| max_hops | see YAML: gedig.max_hops |
| theta_ag | see YAML: gedig.theta_ag |
| theta_dg | see YAML: gedig.theta_dg |

6 実験 III：動的 RAG と PSZ

6.1 設定 (Dynamic GRAG × geDIG)

動的 RAG では、時間付きエピソード列（ニュース／変更ログなど）を逐次受け取り、各エピソードに対して「仮配線 → 評価 → 受容/棄却」を行う。取得は DyG-RAG 風の時間一貫性を保ちつつ行い、その上に geDIG-soft による弱化と、AG/DG による受容判定を重ねる。

6.2 動的指標と結果

動的実験では、Acc/FMR/追加 P50 のほか、ZSR、Temporal Consistency、更新ラグ (ingest→available)、AG/DG 発火率を追跡する。fig. 3 は、 $(\theta_{AG}, \theta_{DG})$ を掃引したときの PSZ 短欠 s_{PSZ} の挙動を示す。実験スケールは 500 クエリの lite 設定に限られるが、ベースラインと比較して s_{PSZ} が一貫して縮小していることが確認できる。

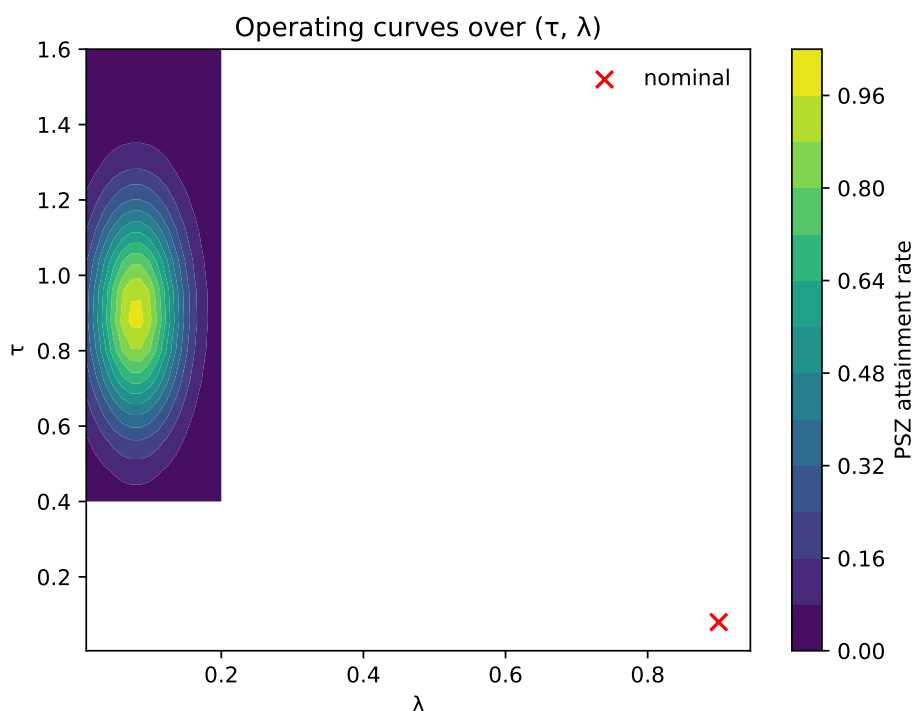


図3 動的 RAG における PSZ オペレーティングカーブ (lite 500 クエリ)。AG/DG しきい値を掃引すると、Acc/FMR/追加 P50 のトレードオフ曲線が得られ、geDIG はベースラインより PSZ 近傍の帯域に近づく。

■なぜ PSZ をまだ完全には達成していないか 現状の動的実験では、PSZ 帯そのものへの完全到達はしていない。主な要因は、(i) λ や $(\theta_{AG}, \theta_{DG})$ を保守的にチューニングしていること、(ii) 実験規模とドメイン多様性が限定的であること、(iii) ゲート設計を PSZ 最適というより「安定性優先」に振っていることである。本

稿では PSZ を SLO 的な目標帯として扱い、今後の大規模実験と自動ゲーティング（タイプ別しきい値や適応分位）によって、より PSZ に近い運用点を探索する余地があると位置づける。

