GRAPH HOPFIELD NETWORKS: ENERGY-BASED NODE CLASSIFICATION WITH ASSOCIATIVE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce *Graph Hopfield Networks* (GHN), whose energy function couples associative memory retrieval with graph Laplacian smoothing for node classification. Gradient descent on this joint energy yields an iterative update interleaving Hopfield retrieval with Laplacian propagation. Memory retrieval provides regime-dependent benefits: up to 2.0 pp on sparse citation networks and up to 5 pp additional robustness under feature masking; the iterative energy-descent architecture itself is a strong inductive bias, with all variants (including the memory-disabled NoMem ablation) outperforming standard baselines on Amazon co-purchase graphs. Tuning $\lambda \leq 0$ enables graph sharpening for heterophilous benchmarks without architectural changes.

1 INTRODUCTION

Modern Hopfield networks (Krotov & Hopfield, 2016; Demircigil et al., 2017; Ramsauer et al., 2021) have renewed interest in associative memory as a computational primitive. Their connection to Transformer attention (Ramsauer et al., 2021) has spurred applications across vision, language, and scientific domains. Yet despite growing theoretical understanding (including hierarchical memory organization (Krotov, 2021), novel energy functions (Hoover et al., 2025), and capacity analyses (Lucibello & Mézard, 2024)), their application to *node classification* on graphs via an energy-based formulation remains, to our knowledge, underexplored. Liang et al. (2022) apply modern Hopfield networks to graph-level tasks, but do not couple memory retrieval to graph structure in a joint energy.

Graph neural networks (GNNs) learn node representations by aggregating neighborhood information (Kipf & Welling, 2017; Velickovic et al., 2018; Hamilton et al., 2017), but degrade under noisy or incomplete edges (Zügner et al., 2018). Associative memory offers an alternative, content-based signal: rather than relying on local structure, it retrieves relevant patterns from feature content.

We propose *Graph Hopfield Networks* (GHN), which combine these two mechanisms in a single energy function:

$$E_{\mathrm{GH}}(\mathbf{X}) = \sum_{v \in \mathcal{V}} \left[-\mathrm{lse}(\beta, \mathbf{M}\mathbf{x}_v) + \tfrac{1}{2}\|\mathbf{x}_v\|^2 \right] + \lambda \,\mathrm{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}), \tag{1}$$

where $\mathrm{lse}(\beta, \mathbf{z}) \coloneqq \beta^{-1} \log \sum_\mu \exp(\beta z_\mu)$ is the log-sum-exp operator. The first term drives memory retrieval toward learned patterns $\mathbf{M}$ and the second encourages smoothness over the graph Laplacian $\mathbf{L}$. Gradient descent on this energy yields an update rule interleaving Hopfield retrieval with Laplacian propagation (Section 2; derivation in Appendix D.2). The iterative energy-descent architecture is itself a strong inductive bias: all GHN variants,including a memory-disabled ablation, outperform every standard baseline on Amazon co-purchase graphs (Section 3.2). Memory acts as a substitute for missing structural signal: it adds up to 2.0 pp on sparse citation networks and up to 5 pp robustness under feature masking (Section 3.3), but is redundant on dense graphs where the Laplacian suffices. Tuning $\lambda \leq 0$ enables graph sharpening for heterophilous benchmarks without architectural changes (Section 3.4).

2 GRAPH HOPFIELD NETWORKS

2.1 ENERGY FUNCTION AND UPDATE RULE

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node features $\mathbf{X} \in \mathbb{R}^{N \times d}$ and (symmetric normalized) graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, we define the *Graph Hopfield energy* as in Eq. 1. The two terms encode complementary objectives: the Hopfield term drives each node's representation toward relevant learned patterns, while the Laplacian term encourages neighboring nodes to have similar representations.

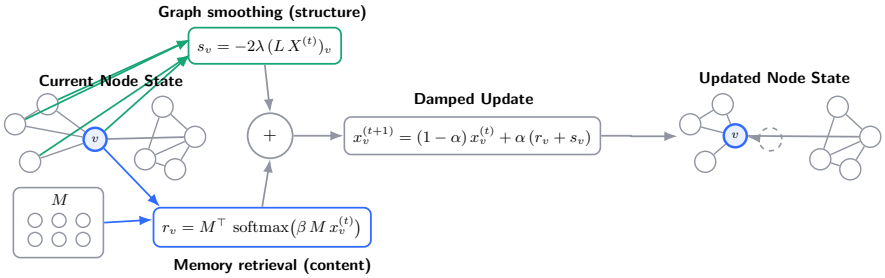

Figure 1: One iteration of the GHN update: node features $\mathbf{x}_v$ are blended with (i) memory retrieval from pattern bank $\mathbf{M}$, and (ii) graph Laplacian smoothing over neighbors. Damping α controls the step size.

Taking the gradient with respect to $\mathbf{x}_v$ yields the update rule:

$$\mathbf{x}_v^{(t+1)} = (1-\alpha)\,\mathbf{x}_v^{(t)} + \alpha\left[\underbrace{\mathbf{M}^\top \mathrm{softmax}\Big(\beta \mathbf{M}\mathbf{x}_v^{(t)}\Big)}_{\text{memory retrieval}} - 2\lambda\ \underbrace{(\mathbf{L}\mathbf{X}^{(t)})_v}_{\text{graph smoothing}}\right] \tag{2}$$

where $\alpha \in (0, 1)$ is a damping coefficient. This naturally interleaves two operations per iteration: memory retrieval from the memory bank $\mathbf{M}$, and structure-based smoothing via the graph Laplacian.

For least-squares retrieval (LSR; Hoover et al. (2025)), the softmax is replaced by normalized Epanechnikov kernel weights: $w_\mu = \mathrm{ReLU}(1 - \frac{\beta}{2}\|\mathbf{x}_v - \mathbf{m}_\mu\|^2) / \sum_{\mu'} \mathrm{ReLU}(1 - \frac{\beta}{2}\|\mathbf{x}_v - \mathbf{m}_{\mu'}\|^2)$. In our implementation, β is treated as a learnable parameter, so retrieval sharpness is adapted during training.

Convergence of the base LSE formulation is analyzed in Appendix D; in practice we observe stable convergence within T=4 iterations across all experiments.

2.2 GATED MEMORY RETRIEVAL

Memory retrieval can produce poor outputs early in training or when queries fall far from any stored pattern; to prevent this from corrupting representations, we introduce a learned gate:

$$\mathbf{g}_v = \sigma(\mathbf{W}_g[\mathbf{x}_v \| \mathbf{r}_v] + \mathbf{b}_g)\,, \qquad \tilde{\mathbf{r}}_v = \mathbf{g}_v \odot \mathbf{r}_v + (1 - \mathbf{g}_v) \odot \mathbf{x}_v, \tag{3}$$

where $\mathbf{r}_v$ is the raw retrieval output, σ is the sigmoid, $\|$ denotes concatenation, and $\odot$ is element-wise multiplication. We initialize $b_g = 2$ ($\mathbf{g}_v \approx 0.88$) to encourage memory use early in training; the gated output $\tilde{\mathbf{r}}_v$ replaces the retrieval term in Eq. 2.

2.3 HIERARCHICAL MEMORY

Flat attention over K patterns can be unstable when K is large relative to the feature dimension (Section 3.2). Inspired by Krotov (2021), we partition K patterns into G groups and use two-stage retrieval: (i) **routing**, soft-assign queries to groups via centroids computed from pattern parameters, then (ii) **retrieval**, standard Hopfield retrieval within each selected group. The full model is an encoder, L stacked GHN layers (each with T iterations and gating), and a linear classifier; retrieval uses tied keys/values. Details in Appendix A.

3 EXPERIMENTS

3.1 SETUP

Datasets. We evaluate on nine benchmarks: three homophilous citation networks (Cora, CiteSeer, PubMed (Sen et al., 2008)), two Amazon co-purchase graphs (Photo and Computers (Shchur et al., 2018)), and four heterophilous graphs (Texas, Wisconsin, Cornell (Pei et al., 2020), and Actor (Tang et al., 2009)). Standard public splits are used for Planetoid; random 60/20/20 splits for Amazon; standard splits for the heterophilous datasets.

Models. We compare GHN variants (**LSE** (flat, softmax retrieval), **LSR** (flat, Epanechnikov retrieval), **Hier**(G) (hierarchical with G groups), and **NoMem** (memory-disabled ablation, Laplacian smoothing only)) against GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), GraphSAGE (Hamilton et al., 2017), APPNP (Klicpera et al., 2019), GIN (Xu et al., 2019), MLP, and GPR-GNN (Chien et al., 2021) (heterophily-specific; results from the original paper on the same evaluation protocol).

