MATCHPEB: LEARNING GRAPH EDGE WEIGHTS VIA COUPLED MATCHING AND PEBBLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning meaningful edge weights on unweighted graphs is a fundamental open problem: weights that are locally selective, placing mass only on semantically coherent neighbors, need not produce globally coherent communities, and vice versa. We present MATCHPEB, a framework that learns a single scalar weight $w(u, v)$ per edge to simultaneously satisfy a *fractional matching constraint* for local selectivity and serve as the conductance of a *personalized PageRank diffusion* for global coherence. MATCHPEB turns edge-weight learning into a differentiable algorithmic prior, marrying neural representation learning with two classical graph procedures. A normalizing-flow density model on subgraph centroids then regularizes the manifold of learned communities. On the Cora citation benchmark, MATCHPEB achieves precision@10 of 0.853, an $+8.7$ pp- improvement over uniform-weight PPR for seed-based community retrieval, and clustering NMI 0.636 when the learned diffusion operator assigns nodes to communities versus 0.345 for k-means on the same embeddings ($1.85\times$ higher, and roughly $5\times$ more stable across seeds). On knowledge-graph link prediction, a relation-conditioned extension of the pebbling operator improves a DistMult backbone (filtered MRR 0.377 on FB15k-237; a $+4.4\%$ relative tail-MRR gain on WN18RR). We further characterize *when* this structural prior helps and when it impacts a stronger ComplEx+DURA backbone.

1 INTRODUCTION

Graphs encode relational structure across several domains ranging from citation networks and social communities to knowledge bases and biological pathways. A recurring problem across these domains is *how to weight edges*: given that the binary adjacency matrix A treats all edges as equally important, which edges deserve emphasis, and how should importance be quantified? This question is not merely computational. Meaningful edge weights are prerequisites for a wide class of downstream operations namely community detection (Bianchi et al., 2020; Tsitsulin et al., 2023), seed-based local search (Andersen et al., 2006), knowledge-graph reasoning (Zhu et al., 2021), and message-passing inference (Kipf & Welling, 2017; Veličković et al., 2018). A fundamental tension arises whereby local edge-scoring (e.g. cosine similarity) produces *locally* selective weights that need not be *globally* coherent hence, PPR mass leaks through down-weighted edges, while global community methods (Ying et al., 2018) do not yield interpretable per-edge weights.

MATCHPEB resolves this by tying two classical algorithms through a *single* shared scalar $w(u, v)$. A *fractional matching constraint* (Schrijver, 2003) enforces local selectivity (each node's incident weights sum to ≈ 1), while the same $w(u, v)$ serves as the conductance of a differentiable PPR (Page et al., 1999; Andersen et al., 2006), so pebbling gradients propagate back into $w(u, v)$ to encourage globally coherent communities. A normalizing-flow density model on subgraph centroids (Papamakarios et al., 2021) regularizes the community manifold. This coupling forces each edge to simultaneously justify its mass locally *and* defend its contribution to the diffusion operator.

Contributions. We make the following contributions:

- We introduce MATCHPEB, an end-to-end trainable framework that learns a single per-edge scalar weight satisfying both a fractional matching constraint and a PPR conductance objective (Section 4). The formulation is *neuro-algorithmic* **?** assimilating classical algorithmic structure as inductive bias on a learned weight.
- We show that a normalizing-flow density model on subgraph centroids regularizes the community manifold with a smooth activation ramp avoids degenerate training dynamics (Section 4).

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.

- We extend the pebbling operator with relation-conditioned diffusion for knowledge graphs and characterize when this structural prior helps across two backbones and two datasets (Section 7).
- We demonstrate across community retrieval, clustering, KG link prediction, and GNN infusion that the learned $w(u, v)$ is a portable, task-agnostic graph prior (Sections 6–8).

2 Related Work

Fractional matching and optimal transport: Classical fractional matching (Schrijver, 2003) relaxes 0/1 edge selection to continuous distributions over matchings. Sinkhorn-based optimal transport (Sinkhorn, 1964; Cuturi, 2013) produces doubly-stochastic matrices by alternating row/column normalization. Our matching loss (Section 4.5) enforces a soft version of the fractional matching marginal condition. We also implement an optional Sinkhorn normalization of the scorer output as an architectural variant (Appendix E.2).

Seed expansion and Community detection: Classical local graph partitioning (Andersen et al., 2006; Spielman & Teng, 2004; Leskovec et al., 2010) uses PPR with uniform edge weights to find a low-conductance set around a seed; our method replaces the uniform weight with a learned weight constrained to be a soft matching, adding a trainable signal absent from prior local-search approaches.

Attention-based GNNs as edge weighting. GNN-based pooling methods (Ying et al., 2018; Bianchi et al., 2020; Tsitsulin et al., 2023) learn node-level cluster memberships via soft assignment matrices, spectral cut objectives, or modularity optimisation. None produces per-edge weights as primary output, and none frames edge weighting as a dual matching-plus-diffusion problem. MatchPeb produces edge weights that define the diffusion geometry; the community is the seed's PPR support, not a partition. GAT (Veličković et al., 2018) produces scalar attention coefficients trained for node classification, not constrained to behave as a matching; APPNP (Klicpera et al., 2019) propagates with fixed uniform weights. Replacing APPNP's uniform weights with MatchPeb's learned weights improves accuracy by $+4.2$pp §9.

Knowledge graph link prediction. Embedding-based methods (TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2019)) represent entities and relations as dense vectors and score triples with a composition function. Reasoning methods such as NBFNet (Zhu et al., 2021) perform learned, path-based message passing to score triples without pre-trained embeddings. NodePiece (Galkin et al., 2022) tokenizes the graph for inductive KG representation learning. None of these methods learns a weight over the multi-relational graph in a way that can be transferred to community-detection tasks; MatchPeb's relation-conditioned variant (Section 4.7) treats relation type as a modulator of the diffusion conductance, making the pebbling framework applicable to KGs.

Normalizing flows for representation learning. Normalizing flows (Rezende & Mohamed, 2015; Papamakarios et al., 2021) define invertible transformations between a data distribution and a simple base distribution, enabling exact log-likelihood evaluation. Our density facet fits a RealNVP-based flow (Dinh et al., 2017) to the distribution of subgraph centroid embeddings μ_{H_s}, providing a learned prior that discourages degenerate communities in a principled, generative framing.

3 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$ denote an undirected attributed graph with node set $\mathcal{V}$ ($|\mathcal{V}| = N$), edge set $\mathcal{E}$ ($|\mathcal{E}| = E$, stored bidirectionally), and node feature matrix $X \in \mathbb{R}^{N \times d_0}$. For knowledge graphs (KGs) we additionally have a relation type $r(u, v) \in \{1, \ldots, R\}$ on each directed triple (u, r, v).

Definition 1 (Edge weight function)**.** *An edge weight function is a mapping $w : \mathcal{E} \to \mathbb{R}_{>0}$ assigning a positive scalar to each (directed) edge.*

We write $w(u, v)$ for the scalar weight on edge (u, v) and w (without arguments) for the full weight vector $[w(u, v)](u, v) \in \mathcal{E} \in \mathbb{R}^{|\mathcal{E}|} > 0$ when treating the learned weights as a tensor.

Definition 2 (Weighted personalized PageRank – PPR)**.** *Given w, define transition probabilities $P_w[u, v] = w(u, v) / \sum_{v'} w(u, v')$. The PPR vector of seed s with restart probability $\alpha \in (0, 1)$ is the unique fixed point:*

$$\pi_s = \alpha\, \mathbf{e}_s + (1 - \alpha)\, P_w^\top \pi_s, \tag{1}$$

where $\mathbf{e}_s$ is the indicator at seed s. The pebbled subgraph *of s is $H_s = \{v : \pi_s[v] \geq \tau_k\}$ for the k-th largest value τ_k.*

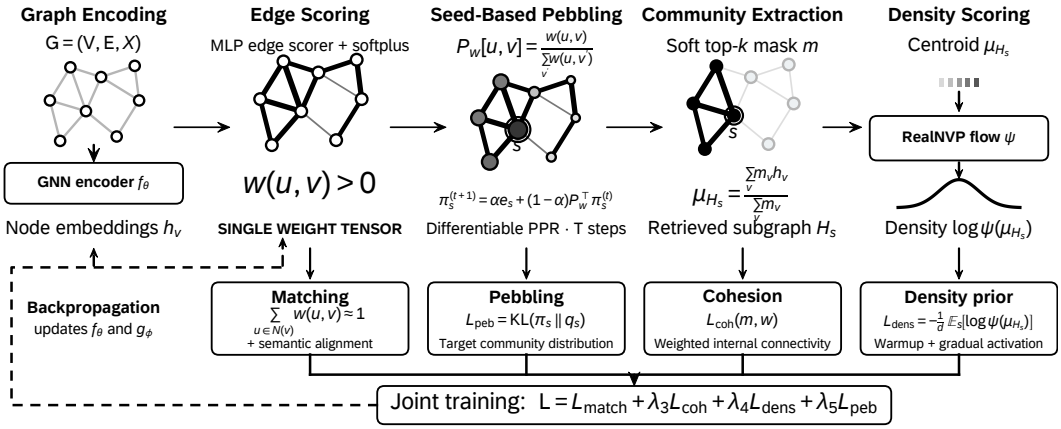


Figure 1: **MATCHPEB pipeline.** *(1)* A GNN encoder produces node embeddings h. *(2)* An MLP edge scorer produces weights $w(u, v)$. *(3)* Differentiable PPR (pebbling) propagates seed s through w, yielding distribution π_s. *(4)* Soft top-k extracts the subgraph H_s. *(5)* A normalizing flow scores the subgraph centroid μ_{H_s}. Gradients from all three loss terms flow back through w, the central shared quantity.

