SCORER, SEARCH, OR INTERFACE? DECISION BOUNDARIES FOR SYMBOLIC PLANNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Maintained symbolic planners evolve around interfaces that can remain fixed while learned scoring and deployment-time search change. This setting calls for a comparison object that separates the interface contract, the shared planning wrapper, the scorer, and the controller budget. We formulate such an object and introduce RAVELWM, an action-conditioned dynamic-graph scorer built from interleaved graph propagation and selective per-node state-space updates. The design carries node state across time, handles graph birth and disappearance explicitly, predicts next-state attributes and candidate edges, and exposes per-constraint risk to a shared plan-ranking rule. We also specify a protocol-local structured-versus-raw fallback threshold and a condition registry that makes an eventual scorer, controller, or interface comparison auditable. The resulting framework provides a precise basis for studying maintenance choices in long-horizon symbolic planning.

1 INTRODUCTION

Accessibility trees, simulator APIs, and operational schemas often become compatibility boundaries before the learned components surrounding them reach a stable design. A planner maintained under such a boundary has several levers: improve a learned rollout scorer, allocate recurring resources to a controller, or revise the interface. Meaningful comparisons require these levers to be named separately. Otherwise, a change in graph extraction, proposal distribution, caching policy, constraint implementation, or timeout behavior can be mistaken for a scorer or search effect.

This paper develops a maintained-planner comparison framework and instantiates its scorer component with RAVELWM. The framework represents the symbolic state as a dynamic attributed graph, fixes the interface and execution contract inside a condition registry, and assigns every evaluation quantity an explicit outcome unit and cost scope. Its central decision object is a structured-versus-raw fallback threshold: a protocol-specific comparison point between a structured scorer and a raw-observation fallback under one declared controller and budget. This definition keeps a threshold attached to the interventions it actually compares.

RAVELWM uses RAVELSCAN layers, in which graph message passing feeds a contractive selective per-node state-space update at every layer. The construction is designed for action-conditioned rollouts with changing node sets: new nodes receive semantic initial state, disappeared nodes are masked from temporal state, and edge predictions use a deterministic action-conditioned candidate set. A Bernoulli risk head supplies per-constraint risk estimates to the plan-ranking rule without conflating them with a categorical distribution.

Our contributions are:

- a formal comparison contract for maintained symbolic planners, separating the interface, shared wrapper, scorer, controller, outcome unit, and cost ledger;
- a protocol-local structured-versus-raw fallback threshold that precisely states the comparison to which a boundary estimate applies;
- RAVELSCAN, an interleaved dynamic-graph scorer with explicit node-lifecycle handling, contractive selective state updates, action-conditioned sparse decoding, and per-constraint risk prediction; and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- a reporting registry that makes condition identity, sampling hierarchy, resource accounting, and external-version provenance auditable before numerical outcomes are interpreted.

2 Related Work

World models support planning by predicting consequences before action execution (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2023). Dynamic graph models represent evolving relational state (Li et al., 2016; Trivedi et al., 2019; Xu et al., 2020; Pareja et al., 2020); selective state-space models provide efficient sequence memory (Gu et al., 2022; Gu & Dao, 2023; Dao & Gu, 2024). RavelScan combines these ingredients for an action-conditioned graph rollout with node birth, node disappearance, and typed edges.

Planning systems also vary their proposal mechanism, search procedure, value estimator, memory, and execution wrapper (Yao et al., 2023b; Shinn et al., 2023; Yao et al., 2023a; Hao et al., 2023; Zhou et al., 2024a). These are meaningful system choices, but they are distinct interventions. Our comparison contract makes the distinction explicit: a scorer swap changes the learned rollout model while a controller experiment changes the search procedure under a separately registered setting. This framing complements agent benchmarks such as ALFWorld, WebArena, ScienceWorld, and VirtualHome (Shridhar et al., 2021; Zhou et al., 2024b; Wang et al., 2022; Puig et al., 2018).