Training. Per-model hyperparameter tuning on each dataset; each model selects its best config by validation accuracy. All results are mean ± std over 10 seeds. Full training protocol and hyperparameter grids are in Appendix A.

3.2 NODE CLASSIFICATION

Table 1: Node classification accuracy (%) with per-model HP tuning. **Bold**: best per column. †: bimodal (>10% std).

	Cora	CiteSeer	PubMed	Photo	Computers
GHN *variants*					
LSE	82.0±0.6	70.9±0.8	77.9±0.4	94.5±0.3	91.3±0.2
LSR	82.0±0.6	70.9±1.0	78.1±0.4	94.4±0.2	91.3±0.4
Hier(8)	82.3±0.4	70.3±1.1	77.6±0.8	94.2±0.2	91.3±0.2
NoMem	80.3±0.4	69.8±1.1	77.2±0.6	**94.7±0.2**	**91.5±0.2**
Baselines					
GCN	82.2±0.4	72.2±0.5	79.4±0.4	89.6±0.3	73.7±13.0†
GAT	**82.8±0.8**	72.3±0.6	78.2±0.4	92.2±0.3	87.9±0.6
GraphSAGE	80.2±0.6	70.5±0.8	77.9±0.4	93.9±0.2	87.4±0.4
APPNP	**82.8±0.7**	**72.4±0.5**	**80.3±0.2**	92.2±0.3	41.9±13.1†
GIN	77.5±1.2	63.8±2.3	77.6±0.7	93.0±0.5	88.9±0.5
MLP	60.0±0.8	60.6±0.6	73.4±0.5	88.5±0.3	75.4±11.9†

Iterative architecture is the primary driver on Amazon. Every GHN variant, including the memory-disabled NoMem ablation, outperforms the best standard baseline on both Amazon datasets (+0.8 pp on Photo, +2.6 pp on Computers). GCN, APPNP, and MLP exhibit bimodal training collapse on Computers (>10% std), a known failure mode that the iterative energy-descent architecture avoids entirely. Memory-enabled variants are within 0.5 pp of NoMem on Amazon, suggesting that on dense graphs (Photo: 238k edges vs. Cora: 10.6k), Laplacian smoothing captures the relevant signal and memory provides marginal additional value.

Memory retrieval adds value on sparse citation graphs. On all three Planetoid datasets, the best memory-enabled variant outperforms NoMem: Hier(8) on Cora (+2.0 pp, non-overlapping $\pm 1\sigma$), LSR on CiteSeer (+1.1 pp), and LSR on PubMed (+0.9 pp), though the latter two gaps overlap in standard deviation. GHN variants are competitive with but do not surpass GAT or APPNP on Planetoid.

LSR is more stable than LSE at moderate K. A $\beta \times K$ phase diagram (Figure 4) shows LSE requires $K{=}256$ to converge on Amazon Photo, while LSR is stable at $K{=}64$ for moderate β (Hoover et al., 2025).

3.3 Robustness to Graph Corruption

We test three corruption types on Amazon Photo: edge removal, feature masking, and additive feature noise.

Table 2: Accuracy (%) under 50% corruption on Amazon Photo. Clean baselines may differ from Table 1 due to a separate HP sweep optimized for robustness. **Bold**: best per column.

	Clean	Edge drop	Feat. mask	Feat. noise
GHN *variants*				
LSE	**94.4**±0.3	93.7±0.4	89.4±1.8	93.9±0.3
LSR	94.3±0.2	93.7±0.5	91.0±2.4	94.1±0.2
Hier(8)	94.2±0.2	93.7±0.5	**91.9**±1.1	93.9±0.1
NoMem	**94.4**±0.3	**94.0**±0.5	86.9±3.1	**94.2**±0.3
Baselines				
GCN	89.7±0.3	89.1±0.5	81.2±1.5	89.7±0.3
GAT	92.2±0.3	91.2±0.5	81.4±5.0	92.2±0.2
GraphSAGE	93.9±0.2	93.0±0.3	73.3±1.8	93.6±0.3
APPNP	92.2±0.3	91.7±0.3	65.0±1.8	91.8±0.3
GIN	92.6±0.4	89.6±0.8	91.1±0.9	92.6±0.5
MLP	88.5±0.3	88.5±0.3	70.7±0.6	86.6±0.2

All memory-enabled GHN variants rank in the top four under every corruption type, while NoMem joins them except under feature masking, where GIN's structure-independent aggregation (91.1%) outperforms NoMem (86.9%) (Table 2). Under edge removal and feature noise, NoMem performs comparably to memory-enabled variants, indicating that iterative Laplacian propagation is the primary robustness mechanism. **Under feature masking, memory adds substantial value.** Hier(8) retains 91.9% at 50% masking versus 86.9% for NoMem, a 5.0 pp gap absent on clean data. This is the clearest evidence that stored patterns compensate for degraded node features, a mechanism unavailable to structure-only smoothing.

On Planetoid, GHN robustness varies by dataset (LSE leads on PubMed; details in Appendix C.1); Figure 2 shows edge-deletion curves.

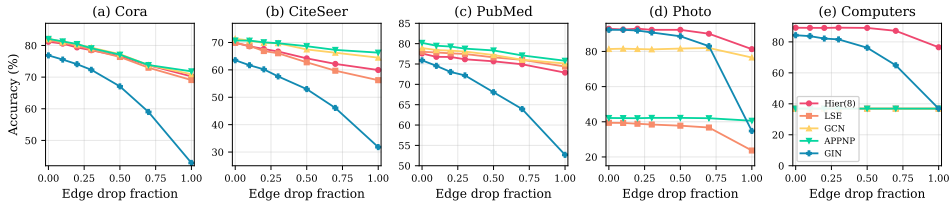

Figure 2: Accuracy under progressive edge deletion on Amazon and Planetoid datasets. GHN variants degrade more gradually on Amazon; Planetoid details in Appendix C.1.

3.4 Heterophilous Graphs

On heterophilous graphs, Laplacian smoothing ($\lambda > 0$) pushes neighbors toward similar representations and can harm classification. We tune $\lambda \in \{-0.1, -0.05, -0.01, 0\}$ to enable graph sharpening (negative λ pushes neighbors apart).

Table 3 shows that with tuned negative λ, GHN variants perform competitively with GPR-GNN, a method specifically designed for heterophily, while substantially outperforming standard GNNs (GCN, GAT, APPNP), which all collapse under graph smoothing. We report published GPR-GNN results on the same splits (†); we did not re-tune or re-run GPR-GNN. GHN-LSE leads on Texas (81.4% vs. 78.4%), while GPR-GNN leads on Wisconsin (82.9% vs. 82.5%) and Cornell (80.3% vs. 78.9%); all differences are within overlapping standard deviations. Notably, GPR-GNN exhibits much higher variance (e.g., ±8.1 on Cornell vs. ±1.7 for GHN-LSR), suggesting that GHN is more stable across splits on these small datasets. The key advantage is architectural: GHN handles both

Table 3: Node classification (%) on heterophilous benchmarks. **Bold**: best per column. †Published results (Chien et al., 2021) on same splits; Actor not reported.

	Texas	Wisconsin	Cornell	Actor
GHN *variants*				
LSE	**81.4±1.5**	82.2±0.6	78.1±3.2	**38.0±0.5**
LSR	80.8±1.5	82.5±1.1	78.9±1.7	37.9±0.5
Hier(8)	80.5±3.3	82.2±1.7	78.1±2.0	37.8±0.6
NoMem	80.5±1.7	82.4±0.9	78.6±2.4	**38.0±0.7**
Heterophily-aware baselines				
GPR-GNN†	78.4±4.4	**82.9±4.2**	**80.3±8.1**	–
Standard baselines				
MLP	80.3±1.8	82.2±0.6	73.0±0.0	37.6±0.4
GraphSAGE	78.4±1.8	80.0±0.8	76.2±1.7	37.0±0.5
GCN	70.3±1.3	55.7±1.0	40.3±0.9	28.8±0.4
GAT	68.6±2.3	58.0±1.9	43.8±4.0	29.4±0.6
APPNP	70.0±1.5	56.7±1.1	43.2±3.6	32.4±0.6
GIN	64.9±0.0	50.4±1.6	55.7±6.3	27.3±0.5

homophilous and heterophilous graphs by tuning a single parameter λ, whereas GPR-GNN requires learning K polynomial filter coefficients specialized for the graph's spectral properties.

All best configs select $\lambda \in \{-0.05, -0.01\}$; Appendix C.2 shows a sharp phase transition (e.g., LSE on Cornell: 73.5% at λ=0 to 44.3% at λ=0.3). Memory-enabled and NoMem variants perform similarly at $\lambda \leq 0$, since the Laplacian term is minimal and all variants reduce to feature-space operations.