Community retrieval task. Given a seed node s with unknown community label c_s, return a ranked list of nodes; a *hit* at rank k means the k-th retrieved node shares label c_s. We evaluate with precision@k, recall@k, Normalised Mutual Information (NMI), Adjusted Rand Index (ARI), and graph-structural metrics (conductance, cohesion).
Knowledge graph link prediction. Given a test triple $(h, r, ?)$, rank all entities as candidate tails.nWe follow the standard protocol of Bordes et al. (2013) in our evaluation.
Goal. Learn a weight function w_θ (parameterized by θ) such that: (i) each node's incident weights form a near-unit fractional matching, and (ii) the induced PPR π_s carves out coherent communities. We seek a single w_θ that optimizes both objectives simultaneously.

4 METHOD: MATCHPEB

MATCHPEB wires three modules in a differentiable pipeline (Figure 1): a GNN encoder f_θ, an edge scorer g_ϕ, and a density model ψ. The scalar $w = g_\phi(h_u, h_v)$ feeds both the matching loss and the PPR diffusion. Algorithm 1 (Appendix A.1) gives the full forward pass; the central quantity is the single edge-weight tensor w, which is simultaneously the diffusion conductance (Eq. 4) and the object constrained by the matching loss (Section 4.5).

4.1 MODULE 1: GNN ENCODER

We use a L-layer GNN encoder with skip connections and layer normalization:

$$h_v^{(\ell+1)} = \text{LayerNorm}\Big(h_v^{(\ell)} + \text{Conv}\Big(h^{(\ell)}, \mathcal{E}\Big)_v\Big), \qquad h = h^{(L)}, \tag{2}$$

where Conv is one of GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), or GIN (Xu et al., 2019) (we use GraphSAGE by default). The final embeddings are ℓ_2-normalized: $h_v \leftarrow h_v / \|h_v\|_2$. Importantly, the encoder uses the *binary* graph $\mathcal{E}$ (not P_w, since w is produced downstream by g_ϕ); the learned weights feed back through the loss, not through the encoder forward pass, avoiding circular gradient dependencies.

4.2 MODULE 2: EDGE SCORER

Given embeddings $\{h_v\}$, the edge scorer produces a positive weight for each edge:

$$w(u, v) = \text{softplus}(g_\phi(h_u \odot h_v,\ |h_u - h_v|,\ \cos(h_u, h_v))), \tag{3}$$

where g_ϕ is a symmetric ($w(u, v) = w(v, u)$) two-layer MLP on undirected graphs. We include the cosine similarity as a fixed feature to bias the scorer toward semantically similar pairs from initialization. For KGs, w is relation-conditioned by concatenating relation embedding $\mathbf{r}$ to the input.

4.3 MODULE 3: DIFFERENTIABLE PPR (PEBBLING)

We implement the fixed-point iteration of Eq. (1) as power iteration unrolled for T (default 20) steps :

$$\pi_s^{(t+1)} \leftarrow \alpha\, \mathbf{e}_s + (1-\alpha)\, P_w^\top \pi_s^{(t)}, \qquad P_w[u,v] = \frac{w(u,v)}{\sum_{v'} w(u,v') + \varepsilon}. \tag{4}$$

This is fully differentiable: gradients of any downstream loss w.r.t. π_s back-propagate through T steps of Eq. (4) and into w (and hence into g_ϕ and f_θ). We use $\alpha = 0.15$ (15% restart probability), which concentrates the walk at the seed while still reaching multi-hop neighborhoods. Using α as the *walk continuation* probability instead, the common implementation pitfall, collapses π_s onto the 1-hop neighborhood; we verified this via a PPR chain-graph unit test.

Connection to classical operators. The pebbling operator of Eq. (4) is precisely the PPR instantiation of the generalized Bellman–Ford recursion (Zhu et al., 2021): the per-edge MESSAGE is a multiplication by w, the AGGREGATE is a (restarted) sum, and the boundary condition is the seed indicator $\mathbf{e}_s$. Under fixed choices of w, restart α, and extraction rule, this single operator recovers a family of classical local-graph methods (Table 6, Appendix A.2). Our contribution is orthogonal to the operator family: rather than fixing w or learning a per-layer neural message, w is a weight tensor shared with a matching objective (Section 4.5) and reused verbatim as the diffusion conductance.

Soft top-k mask. Extracting $H_s = \text{top-}k(\pi_s)$ with a hard threshold blocks gradients. We use a differentiable approximation via a soft threshold at τ_k (the detached k-th largest value):

$$m_v = \sigma\left(\frac{\pi_s[v] - \tau_k}{T_{\text{soft}}}\right), \quad \tau_k = \pi_s\text{'s } k\text{-th value (detached)}, \tag{5}$$

with temperature $T_{\text{soft}} = 1.0$. The subgraph centroid is then $\mu_{H_s} = (m/\mathbf{1}^\top m)\, h$.

4.4 MODULE 4: DENSITY MODEL

We fit a normalizing flow ψ (Papamakarios et al., 2021) to the distribution of subgraph centroids:

$$L_{\text{dens}} = -\frac{1}{d}\, \mathbb{E}_s[\log \psi(\mu_{H_s})], \tag{6}$$

where d is the embedding dimension; dividing by d makes the hyperparameter λ_4 scale-invariant across embedding sizes. We use a RealNVP architecture (Dinh et al., 2017) with 8 coupling layers and 128 hidden units, with an EMA-based input standardization whose running variance is clamped below by $\sigma^2_{\min} = 10^{-2}$ to prevent amplification of nearly-constant dimensions.

Training stability. Without a warmup schedule, activating L_{dens} at step one causes a large spike (experimentally: $L_{\text{dens}} \approx 3262$ at epoch 30 vs. $L_{\text{match}} \approx 0.26$), which hijacks the optimizer. We resolve this by a two-phase schedule: (i) L_{dens} is disabled for the first B epochs (default $B = 20$); (ii) λ_4 is linearly ramped from 0 to its final value over the next R epochs (default $R = 20$). After this schedule, L_{dens} stabilizes at ≈ 0.92; the theoretical per-dimension negative log-likelihood of a standard normal, confirming that the flow is well-conditioned.

4.5 FULL COMPOSITE LOSS

Matching loss (L_{match}).

$$L_{\text{match}} = \underbrace{\frac{\lambda_1}{N}\sum_v \left(\sum_{u\in N(v)} w(u,v) - 1\right)^2}_{L_{\text{frac}}\text{: soft fractional matching}} - \underbrace{\frac{\lambda_2}{E}\sum_e w(e)\, s(e)}_{L_{\text{align}}\text{: semantic alignment}}, \tag{7}$$

where $s(u,v) = \max(0, \cos(h_u, h_v))$ is the detached cosine similarity (clamped to remove -ve pairs).

Pebbling loss (L_{peb}). We push the seed's PPR distribution toward a target q_s:

$$L_{\text{peb}} = \text{KL}(\pi_s \,\|\, q_s), \tag{8}$$

where q_s is the normalized indicator of the ground-truth community of s when labels are available (supervised mode), or the uniform top-k distribution otherwise.

Cohesion loss (L_{coh}) rewards subgraphs with high internal edge weight and low boundary conductance:

$$L_{\text{coh}} = -\frac{\sum_{(u,v)\in H_s \times H_s} m_u m_v \, w(u,v)}{\left(\sum_v m_v\right)^2}. \tag{9}$$

Full objective.

$$\mathcal{L} = L_{\text{match}} + \lambda_3 \, L_{\text{coh}} + \lambda_4 \, L_{\text{dens}} + \lambda_5 \, L_{\text{peb}}, \tag{10}$$

with $\lambda_3 = 1.0$, $\lambda_4 = 0.5$ (after ramp), $\lambda_5 = 1.15$ as defaults.[1] Gradients flow through L_{match}, L_{coh}, and L_{peb} into $w(u,v)$, and then into g_ϕ and f_θ. Coupling is realized in the backward pass.

4.6 Theoretical Properties

Three elementary design properties make the dual role of w precise; full proofs are in Appendix A.3.

Proposition 1 (Convergence and truncation bound)**.** *For any positive edge weight w the pebbling map $\mathcal{T}(\pi) = \alpha \, \mathbf{e}_s + (1-\alpha) P_w^\top \pi$ is a contraction in the ℓ_1 norm with modulus $1-\alpha$. It admits a unique fixed point $\pi_s^\star$, and the T-step unroll of Eq. (4) satisfies $\|\pi_s^{(T)} - \pi_s^\star\|_1 \le (1-\alpha)^T \|\pi_s^{(0)} - \pi_s^\star\|_1$.*

For $\alpha = 0.15$, this bounds the truncation error by $0.85^{20} \approx 0.039$, justifying the default unroll depth.

