# GRAFT: Grounding Cold-Start Nodes via Factorized Structural Alignment

Srinivas Virinchi, Aman Gulati, Gokul Swamy, and Anoop Saladi

Amazon.com Inc.
{virins,amangula,swagokul,saladias}@amazon.com

**Abstract.** Graph Neural Networks (GNNs) break down on zero-degree nodes, as message passing requires neighbors. Without interaction history, unseen entities are sub-optimally embedded, leaving them weakly anchored in the latent space, creating a cold-start bottleneck in retrieval. To address this, we propose GRAFT, a factorized architecture that unifies structural and feature transformations into a shared weight space. Unlike distillation methods that merely mimic teacher outputs, GRAFT decomposes the model into symmetric Neighbor Aggregation (NA) and Feature Encoding (FE) sub-layers, forcing the feature logic to internalize the graph manifold during training. This shared parameterization enables relational grounding with stable $O(1)$ inference for cold-start nodes by bypassing structural dependencies. Production A/B tests demonstrate a 147% surge in cold-start sales, 21% wider product coverage, and $1.6M in annual savings, all while maintaining sub-second latency.

**Keywords:** Cold-Start, Factorized GNN, Structural Alignment

## 1 Introduction

While GNNs excel on dense topologies, they suffer representational collapse on zero-degree entities where message passing is undefined. In high-velocity e-commerce, new entities lack interaction history, rendering neighborhood aggregation infeasible. Traditional GNNs fail here because their parameters, optimized for aggregated distributions, produce unstable embeddings when applied to isolated features. Current solutions remain inadequate: distillation inherits representational instability by forcing students to mimic erratic teacher outputs on sparse nodes, while graph reconstruction introduces prohibitive inference latency. This creates a critical bottleneck for real-time tasks like product recommendation and defect detection, where metadata is often too sparse to act as a reliable anchor.

We formalize this as the strict cold-start problem: embedding an isolated entity into a relational manifold in the absence of structural context during inference. This setting requires models to generalize relational inductive biases using only intrinsic features. For example, a newly listed product must be accurately positioned for recommendation, or a new customer's package must be assessed for defect detection, without any interaction history.

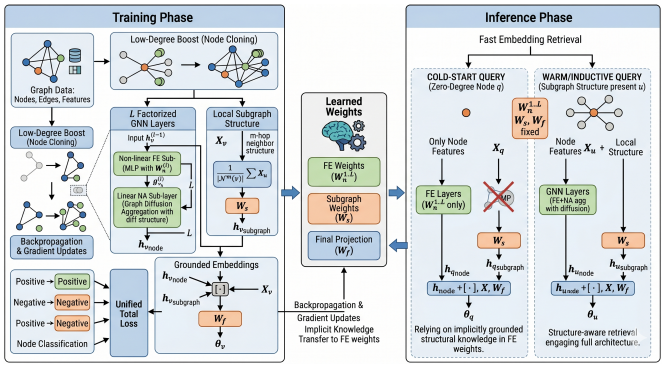


**Fig. 1.** The GRAFT Unified Factorized Architecture.

To address this, we propose GRAFT (Figure 1), a factorized architecture unifying structural and feature transformations via shared weights. By decomposing layers into symmetric Neighbor Aggregation (NA) and Feature Encoding (FE) sub-layers, GRAFT forces feature logic to internalize the graph manifold. At inference, the model bypasses the structural path if neighbors are absent, ensuring stable $O(1)$ inference via the feature pathway.