4 Discussion and Future Directions

The central finding of this work is a decomposition: on graphs, *iterative energy descent matters more than what you descend on*. NoMem, an energy-descent architectural ablation that removes the Hopfield term entirely, matches or exceeds memory-enabled variants on both Amazon datasets and ties on Actor, yet still outperforms every standard baseline on Amazon. This suggests that the iterative damped-update architecture provides a strong inductive bias that stabilizes training on graphs where feedforward baselines collapse and enables graceful degradation under edge corruption. For associative memory, the regime-dependent benefits are key: memory adds value on sparse graphs and under feature masking, while the ablation clarifies the respective roles of architecture vs. retrieval.

Memory's contribution is better understood as *substitutive* rather than complementary to graph structure. On dense graphs (Photo: 238k edges), the Laplacian already provides sufficient signal for classification, and memory is redundant. On sparse graphs (Cora: 10.6k edges) or under feature corruption, the structural signal is insufficient and memory fills the gap, up to 2.0 pp on clean Cora and 5.0 pp under 50% feature masking. This substitution view explains why the gate learns a near-constant blending ratio (Appendix C.6): the optimal balance between memory and structure is a property of the *graph regime*, not of individual nodes, so a static gate suffices.

Limitations. GHN does not outperform GAT/APPNP on clean Planetoid graphs; on heterophilous benchmarks, gains over MLP are modest and often within one standard deviation. Flat LSE/LSR require K=256 to avoid collapse on Amazon; we evaluate only random (not adversarial) corruption; $O(NK)$ retrieval adds overhead (approximately 1.5–2$\times$ wall-clock time per epoch relative to GAT, estimated across datasets). The convergence theory (Appendix D) requires $\beta\|\mathbf{M}\|^2 < 2$, a condition violated at trained operating points for most datasets (Table 11); in practice, stability is achieved empirically within T=4 iterations. Promising extensions include per-node λ, replacing the gate with a fixed scalar blend, adversarial training of memory patterns, sparse retrieval for scaling, and comparison with graph transformers, the most direct architectural competitors.

REFERENCES

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021.

Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168(2):288–299, 2017. doi: 10.1007/s10955-017-1816-y. URL https://arxiv.org/abs/1702.01929.

Fangda Gu, Heng Chang, Weize Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2009.06211.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.02216.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Dmitry Krotov, Duen Horng Chau, and Mohammed J Zaki. Energy transformer. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2302.07253.

Benjamin Hoover, Zhaoyang Shi, Krishnakumar Balasubramanian, Dmitry Krotov, and Parikshit Ram. Dense associative memory with epanechnikov energy. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.10801.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.02907.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.05997.

Dmitry Krotov. Hierarchical associative memory. *arXiv preprint arXiv:2107.06446*, 2021. URL https://arxiv.org/abs/2107.06446.

Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1606.01164.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1801.07606.

Yuchen Liang, Dmitry Krotov, and Mohammed J Zaki. Modern hopfield networks for graph embedding. *Frontiers in Big Data*, 5, 2022. doi: 10.3389/fdata.2022.1044709. URL https://arxiv.org/abs/2208.14376.

Carlo Lucibello and Marc Mézard. Exponential capacity of dense associative memories. *Physical Review Letters*, 132(7):077301, 2024. doi: 10.1103/PhysRevLett.132.077301. URL https://arxiv.org/abs/2304.14964.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.10947.

Hongbin Pei, Bingzhe Wei, Bairu Chang, Yunhe Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2002.05287.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.12454.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2008.02217.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop at NeurIPS*, 2018. URL https://arxiv.org/abs/1811.05868.

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 807–816, 2009. doi: 10.1145/1557019.1557108.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.14522.

Petar Velickovic, 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.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.00826.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform bad for graph representation? In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.05234.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2847–2856, 2018. URL https://arxiv.org/abs/1805.07984.

A Experimental Details

Table 4 gives the full hyperparameter grid. The full model is encoder $\rightarrow L$ Graph Hopfield layers (each with T iterations and gating) $\rightarrow$ classifier; retrieval uses tied keys/values, skip connections (0.1), dropout, and LayerNorm. Model-specific tuning axes: GAT heads $\in \{4, 8\}$, APPNP $\alpha \in \{0.1, 0.2\}$, GHN-LSE/LSR $K \in \{64, 256\}$. GHN uses λ=0.3 on homophilous datasets and $\lambda \in \{-0.1, -0.05, -0.01, 0\}$ on heterophilous datasets.

Table 4: Hyperparameters for GHN and training protocol. Per-model HP tuning (Table 1) selects the best config per dataset from the grid ranges shown.

Parameter	**Value / Grid**
GHN *(all variants)*	
Hidden dimension	$\{64, 128\}$
Number of patterns K	$\{64, 256\}$ (LSE/LSR)
Inverse temperature β (init)	1.0 (learnable)
Laplacian weight λ	0.3
Damping α	0.3
Iterations per layer T	4
Number of layers L	2
Number of groups G (Hier)	8
Memory heads H	$\{1, 2, 4, 8\}$ (ablated; Appendix C.5)
Dropout	$\{0.3, 0.5\}$
Gate bias init	2.0
Keys/values	Tied
Skip weight	0.1
Training (all models)	
Learning rate	$\{0.001, 0.005, 0.01\}$
Weight decay	$\{10^{-4}, 5{\times}10^{-4}, 10^{-3}\}$
Optimizer	Adam
Epochs	300
Early stopping patience	50

Baseline collapse on Amazon Computers. The bimodal collapse of GCN, APPNP, and MLP on Amazon Computers (Table 1) persists across our full hyperparameter grid: we swept learning rates $\{0.001, 0.005, 0.01\}$, weight decay $\{10^{-4}, 5{\times}10^{-4}, 10^{-3}\}$, and dropout $\{0.3, 0.5\}$, with no configuration eliminating the failure mode. Collapse manifests as some seeds converging normally ($\sim$87%) while others diverge to near-chance ($\sim$50%), producing the $>$10% standard deviations reported. This instability is consistent with known sensitivity of spectral-normalization-based propagation (GCN, APPNP) on graphs with heterogeneous degree distributions (Shchur et al., 2018); Amazon Computers has a heavy-tailed degree distribution (max degree $>$400) that amplifies initialization-dependent convergence. We did not apply additional stabilization techniques (e.g., batch normalization, learning rate warmup) beyond those in standard implementations, since our goal was to compare architectures under a shared training protocol rather than to maximize each baseline individually.

B Related Work

Hopfield networks and associative memory. Modern (dense) Hopfield networks (Krotov & Hopfield, 2016; Demircigil et al., 2017; Ramsauer et al., 2021) store exponentially many patterns and connect directly to Transformer attention. Extensions include hierarchical memory (Krotov, 2021), the Energy Transformer (Hoover et al., 2023), and the Epanechnikov/LSR energy (Hoover et al., 2025). Liang et al. (2022) apply modern Hopfield retrieval to graph *embedding* (graph-level classification), while we focus on *node* classification with a joint energy that explicitly couples retrieval and graph smoothing.

Graph transformers and content–structure coupling. Graph transformers such as Graphormer (Ying et al., 2021) and subsequent scalable variants (Rampášek et al., 2022) combine content-based attention with structural biases. Our goal is different: instead of replacing message

passing with full transformer blocks, we formulate node updates as iterative descent steps on a joint energy, making the retrieval/smoothing tradeoff explicit and analyzable.

Implicit and iterative graph models. Implicit/equilibrium GNNs solve fixed-point equations over graph representations (Gu et al., 2020). Our fixed-point view is related, but the operator is derived from a Hopfield-inspired retrieval energy plus Laplacian regularization, yielding explicit smoothness/contraction conditions for the base dynamics (Appendix D). Our iterative formulation also connects to analyses of depth-related degradation and expressivity loss in graph propagation (Li et al., 2018; Oono & Suzuki, 2020).

Robust graph learning. GNN robustness is typically studied under adversarial structural perturbations (Zügner et al., 2018). Our setting is complementary: we study random corruption to isolate the effect of structural reliability on model accuracy, without adversarial assumptions. Our finding that aggregation design (sum vs. mean vs. teleportation) largely determines robustness connects to analyses of information flow bottlenecks in GNNs (Topping et al., 2022).