3 Maintained-Planner Comparison Contract

At step t, an agent receives observation o_t and executes action $a_t \in \mathcal{A}$. A graph builder g maps the observation history to a dynamic attributed graph

$$G_t = (V_t, E_t, X_t^V, X_t^E, \mathcal{T}_t),$$

where $E_t \subseteq V_t \times \mathcal{R}_b \times V_t$ contains typed relations from a benchmark-specific vocabulary $\mathcal{R}_b$. Nodes have stable identifiers while they remain present, and $\mathcal{T}_t$ contains textual metadata. A benchmark-specific constraint library is

$$\mathcal{C}_b = \{c_{b,m}\}_{m=1}^{M_b}, \qquad c_{b,m}(G_t, a_t) \in \{0, 1\},$$

where M_b is benchmark-specific and $c_{b,m} = 0$ denotes a violation.

3.1 Condition identity

A reported comparison must declare a condition record

$$\kappa = (g, \mathcal{C}_b, \gamma, \pi, K, H, e, \theta, \rho, \omega),$$

where γ is the action-conditioned edge-candidate grammar, π is the proposal distribution, (K, H) is the candidate-and-horizon budget, e is the execution and replanning wrapper, θ identifies the learned scorer, ρ is the controller, and ω contains the task identifiers, training seeds, evaluation seeds, hardware configuration, and timeout rule. A scorer comparison holds $(g, \mathcal{C}_b, \gamma, \pi, K, H, e, \rho, \omega)$ fixed and varies θ; a controller comparison holds the remaining fields fixed and varies ρ. This record is the unit of causal interpretation.

For an episode-level binary outcome $Y \in \{0, 1\}$, let

$$R(\kappa) = \mathbb{E}[Y \mid \kappa].$$

The outcome definition, terminal rule, and denominator belong to ω. If an analysis compares paired episodes, it must retain the matched episode identifiers and state the tie rule; a continuous diagnostic must be named separately from the binary success outcome.

3.2 Fallback threshold and cost ledger

Let $\mathcal{Q} \subseteq \mathbb{R}$ be an ordered, registered interface-quality scale, oriented so that larger values denote higher quality. For $q \in \mathcal{Q}$, a fixed controller ρ, and a declared budget b, define

$$D_{\mathrm{str}}(q, b) = R(\kappa_{\mathrm{str}}(q, b)) - R(\kappa_{\mathrm{raw}}(q, b)), \qquad q_{\mathrm{fb}} = \inf\{q \in \mathcal{Q} : D_{\mathrm{str}}(q, b) \geq 0\}. \tag{1}$$

Here q is a registered interface-quality coordinate, and κ_{str} and κ_{raw} share the declared condition fields and differ only by the registered structured and raw-observation alternatives. Thus q_{fb} is a structured-versus-raw fallback threshold for that registry, not a benchmark-intrinsic constant. A reported threshold additionally requires a predeclared quality grid, corruption allocation, held-out partition, monotone fitting family, interpolation rule, and interval construction. If those choices do not establish a single crossing on the registered support, the comparison is reported as the curve $D_{\mathrm{str}}(q, b)$ rather than as a threshold.

The associated cost ledger records four composable scopes in one declared unit u (wall-clock time, accelerator time, or FLOPs): $C^u_{\mathrm{step}} = C^u_{\mathrm{build}} + C^u_{\mathrm{score}}$, $C^u_{\mathrm{rollout}} = \sum_{h=1}^{H} C^u_{\mathrm{step},h}$, $C^u_{\mathrm{replan}} = C^u_{\mathrm{propose}} + \sum_{k=1}^{K} C^u_{\mathrm{rollout},k}$, and $C^u_{\mathrm{episode}} = C^u_{\mathrm{environment}} + \sum_{r=1}^{R_{\mathrm{ep}}} C^u_{\mathrm{replan},r}$, where R_{ep} is the registered number of replans attributed to the episode. Each entry stores its device, precision, batch size, parallelism, warm-up rule, and timeout treatment. Units are reported as separate ledger columns rather than combined into a scalar, and a budget comparison is valid only when both alternatives reference the same ledger scope and configuration.