## 2 GRAFT Framework

We present GRAFT, a factorized architecture for strict cold-start recommendation that grounds zero-degree query nodes $q$ into a latent manifold using only static features $\mathcal{X}_q$. As shown in Figure 1, GRAFT unifies structural and feature transformations via a shared weight space ($W_n$). Each layer $l$ is decomposed into a non-linear Feature Encoding (FE) sub-layer, $h_{v_n}^{(l)} = \sigma(h_v^{(l-1)} W_n^{(l)})$, and a linear Neighbor Aggregation (NA) path which is implemented via a matrix exponential diffusion as $h_{v_{\text{node}}}^{(l)} = \frac{1}{Z} \sum_{i=0}^{k} \frac{(\mathcal{D}^T)^i h_{v_n}^{(l)}}{i!}$, where $\mathcal{D}$ is the transition matrix, $k$ is the diffusion depth, and $Z$ is a normalization constant. *The trick lies in this shared parameterization, which factorizes the architecture by decoupling message-passing from feature transformations, forcing the feature-only logic to distill and internalize topological signals from neighbors during training.* The final embedding $\theta_v$ fuses the node representation, an $m$-hop pooled subgraph embedding $h_{v_{\text{subgraph}}}$, and a raw feature residual: $\theta_v = W_f[h_{v_{\text{node}}} \,||\, \mathcal{X}_v \,||\, h_{v_{\text{subgraph}}}]$.

Training optimizes a dual objective across edges $(u, v) \in \mathcal{E}$ and $r$ randomly sampled negative nodes $z_j$: (1) an angular loss to enforce manifold alignment, $\mathcal{L}_{\text{a}} = -\sum_{(u,v)} \sum_{j=1}^{r} [\log(\cos(\theta_u, \theta_v)) + \log(1 - \cos(\theta_u, \theta_{z_j}))]$, and (2) a distance-based loss, $\mathcal{L}_{\text{m}} = \sum_{(u,v)} \sum_{j=1}^{r} \max(0, M - \theta_u \cdot \theta_v + \theta_u \cdot \theta_{z_j})$, which enforces a separation margin $M$. During inference, GRAFT bypasses structural aggregation for zero-degree query nodes $q$, enabling stable inference. This produces relationally grounded embeddings via a fusion of node, subgraph, and residual signals: $\theta_q = W_f[h_{q_{\text{node}}} \,||\, \mathcal{X}_q \,||\, h_{q_{\text{subgraph}}}]$.

## 3 Results

***Datasets and Implementation:*** We evaluate GRAFT on two large scale proprietary graphs with over 1M nodes and 100M edges. The product graph ($\mathcal{G}_p$) is a dense co-purchase network for recommendation and product classification across 566 product types, while the order graph ($\mathcal{G}_o$) is a transactional network where nodes represent orders and edges connect shared entities (e.g., email, phone, credit card). Binary labels capture defects such as fulfillment failures and fraudulent claims under extreme class imbalance. Product features are derived from a fine tuned SBERT [5] model, while defect graph features use aggregated statistics over shared entities. We implement GRAFT in PyTorch Geometric with $L = 2$ layers over 2-hop subgraphs, using 256 dimensional embeddings trained up to 50 epochs on an NVIDIA A10G GPU; results are averaged over 10 runs. Owing to confidentiality, absolute values are scaled by constants $x$ and $y$, and we primarily report relative performance gains.

***Evaluation:*** We benchmark GRAFT against competitive state-of-the-art baselines: structural GNNs (*GATv2* [1], *LightGCN* [3]), feature-only models (*MLP* [6], *PMLP* [7]), and cold-start frameworks based on distillation (*LLP* [2], *TailGNN* [4], *ColdBrew* [8]).