C Additional Results

C.1 Edge Deletion on Planetoid

Table 5 quantifies relative accuracy drop under edge deletion on the Planetoid datasets. GHN robustness varies by dataset: LSE leads on PubMed (4.6% drop), while on CiteSeer GHN variants are less robust (14–20% drop) than GCN (10%) and APPNP (6%). GIN is uniquely fragile under full edge removal (31–50% loss) due to its injective sum aggregation (Xu et al., 2019); GCN (mean) and APPNP (teleportation) degrade more gracefully (5–14%). GIN shows the opposite profile under feature masking: robust (91.1% at 50% masking) but fragile under edge drop, reflecting that injective aggregation exploits structure maximally while memory retrieval provides a structure-independent channel.

Table 5: Relative accuracy drop (%) under edge deletion on Planetoid. Smaller magnitude = greater robustness. **Bold**: most robust per dataset.

	50% edge drop			100% edge drop		
	Cora	CiteSeer	PubMed	Cora	CiteSeer	PubMed
GHN-LSE	−6.5	−10.2	**−1.8**	−15.5	−19.5	**−4.6**
GHN-Hier(8)	**−5.6**	−8.1	−2.5	−13.5	−14.2	−6.1
GCN	−5.9	−5.5	−2.2	−13.6	−9.7	−4.8
APPNP	−6.1	**−3.0**	−2.3	**−12.5**	**−6.3**	−5.5
GIN	−12.7	−16.5	−10.3	−44.1	−49.9	−30.5

Feature masking across datasets. Figure 3 shows feature masking degradation across all five datasets. Unlike edge deletion, all models (including GHN) degrade at comparable rates as features are masked on these datasets. However, on the Amazon datasets with $K{=}256$ and tuned hyperparameters (Table 2), memory-enabled variants show substantially greater robustness to feature masking than NoMem (e.g., Hier: 91.9% vs. NoMem: 86.9% at 50% masking on Photo), suggesting that higher pattern capacity provides redundant representations that compensate for missing features, a benefit not observed at $K{=}64$ on Planetoid.

LSE vs LSR phase diagram. Figure 4 shows the $\beta \times K$ phase diagram on Amazon Photo. LSE (left) is stable only at $K{=}256$; at $K{\leq}128$ it exhibits bimodal collapse where some seeds converge and others do not, regardless of β. LSR (right) achieves stable $\geq$93% accuracy at $K{=}64$ for $\beta \in \{0.2, 0.5, 1.0, 5.0\}$, but is also bimodal at very low β (< 0.2) and at $\beta{=}2.0$.

Memory contribution on Amazon. Under per-model hyperparameter tuning (Table 1), NoMem slightly outperforms all memory-enabled variants on both Amazon datasets (94.7% vs. 94.5% on Photo; 91.5% vs. 91.3% on Computers). This contrasts with the consistent memory benefit on Planetoid and suggests that the Amazon co-purchase graphs have richer edge structure (Photo: 238k edges vs. Cora: 10.6k edges) making Laplacian smoothing alone sufficient. The GHN architecture's

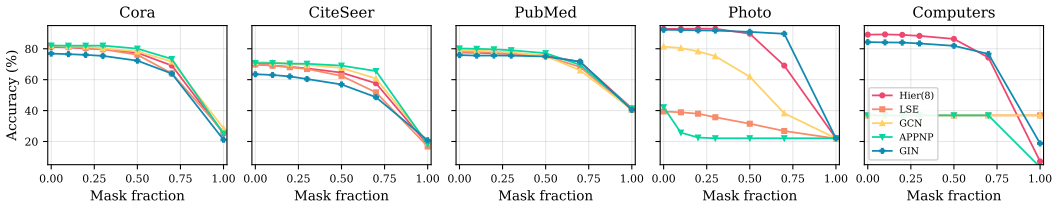

Figure 3: Accuracy under feature masking across five datasets. Tuning selects K=64 on Planetoid and K=256 on Amazon. All models degrade at comparable rates on Planetoid; on Amazon, memory-enabled variants show substantially greater robustness (Table 2).

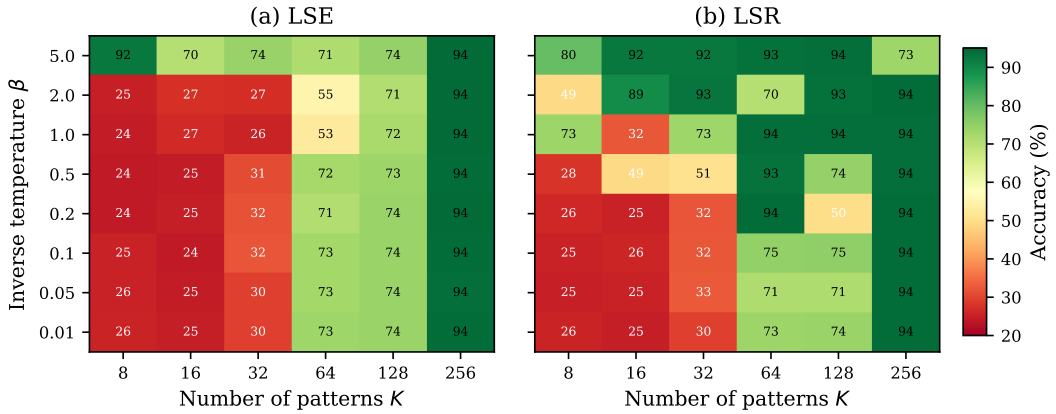

Figure 4: Phase diagram of LSE vs. LSR energy on Amazon Photo. Each cell shows mean test accuracy over 10 seeds. High variance in some cells reflects bimodal behavior (seeds either converge to $\sim$94% or collapse). At K=64, LSR is stable for moderate β while LSE is bimodal for all tested β.

advantage on Amazon is its iterative Laplacian propagation, not memory retrieval per se; all GHN variants outperform the best standard baseline by 0.8–2.6 pp.

C.2 Laplacian Weight λ Ablation

Table 6 shows GHN-LSE accuracy across $\lambda \in \{0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 1.0\}$ on all seven datasets. On homophilous datasets (Cora, CiteSeer, PubMed), accuracy increases monotonically from λ=0 to a peak at λ=0.2–0.3, then degrades mildly. On heterophilous datasets (Texas, Wisconsin, Cornell, Actor), accuracy decreases monotonically with increasing λ, confirming that Laplacian smoothing is harmful when neighbors have different labels.

Table 6: GHN-LSE test accuracy (%) as a function of λ. **Bold**: best λ per dataset.

Dataset	0	0.01	0.05	0.1	0.2	0.3	0.5	1.0
Cora	57.7	62.7	71.7	76.1	80.2	**81.8**	81.6	80.6
CiteSeer	55.6	59.1	65.1	67.6	**70.2**	69.6	68.9	69.4
PubMed	73.0	73.2	74.5	75.2	**78.2**	77.7	76.9	76.7
Texas	**74.1**	71.6	70.0	64.9	66.8	67.0	66.8	64.9
Wisconsin	79.8	**80.8**	72.2	67.1	60.0	55.1	45.3	44.9
Cornell	**73.5**	72.4	63.8	46.2	41.1	44.3	42.2	41.4
Actor	**37.5**	37.3	36.5	35.5	30.8	28.1	25.5	25.0

The results for other GHN variants (LSR, Hier, NoMem) follow the same pattern. On Cora, GHN-LSR peaks at λ=0.3 (81.5%) and GHN-Hier(8) at λ=0.5 (81.7%). On heterophilous datasets, all variants achieve their best performance at λ=0 or λ=0.01.

C.3 Negative λ (Graph Sharpening)

Setting $\lambda < 0$ reverses the Laplacian regularization: instead of pulling neighbors together, it pushes them apart ("graph sharpening"). Table 7 shows that small negative values ($\lambda \in \{-0.05, -0.01\}$) can improve upon λ=0 on heterophilous datasets.

Table 7: GHN accuracy (%) with negative λ on heterophilous datasets. Best result per dataset-model in **bold**.

		λ=−0.3	λ=−0.1	λ=−0.05	λ=−0.01
Texas	LSE	75.9	77.3	**78.6**	74.6
	LSR	75.9	77.3	**79.5**	78.6
Wisconsin	LSE	76.9	81.0	**81.8**	79.0
	LSR	78.8	**80.8**	80.4	79.0
Cornell	LSE	61.9	66.5	74.6	**77.0**
	LSR	63.2	69.5	74.3	**77.8**
Actor	LSE	34.7	37.5	**38.0**	37.5
	LSR	34.9	37.2	37.3	**37.8**

The optimal negative λ is dataset-dependent: -0.05 works best on Texas and Wisconsin, while -0.01 is preferred on Cornell. Gains over λ=0 are modest but consistent (0.5–3.5 pp). Note that the accuracies here differ from Table 3 because this ablation varies λ with other hyperparameters (hidden dimension, dropout, K) held at a single default configuration, whereas Table 3 reports per-model HP tuning across the full grid. Very negative values (λ=−0.3) degrade performance, suggesting that excessive sharpening is as harmful as excessive smoothing.

