

Controlled Fine-Tuning for Two-Tower Recommenders: Why Gradual Unfreezing Can Fail and Cosine Regularization Suffices

Gabriel Dershowitz

Yeshiva University

gabrieldershowitz@gmail.com

Djallel Bouneffouf

IBM Research

Djallel.Bouneffouf@ibm.com

Abstract

Controlled fine-tuning of pretrained text encoders—via gradual layer-wise unfreezing and anchoring—is widely assumed to benefit downstream tasks. We test this assumption in two-tower recommendation and, across three seeds, find a mixed picture. **First**, any fine-tuning dramatically beats a frozen encoder ($\geq 3.0\times$ cold Recall@10 lift); freezing the item tower is strongly counterproductive. **Second**, gradual unfreezing lowers warm-setting performance relative to vanilla fine-tuning under realistic budgets (12 epochs), with no measurable cold-start benefit. **Third**, our adaptive cosine-anchored regularizer shows no measurable benefit over vanilla fine-tuning, and the full adaptive recipe underperforms simpler alternatives. Thus, for two-tower retrieval, vanilla fine-tuning is a strong, low-complexity default.

1 Introduction

Two-tower recommenders separately encode users and items into a shared embedding space and have become standard for large-scale retrieval. A natural extension replaces the item tower with a pretrained text encoder to inherit semantic priors and improve generalization to rare items. A consistent empirical finding, however, is that frozen pretrained encoders in DSSM-style models are routinely beaten by simple ID-based matrix factorization on warm recommendation (Yuan et al., 2023).

We frame this as an *alignment problem* between towers: the user and item towers must converge on a shared geometry, and freezing one forces the other to contort itself to match a fixed external geometry never optimized for the recommendation signal. The obvious remedy—unfreeze and fine-tune end-to-end—induces catastrophic forgetting of the very prior that motivated using a pretrained encoder (Howard and Ruder, 2018).

We investigate whether *controlled unfreezing*—exposing the item tower’s capacity to adapt gradu-

ally, anchoring its representations to the pretrained state, and modulating the anchoring strength using validation feedback—is the right tool. Our method, Adaptive E5, combines three ingredients: (1) gradual layer-wise unfreezing of the E5 encoder triggered by NDCG plateaus—a recsys adaptation of ULMFiT (Howard and Ruder, 2018); (2) adaptive cosine-anchored regularization with a feedback-driven coefficient λ and cooldown windows around unfreeze events; and (3) a projection head identical to the frozen-encoder baseline’s, so the only difference is the fine-tuning procedure.

We evaluate on Yelp, Amazon Video Games, and MovieLens-1M under warm and inductive cold-start protocols adapted from A-LLMRec (Kim et al., 2024). Our method improves consistently over Frozen E5 on every dataset and regime, with the largest gains on the most text-rich domain. Our ablation further shows that *any* fine-tuning dramatically beats frozen encoders, and that cosine regularization alone is the strongest single component. That no single component of our adaptive recipe reliably outperforms vanilla fine-tuning—though gradual unfreezing alone yields the best cold-start numbers we observe, at a cost to warm performance.

Negative results contribution

This paper provides three findings that align with the Insights 2026 workshop theme. First, we show that the common practice of keeping the item tower frozen (Yuan et al., 2023) is strongly counterproductive: any fine-tuning yields large gains. Second, gradual layer-wise unfreezing—a technique many practitioners would reasonably try—lowers warm-setting performance under typical training budgets, with no measurable cold-start benefit. Third, our ablation finds no component of the adaptive recipe that reliably outperforms vanilla fine-tuning; the full adaptive schedule underperforms simpler alternatives. These results complicate the assump-

tion, inherited from language-model fine-tuning (Howard and Ruder, 2018), that controlled unfreezing straightforwardly transfers to retrieval-style tasks.

2 Related Work

MoRec (Yuan et al., 2023) shows that frozen-encoder DSSM models trail ID-based MF on warm recommendation, while sequential backbones benefit more reliably from textual input. A-LLMRec (Kim et al., 2024) addresses cold-start by aligning a pretrained recommender’s collaborative signal with an LLM, showing text helps most when interaction signal is sparse. We hold the two-tower architecture fixed and ask whether a careful fine-tuning recipe alone closes the gap MoRec documents.

Beyond architecture-level comparisons, the fundamental challenge of balancing adaptation with preservation of prior knowledge has been extensively studied in contextual bandit and recommender systems research (Lin et al., 2018; Bouneffouf et al., 2017; Balakrishnan et al.). These works highlight that adaptive algorithms must carefully manage the trade-off between incorporating new data and retaining previously learned representations—a tension that directly informs our investigation of controlled fine-tuning strategies.

ULMFiT (Howard and Ruder, 2018) introduced gradual unfreezing and discriminative learning rates for language model fine-tuning, motivated by catastrophic forgetting under uniform fine-tuning. Subsequent regularized fine-tuning approaches anchor parameters or representations to the pretrained state. Our method adapts both ideas to recommendation: the unfreezing trigger is validation NDCG, and the anchor coefficient is itself adaptive.

3 Method

Architecture. Let $\mathcal{U}, \mathcal{I}$ denote user and item sets. The user tower is a learned embedding table $\mathbf{u}_u = W_u \mathbf{e}_u \in \mathbb{R}^d$ with $d=128$. The item tower is $\mathbf{e}_i = g_\phi(f_\theta(t_i))$, where t_i is the item’s textual description (title + category meta-data), f_θ is `intfloat/e5-base` (12 layers, 768-dim, 110M params), and g_ϕ is a two-layer GELU MLP (768→512→128). Both outputs are L2-normalized, so $s(u, i) = \hat{\mathbf{u}}_u^\top \hat{\mathbf{e}}_i$. The head g_ϕ is architecturally identical to the one used in our Frozen-E5 baseline.