**Table 1.** Cold-Start Evaluation. **Bold**: best; Underlined: second best.

| Model | Defect Detection ($\mathcal{G}_o$) | | | | | Product Recommendation ($\mathcal{G}_p$) | | | Product Type Classification | |
|---|---|---|---|---|---|---|---|---|---|---|
| | F1 | Prec@1% | Prec@2% | Recall@1% | Recall@2% | hitrate@10 | mrr@10 | rocauc | accuracy | rocauc |
| GATv2 | 44.52y | 1455y | 1478y | 73y | 153y | 2.12x | 0.86x | 792x | 282x | 291x |
| LightGCN | 44.55y | 1460y | 1480y | 74y | 154y | 2.15x | 0.87x | 795x | 285x | 293x |
| MLP | 44.05y | 1400y | 1420y | 70y | 150y | 2.00x | 0.80x | 770x | 260x | 270x |
| PMLP | 44.65y | 1462y | 1483y | 74y | 154y | 2.18x | 0.9x | 799x | 294.5x | 296.1x |
| LLP | 44.60y | 1468y | 1488y | 75y | 155y | 2.16x | 0.88x | 797x | 292x | 295x |
| TailGNN | 44.62y | 1470y | 1490y | 75y | 155y | 2.17x | 0.89x | 798x | 293x | 295x |
| ColdBrew | 44.63y | 1471y | 1491y | 75y | 155y | 2.17x | 0.89x | 798x | 294x | 296x |
| GRAFT | **45.32y** | **2052y** | **1967y** | **106y** | **204y** | **2.2x** | **1.1x** | **810x** | **366x** | **400x** |
| % Improvement | +1.50% | +4.32% | +31.84% | +41.33% | +31.61% | +0.92% | +22.22% | +1.38% | +24.28% | +35.09% |

As shown in Table 1, GRAFT outperforms all structural and distillation baselines. On defect detection ($\mathcal{G}_o$), it achieves a +31.84% gain in Precision@2% and a +41.33% boost in Recall@1%, significantly improving the recovery of rare defects. For product recommendation ($\mathcal{G}_p$), GRAFT increases MRR@10 by +22.22%, surfacing relevant cold-start items earlier in ranked results. Finally, the +24.28% accuracy gain in product type classification indicates superior class separation across the 566 fine-grained categories.

***Production Deployment and Impact*** We conducted a comprehensive two-week A/B test to evaluate GRAFT's effectiveness in creating recommendations for new products. For the control group, recommendations from our in-house production GNN model (based on GATv2 [1] serve as the baseline, whereas the treatment group receives recommendations generated by GRAFT. The results demonstrated GRAFT's superior performance across multiple key metrics, with sales increasing by 147%, profit margins improving by 73%, and product

coverage for new items expanding by 21% compared to the control group (p-value $< 0.05$). Similarly, for the defect detection use-case, GRAFT is estimated to save approximately $1.67 million annually by proactively identifying defects. GRAFT is deployed as a SageMaker endpoint serving real time embeddings for product recommendation and defect detection, with sub second inference latency.

## 4 Learning and Conclusion

GRAFT's deployment yields five core insights: *(1) Factorization over Distillation:* Shared weights bake relational priors into feature logic, preventing representational collapse caused by mimicking unstable teacher logits on isolated nodes. *(2) Expressivity vs. Stability:* Non-linear encoding ensures model expressivity, while a linear aggregation path provides inference stability by keeping structural operators factorable when neighbors are missing. *(3) Linear Decoupling:* A linear structural path enables runtime bypassing, guaranteeing $O(1)$ complexity and sub-second latency ($< 1$s) to meet strict production SLAs. *(4) Internalized Reasoning:* Shared parameterization forces feature paths to internalize topological signals, allowing cold-start nodes to generate grounded embeddings from static features alone. *(5) Structural Safety Nets:* Subgraph embeddings stabilize training in sparse graphs, preserving relational context when explicit edges are absent at inference. In conclusion, GRAFT bridges GNN expressivity and MLP efficiency, delivering a 147% increase in cold-start sales and $1.67M in annual savings.

## References

1. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks? arXiv preprint arXiv:2105.14491 (2021)
2. Guo, Z., Shiao, W., Zhang, S., Liu, Y., Chawla, N.V., Shah, N., Zhao, T.: Linkless link prediction via relational distillation. In: ICML (2023)
3. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Liao, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). pp. 639–648 (2020)
4. Liu, Z., Nguyen, T.K., Fang, Y.: Tail-gnn: Tail-node graph neural networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1109–1119 (2021)
5. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: EMNLP (2019)
6. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
7. Yang, C., Wu, Q., Wang, J., Yan, J.: Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. arXiv preprint arXiv:2212.09034 (2022)
8. Zheng, W., Huang, E.W., Rao, N., Katariya, S., Wang, Z., Subbian, K.: Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. In: ICLR (2022)