C.4 Iteration Count T Ablation

Table 8 shows how the number of Hopfield update iterations T per layer affects accuracy on one homophilous (Cora) and two heterophilous (Texas, Wisconsin) datasets, using λ=0.3.

Table 8: GHN-LSE accuracy (%) vs. iterations T with λ=0.3. **Bold**: best T.

	T=1	T=2	T=4	T=8	T=16
Cora	74.5±1.1	78.4±0.5	**81.8±1.4**	56.1±26.1	30.9±2.0
Texas	**73.5±2.8**	72.2±2.9	67.0±1.7	65.9±1.4	65.4±1.1
Wisconsin	**63.7±1.0**	58.6±2.3	55.1±5.9	44.1±1.9	46.3±1.4

On Cora, LSE peaks at T=4 and collapses at $T \geq 8$ (the large standard deviation at T=8 indicates bimodal convergence). On heterophilous datasets, T=1 is best; additional iterations with λ=0.3 apply progressively more harmful smoothing. This is consistent with the λ ablation results: on heterophilous graphs, minimizing the influence of the Laplacian term (whether by setting λ=0 or T=1) is essential.

C.5 Memory Heads H Ablation

Table 9 shows GHN-LSE accuracy with $H \in \{1, 2, 4, 8\}$ memory heads on four datasets.

Table 9: GHN-LSE accuracy (%) vs. number of memory heads H.

	H=1	H=2	H=4	H=8
Cora	81.5±0.9	81.6±1.0	81.8±1.4	80.3±0.4
CiteSeer	69.6±1.4	70.0±1.6	69.6±0.7	69.5±1.5
PubMed	77.6±0.5	77.6±0.6	77.7±0.9	77.9±1.0
Texas	64.9±1.8	65.9±2.9	67.0±1.7	67.6±0.0

The number of memory heads has minimal effect on accuracy: across all four datasets, the difference between the best and worst H is at most 1.5 pp on Cora and 2.7 pp on Texas. This supports using $H{=}1$ as the default for simplicity without sacrificing performance.

C.6 Gate Analysis under Feature Corruption

We train GHN-LSE on Amazon Photo (300 epochs) and extract the learned gate value $g_v = \sigma(W_g[\mathbf{x}_v \| \hat{\mathbf{x}}_v] + b_g)$ under feature masking at levels 0%, 10%, 30%, 50%, and 70%. Table 10 reports the mean gate value (averaged over nodes and iterations) at each corruption level.

Table 10: Mean gate value $\bar{g}$ of GHN-LSE on Amazon Photo under increasing feature masking. Higher g means more weight on memory retrieval.

Mask %	0%	10%	30%	50%	70%
$\bar{g}$	$0.727_{\pm 0.039}$	$0.727_{\pm 0.039}$	$0.726_{\pm 0.038}$	$0.725_{\pm 0.038}$	$0.723_{\pm 0.037}$
Accuracy (%)	$93.6_{\pm 0.5}$	$93.6_{\pm 0.5}$	$93.2_{\pm 0.5}$	$90.6_{\pm 1.3}$	$67.9_{\pm 12.8}$

The gate remains nearly constant ($\Delta\bar{g} \leq 0.004$) even as accuracy degrades substantially under heavy corruption. This indicates the gate does not dynamically adapt its memory/feature blending ratio at inference time; instead, robustness arises because the stored prototype patterns provide useful class-level information even when node features are degraded.

D Theoretical Analysis for Base LSE Dynamics

The standard Hopfield energy is guaranteed to decrease under asynchronous updates of a single query (Ramsauer et al., 2021). Our coupled scheme applies synchronous updates across all nodes with the Laplacian term introducing inter-node coupling, so this guarantee does not directly apply. Below we provide formal results for the *base* GHN energy with LSE retrieval and fixed $(\mathbf{M}, \beta, \lambda, \mathbf{L})$ during iterative updates: explicit smoothness and contraction conditions for gradient-descent and fixed-point updates. These results do not cover the full training-time system with gating, hierarchical routing, and jointly learned parameters; for the full model we rely on empirical stability (stable convergence within $T{=}4$ iterations in all experiments). The theory assumes $\lambda \geq 0$; for $\lambda < 0$ (graph sharpening on heterophilous benchmarks), the Laplacian term becomes indefinite and our convexity/contraction guarantees no longer apply. We keep $|\lambda|$ small ($\lambda \in \{-0.1, -0.05, -0.01\}$) and rely on damping, early stopping, and empirical stability in those experiments. Table 11 reports $\beta\|\mathbf{M}\|_\sigma^2$ (product of learned β and spectral norm squared of $\mathbf{M}$) at trained operating points, averaged over 5 seeds. The convexity condition $\beta\|\mathbf{M}\|^2 < 2$ is satisfied only for Cornell (0.75) and CiteSeer (1.88); all other datasets exceed the threshold, with Amazon Photo reaching 35.96. This places the system well outside the strongly convex regime of Proposition D.5 in typical use; gradient descent nonetheless converges reliably within $T{=}4$ iterations in all experiments, suggesting tighter analysis (e.g., local convexity near attractors or Łojasiewicz-type arguments) may close this gap.

Table 11: Learned $\beta\|\mathbf{M}\|_\sigma^2$ at trained operating points (mean over 5 seeds). The convexity condition requires this product to be < 2.

Dataset	β	$\|\mathbf{M}\|_\sigma^2$	$\beta\|\mathbf{M}\|_\sigma^2$
Cora	1.11	1.94	2.15
CiteSeer	1.09	1.71	**1.88**
PubMed	1.36	5.24	6.54
Amazon Photo	0.99	36.37	35.96
Amazon Comp.	1.12	2.11	2.36
Texas	1.25	3.88	4.65
Wisconsin	1.05	2.48	2.60
Cornell	1.07	0.68	**0.75**
Actor	1.00	4.61	4.72

D.1 SETUP

Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be the node representation matrix with rows $\mathbf{x}_v$, and let $\mathbf{M} \in \mathbb{R}^{K\times d}$ be the memory matrix. For fixed $\beta > 0$ and $\lambda \geq 0$, define

$$E_{\text{base}}(\mathbf{X}) = \sum_{v=1}^{N}\left[-\beta^{-1}\log\sum_{\mu=1}^{K}\exp\big(\beta\,\mathbf{m}_\mu^\top\mathbf{x}_v\big) + \frac{1}{2}\|\mathbf{x}_v\|^2\right] + \lambda\,\mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}), \tag{4}$$

where $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the symmetric normalized Laplacian.

Proposition D.1 (Normalized Laplacian quadratic form)**.** *If* $\mathbf{A}$ *is symmetric and* $\mathbf{D}_{vv} = d_v > 0$*, then*

$$\mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) = \frac{1}{2}\sum_{u,v}A_{uv}\left\|\frac{\mathbf{x}_u}{\sqrt{d_u}} - \frac{\mathbf{x}_v}{\sqrt{d_v}}\right\|^2 \geq 0.$$

Hence $\mathbf{L} \succeq 0$ *and the Laplacian term is convex.*

Proof. Define $\mathbf{Y} := \mathbf{D}^{-1/2}\mathbf{X}$, so $\mathbf{y}_v = \mathbf{x}_v/\sqrt{d_v}$. Using $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$,

$$\mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) = \mathrm{tr}(\mathbf{X}^\top\mathbf{X}) - \mathrm{tr}(\mathbf{Y}^\top\mathbf{A}\mathbf{Y}) = \sum_v\|\mathbf{x}_v\|^2 - \sum_{u,v}A_{uv}\,\mathbf{y}_u^\top\mathbf{y}_v.$$

Now expand

$$\frac{1}{2}\sum_{u,v}A_{uv}\|\mathbf{y}_u - \mathbf{y}_v\|^2 = \frac{1}{2}\sum_{u,v}A_{uv}\left(\|\mathbf{y}_u\|^2 + \|\mathbf{y}_v\|^2 - 2\mathbf{y}_u^\top\mathbf{y}_v\right) \tag{5}$$

$$= \sum_u\left(\sum_v A_{uv}\right)\|\mathbf{y}_u\|^2 - \sum_{u,v}A_{uv}\mathbf{y}_u^\top\mathbf{y}_v \tag{6}$$

$$= \sum_u d_u\left\|\frac{\mathbf{x}_u}{\sqrt{d_u}}\right\|^2 - \sum_{u,v}A_{uv}\frac{\mathbf{x}_u^\top\mathbf{x}_v}{\sqrt{d_u d_v}} \tag{7}$$

$$= \sum_u\|\mathbf{x}_u\|^2 - \sum_{u,v}A_{uv}\frac{\mathbf{x}_u^\top\mathbf{x}_v}{\sqrt{d_u d_v}} \tag{8}$$

$$= \mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}). \tag{9}$$

This proves the identity. The right-hand side is a sum of squared norms with nonnegative weights, so it is nonnegative. □