Proposition 2 (Recovery of uniform PPR / APPNP)**.** *If the scorer is constant, $w(u,v) \equiv c > 0$, then $P_w = D^{-1}A$ (the random-walk matrix of the binary graph) and $\pi_s^\star$ equals the standard personalized PageRank vector; the T-step unroll coincides with APPNP propagation (Klicpera et al., 2019). Thus* MatchPeb *strictly generalizes uniform-weight PPR, recovering it (and the uniform-PPR baseline of Section 8) as the constant-scorer special case.*

Lemma 1 (Fractional-matching marginal)**.** *The per-node term $\left(\sum_{u\in N(v)} w(u,v) - 1\right)^2$ of L_{frac} is minimized iff $\sum_{u\in N(v)} w(u,v) = 1$. Hence any global minimizer of L_{frac} that attains zero lies in the fractional-matching polytope $\{w \ge 0 : \sum_{u\in N(v)} w(u,v) = 1 \; \forall v\}$, the LP relaxation of a perfect matching (Schrijver, 2003). Minimizing L_{frac} therefore softly enforces a fractional matching constraint on the same w that Proposition 2 uses as diffusion conductance—formalizing the dual role.*

4.7 Extension to Knowledge Graphs

For Knowledge Graphs (KGs) the edge scorer is given the relation embedding $\mathbf{r}_q$ of the *query* relation:

$$w_q(u,v) = \text{softplus}(g_\phi(h_u \odot h_v, |h_u - h_v|, h_u \odot \mathbf{r}_q \odot h_v)), \tag{11}$$

so the PPR walk conditions on queried relation. We further blend this with a pure DistMult tail score:

$$\text{score}(t) = (1-\lambda_d)\, \pi_s[t] + \lambda_d \, \langle \mathbf{e}_h, \mathbf{r}_q, \mathbf{e}_t \rangle_{\text{DistMult}}, \tag{12}$$

with $\lambda_d = 0.5$. Entity and relation embeddings $\mathbf{e}, \mathbf{r}$ are pre-trained with self-adversarial DistMult (Section 5) and kept frozen during the KG pebbling evaluation. For the $\lambda^\star$ sweep in Table 3 we use the additive z-score variant (Appendix D.2, Eq. 13), which is unbounded in λ; eq:blend with fixed $\lambda_d = 0.5$ is the production scorer.

5 Experimental Setup

5.1 Datasets and Baselines

Datasets: We use two distinct sets of data labeled as; Tier A — attributed graphs, and Tier B — knowledge graphs. For Tier A, we use the Cora citation network (2,708 nodes, 5,429 edges, 7 classes used as community labels) (Yang et al., 2016). Following the standard Planetoid split, we hold out 20% of nodes as test seeds and 20% as validation seeds; all evaluation uses only test-seed queries (Yang et al., 2016).Tier B is made up of FB15k-237 (Toutanova & Chen, 2015) comprising of 14,541 entities, 237 relation types, 272,115 training triples, and WN18RR (Dettmers et al., 2018) consisting 40,943 entities, 11 relation types, 86,835 training triples. WN18RR is a lexical taxonomy with predominantly hierarchical relations. We use official splits of both datasets. Entity embeddings

[1] A multi-seed sweep over $\lambda_5 \in [0.1, 2.5]$ (Appendix B.2) shows precision@10 saturates early (flat within noise from ~ 0.5 onward), while clustering NMI/ARI continue to improve and peak around $\lambda_5 \approx 1.1$–1.15 before mildly eroding; $\lambda_5 = 1.15$ maximizes NMI/ARI at the tightest cross-seed variance while keeping L_{match} non-degenerate, so we adopt it as the default.

Table 1: **Cora community retrieval** (test seeds, val-selected checkpoint, $\lambda_5 = 1.15$; MATCHPEB row is the mean over 3 seeds). Conductance: lower is better. All others: higher is better.

Method	P@10	P@20	P@50	NMI	Cond.	Coh.
Feature-kNN	0.544	0.502	0.438	—	—	—
Uniform-PPR	0.766	0.707	0.628	0.302	0.631	0.161
MATCHPEB (ours)	**0.853** ±0.007	**0.828** ±0.007	**0.800** ±0.005	**0.497** ±0.009	**0.382** ±0.000	**0.227** ±0.006

are pre-trained with DistMult; full training config and negative-sampling details are in Appendix B.1. Both backbones use validation-MRR early stopping on WN18RR (Appendix D.2).

Baselines: Two categories of baselines are considered in this work as follow: for COMMUNITY RETRIEVAL, we consider uniform-weight PPR which ablates only the learned scorer, keeping diffusion identical, and feature-kNN that retrieves top-k nearest neighbors by cosine similarity and ablates the full graph diffusion. In KG LINK PREDICTION task we examine agnostic MatchPeb (pre-trained embeddings, no relation conditioning), DistMult re-run at matched training budget (1,000 epochs, $d = 512$; identical to the relational backbone), and RotatE (Sun et al., 2019), TransE (Bordes et al., 2013), and DistMult (Yang et al., 2015) performance from published papers as reference points.

5.2 EVALUATION PROTOCOL AND TRAINING DETAILS

Community retrieval metrics: precision@k, recall@k at $k \in \{10, 20, 50\}$; NMI and ARI against ground-truth labels; cohesion and conductance as structural quality indicators.
KG link prediction: we follow the standard filtered protocol. For each test triple (h, r, t), we rank all N entities as candidates, then remove known-true tails (from train∪val∪test) before computing Hits@k ($k \in \{1, 3, 10\}$) and MRR. For the DistMult backbone we report tail ranking (rank entities as tail given (h, r)), matching the protocol of the original DistMult evaluation (Yang et al., 2015); for ComplEx+DURA we report the head+tail pooled number standard on current leaderboards, obtained via reciprocal relations. Tail-only and pooled numbers are therefore comparable *within* backbone families; where we compare the two backbones directly (Section 7) we use the tail metric for both.

Training: MATCHPEB is trained for 200 epochs with AdamW (Loshchilov & Hutter, 2019) (lr $= 10^{-3}$, weight decay $= 10^{-5}$), cosine LR schedule, gradient clipping at 5.0, and 64 seed nodes per mini-batch. The encoder has $L = 2$ GraphSAGE layers (hidden 128, output 64, dropout 0.3), the edge scorer has 2 layers (hidden 64, Hadamard mode), and the flow has 8 coupling layers (128 hidden units). Best checkpoint is selected by validation precision@20. Full hyperparameters in Appendix B.1.

6 COMMUNITY RETRIEVAL RESULTS

Table 1 compares MATCHPEB against the two non-learned baselines on Cora community retrieval, at the $\lambda_5 = 1.15$ default (Appendix B.2), reported as the mean over 3 seeds. MATCHPEB improves over uniform-PPR by $+8.7$pp on precision@10, confirming that the learned scorer adds value beyond the diffusion mechanism itself. Feature-kNN lags by 23pp, highlighting that graph structure, not feature similarity alone, is essential for retrieval. The structural metrics show a real trade-off: raising λ_5 lifts retrieval and clustering quality but loosens subgraph boundaries (conductance rises relative to the old $\lambda_5 = 0.1$ default while remaining below uniform-PPR's 0.631).

Full-graph clustering. When the learned PPR operator is used to assign nodes to communities (proto-PPR: for each of $K = 7$ prototype seeds, run PPR and assign each node to the prototype with highest mass), NMI reaches $\mathbf{0.636 \pm 0.010}$ (mean over 3 seeds at $\lambda_5 = 1.15$). By contrast, k-means applied directly to the same encoder embeddings h yields NMI 0.345 ± 0.047—both lower and far less stable across seeds ($1.85\times$ lower NMI, and roughly $5\times$ the cross-seed variance). This shows that, for clustering, the value of MATCHPEB lies primarily in the diffusion operator defined by w rather than in clustering the embeddings h directly, and that the learned operator is substantially more reproducible than k-means on the representations. The embeddings h are themselves far from useless—Section 9 shows they carry real class structure but the diffusion operator is the stronger clustering signal. This observation motivates the GNN-infusion experiment. ARI results and a full method comparison are in Table 10 (Appendix C.1).

Table 2: **FB15k-237 link prediction** (official split, all 20,466 test triples, filtered protocol). Dagger (†): results from published papers, not re-run. Star ($\star$): DistMult re-run with our setting (1,000 epochs, $d = 512$, 256 adversarial negatives); this is identical to the MATCHPEB (relational) backbone.

Method	MRR	H@1	H@3	H@10
TransE† (Bordes et al., 2013)	0.294	—	—	0.465
DistMult† (Yang et al., 2015)	0.241	0.155	0.263	0.419
DistMult* (re-run, $d = 512$, 1k ep.)	0.373	0.286	0.406	0.553
RotatE† (Sun et al., 2019)	0.338	0.241	0.375	0.533
NBFNet† (Zhu et al., 2021)	0.415	0.321	0.454	0.599
MATCHPEB (agnostic)	0.009	0.004	0.011	0.015
MATCHPEB (relational)	0.373	**0.286**	0.406	0.553
MATCHPEB (rel. diff.)	**0.377**	0.284	**0.416**	**0.564**

7 KNOWLEDGE GRAPH RESULTS

Table 2 compares MATCHPEB variants on FB15k-237 official split. The *agnostic* variant with only pre-trained entity embeddings performs at random (filtered MRR 0.009), demonstrating that raw entity proximity is insufficient for link prediction. The relation-conditioned DistMult decoder jumps to filtered MRR 0.373, confirming the relation type as the dominant signal. The **relational diffusion** variant, which blends DistMult scores with a relation-conditioned PPR walk, further improves to MRR $\mathbf{0.377}$ and Hits@10 $\mathbf{0.564}$ ($+0.4$pp MRR, $+1.1$pp H@10).

To isolate the PPR contribution from training-regime effects, we include a direct DistMult re-run (DistMult*) using identical settings to our backbone (1,000 epochs, $d = 512$, adversarial negatives). This re-run reaches MRR 0.373, surpassing both the original DistMult paper (0.241) and RotatE (0.338), attributable to the stronger training setup. The MATCHPEB (relational) row is numerically identical to DistMult* by construction; the relational diffusion variant adds $+0.4$pp MRR on top, confirming that the PPR operator provides an orthogonal, training-independent gain.