3.3 Sampling and uncertainty

Training seed, evaluation seed, and task identity are separate axes. Let $s \in \{1, \dots, S\}$ index training seeds and let $\mathcal{I}_s$ be the complete evaluation units nested in seed s (including each task and evaluation-seed identifier). For a binary outcome, the seed-level mean is $\widehat{R}_s = |\mathcal{I}_s|^{-1} \sum_{i \in \mathcal{I}_s} Y_{s,i}$ and a reported aggregate is $\widehat{R} = \sum_s w_s \widehat{R}_s$, with declared weights $w_s \geq 0$ and $\sum_s w_s = 1$. A paired analysis retains one matched identifier for both alternatives and declares its tie rule separately from this outcome estimator. A cluster bootstrap resamples complete units within each training seed and recomputes the declared aggregate; if the inferential target includes training-seed variation, it also resamples seed-level units. The registry states the estimator, weights, seed treatment, interval construction, and number of resamples. A graph-quality threshold additionally records the quality grid, corruption allocation, monotonicity constraint, fit procedure, and held-out split before its interval is interpreted.

4 Interleaved Dynamic-Graph Scoring

4.1 State update

Let $H_t^{(0)}$ be the action-conditioned node representation at time t. For layers $l = 0, \dots, L-1$, RavelScan first applies graph propagation on the current topology and then updates a per-node temporal state:

$$\widetilde{H}_t^{(l)} = \mathrm{GATConv}(H_t^{(l)}, E_t, Z_t^E), \tag{2}$$

$$\bar{h}_t^{(l+1)}[i] = A_t^{(l+1)} \bar{h}_{t-1}^{(l+1)}[i] + B_t^{(l+1)} \widetilde{H}_t^{(l)}[i], \tag{3}$$

$$H_t^{(l+1)}[i] = C_t^{(l+1)} \bar{h}_t^{(l+1)}[i] + D^{(l+1)} \widetilde{H}_t^{(l)}[i]. \tag{4}$$

Here $\mathrm{GATConv}$ denotes graph-attention propagation (Veličković et al., 2018). The update order in Equations (2) and (3) gives the temporal state access to current-topology messages in the same layer.

The selective transition uses

$$\Delta_t^{(l+1)} = \mathrm{softplus}(W_\Delta^{(l+1)} \widetilde{H}_t^{(l)} + b_\Delta^{(l+1)}), \qquad A_t^{(l+1)} = \exp\left(\Delta_t^{(l+1)} \odot \bar{A}^{(l+1)}\right), \tag{5}$$

with $\bar{A}^{(l+1)} = -\exp(A_{\log}^{(l+1)})$. Since $\Delta_t^{(l+1)} > 0$ and every diagonal entry of $\bar{A}^{(l+1)}$ is negative, each diagonal entry of $A_t^{(l+1)}$ lies strictly between zero and one. This is the contractive diagonal convention used for the selective recurrence (Gu & Dao, 2023).

For a node i born at t, the state supplied on the right-hand side of Equation (3) is the birth initializer $\bar{h}_{t-1}^{(l+1)}[i] = W_{\mathrm{init}} Z_{\mathrm{sem}}[i]$. For a node absent from V_t, the model masks its state and incident messages; no state is carried into a later, unrelated identifier. This convention fixes the state-time semantics at node birth and disappearance.

4.2 ACTION-CONDITIONED DECODING AND RISK

The decoder predicts node attributes and typed edges for the next graph. Let $N_2(G_t) \subseteq V_t \times V_t$ be the ordered node pairs connected by a two-hop walk, and let $\mathcal{R}_\gamma(u, v, G_t, a_t) \subseteq \mathcal{R}_b$ be the relation types permitted by the registered candidate grammar, with $|\mathcal{R}_\gamma| \leq r_{\max}$. Rather than score all node-pair–relation triples, it evaluates the deterministic candidate set

$$\mathcal{E}_{\text{cand}}(G_t, a_t) = E_t \cup \{(u, r, v) : (u, v) \in N_2(G_t),\ r \in \mathcal{R}_\gamma(u, v, G_t, a_t)\} \cup \mathcal{C}_a(G_t, a_t),$$

where $\mathcal{C}_a(G_t, a_t) \subseteq V_t \times \mathcal{R}_b \times V_t$ contains action-referenced triples. For every (u, r, v) in this set, the decoder produces

$$\hat{e}_{u,r,v} = \sigma\Big(\text{MLP}_E([H_t^{(L)}[u]; Z_r; H_t^{(L)}[v]])\Big).$$

The worst-case number of typed two-hop candidates is $O(|V_t| d_{\max}^2 r_{\max})$; consequently the worst-case decoding cost is

$$O\big(|E_t| + |V_t| d_{\max}^2 r_{\max} + |\mathcal{C}_a(G_t, a_t)|\big).$$

The backbone is linear in rollout horizon for fixed graph operations, while this decoder bound makes both the degree and relation-grammar dependence explicit.