D.2 GRADIENT AND HESSIAN STRUCTURE

Proposition D.2 (Gradient formula)**.** *For each node* v*,*

$$\nabla_{\mathbf{x}_v}E_{\text{base}}(\mathbf{X}) = -\mathbf{M}^\top\mathrm{softmax}(\beta\,\mathbf{M}\mathbf{x}_v) + \mathbf{x}_v + 2\lambda(\mathbf{L}\mathbf{X})_v. \tag{10}$$

Therefore the damped update in Eq. 2 is exactly gradient descent on E_{base} *with step size* α*.*

Proof. Write

$$E_{\text{base}}(\mathbf{X}) = \sum_{u=1}^{N}e_u(\mathbf{x}_u) + \lambda\,\mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}), \quad e_u(\mathbf{x}) = -\beta^{-1}\log\sum_{\mu=1}^{K}e^{\beta\mathbf{m}_\mu^\top\mathbf{x}} + \frac{1}{2}\|\mathbf{x}\|^2.$$

Only $e_v(\mathbf{x}_v)$ and the Laplacian term depend on $\mathbf{x}_v$. For retrieval, set $\mathbf{z} = \mathbf{M}\mathbf{x}_v$ and use $\nabla_{\mathbf{z}}\left[\beta^{-1}\log\sum_\mu e^{\beta z_\mu}\right] = \mathrm{softmax}(\beta\mathbf{z})$. Chain rule gives $\nabla_{\mathbf{x}_v}\mathrm{lse}(\beta, \mathbf{M}\mathbf{x}_v) = \mathbf{M}^\top\mathrm{softmax}(\beta\mathbf{M}\mathbf{x}_v)$. The quadratic term contributes $\mathbf{x}_v$. For the Laplacian term,

$$\nabla_{\mathbf{X}}\mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) = (\mathbf{L} + \mathbf{L}^\top)\mathbf{X} = 2\mathbf{L}\mathbf{X},$$

since $\mathbf{L}$ is symmetric. Combining terms yields Eq. 10. Then

$$\mathbf{x}_v^{(t+1)} = \mathbf{x}_v^{(t)} - \alpha\nabla_{\mathbf{x}_v}E_{\text{base}}(\mathbf{X}^{(t)}) = (1-\alpha)\mathbf{x}_v^{(t)} + \alpha\Big[\mathbf{M}^\top\mathrm{softmax}(\beta\mathbf{M}\mathbf{x}_v^{(t)}) - 2\lambda(\mathbf{L}\mathbf{X}^{(t)})_v\Big].$$

□

Lemma D.3 (Retrieval Jacobian). *Let* $s(\mathbf{x}) = \mathbf{M}^\top \mathrm{softmax}(\beta \mathbf{M}\mathbf{x})$ *and* $\mathbf{p}(\mathbf{x}) = \mathrm{softmax}(\beta \mathbf{M}\mathbf{x})$. *Then*

$$\nabla s(\mathbf{x}) = \beta\, \mathbf{M}^\top \Sigma(\mathbf{p}(\mathbf{x}))\mathbf{M}, \quad \Sigma(\mathbf{p}) = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top.$$

Proof. For softmax $\mathbf{p}$, componentwise Jacobian is

$$\frac{\partial p_i}{\partial z_j} = p_i(\delta_{ij} - p_j),$$

so in matrix form $\nabla_{\mathbf{z}} \mathrm{softmax}(\mathbf{z}) = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$. With $\mathbf{z} = \beta \mathbf{M}\mathbf{x}$,

$$\nabla s(\mathbf{x}) = \mathbf{M}^\top \left(\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top\right)(\beta \mathbf{M}) = \beta\, \mathbf{M}^\top \Sigma(\mathbf{p})\mathbf{M}.$$

□

Lemma D.4 (Softmax covariance bound). *For any probability vector* $\mathbf{p} \in \Delta^{K-1}$, *define* $\Sigma(\mathbf{p}) = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$. *Then* $\Sigma(\mathbf{p}) \succeq 0$ *and* $\|\Sigma(\mathbf{p})\| \le \frac{1}{2}$.

Proof. $\Sigma(\mathbf{p})$ is the covariance matrix of a categorical one-hot variable, hence PSD. For unit $\mathbf{u}$,

$$\mathbf{u}^\top \Sigma(\mathbf{p})\mathbf{u} = \sum_i p_i u_i^2 - \left(\sum_i p_i u_i\right)^2 = \mathrm{Var}(u_I) \le \frac{(\max_i u_i - \min_i u_i)^2}{4} \le \frac{1}{2},$$

by Popoviciu's inequality and $\max_i u_i - \min_i u_i \le \sqrt{2}\|\mathbf{u}\| = \sqrt{2}$. Taking the supremum over unit $\mathbf{u}$ gives $\|\Sigma(\mathbf{p})\| \le \frac{1}{2}$. □

Proposition D.5 (Convexity and strong convexity regime). *Let* $\mu := 1 - \frac{\beta\|\mathbf{M}\|^2}{2}$. *Then the Hessian satisfies*

$$\nabla^2 E_{\mathrm{base}}(\mathbf{X}) \succeq \mu\, \mathbf{I}_{Nd} + 2\lambda(\mathbf{L} \otimes \mathbf{I}_d).$$

Consequently:

1. *if* $\beta\|\mathbf{M}\|^2 \le 2$, E_{base} *is convex;*
2. *if* $\beta\|\mathbf{M}\|^2 < 2$, E_{base} *is* μ*-strongly convex.*

Proof. For each node,

$$\nabla^2_{\mathbf{x}_v} \mathrm{lse}(\beta, \mathbf{M}\mathbf{x}_v) = \beta\, \mathbf{M}^\top \Sigma(\mathbf{p}_v)\mathbf{M}, \quad \mathbf{p}_v := \mathrm{softmax}(\beta \mathbf{M}\mathbf{x}_v).$$

By Lemma D.4,

$$\mathbf{0} \preceq \beta \mathbf{M}^\top \Sigma(\mathbf{p}_v)\mathbf{M} \preceq \frac{\beta}{2}\|\mathbf{M}\|^2 \mathbf{I}_d.$$

Hence

$$\nabla^2_{\mathbf{x}_v}\left[-\mathrm{lse}(\beta, \mathbf{M}\mathbf{x}_v) + \frac{1}{2}\|\mathbf{x}_v\|^2\right] = \mathbf{I}_d - \beta \mathbf{M}^\top \Sigma(\mathbf{p}_v)\mathbf{M} \succeq \left(1 - \frac{\beta\|\mathbf{M}\|^2}{2}\right)\mathbf{I}_d.$$

Stacking all nodes and adding the Laplacian Hessian contribution $2\lambda(\mathbf{L} \otimes \mathbf{I}_d) \succeq 0$ yields

$$\nabla^2 E_{\mathrm{base}}(\mathbf{X}) \succeq \left(1 - \frac{\beta\|\mathbf{M}\|^2}{2}\right)\mathbf{I}_{Nd} + 2\lambda(\mathbf{L} \otimes \mathbf{I}_d).$$

This implies convexity when $\beta\|\mathbf{M}\|^2 \le 2$, and μ-strong convexity when $\beta\|\mathbf{M}\|^2 < 2$. □

D.3 Smoothness and Existence of Minimizers

Proposition D.6 (L-smoothness)**.** *∇E_{base} is Lipschitz in Frobenius norm:*

$$\|\nabla E_{\text{base}}(\mathbf{X}) - \nabla E_{\text{base}}(\mathbf{Y})\|_F \le L_{\text{lip}} \|\mathbf{X} - \mathbf{Y}\|_F,$$

with

$$L_{\text{lip}} = \frac{\beta}{2}\|\mathbf{M}\|^2 + 1 + 2\lambda\|\mathbf{L}\|.$$

Proof. Define row-wise retrieval $s(\mathbf{x}) = \mathbf{M}^\top \text{softmax}(\beta \mathbf{M}\mathbf{x})$ and $[S(\mathbf{X})]_v = s(\mathbf{x}_v)$. From Proposition D.2,

$$\nabla E_{\text{base}}(\mathbf{X}) = \mathbf{X} - S(\mathbf{X}) + 2\lambda \mathbf{L}\mathbf{X}.$$