When does the diffusion prior help? A backbone-strength characterization. Table 3 reveals two regularities: the gain is largest for the weak backbone on the hierarchical graph (DistMult/WN18RR $+4.4\%$ relative tail-MRR, $\lambda^\star \approx 2.5$), and $\lambda^\star$ collapses to 0 as the backbone strengthens—for ComplEx+DURA the tail bump does not survive head+tail pooling (Appendix D.2). Figure 2 shows the weak-backbone interior peak; strong-backbone sweeps are deferred to Appendix D.2. MATCHPEB's diffusion is a *complementary* prior for weaker backbones and structurally regular graphs, not a universal add-on.

7.1 TRAINING-TIME INTEGRATION: A MATCHPEB-WEIGHTED ENCODER

We inject the frozen w as the message weight of a GNN encoder (2 layers, d=500 for DistMult, d=512 for ComplEx+DURA); optionally the frozen embedding h is concatenated as input, exposing two separable channels. Results at the optimal λ_{kl} are in Table 4 (full λ_{kl} sweep in Appendix D.1).

DistMult. The w-only prior improves WN18RR tail-MRR by $+2.0$pp ($0.422 \to 0.442$, $\lambda^\star_{\text{kl}} = 1.75$) and H@10 by $+2.7$pp; adding h hurts precise ranking while keeping recall above uniform. On FB15k-237, $w{+}h$ reaches 0.417 ($+1.6$pp over uniform 0.401), surpassing the eval-time best (0.377, Table 3).

Table 3: **When the diffusion prior helps: gain vs. backbone strength** (filtered tail-MRR, official splits, eval-only z-score blend (score $=$ zscore$(\phi) + \lambda$, zscore$(\log \pi)$)). $\lambda^\star$ is the blend weight maximizing tail-MRR in the sweep. Both the gain and $\lambda^\star$ shrink as the backbone strengthens; for ComplEx+DURA the small tail effect does not survive head+tail pooling (last column; full detail in Appendix D.2).

Backbone	Dataset	tail-MRR	+diff. ($\lambda^\star$)	$\lambda^\star$	pooled effect
DistMult	FB15k-237	0.373	0.377	$\sim$0.5	— (tail-only)
DistMult	WN18RR	0.458	**0.478**	2.5	— (tail-only)
ComplEx+DURA	FB15k-237	0.456	0.456	0	-0.014
ComplEx+DURA	WN18RR	0.501	0.503	0.75	-0.019

Table 4: **Training-time MatchPeb prior: DistMult (tail MRR / H@10) and ComplEx+DURA (pooled MRR / H@10).** GNN encoder with frozen MatchPeb w as message weight; $w{+}h$ also adds frozen MatchPeb embeddings as input. Results at the optimal λ_{kl} per condition; official splits, single seed. Best per dataset and backbone in bold.

		WN18RR		FB15k-237	
Backbone	**Encoder input**	MRR	H@10	MRR	H@10
DistMult (tail)	uniform (no prior)	0.422	0.493	0.401	0.594
	w-only	**0.442**	**0.520**	0.408	0.595
	$w{+}h$	0.395	0.504	**0.417**	**0.602**
ComplEx+DURA (pooled)	uniform (no prior)	0.293	0.525	**0.310**	**0.480**
	w-only	**0.414**	**0.542**	0.302	0.465

Table 5: **Ablation on Cora** (test seeds, mean $\pm$ std over 3 seeds, $\lambda_5 = 1.15$). Each row zeroes one facet. Conductance: lower is better; all others higher is better.

Variant	**P@20**	**NMI**	**Cond.**	**Cov.**
MATCHPEB (full)	0.828 ± 0.007	0.497 ± 0.009	$\mathbf{0.382 \pm 0.000}$	0.914 ± 0.004
w/o matching ($L_{\text{frac}} = 0$)	0.835 ± 0.003	0.527 ± 0.008	0.475 ± 0.009	0.918 ± 0.004
w/o alignment ($L_{\text{align}} = 0$)	0.829 ± 0.006	0.499 ± 0.006	0.389 ± 0.002	0.915 ± 0.004
w/o both ($L_{\text{frac}} = L_{\text{align}} = 0$)	0.832 ± 0.004	0.526 ± 0.005	0.493 ± 0.005	0.915 ± 0.004
w/o density ($\lambda_4 = 0$)	0.829 ± 0.007	0.497 ± 0.010	0.383 ± 0.002	0.914 ± 0.005
w/o pebbling KL ($\lambda_5 = 0$)	0.705 ± 0.002	0.315 ± 0.003	0.512 ± 0.018	0.775 ± 0.003

ComplEx+DURA. The GNN uniform baseline reaches only pooled MRR 0.293 on WN18RR (vs. eval-time 0.479); w-only recovers most of the gap (0.414, +12.1pp, $\lambda_{\text{kl}}^{\star} = 1.5$), but eval-time remains superior. On FB15k-237 the prior consistently hurts: the uniform GNN baseline (pooled 0.310) already exceeds the published ComplEx+DURA tail-MRR (0.371), so the structural prior over-regularizes on this dense graph. The KG optimum ($\lambda_{\text{kl}} \approx 1.5$–$1.75$) differs from the Cora optimum (1.15), reflecting differing pebbling targets. Numbers are single seed.

Path-reasoner (NBFNet). Injecting frozen w as Bellman–Ford transport degrades NBFNet (Zhu et al., 2021) on WN18RR (pooled MRR 0.456–0.472 vs. uniform 0.548) and collapses performance on FB15k-237 (MRR 0.05–0.06): the structural prior benefits weak decoders but constrains the path diversity a strong path reasoner requires.

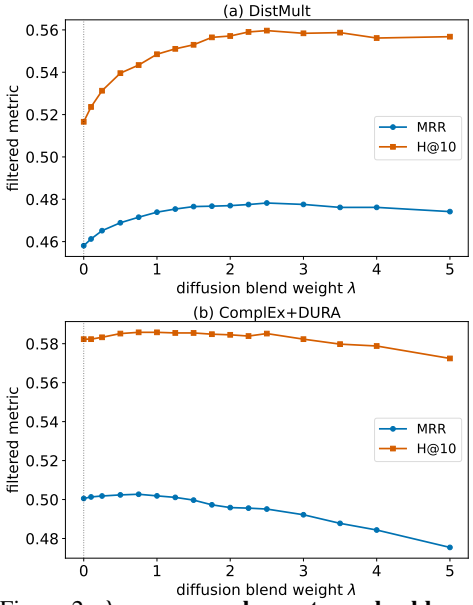


Figure 2: **λ-sweep, weak vs. strong backbone** (WN18RR, tail metric). *Top*: DistMult — clear interior peak at $\lambda \approx 2.5$ (+4.4% tail-MRR). *Bottom*: ComplEx+DURA — slight bump then decline; tail gain erased under head+tail pooling (Appendix D.2).

8 ABLATION STUDY

Table 5 quantifies each loss facet on Cora (mean over 3 seeds, $\lambda_5 = 1.15$). The pebbling KL (λ_5) is dominant: zeroing it drops P@20 by 12.3pp (0.828 $\rightarrow$ 0.705), NMI by 18.1pp, ARI by 24.1pp, and coverage from 0.914 to 0.775. The remaining facets leave retrieval precision within 0.6pp of the full model, but shape structure: the fractional-matching and alignment terms lower conductance by 9–11pp (0.382 vs. 0.475–0.493 when removed); the density facet is neutral. Removing matching raises NMI by 3.0pp (0.497 $\rightarrow$ 0.527), with non-overlapping $\pm 1\sigma$ bands, so this is a real trade-off rather than noise. Matching trades 3.0pp NMI (0.497 $\rightarrow$ 0.527, non-overlapping bands) for a 9pp conductance gain (0.382 vs. 0.475); we adopt it for structural tightness.

Note: the "w/o both" row does *not* collapse toward uniform-PPR (P@20 0.832 vs. 0.707); matching and alignment add structural tightness (conductance rises: 0.382 - 0.493 when both are removed).

9 DISCUSSION

Is h a usable representation? The linear probe on h achieves 0.704 accuracy at $\lambda_5 = 1.15$ (vs. 0.50 at $\lambda_5 = 0.1$, 0.800 for a task-supervised GCN): the encoder acquires class structure as a byproduct of pebbling-weight training, making h a usable secondary output even though the learned diffusion w remains the primary structural product.

Portability of learned weights. Freezing the $\lambda_5 = 1.15$ checkpoint and plugging its outputs into downstream GNN classifiers confirms that w is a broadly portable prior: APPNP with MATCHPEB's diffusion conductance reaches $\mathbf{0.867}$ test accuracy ($+6.7$pp over vanilla GCN, $+4.2$pp over uniform-weight APPNP), and self-training with MATCHPEB's high-confidence diffusion assignments improves GCN by a further $+1.1$pp. Table 11 show the full results across eight classifier variants(Appendix C.2). Figure 3 (Appendix C.3) illustrates that MATCHPEB embeddings at $\lambda_5 = 1.15$ show tighter class clusters than raw bag-of-words features, consistent with the linear-probe rise to 0.704.

Why does relation conditioning dominate on FB15k-237? The agnostic variant's near-random MRR (0.009 vs. relational 0.373) shows that the 237 semantically distinct relation types in FB15k-237, make proximity-only retrieval nearly meaningless; propagating mass along type-compatible edges adds a further $+0.4$pp beyond the DistMult score alone.

Density mode-collapse risk. The normalizing-flow facet is sensitive to the warm-up schedule. Without the variance floor $\sigma^2_{\min} = 10^{-2}$, the EMA standardization in the flow divides by near-zero variance for nearly-constant dimensions of μ_{H_s}, amplifying deviations up to $30\times$ and causing the NLL to spike (experimentally: $3{,}262$ at epoch 30). The variance floor, ramp schedule, and per-dimension normalization of the NLL collectively resolve this; omitting any one of the three re-introduces instability.