For each $c_{b,m}$, the risk head predicts

$$\hat{c}_{b,m}(G_t, a_t) \approx \Pr[c_{b,m}(G_t, a_t) = 0].$$

The vector $\hat{\mathbf{c}} \in (0, 1)^{M_b}$ contains per-constraint Bernoulli probabilities, not a joint categorical distribution. The planner can use the worst-constraint aggregate $\hat{c}(G, a) = \max_m \hat{c}_{b,m}(G, a)$ and rank a candidate sequence by

$$S_k = \sum_{h=1}^{H} \Big[w_{\text{goal}} r_{\text{goal}}(\hat{G}_{t+h}) - w_{\text{cons}} \hat{c}(\hat{G}_{t+h-1}, a_{t+h-1}^{(k)}) \Big]. \tag{6}$$

The registry includes the coefficients, thresholding rule, and all replanning triggers used with this score.

4.3 TRAINING OBJECTIVE

Let $\mathcal{L}_{\text{pred}}$ be the normalized next-state loss, $\mathcal{L}_{\text{align}}$ a semantic alignment loss, and $\mathcal{L}_{\text{cons}}$ the sum of per-constraint binary cross-entropies. Let ξ_{cf} be the registered distribution over feasible triples (G, a, a') of semantically related actions. For $p, q \in (0, 1)$, define the Bernoulli divergence

$$d_{\text{Ber}}(p \| q) = p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q}.$$

The counterfactual consistency term is

$$\mathcal{L}_{\text{cf}} = \mathbb{E}_{\xi_{\text{cf}}} \left[\frac{1}{M_b} \sum_{m=1}^{M_b} \{ d_{\text{Ber}}(\hat{c}_{b,m}(G, a) \| \hat{c}_{b,m}(G, a')) + d_{\text{Ber}}(\hat{c}_{b,m}(G, a') \| \hat{c}_{b,m}(G, a)) \} \right]. \tag{7}$$

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} + \lambda_{\text{cf}} \mathcal{L}_{\text{cf}}. \tag{8}$$

The registered distribution ξ_{cf} includes its feasibility filter and action-pairing rule. All coefficients belong to the condition record; reporting them symbolically here prevents the objective definition from silently diverging from a registered configuration.

5 EVALUATION REGISTRY

The framework is intended to make empirical claims traceable rather than to treat similarly named rows as interchangeable. Each reported result therefore requires the following records:

- an intervention record identifying the parser build, graph schema, constraint-library revision, scorer, controller, proposer, execution wrapper, and task/seed mapping;

- a decoder record identifying the action-conditioned candidate grammar, relation vocabulary, and the applicable bound $r_{\max}$;
- an outcome record defining success, failure, timeout, tie handling, denominator, and the relationship between discrete outcomes and continuous diagnostics;
- a resource record connecting step, rollout, replan, and episode costs under one unit, hardware, and parallelism specification;
- an inference record specifying the sampling hierarchy, paired units, estimator, seed treatment, weights, and interval construction; and
- a provenance record resolving every external comparator to an immutable release, commit, checkpoint, configuration, and evaluation mapping.

This registry also separates natural interface conditions from controlled graph-quality perturbations. A threshold fit is associated only with the quality grid and corruption process declared in its own intervention record. The same discipline applies to wrapper policies: a score model and an execution-time policy are registered as different components, so an observed difference is attached to its actual intervention.

6 Discussion

Maintained planning systems combine learned prediction with interfaces and wrappers that can change on different schedules. The comparison contract turns that fact into a technical object: it specifies which component is swapped, which components remain shared, how an outcome is defined, and which cost scope is active. This supports direct comparisons among scorer, controller, and interface alternatives while preserving the identity of each condition.