Hence

$$\|\nabla E(\mathbf{X}) - \nabla E(\mathbf{Y})\|_F \le \|\mathbf{X} - \mathbf{Y}\|_F + \|S(\mathbf{X}) - S(\mathbf{Y})\|_F + 2\lambda\|\mathbf{L}\| \, \|\mathbf{X} - \mathbf{Y}\|_F \tag{11}$$

$$= (1 + 2\lambda\|\mathbf{L}\|) \, \|\mathbf{X} - \mathbf{Y}\|_F + \|S(\mathbf{X}) - S(\mathbf{Y})\|_F. \tag{12}$$

By Lemma D.3 and Lemma D.4,

$$\|\nabla s(\mathbf{x})\| = \|\beta \mathbf{M}^\top \Sigma(\mathbf{p}) \mathbf{M}\| \le \beta \|\mathbf{M}\|^2 \|\Sigma(\mathbf{p})\| \le \frac{\beta}{2}\|\mathbf{M}\|^2.$$

Thus s is $\frac{\beta}{2}\|\mathbf{M}\|^2$-Lipschitz by the mean value theorem:

$$\|s(\mathbf{x}) - s(\mathbf{y})\| \le \frac{\beta}{2}\|\mathbf{M}\|^2 \|\mathbf{x} - \mathbf{y}\|.$$

Apply this row-wise:

$$\|S(\mathbf{X}) - S(\mathbf{Y})\|_F^2 \le \left(\frac{\beta}{2}\|\mathbf{M}\|^2\right)^2 \|\mathbf{X} - \mathbf{Y}\|_F^2,$$

so

$$\|S(\mathbf{X}) - S(\mathbf{Y})\|_F \le \frac{\beta}{2}\|\mathbf{M}\|^2 \|\mathbf{X} - \mathbf{Y}\|_F.$$

Substitute into Eq. 12 to obtain the claimed constant. □

Corollary D.7 (Step-size stability condition)**.** *For gradient descent $\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} - \eta \nabla E_{\text{base}}(\mathbf{X}^{(t)})$, any $\eta \in (0, 2/L_{\text{lip}})$ guarantees monotone descent. Since $\|\mathbf{L}\| \le 2$ for the normalized Laplacian, a convenient sufficient bound is*

$$\eta < \frac{2}{\frac{\beta}{2}\|\mathbf{M}\|^2 + 1 + 4\lambda}.$$

Proof. The first claim is the standard smooth-gradient step-size condition. The second follows from Proposition D.6 and $\|\mathbf{L}\| \le 2$. □

Proposition D.8 (Coercivity and existence of a minimizer)**.** *$E_{\text{base}}(\mathbf{X}) \to +\infty$ as $\|\mathbf{X}\|_F \to \infty$. Therefore E_{base} attains at least one global minimizer.*

Proof. For each node, with $\mathbf{z} = \mathbf{M}\mathbf{x}_v$,

$$\text{lse}(\beta, \mathbf{M}\mathbf{x}_v) = \beta^{-1} \log \sum_{\mu=1}^{K} e^{\beta z_\mu} \le \beta^{-1} \left(\log K + \beta \max_\mu z_\mu\right) \le \beta^{-1} \log K + \|\mathbf{M}\| \, \|\mathbf{x}_v\|.$$

Hence

$$E_{\text{base}}(\mathbf{X}) \ge \frac{1}{2}\|\mathbf{X}\|_F^2 - \|\mathbf{M}\| \sum_v \|\mathbf{x}_v\| - N\beta^{-1} \log K + \lambda \text{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}).$$

Using $\sum_v \|\mathbf{x}_v\| \le \sqrt{N}\|\mathbf{X}\|_F$ and $\text{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) \ge 0$ from Proposition D.1,

$$E_{\text{base}}(\mathbf{X}) \ge \frac{1}{2}\|\mathbf{X}\|_F^2 - \|\mathbf{M}\|\sqrt{N} \, \|\mathbf{X}\|_F - N\beta^{-1} \log K.$$

The right-hand side is a coercive quadratic lower bound in $\|\mathbf{X}\|_F$, so it diverges to $+\infty$ as $\|\mathbf{X}\|_F \to \infty$. Thus E_{base} is coercive. By continuity, it attains a global minimizer. □

D.4 Gradient-Descent Convergence

Theorem D.9 (Monotone descent and convergence to stationarity)**.** *Assume $\eta \in (0, 2/L_{\text{lip}})$ and iterate*

$$\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} - \eta \nabla E_{\text{base}}(\mathbf{X}^{(t)}).$$

Then:

1. *$E_{\text{base}}(\mathbf{X}^{(t)})$ is non-increasing and*

$$E_{\text{base}}(\mathbf{X}^{(t+1)}) \leq E_{\text{base}}(\mathbf{X}^{(t)}) - \eta \left(1 - \frac{\eta L_{\text{lip}}}{2}\right) \|\nabla E_{\text{base}}(\mathbf{X}^{(t)})\|_F^2;$$

2. *$\sum_{t=0}^{\infty} \|\nabla E_{\text{base}}(\mathbf{X}^{(t)})\|_F^2 < \infty$, so $\|\nabla E_{\text{base}}(\mathbf{X}^{(t)})\|_F \to 0$;*
3. *every accumulation point is stationary.*

Proof. By Proposition D.6, E_{base} is L_{lip}-smooth, so for any $\mathbf{X}, \mathbf{Y}$,

$$E(\mathbf{Y}) \leq E(\mathbf{X}) + \langle \nabla E(\mathbf{X}), \mathbf{Y} - \mathbf{X} \rangle + \frac{L_{\text{lip}}}{2} \|\mathbf{Y} - \mathbf{X}\|_F^2. \tag{13}$$

Set $\mathbf{Y} = \mathbf{X} - \eta \nabla E(\mathbf{X})$:

$$E(\mathbf{X} - \eta \nabla E(\mathbf{X})) \leq E(\mathbf{X}) - \eta \|\nabla E(\mathbf{X})\|_F^2 + \frac{L_{\text{lip}} \eta^2}{2} \|\nabla E(\mathbf{X})\|_F^2 \tag{14}$$

$$= E(\mathbf{X}) - \eta \left(1 - \frac{\eta L_{\text{lip}}}{2}\right) \|\nabla E(\mathbf{X})\|_F^2. \tag{15}$$

Applying this at $\mathbf{X} = \mathbf{X}^{(t)}$ proves item (1). Let $c := \eta(1 - \eta L_{\text{lip}}/2) > 0$. Summing from $t = 0$ to $T - 1$ gives

$$E(\mathbf{X}^{(0)}) - E(\mathbf{X}^{(T)}) \geq c \sum_{t=0}^{T-1} \|\nabla E(\mathbf{X}^{(t)})\|_F^2.$$

Using lower boundedness (Proposition D.8) and letting $T \to \infty$ yields $\sum_t \|\nabla E(\mathbf{X}^{(t)})\|_F^2 < \infty$, so norms vanish: item (2). For item (3), if $\mathbf{X}^{(t_k)} \to \mathbf{X}^\star$, then by continuity of ∇E,

$$\nabla E(\mathbf{X}^\star) = \lim_{k \to \infty} \nabla E(\mathbf{X}^{(t_k)}) = \mathbf{0}.$$

□

Theorem D.10 (Rates)**.** *Under the assumptions of Theorem D.9, let $E_{\text{inf}} := \inf_{\mathbf{X}} E_{\text{base}}(\mathbf{X})$ and $c := \eta(1 - \eta L_{\text{lip}}/2)$. Then for any $T \geq 1$,*

$$\min_{0 \leq t \leq T-1} \|\nabla E_{\text{base}}(\mathbf{X}^{(t)})\|_F^2 \leq \frac{E_{\text{base}}(\mathbf{X}^{(0)}) - E_{\text{inf}}}{cT}.$$

If $\beta \|\mathbf{M}\|^2 < 2$ (so E_{base} is μ-strongly convex with $\mu = 1 - \beta\|\mathbf{M}\|^2/2$), then for $\eta \in (0, 1/L_{\text{lip}}]$ and minimizer $\mathbf{X}^\star$,

$$\|\mathbf{X}^{(t)} - \mathbf{X}^\star\|_F \leq (1 - \eta\mu)^t \|\mathbf{X}^{(0)} - \mathbf{X}^\star\|_F,$$

and

$$E_{\text{base}}(\mathbf{X}^{(t)}) - E_{\text{base}}(\mathbf{X}^\star) \leq (1 - \eta\mu)^t \left(E_{\text{base}}(\mathbf{X}^{(0)}) - E_{\text{base}}(\mathbf{X}^\star)\right).$$

Proof. For the first claim, Theorem D.9 gives

$$E(\mathbf{X}^{(t+1)}) \leq E(\mathbf{X}^{(t)}) - c\|\nabla E(\mathbf{X}^{(t)})\|_F^2.$$

Summing from $t = 0$ to $T - 1$ and using $E(\mathbf{X}^{(T)}) \geq E_{\text{inf}}$,

$$c \sum_{t=0}^{T-1} \|\nabla E(\mathbf{X}^{(t)})\|_F^2 \leq E(\mathbf{X}^{(0)}) - E_{\text{inf}}.$$

Divide by T and lower bound the average by the minimum.