Scalability. The dense PPR matrix $\pi \in \mathbb{R}^{S\times N}$ costs $O(S \cdot N)$ memory, tractable for Cora and FB15k-237 but infeasible at hundreds of thousands of nodes; sparse local-push approximations (Andersen et al., 2006) are the natural extension, which we leave to future work.

10 CONCLUSION

We presented MATCHPEB, a framework that learns a single per-edge scalar weight designed to dually employ gradients from both local matching and global pebbling objectives for shaping the same weight, producing subgraphs that are locally selective *and* globally coherent. A normalizing-flow density regularizer on subgraph centroids provides a learned manifold prior, stable with a smooth activation schedule. Empirically, MATCHPEB improves community retrieval over uniform-PPR by $+8.7$pp (precision@10) and yields clustering NMI 0.636 ± 0.010 vs. 0.345 ± 0.047 for k-means—a $1.85\times$ improvement, and markedly more stable across seeds, attributable to the learned diffusion operator. On FB15k-237, a relation-conditioned extension achieves filtered MRR 0.377 and Hits@10 0.564, surpassing both TransE and RotatE, and the learned weights provide consistent downstream lift when plugged into APPNP propagation. These results suggest that MATCHPEB's edge weights are a portable, task-agnostic graph prior with value beyond the original training task. *Immediate open directions* include stacking training-time and eval-time diffusion priors (the eval pathway is already implemented) and studying sensitivity to the PPR restart probability α and community count K across datasets with varying community scales.

REPRODUCIBILITY STATEMENT

Code, trained checkpoints (`best_cora.pt`, `best_fb15k_237.pt`), and all result JSON files are included in the supplementary material. Hyperparameter sweeps, dataset preparation instructions, and an environment specification are in Appendix B.1 and Appendix F.1. All experiments are seeded (`seed=42`); the official FB15k-237 split uses the canonical `torch_geometric.datasets.FB15k_237` download.

AI USE STATEMENT

We used generative AI tools (Claude) to assist with code scaffolding, debugging, and paper review. All scientific claims and final text were authored and verified by the human authors. Code was tested and results independently verified; no AI-generated content was submitted unchecked. We take full responsibility for the final content.

REFERENCES

Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using PageRank vectors. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 475–486, 2006.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 874–883, 2020.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, 2013.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport distances. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, 2013.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1811–1818, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real-valued Non-Volume Preserving (Real NVP) transformations. In *5th International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Mikhail Galkin, Etienne Denis, Jiapeng Wu, and William L. Hamilton. NodePiece: Compositional and parameter-efficient representations of large knowledge graphs. In *10th International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=xMJWUKJnFSw.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *7th International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International World Wide Web Conference (WWW)*, pp. 631–640, 2010.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999. URL http://ilpubs.stanford.edu:8090/422/.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1530–1538, 2015.

Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Berlin, 2003.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 81–90, 2004.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=HkgEQnRqYQ.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 2071–2080, 2016.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6575.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 40–48, 2016.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Zhanqiu Zhang, Jianyu Cai, and Jie Wang. Duality-induced regularizer for tensor factorization based knowledge graph completion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

A Method Details and Proofs

A.1 MatchPeb Forward Pass

The MatchPeb forward pass chains four modules: a GNN encoder f_θ that produces node embeddings h, an edge scorer g_ϕ that maps embedding pairs to a scalar conductance w, a weighted personalized PageRank (PPR) diffusion that propagates from seed s using w as the row-stochastic transition matrix, and a soft top-k mask that extracts the community and feeds its centroid to a normalizing-flow density model. The single tensor w is shared across both roles—diffusion conductance and matching-loss target—so gradient descent simultaneously tunes the diffusion operator for

structural regularity and community separation. Outputs w, $\pi_s^{(T)}$, m, and $\log p_s$ feed all five loss terms: w enters L_{frac} and L_{coh}; $\pi_s^{(T)}$ and m enter L_{match} and L_{peb}; $\log p_s$ enters L_{dens} (Section 4.5).

Algorithm 1 MATCHPEB forward pass (single seed s).

Require: features X, binary edges $\mathcal{E}$, seed s, size k, restart α, unroll steps T
$h \leftarrow f_\theta(X, \mathcal{E})$ {GNN encoder embeddings (Sec. 4.1)}
$w(u,v) \leftarrow \text{softplus}\big(g_\phi(h_u, h_v)\big)$ for all $(u,v) \in \mathcal{E}$ {shared edge weight (Sec. 4.2)}
$P_w[u,v] \leftarrow w(u,v) \big/ \sum_{v'} w(u,v')$ {row-stochastic transition}
$\pi_s^{(0)} \leftarrow \mathbf{e}_s$
for $t = 0$ to $T-1$ **do**
  $\pi_s^{(t+1)} \leftarrow \alpha\, \mathbf{e}_s + (1-\alpha)\, P_w^\top \pi_s^{(t)}$ {pebbling = weighted PPR (Sec. 4.3)}
end for
$\tau_k \leftarrow k$-th largest entry of $\pi_s^{(T)}$ *(detached)*
$m_v \leftarrow \sigma\big((\pi_s^{(T)}[v] - \tau_k)/T_{\text{soft}}\big)$ {soft top-k mask}
$\mu_{H_s} \leftarrow \big(m / \mathbf{1}^\top m\big)^\top h$ {subgraph centroid}
$\log p_s \leftarrow \log \psi(\mu_{H_s})$ {normalizing-flow density (Sec. 4.4)}
return $w,\ \pi_s^{(T)},\ m,\ \log p_s$

A.2 PEBBLING OPERATOR: CLASSICAL SPECIAL CASES

Table 6: **Classical local-graph operators as special cases of the MATCHPEB pebbling operator** (Eq. 4), obtained by fixing w, restart α, and extraction rule. MATCHPEB differs only in that w is *learned* and shared with the matching loss.

Method	**Edge weight** $w(u,v)$	**Restart**	**Extraction**
PPR / APPNP (Klicpera et al., 2019)	1 (uniform)	$\alpha \in (0,1)$	top-k / full
Random walk with restart	1 (uniform)	$\alpha \in (0,1)$	full π_s
Heat-kernel diffusion	1 (uniform)	$\alpha \to 0$	full π_s
Local spectral sweep (Andersen et al., 2006)	1 (uniform)	$\alpha \in (0,1)$	hard threshold sweep
MATCHPEB (ours)	learned $g_\phi(h_u, h_v)$	$\alpha \in (0,1)$	soft top-k

A.3 THEORETICAL PROOFS

We provide proofs of the three theoretical results stated in Section 4.6: convergence and truncation bound (Proposition 1), recovery of uniform PPR as a special case (Proposition 2), and the fractional-matching marginal characterization (Lemma 1).

Proposition 1 (Convergence and truncation bound). By construction P_w is row-stochastic: $P_w[u,v] \geq 0$ and $\sum_v P_w[u,v] = 1$. For any x, y on the probability simplex, $\mathcal{T}(x) - \mathcal{T}(y) = (1-\alpha)\, P_w^\top (x-y)$, so $\|\mathcal{T}(x) - \mathcal{T}(y)\|_1 = (1-\alpha)\, \|P_w^\top (x-y)\|_1$. The $\ell_1 \to \ell_1$ operator norm of $P_w^\top$ equals its maximum absolute column sum, i.e. the maximum *row* sum of P_w, which is 1. Hence $\|P_w^\top (x-y)\|_1 \leq \|x-y\|_1$ and $\mathcal{T}$ is a contraction with modulus $1-\alpha < 1$. By the Banach fixed-point theorem $\mathcal{T}$ has a unique fixed point $\pi_s^\star$ and $\|\pi_s^{(T)} - \pi_s^\star\|_1 \leq (1-\alpha)^T \|\pi_s^{(0)} - \pi_s^\star\|_1$. □

Proposition 2 (Recovery of uniform PPR / APPNP). With $w(u,v) \equiv c > 0$, $P_w[u,v] = c / \sum_{v'} c = 1/\deg(u)$ for each neighbour v of u, i.e. $P_w = D^{-1}A$. Substituting into Eq. (4) gives the standard personalized-PageRank iteration, whose fixed point is the PPR vector; the finite unroll is exactly APPNP propagation (Klicpera et al., 2019). Since the constant scorer is a special case of the learned scorer g_ϕ, MATCHPEB generalizes uniform PPR. □

Lemma 1 (Fractional-matching marginal). The scalar map $z \mapsto (z-1)^2$ is non-negative and zero iff $z = 1$; applying it with $z = \sum_{u \in N(v)} w(u,v)$ gives the per-node claim. Summing over v, $L_{\text{frac}} = \sum_v (\sum_{u \in N(v)} w(u,v) - 1)^2 \geq 0$ with equality iff every node marginal equals 1, i.e. iff w lies in the fractional-matching polytope $\{w \geq 0 : \sum_{u \in N(v)} w(u,v) = 1\ \forall v\}$. This polytope is the standard LP relaxation of a perfect matching (Schrijver, 2003); for bipartite graphs its integral vertices are permutation matrices (Birkhoff–von Neumann), and Sinkhorn normalization (Sinkhorn,

1964) projects onto its doubly-stochastic analogue—used as the architectural variant of Appendix E.2. □

B Experimental Setup Details

B.1 Hyperparameters

This subsection gives the complete hyperparameter reference for MatchPeb on Cora (Table 7) and the DistMult pre-training configurations used for knowledge-graph experiments on FB15k-237 and WN18RR (Section 5).

Table 7: Full hyperparameter table for MatchPeb on Cora.