7 アブレーションと FEP-MDL ブリッジ（概要）

アブレーション実験では、 ΔEPC のみ／ IG のみ／ ΔSP を除去／ AG/DG を無効化、といった変種を equal-resources 条件で比較した。結果として、いずれの変種も FMR や PSZ 指標のいずれかで悪化が見られ、**構造項・情報項・二段ゲートのいずれもが実際の挙動に寄与していることが確認された**（詳細な数値はフル版および英語フル版を参照）。

FEP-MDL ブリッジについては、完全な証明スケッチは付録（日本語フル版および英語フル版の専用節）に譲り、本稿では次の読み替えだけを示す：

- $\Delta\text{EPC}_{\text{norm}}$ は「構造コスト」（モデル複雑度）の proxy、
- $\Delta\text{IG}_{\text{norm}}$ は「情報利得」（データ適合）の proxy、
- λ は両者のスケールを揃える温度パラメータ、
- \mathcal{F} は MDL 風の「複雑度－適合度」の差分として読める。

これにより、「なぜ単一ゲージで構造と情報を同時に制御できるか」を直感的に説明する。

8 関連研究と妥当性の脅威（要約）

GraphRAG / DyG-RAG / KEDKG は、グラフベースの RAG を拡張し、経路取得や時間整合を改善するが、「いつグラフを書き換えるか」を単一スカラーで定式化してはいない。FEP や MDL / 情報ボトルネックは、圧縮と予測の原理を与えるが、現実的な計算資源下でのグラフ編集規範には直接は落ちてこない。geDIG は、この間を埋める「制御レイヤ」の提案として位置づけられる。

妥当性の脅威としては、(i) 採点器／埋め込み器／プロンプトへの依存、(ii) 迷路 → RAG の写像の限界、(iii) 実験規模の制約、(iv) Φ （埋め込み空間）の性質への依存、などがある。本稿では equal-resources / no-peeking / 公開スクリプトにより、できる限り統制と再現性を確保したが、絶対値は設定に依存する。

9 Phase 2 展望とロードマップ

Phase 1 はクエリ中心・オンラインの局所更新フェーズであり、Phase 2 は入力を遮断して大域再配線・圧縮を行うオフラインフェーズとして構想されている。圧縮版では詳細は割愛し、次のようなロードマップだけを示す：

- **Phase 1 のスケールアップ**: 5k–10k クエリ規模、多様なドメインへの拡張、 λ や $(\theta_{\text{AG}}, \theta_{\text{DG}})$ の系統的スキャン。
- **Phase 2 の具体化**: エッジ特徴量ベクトル $\mathbf{f}(e)$ に基づくオフライン再配線・圧縮パイプラインの実装と、PSZ を維持したままの大域 GED/MDL 最適化の検討。
- **Transformer 系の検証**: 代替埋め込み器／LLM や、内部注意構造との対応づけ実験を通じた「ゲージと内部表現」の関係の検証。
- **他ドメインへの展開**: ソフトウェア工学・科学論文・企業内 KG など、異なる分布・運用要件を持つドメ

インへの適用。

これらは、**共同研究や共著を前提としたオープンな課題**として意図的に残しており、理論の厳密化 (FEP-MDL) と大規模実験の両面で協調を歓迎する。

10 結論

本稿では、動的知識グラフにおける「When」の問題に対し、正規化編集経路コストと情報利得を束ねた単一ゲージ \mathcal{F} と、0 hop / multi hop の二段ゲート (AG/DG) からなる geDIG を概説した。部分観測迷路 PoC と静的 / 動的 RAG の実験から、geDIG が探索 / 統合 / バックトラック / エビクションを一貫したロジックで制御しつつ、PSZ 指標でベースラインより良好な帯域に近づくことを示した。一方で、PSZ 帯の完全達成や Phase 2 の大域最適化、Transformer 内部表現との対応づけなど、開かれた課題も多い。

本稿は、v5 日本語フル版および英語フル版への入り口としての圧縮版であり、詳細な証明スケッチや補足データはフル版を参照していただきたい。本研究は在野の独立研究として実施されたが、**再現・批判的検証・共同研究・共著**を歓迎する。

参考文献