For strong convexity, let $\mathbf{X}^\star \in \arg\min_{\mathbf{X}} E(\mathbf{X})$ (unique). By smoothness and strong convexity, for each t there exists symmetric $\mathbf{A}^{(t)}$ with $\mu\mathbf{I} \preceq \mathbf{A}^{(t)} \preceq L_{\text{lip}}\mathbf{I}$ such that

$$\nabla E(\mathbf{X}^{(t)}) - \nabla E(\mathbf{X}^\star) = \mathbf{A}^{(t)}(\mathbf{X}^{(t)} - \mathbf{X}^\star).$$

Since $\nabla E(\mathbf{X}^\star) = 0$,

$$\mathbf{X}^{(t+1)} - \mathbf{X}^\star = \left(\mathbf{I} - \eta\mathbf{A}^{(t)}\right)(\mathbf{X}^{(t)} - \mathbf{X}^\star).$$

Therefore

$$\|\mathbf{X}^{(t+1)} - \mathbf{X}^\star\|_F \le \max_{\lambda\in[\mu,L_{\text{lip}}]} |1-\eta\lambda| \, \|\mathbf{X}^{(t)} - \mathbf{X}^\star\|_F.$$

For $\eta \le 1/L_{\text{lip}}$, the maximum is $1-\eta\mu$, giving the iterate rate. For function values, combine

$$E(\mathbf{X}^{(t+1)}) - E(\mathbf{X}^\star) \le E(\mathbf{X}^{(t)}) - E(\mathbf{X}^\star) - \eta\left(1 - \frac{\eta L_{\text{lip}}}{2}\right)\|\nabla E(\mathbf{X}^{(t)})\|_F^2,$$

with gradient domination for μ-strongly convex functions,

$$\|\nabla E(\mathbf{X}^{(t)})\|_F^2 \ge 2\mu\left(E(\mathbf{X}^{(t)}) - E(\mathbf{X}^\star)\right),$$

and $\eta \le 1/L_{\text{lip}}$ to conclude

$$E(\mathbf{X}^{(t+1)}) - E(\mathbf{X}^\star) \le (1-\eta\mu)\left(E(\mathbf{X}^{(t)}) - E(\mathbf{X}^\star)\right).$$

□

D.5 Fixed-Point Contraction and Damping

Define

$$[T(\mathbf{X})]_v = \mathbf{M}^\top \text{softmax}(\beta\mathbf{M}\mathbf{x}_v) - 2\lambda(\mathbf{L}\mathbf{X})_v.$$

Remark D.11 (Gradient descent vs. fixed-point iteration). It is useful to separate two schemes:

1. **Gradient descent**: $\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} - \eta\nabla E_{\text{base}}(\mathbf{X}^{(t)})$. Under L_{lip}-smoothness and $\eta \in (0, 2/L_{\text{lip}})$, this guarantees monotone energy descent.
2. **Fixed-point iteration**: $\mathbf{X}^{(t+1)} = T(\mathbf{X}^{(t)})$. This does not automatically imply descent; global convergence is guaranteed only if T is a contraction.

Theorem D.12 (Fixed-point contraction condition)**.** *The map T is globally Lipschitz with*

$$\text{Lip}(T) \le \rho := \frac{\beta}{2}\|\mathbf{M}\|^2 + 2\lambda\|\mathbf{L}\|.$$

If $\rho < 1$, then T is a contraction: it has a unique fixed point $\mathbf{X}^\star$ and $\mathbf{X}^{(t+1)} = T(\mathbf{X}^{(t)})$ converges geometrically to $\mathbf{X}^\star$ from any initialization.

Proof. Write $T(\mathbf{X}) = S(\mathbf{X}) - 2\lambda\mathbf{L}\mathbf{X}$. By the proof of Proposition D.6, S is $\frac{\beta}{2}\|\mathbf{M}\|^2$-Lipschitz, and $\|\mathbf{L}(\mathbf{X}-\mathbf{Y})\|_F \le \|\mathbf{L}\|\,\|\mathbf{X}-\mathbf{Y}\|_F$. Therefore

$$\|T(\mathbf{X}) - T(\mathbf{Y})\|_F \le \left(\frac{\beta}{2}\|\mathbf{M}\|^2 + 2\lambda\|\mathbf{L}\|\right)\|\mathbf{X}-\mathbf{Y}\|_F.$$

Hence $\text{Lip}(T) \le \rho$. If $\rho < 1$, Banach's fixed-point theorem gives existence and uniqueness of $\mathbf{X}^\star$ and

$$\|\mathbf{X}^{(t)} - \mathbf{X}^\star\|_F \le \rho^t\|\mathbf{X}^{(0)} - \mathbf{X}^\star\|_F.$$

□

Corollary D.13 (Damped map)**.** *For $T_\alpha(\mathbf{X}) = (1-\alpha)\mathbf{X} + \alpha T(\mathbf{X})$ with $\alpha \in (0,1]$,*

$$\text{Lip}(T_\alpha) \le (1-\alpha) + \alpha\rho.$$

Hence if $\rho < 1$, every $\alpha \in (0,1]$ yields a contraction.

Proof. For any $\mathbf{X}, \mathbf{Y}$,

$$\|T_\alpha(\mathbf{X}) - T_\alpha(\mathbf{Y})\|_F \le (1-\alpha)\|\mathbf{X}-\mathbf{Y}\|_F + \alpha\|T(\mathbf{X}) - T(\mathbf{Y})\|_F,$$

then apply Theorem D.12. □

Corollary D.14 (Convenient contraction condition for normalized Laplacian)**.** *Since $\|\mathbf{L}\| \le 2$ for normalized $\mathbf{L}$, a sufficient condition for undamped fixed-point contraction is*

$$\frac{\beta}{2}\|\mathbf{M}\|^2 + 4\lambda < 1.$$

Proof. Immediate from Theorem D.12 and $\|\mathbf{L}\| \le 2$. □

D.6 Critical Points and Initialization

Proposition D.15 (Critical points)**.** *$\mathbf{X}^\star$ is first-order critical iff $\nabla E_{\text{base}}(\mathbf{X}^\star) = 0$. Equivalent fixed-point form:*

$$\mathbf{X}^\star = T(\mathbf{X}^\star).$$

If $\nabla^2 E_{\text{base}}(\mathbf{X}^\star) \succ 0$, then $\mathbf{X}^\star$ is a strict local minimizer; if $\nabla^2 E_{\text{base}}(\mathbf{X}^\star)$ has a negative eigenvalue, then $\mathbf{X}^\star$ is a strict saddle.

Proof. First-order: by definition, $\mathbf{X}^\star$ is critical iff $\nabla E_{\text{base}}(\mathbf{X}^\star) = 0$. Using Proposition D.2, this is equivalent to $\mathbf{X}^\star = T(\mathbf{X}^\star)$.

For second-order classification, Taylor-expand around $\mathbf{X}^\star$:

$$E(\mathbf{X}^\star + \Delta) = E(\mathbf{X}^\star) + \langle \nabla E(\mathbf{X}^\star), \Delta\rangle + \frac{1}{2}\langle \Delta, \nabla^2 E(\mathbf{X}^\star)\Delta\rangle + o(\|\Delta\|_F^2).$$

At a critical point the linear term is zero. If $\nabla^2 E(\mathbf{X}^\star) \succ 0$, the quadratic term is positive for all sufficiently small nonzero Δ, so $\mathbf{X}^\star$ is a strict local minimizer. If $\nabla^2 E(\mathbf{X}^\star)$ has a negative eigenvalue, pick $\Delta = \epsilon \mathbf{v}$ along a corresponding eigenvector and small $\epsilon > 0$ to make the quadratic term negative, yielding a strict saddle. □

Proposition D.16 (When initialization does not matter)**.** *If $\beta\|\mathbf{M}\|^2 < 2$, gradient descent converges to the unique global minimizer from any initialization. If $\rho < 1$ in Theorem D.12, fixed-point iteration converges to the unique fixed point from any initialization.*

Proof. The first claim is Theorem D.10 in the strongly convex regime. The second claim follows from Banach's fixed-point theorem. □

Remark D.17 (When initialization can matter). Outside the strongly convex/contraction regimes, E_{base} can be nonconvex with multiple stationary points. Then basin geometry and algorithmic choices (step size, damping, stopping) can change which stationary point is reached.