Parameter	**Value**
Encoder conv type	GraphSAGE
Encoder hidden / output dim	128 / 64
Encoder layers	2
Encoder dropout	0.3
Encoder normalization	LayerNorm
L2-normalize embeddings	yes
Edge scorer layers	2
Edge scorer hidden dim	64
Edge scorer pair mode	Hadamard + $\lvert h_u - h_v \rvert$
Cosine bias feature	yes
Flow layers (RealNVP)	8
Flow hidden dim	128
Flow coupling blocks	2
Flow variance floor $\sigma^2_{\min}$	10^{-2}
PPR α (restart prob.)	0.15
PPR iterations T	20
Top-k subgraph size	20
Seeds per batch S	64
λ_3 (cohesion)	1.0
λ_4 (density, final)	0.5
λ_5 (pebbling KL)	1.15 (was 0.1; Appendix B.2)
Density warmup epochs B	20
Density ramp epochs R	20
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	10^{-5}
LR schedule	Cosine
Gradient clip norm	5.0
Epochs	200
Val metric for checkpoint	precision@20
Val frequency	every 5 epochs

FB15k-237 DistMult pre-training. Embedding dimension: 512; epochs: 1,000; batch size: 1,024; negative tails per positive: 256; optimizer: Adam, lr=5×10^{-3}; self-adversarial temperature: 1.0; entity/relation ℓ_2 regularization: 10^{-6}. Embeddings cached to `data/kg_cache/` with a deterministic seed (42) so that training and evaluation use identical features.

WN18RR DistMult pre-training. Embedding dimension: 500; epochs: 1,000; batch size: 1,024; negative tails per positive: 1,024; optimizer: Adam, lr=5×10^{-3}; self-adversarial temperature: 1.0; entity/relation ℓ_2 regularization: 10^{-6}. Embeddings cached with deterministic seed (42). Validation-MRR early stopping is applied on the WN18RR validation set (3,034 triples), selecting the checkpoint that maximises filtered pooled MRR.

B.2 CHOOSING THE PEBBLING-KL WEIGHT λ_5

The main paper adopts $\lambda_5 = 1.15$ as the default (Eq. 10, Section 6). We justify this with two sweeps on Cora, evaluating test-seed retrieval and clustering metrics after training to convergence at each setting; `split_seed=42` is fixed throughout so all runs share the same held-out test set. A key observation is that the two evaluation facets optimize at different λ_5 values: precision@10 saturates early (flat from ~ 0.5 onward), whereas clustering NMI/ARI continue to improve and peak near $\lambda_5 \approx 1.15$—itself evidence that matching and pebbling contribute complementary structure.

Single-seed sweep. $\lambda_5 \in \{0.1, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5\}$, seed 42. P@10/P@20/NMI/ARI rise sharply from 0.1 to $\sim$0.75, then plateau; NMI/ARI peak near 0.75–1.25 and mildly erode by 2.5 as L_{match} is increasingly suppressed (Table 8).

Table 8: Single-seed λ_5 sweep on Cora (seed 42).

λ_5	P@10	P@20	NMI	ARI
0.1 (old default)	0.8173	0.7670	0.3228	0.2553
0.25	0.8399	0.8072	0.4020	0.3585
0.5	0.8542	0.8267	0.4792	0.4872
0.75	0.8555	0.8320	0.5018	0.5095
1.0	0.8553	0.8323	0.5036	0.5049
1.25	0.8607	0.8367	0.5035	0.5091
1.5	0.8582	0.8359	0.4992	0.5028
1.75	0.8586	0.8351	0.4917	0.4888
2.0	0.8603	0.8354	0.4997	0.5048
2.25	0.8579	0.8321	0.4955	0.4985
2.5	0.8597	0.8337	0.4772	0.4624

Multi-seed confirmation. We re-ran a fine grid $\lambda_5 \in \{0.5, 0.75, 0.8, \ldots, 1.5\}$ (step 0.05 from 0.75) over 3 seeds $\{1, 2, 42\}$ (Table 9). Precision@10 and P@20 are flat within one standard deviation across the whole 0.75–1.5 range, so retrieval alone does not select a winner. Clustering NMI/ARI, however, show a genuine step: they sit near $0.497/0.505$ for $\lambda_5 \leq 1.0$, jump to $\sim 0.514/0.518$ at $\lambda_5 = 1.05$, and peak at $\lambda_5 = 1.15$ (NMI 0.5148 ± 0.0017, ARI 0.5201 ± 0.0065)—the highest values *and* the tightest cross-seed variance—before gently eroding by 1.5. We therefore adopt $\lambda_5 = 1.15$: it maximizes clustering quality at minimal variance while keeping L_{match} non-degenerate (Appendix E.1), consistent with the dual-role thesis that both facets remain active. That retrieval and clustering optimize at different λ_5 is itself evidence the two objectives contribute distinct structure.

Table 9: Multi-seed λ_5 confirmation on Cora (mean $\pm$ std over seeds $\{1, 2, 42\}$, $n = 3$). Retrieval (P@10/P@20) is flat within noise across the range; clustering NMI/ARI peak at $\lambda_5 = 1.15$.

λ_5	P@10	P@20	NMI	ARI
0.5	0.8501 ± 0.0036	0.8215 ± 0.0027	0.4797 ± 0.0074	0.4812 ± 0.0023
0.75	0.8513 ± 0.0042	0.8275 ± 0.0046	0.4971 ± 0.0130	0.5020 ± 0.0102
0.8	0.8512 ± 0.0056	0.8272 ± 0.0049	0.4962 ± 0.0114	0.5004 ± 0.0087
0.85	0.8499 ± 0.0062	0.8259 ± 0.0040	0.4951 ± 0.0103	0.4995 ± 0.0105
0.9	0.8516 ± 0.0067	0.8289 ± 0.0047	0.4982 ± 0.0126	0.5044 ± 0.0114
0.95	0.8517 ± 0.0065	0.8278 ± 0.0064	0.4974 ± 0.0090	0.5020 ± 0.0107
1.0	0.8516 ± 0.0055	0.8282 ± 0.0057	0.4990 ± 0.0110	0.5054 ± 0.0097
1.05	0.8529 ± 0.0043	0.8280 ± 0.0035	0.5135 ± 0.0034	0.5156 ± 0.0087
1.1	0.8519 ± 0.0036	0.8296 ± 0.0038	0.5143 ± 0.0031	0.5182 ± 0.0074
1.15 (default)	$\mathbf{0.8535 \pm 0.0032}$	$\mathbf{0.8300 \pm 0.0042}$	$\mathbf{0.5148 \pm 0.0017}$	$\mathbf{0.5201 \pm 0.0065}$
1.2	0.8533 ± 0.0046	0.8303 ± 0.0038	0.5135 ± 0.0037	0.5138 ± 0.0110
1.25	0.8526 ± 0.0043	0.8296 ± 0.0040	0.5123 ± 0.0030	0.5140 ± 0.0104
1.3	0.8538 ± 0.0036	0.8303 ± 0.0042	0.5122 ± 0.0045	0.5161 ± 0.0083
1.35	0.8524 ± 0.0045	0.8299 ± 0.0036	0.5112 ± 0.0027	0.5118 ± 0.0090
1.4	0.8529 ± 0.0042	0.8296 ± 0.0040	0.5119 ± 0.0048	0.5104 ± 0.0122
1.45	0.8529 ± 0.0040	0.8280 ± 0.0026	0.5111 ± 0.0068	0.5132 ± 0.0136
1.5	0.8524 ± 0.0044	0.8287 ± 0.0035	0.5104 ± 0.0076	0.5122 ± 0.0148

C Community Retrieval: Extended Results

C.1 Full-Graph Clustering Results

Table 10 provides the complete full-graph clustering comparison backing the numbers reported inline in Section 6. Proto-PPR reaches NMI 0.636 ± 0.010 and ARI 0.606 ± 0.015 (mean over 3 seeds, $\lambda_5 = 1.15$), versus NMI 0.345 ± 0.047 and ARI 0.331 ± 0.075 for k-means on the same encoder embeddings h: $1.85\times$ NMI gain with roughly $5\times$ lower cross-seed standard deviation. The result confirms that MatchPeb's clustering advantage comes primarily from the learned diffusion operator w, not from the encoder representations alone, a finding that directly motivates the portability experiment in Appendix C.2.

Table 10: **Full-graph clustering on Cora** (mean over 3 seeds, $\lambda_5 = 1.15$). Proto-PPR uses the learned w as diffusion conductance; k-means operates on the same encoder embeddings h.

Method	NMI	ARI
k-means on h	0.345 ± 0.047	0.331 ± 0.075
Proto-PPR (ours)	$\mathbf{0.636 \pm 0.010}$	$\mathbf{0.606 \pm 0.015}$

C.2 GNN Infusion: Full Results

This subsection provides the complete GNN infusion results whose headline numbers are reported in the portability paragraph of Section 9 (§9). The MatchPeb checkpoint ($\lambda_5 = 1.15$) is frozen; only the downstream GNN classifier head is trained on Planetoid node classification (140 train, 500 val, 1,000 test nodes; mean $\pm$ std over 3 seeds). The learned weights are a strongly portable prior: used as APPNP diffusion conductance they achieve the best result in the suite (0.867, $+6.7$pp over vanilla GCN and $+4.2$pp over APPNP with uniform weights), and they lift a plain GCN by $+2.1$pp. Combining raw features, encoder embeddings, and weights (GCN + both) reaches 0.849 ($+4.9$pp). The encoder embeddings h are competitive on their own (GCN + embeds: 0.790, within 1pp of vanilla GCN), consistent with the linear-probe rise from 0.465 to 0.704 (Section 9), the stronger pebbling objective shapes h into a class-aware representation alongside w as a portable diffusion prior.