The RavelScan design complements that contract with a graph-rollout model whose update order is explicit. Current-topology propagation precedes each selective temporal update, node lifecycles have a defined state convention, independent constraint risks use an appropriate Bernoulli divergence, and the decoder's degree dependence is visible in its complexity bound. These definitions allow an implementation and a later evaluation registry to be checked without relying on an ambiguous shorthand label.

7 Conclusion

Symbolic planners maintained behind frozen interfaces need comparisons that identify the intervention, outcome, and resource scope. This paper provides such a contract, formalizes a structured-versus-raw fallback threshold, and introduces an interleaved dynamic-graph scorer with explicit state, risk, and decoding definitions. Together, these components make maintenance studies easier to reproduce, audit, and extend across planning environments.

AI Use Statement

Generative AI was used only to draft and edit manuscript prose and prepare presentation elements. It was not used to generate synthetic data, develop theoretical or conceptual frameworks, formulate or prove mathematical claims, refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, process datasets, analyze qualitative data, or interpret results. The authors reviewed all AI-assisted work and take responsibility for the final content.

Reproducibility Statement

Sections 3–5 specify the condition record, estimator inputs, cost scopes, state update, objective, and provenance fields required to reproduce a registered comparison.

References

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine*

Learning, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. URL https://arxiv.org/abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.00396. arXiv:2111.00396.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 31, 2018. URL https://arxiv.org/abs/1809.01999. arXiv:1809.01999.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603. arXiv:1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104. arXiv:2301.04104.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507/.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05493. arXiv:1511.05493.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1902.10191. arXiv:1902.10191.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. VirtualHome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Puig_VirtualHome_Simulating_Household_CVPR_2018_paper.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768. arXiv:2010.03768.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. DyRep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyePrhR5KX.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10903. arXiv:1710.10903.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. URL https://arxiv.org/abs/2203.07540. arXiv:2203.07540.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2002.07962. arXiv:2002.07962.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.03629. arXiv:2210.03629.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *International Conference on Machine Learning*, pp. 62138–62160, 2024a. URL https://proceedings.mlr.press/v235/zhou24r.html.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024b. URL https://arxiv.org/abs/2307.13854. arXiv:2307.13854.

A Registry Schema

A registry entry stores the following fields for each result identifier:

- `id`, `benchmark`, and `split`;
- `task_ids`, `training_seeds`, and `evaluation_seeds`;
- `parser_build` and `schema_revision`;
- `constraint_revision` and `proposer_revision`;
- `scorer_revision` and `controller_revision`;
- `wrapper_revision`, `K`, `H`, and `candidate_grammar`;
- `outcome_definition` and `timeout_rule`;
- `pairing_key`, `aggregation_weights`, and `inference_plan`;
- `hardware`, `precision`, and `parallelism`; and
- `cost_scope` and `external_provenance`.

For a graph-quality study, the same entry additionally contains the oracle used only for the quality measurement, the quality grid, corruption generator and allocation, fitting family, monotonicity constraint, held-out partition, and interval construction. This makes a natural parser measurement and a controlled corruption measurement distinct records even when both are summarized by a quantity called graph quality.

B Contractive Selective-SSM Check

For a scalar diagonal entry, Equation (5) has the form

$$A = \exp(\Delta \bar{A}), \qquad \Delta > 0, \qquad \bar{A} = -\exp(A_{\log}) < 0.$$

Hence $\Delta \bar{A} < 0$ and $0 < A < 1$. The recurrence in Equation (3) therefore uses a contractive diagonal transition. The current graph message enters through $\widetilde{H}_t^{(l)}$ before that transition is applied, and the birth initializer is supplied at time $t-1$ only for an identifier newly present at t.

C Candidate-Set Complexity

For a graph with maximum degree $d_{\max}$, each source node can reach at most $d_{\max}^2$ ordered two-hop destinations. Thus $|N_2(G_t)| = O(|V_t| d_{\max}^2)$ in the worst case. The registered grammar permits at most $r_{\max}$ relation types for each such pair, so the typed two-hop expansion is $O(|V_t| d_{\max}^2 r_{\max})$. The action candidate set is reported separately because its size depends on the declared action grammar. This yields the decoder bound in the main text and leaves any tighter empirical bound to a registry entry that documents the applied cap or observed degree distribution.