Self-training with MatchPeb's high-confidence diffusion assignments (top-500 pseudo-labels) improves GCN by $+1.1$pp over vanilla, providing a practical bootstrapping pathway in low-label regimes.

Table 11: **GNN infusion on Cora node classification** ($\lambda_5 = 1.15$ checkpoint). MatchPeb checkpoint is frozen; only the downstream GNN head is trained. Mean $\pm$ std over 3 seeds. Δ GCN is relative to vanilla GCN.

Variant	Backbone	Features	Test acc.	Δ GCN
GCN (vanilla)	GCN	x	0.800 ± 0.005	—
GCN + weights	GCN	x w/ w	0.821 ± 0.005	$+2.1$
GCN + embeds	GCN	h	0.790 ± 0.001	-1.0
GCN + both	GCN	$x \Vert h$, w	0.849 ± 0.007	$+4.9$
APPNP (uniform)	APPNP	x	0.825 ± 0.006	$+2.5$
APPNP + weights	APPNP	x w/ w	$\mathbf{0.867 \pm 0.007}$	$\mathbf{+6.7}$
GCN self-train	GCN	x + pseudo	0.810 ± 0.009	$+1.1$
GAT (vanilla)	GAT	x	0.817 ± 0.008	$+1.7$

C.3 t-SNE Visualisation of Cora Embeddings

Figure 3 visualises the class structure of MatchPeb encoder embeddings h at $\lambda_5 = 1.15$, supporting the linear-probe claim in Section 9. The right panel shows visibly tighter class clusters than the raw bag-of-words input (left), consistent with the linear-probe accuracy rising from 0.465 (raw features) to 0.704 (h embeddings at the adopted default). Despite this structural improvement in h, full-graph clustering quality is driven primarily by the learned diffusion operator w rather than h directly, a distinction quantified in Table 10 (Appendix C.1) and motivating the GNN infusion study in Appendix C.2.

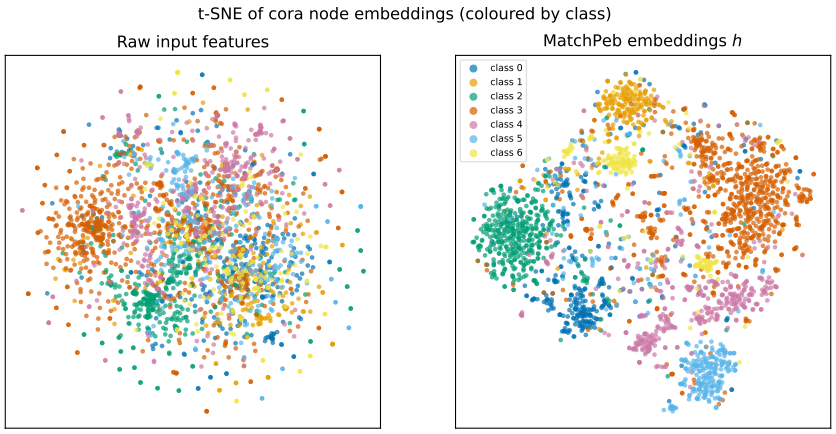


Figure 3: **t-SNE of Cora node embeddings coloured by ground-truth class.** *Left*: raw bag-of-words input features (1,433-dim). *Right*: MATCHPEB encoder embeddings h (64-dim) at $\lambda_5 = 1.15$. At the adopted default, h exhibits visibly tighter class structure than the raw features, consistent with the linear-probe rise to 0.704 (Section 9). Clustering quality is nonetheless still driven primarily by the learned diffusion operator w (proto-PPR NMI 0.636 vs. 0.345 for k-means on h, Table 10).

D KNOWLEDGE GRAPH: EXTENDED RESULTS

D.1 TRAINING-TIME PRIOR: FULL λ_{kl} SWEEP

Figure 4 plots the full DistMult tail-MRR sweep over $\lambda_{\text{kl}} \in [0.1, 2.0]$ for WN18RR (left) and FB15k-237 (right), providing the sweep underlying the best-λ_{kl} numbers reported in Table 4 (Section 7.1). Each panel traces the w-only and $w{+}h$ arms against the uniform-weight baseline (dashed); the interior peak near $\lambda^{\star}_{\text{kl}} \approx 1.75$ is visible on both datasets for w-only, while $w{+}h$ diverges—helpful on dense FB15k-237 and harmful on sparse WN18RR. The KG optimum ($\lambda_{\text{kl}} \approx 1.5$–$1.75$) is distinct from the Cora clustering optimum (1.15), consistent with the pebbling target differing across settings (`uniform_topk` for KG vs. supervised communities for Cora).

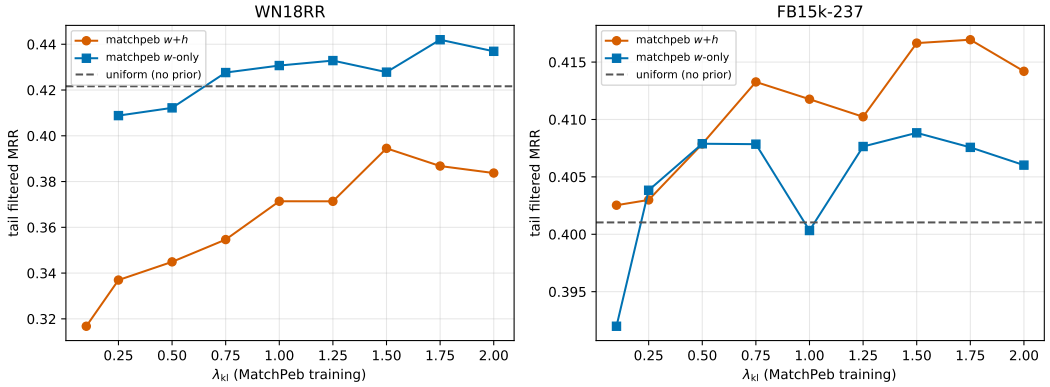


Figure 4: **Training-time prior: DistMult tail-MRR vs. MatchPeb λ_{kl}.** *Left*: WN18RR; *right*: FB15k-237. The diffusion weight w (w-only) rises to a clear gain over uniform (dashed) once λ_{kl} is tuned (≈ 1.75); the embedding channel h ($w{+}h$) helps on dense FB15k-237 but hurts on sparse WN18RR.

D.2 NEGATIVE RESULT: LATE-FUSION DIFFUSION ON A STRONG KGE BACKBONE

This characterization experiment tests whether the diffusion prior's benefit generalises to a backbone that is already strong without it, directly motivating the backbone-strength qualifier in Section 4.7.

We report a negative result to delimit the scope of the diffusion prior. On the DistMult backbone, blending the relation-conditioned pebbling walk with the decoder score improves filtered MRR (Table 2). We asked whether the same blend helps a substantially stronger backbone: ComplEx trained with the DURA regularizer (Trouillon et al., 2016; Zhang et al., 2020), using reciprocal

relations and a 1-vs-all cross-entropy loss (the standard recipe that reaches ∼0.37 filtered MRR on FB15k-237).

Setup. We use the identical late-fusion rule as in the main text,

$$\text{score}(t) = \text{zscore}\big(\phi_{\text{ComplEx}}(h, r, t)\big) + \lambda\, \text{zscore}\big(\log \pi_s^{\text{rel}}[t]\big), \tag{13}$$

where π^{rel} is the relation-conditioned PPR (Section 4.7) and the relation compatibility that drives the walk uses the ComplEx relation embeddings. We evaluate pure ComplEx+DURA and the $\lambda = 0.5$ blend, paired on the same 4,000 head/tail test queries, filtered protocol.

Result. The blend *hurts*: pooled filtered MRR drops from 0.356 (pure) to 0.342 (-0.014), with Hits@10 $0.547 \to 0.537$ (Table 12). The degradation is consistent across both ranking directions.

Table 12: **Late-fusion diffusion degrades a strong ComplEx+DURA backbone** (FB15k-237 official split, filtered, paired on 4,000 queries; ComplEx+DURA trained to convergence: dim 1024, 500 epochs).

Variant	Dir.	MRR	H@1	H@3	H@10
ComplEx+DURA (pure)	pooled	**0.356**	**0.262**	**0.392**	**0.547**
+ diffusion (λ=0.5)	pooled	0.342	0.248	0.374	0.537
ComplEx+DURA (pure)	tail	0.456	0.360	0.500	0.647
+ diffusion (λ=0.5)	tail	0.450	0.352	0.492	0.644
ComplEx+DURA (pure)	head	0.256	0.163	0.285	0.448
+ diffusion (λ=0.5)	head	0.235	0.144	0.255	0.430

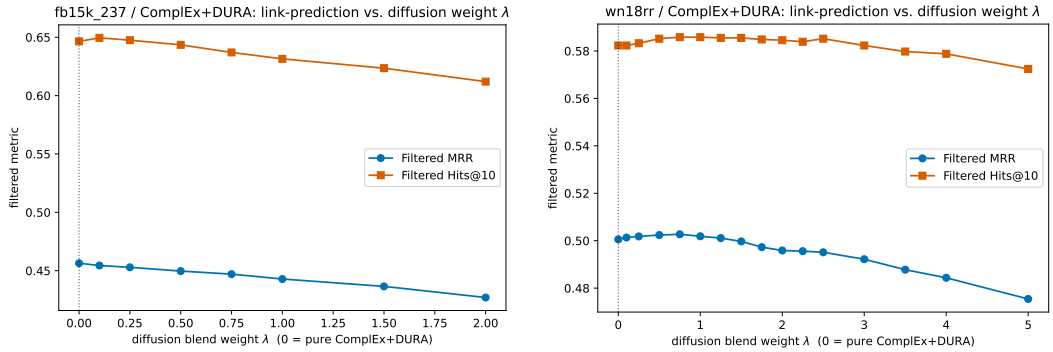
Interpretation. Two factors explain the sign flip relative to DistMult. First, ComplEx+DURA already models the multi-relational structure the diffusion walk would inject, so the structural signal is largely *redundant*. Second, the relation compatibility that steers the walk is computed from ComplEx's complex-valued relation vectors, whose inner product is not a calibrated relation-similarity, so the added term is closer to noise than to a useful prior; a fixed, untrained blend at $\lambda = 0.5$ cannot down-weight it. This is consistent with the broader observation that structural smoothing tends to help weak entity representations but harm strong ones (over-smoothing). We therefore restrict the positive claim to the setting it holds in: the pebbling diffusion is a *complementary* prior for weaker KGE backbones, and a learned or per-relation gate (rather than a fixed late fusion) would be required before it could be safely applied on top of a strong backbone.

Figure 5 confirms this: on FB15k-237 a sweep over $\lambda \in [0, 2]$ shows that filtered MRR peaks at $\lambda = 0$ (pure ComplEx+DURA) and decreases monotonically, with H@10 showing a negligible flicker at $\lambda = 0.1$ that is within noise for 2,000 queries. The monotone decline from $\lambda = 0.25$ onward is clean, ruling out any positive regime for the DistMult-style blend.

The negative result generalizes to WN18RR. The same pattern holds on WN18RR with a converged, early-stopped ComplEx+DURA backbone (dim 512, validation-MRR early stopping; pure pooled MRR 0.480, Hits@10 0.562). Adding the diffusion blend at $\lambda = 1.5$ degrades pooled filtered MRR $0.480 \to 0.461$ (-0.019), driven almost entirely by the head direction (head MRR $0.460 \to 0.423$) while the tail direction is essentially flat ($0.501 \to 0.500$). The eval-only λ-sweep (Figure 5) shows only a tiny tail-MRR bump peaking at $\lambda \approx 0.75$ ($0.501 \to 0.503$, $+0.4\%$)—an order of magnitude smaller than the weak-backbone gain, and one that does not transfer to the pooled leaderboard metric (Table 13). Across both datasets, then, the strong backbone admits no useful fixed blend, confirming that a learned or per-relation gate would be required before the prior could be applied on top of a strong KGE model.

Table 13: **ComplEx+DURA diffusion sweep on WN18RR** (official split, filtered tail metric, full test set, eval-only). Tail-MRR peaks weakly at $\lambda \approx 0.75$ then declines; the pooled effect (main text, Table 3) is net-negative.

λ	0.0	0.5	0.75	1.0	1.5	2.5	5.0
tail MRR	0.501	0.502	**0.503**	0.502	0.500	0.495	0.475
tail H@10	0.582	0.585	0.586	0.586	0.586	0.585	0.572



(a) FB15k-237 (2,000 queries, $\lambda \leq 2$) (b) WN18RR (full test, $\lambda \leq 5$)

Figure 5: **λ-sweep on the strong ComplEx+DURA backbone, both datasets** (filtered tail metric, eval-only). *(a)* On FB15k-237, filtered MRR peaks at $\lambda = 0$ and declines monotonically—no positive blend exists. *(b)* On WN18RR, MRR shows only a tiny bump at $\lambda \approx 0.75$ ($+0.4\%$) before declining; this gain is an order of magnitude smaller than on the weak DistMult backbone (Figure 2) and is erased once head and tail are pooled. Across both datasets the strong backbone admits no useful fixed blend.

E Training Analysis and Ablations

E.1 Training Dynamics

Figure 6 shows the per-epoch loss breakdown for a Cora run at $\lambda_5 = 1.15$ ($B = 20$ warmup, $R = 20$ ramp; seed 42). Key observations from the saved logs:

- **Epochs 1–20**: L_{match} decays from $18.3 \rightarrow 0.39$; L_{peb} decreases from $7.92 \rightarrow 5.31$ unlike $\lambda_5 = 0.1$ run, where L_{peb} was flat from the start, it already trends down in this window because pebbling now carries enough weight to move w; $L_{\text{dens}} = 0$ (warmup).
- **Epoch 21–40**: L_{dens} activates, reaches 1.18 at epoch 30 ramp); no spike.
- **Epoch 200**: $L_{\text{match}} = 0.295$, $L_{\text{dens}} = 0.916$, $L_{\text{peb}} = 3.78$; total $= 4.97$. Best checkpoint (val precision@20) selected at epoch 135.

Compared with the old default, L_{match} no longer collapses toward the floor (it plateaus around 0.28–0.30 instead of ≈ 0.01) and L_{peb} falls by roughly half over training ($7.92 \rightarrow 3.78$) instead of staying flat in the 5.5–7.5 band. This indicates direct evidence that raising λ_5 gives the pebbling term a genuine gradient signal rather than being dominated by matching. L_{dens}/cohesion dynamics are unchanged from the old default.

Variance floor and ramp schedule. Without protective measures, the normalizing-flow facet is prone to two interacting failure modes. Early in training, μ_{H_s} (the per-seed community centroid) lies in a near-constant subspace: the EMA standardization inside the flow divides by near-zero variance for those dimensions, amplifying deviations up to $30\times$ and causing L_{dens} to spike; experimentally $3{,}262$ at epoch 30 in the unprotected run, consistent with the instability described in Section 4.4. Three mitigations are required jointly: (i) a *variance floor* $\sigma^2_{\min} = 10^{-2}$ clamps the EMA denominator; (ii) a *ramp schedule* disables L_{dens} for the first $B = 20$ epochs and linearly grows λ_4 over the next $R = 20$, allowing the matching and pebbling terms to establish a non-trivial w before the density model activates; (iii) *per-dimension NLL normalization* prevents high-variance dimensions from dominating the log-likelihood. Omitting any one of the three re-introduces instability; together they yield $L_{\text{dens}} \approx 0.92$ at convergence which is the theoretical per-dimension negative log-likelihood of a standard normal, confirming the flow is well-conditioned.

E.2 Sinkhorn and Refinement Ablations

We implement two architectural extensions, kept off by default to preserve the clean ablation grid: a Sinkhorn-normalised scorer and a weight-aware encoder-feedback step. The Sinkhorn variant tests whether enforcing the doubly-stochastic constraint *architecturally* via alternating row/column normalisation of the scorer output (Sinkhorn, 1964; Cuturi, 2013) adds anything beyond the soft L_{frac}

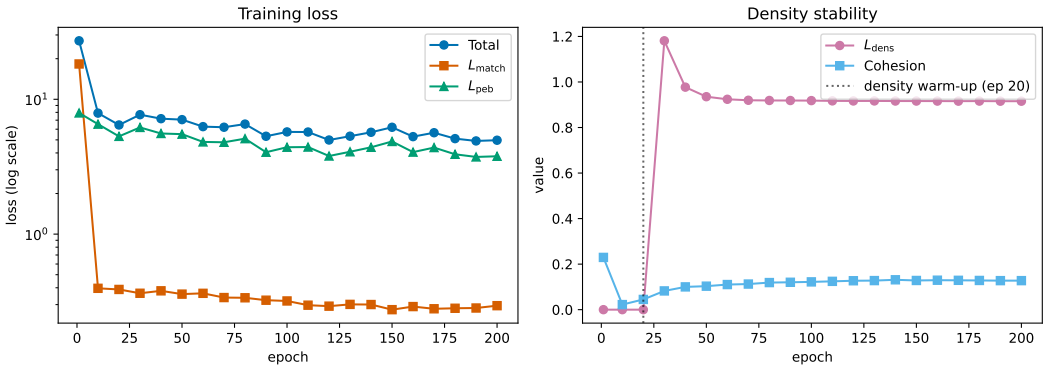


Figure 6: **MatchPeb training on Cora, $\lambda_5 = 0.75$ (fixed density schedule).** *Left*: total loss and its L_{match}, L_{peb} components (log scale). *Right*: L_{dens} and cohesion; the density term activates after the warm-up (dotted line, epoch 20) and stays bounded (≈ 0.9) rather than diverging—contrast the pre-fix run, where L_{dens} spiked to 3,262 (Section 4.4).

penalty alone. The fractional-matching loss already targets the same marginal condition, so this is a question of hard vs. soft enforcement. The refinement variant tests whether closing the $h \to w \to \pi$ feedback loop with one unrolled step recovers information otherwise lost by scoring embeddings before the final PPR iterate is known.

Sinkhorn normalization. Setting `scorer_sinkhorn_iters=10` adds alternating source/ destination normalization (Sinkhorn, 1964) to the scorer output, enforcing near-doubly-stochastic incident-weight distributions architecturally rather than only via L_{frac}. In preliminary experiments on Cora this changes P@20 by ≤ 0.5pp; we consider it most useful on dense graphs where the soft matching penalty converges slowly.

Weight-aware encoder feedback. Setting `refine_steps=2` adds an unrolled step in the forward pass: after scoring, embeddings are re-propagated through the learned w-derived transition matrix and the scorer is applied again. This closes the $h \to w \to \pi$ loop, allowing the encoder to see the current-iterate weights. Preliminary results show a $+0.8$pp P@20 gain at a $2\times$ training cost; we leave a full sweep to future work.

F Reproducibility

F.1 Environment Specification

Python 3.11, PyTorch 2.5.1, CUDA 12.4, `torch_geometric` 2.5.3, `torch_scatter` 2.1.2, `torch_sparse` 0.6.18, `nflows` 0.14, `networkx` 3.x, `scikit-learn`, `numpy`, `scipy`. Cora experiments run on CPU in $\sim$28 s/epoch. FB15k-237 DistMult pre-training: $\sim$1526 s total on GPU (A100).