Training objective. We use symmetric in-batch InfoNCE with temperature $\tau=0.07$: $\mathcal{L}_{\text{NCE}} =$

$\frac{1}{2}(\mathcal{L}^{u \rightarrow i} + \mathcal{L}^{i \rightarrow u})$, where

$$\mathcal{L}^{u \rightarrow i} = -\frac{1}{B} \sum_{k=1}^B \log \frac{\exp(\hat{\mathbf{u}}_k^\top \hat{\mathbf{e}}_{i_k} / \tau)}{\sum_j \exp(\hat{\mathbf{u}}_k^\top \hat{\mathbf{e}}_{i_j} / \tau)}. \quad (1)$$

To anchor the item tower to its pretrained state, we keep a frozen snapshot f_{θ_0} taken before any fine-tuning and add a cosine drift penalty:

$$\mathcal{L}_{\text{reg}} = \frac{1}{B} \sum_{k=1}^B (1 - \cos(f_\theta(t_{i_k}), f_{\theta_0}(t_{i_k}))). \quad (2)$$

The total loss is $\mathcal{L} = \mathcal{L}_{\text{NCE}} + \lambda \mathcal{L}_{\text{reg}}$, where λ is time-varying.

Adaptive λ schedule. Let $\delta_e = \text{NDCG@10}(e) - \text{NDCG@10}(e-1)$. We multiply λ by 0.9 when $\delta_e > 0$ and by 1.1 otherwise, clipped to $[\lambda_{\min}, \lambda_{\max}] = [0.01, 100]$ with $\lambda_0 = 1.0$. When validation improves we relax the anchor; when it regresses we tighten it. Around each unfreeze event we hold λ fixed for $C=2$ epochs (cooldown) to avoid over-correcting during the optimization-landscape shift.

Gradual unfreezing. We unfreeze the E5 transformer one layer at a time, top-down. Initially all 12 layers are frozen; only g_ϕ is trainable. A layer patience counter c increments each epoch validation NDCG@10 fails to exceed its running best. When $c \geq \text{LAYER_PATIENCE}=5$, we unfreeze one more layer, reset c , and begin the cooldown. Unfrozen layers use $\eta_{\text{enc}}=10^{-4}$ vs. $\eta_{\text{user}}=10^{-3}$ for the user table and head, following ULMFiT. Training halts when all layers are unfrozen and $c \geq \text{LAYER_PATIENCE}$, or when a global patience counter reaches `GLOBAL\_PATIENCE`=3.

Hypothesis: fine-tuning’s benefits without its risks

We hypothesize that combining gradual unfreezing with adaptive cosine-anchored regularization lets Adaptive E5 capture the benefits of fine-tuning while limiting its principal risk: catastrophic forgetting of the pretrained encoder’s prior. Gradual unfreezing preserves that prior early in training and only gradually exposes new capacity for task-specific adaptation; the cosine anchor keeps representations close to their pretrained state throughout, relaxing when validation improves and tightening when it regresses. We therefore expect Adaptive E5 to outperform both extremes: the frozen encoder, which cannot adapt at all, and vanilla end-to-end

fine-tuning, which adapts freely but risks overwriting the prior that motivated using a pretrained encoder in the first place.

4 Experimental Setup

Datasets. **Yelp Open Dataset:** top 10,000 users $\times$ 5,000 businesses; item text is name + categories + location attributes. **Amazon Video Games (Hou et al., 2024):** top 10,000 $\times$ 5,000; item text is title + category metadata. **MovieLens-1M:** full ($\approx 6,000 \times 3,700$); item text is title + genres. ML-1M has weak item text relative to Yelp and Amazon. All datasets use a minimum-interaction filter of 5 for users and items.

Split and metrics. Leave-one-out temporal split: each user’s most recent interaction is the test item, the next is validation, the rest are training. We rank against the full corpus with seen training items masked rather than sampled negatives (Krichene and Rendle, 2020), and report Recall@10 (R@10) and NDCG@10 (N@10).

Cold-start protocol. Training is unchanged across all methods. At evaluation only, we sample 15% of test users with ≥ 4 training interactions, retain $K=3$ random training items per sampled user, and re-derive the user representation from those K items. Two-tower methods use a cold user vector $= \ell_2\text{-norm}(\frac{1}{K} \sum_j \hat{\mathbf{e}}_{ij})$; SASRec consumes the K items in temporal order and uses the final hidden state. The cold pool is fixed by COLD_SEED=42 so cross-seed variance reflects training randomness only.

Baselines. **BPR-MF** via RecBole (Zhao et al., 2021); **Two-Tower (ID):** our two-tower architecture with a learned item-ID table ($d=128$); **Frozen E5** (primary control): identical architecture and loss to Adaptive E5 but with the encoder frozen; **SASRec** (Kang and McAuley, 2018); **Adaptive E5** (ours).

Training. All two-tower models use AdamW, batch 512, gradient accumulation over 2 steps, weight decay 10^{-2} , $\tau=0.07$, $\eta_{\text{user}}=10^{-3}$, $\eta_{\text{enc}}=10^{-4}$. SASRec: dim 64, 2 layers, 1 head, max sequence length 50, 30 epochs. All stochastic results are means over $n=3$ seeds.

5 Results

Warm setting. SASRec dominates warm recall everywhere, peaking at R@10 = 0.250 on

MovieLens-1M—roughly $4\times$ the next-best method. Among text-encoder methods, Adaptive E5 substantially outperforms Frozen E5: +119% warm R@10 on Yelp, +81% on MovieLens, and +5% on Amazon. On Amazon—sparse interactions, rich text—Adaptive E5 also surpasses ID-only in warm R@10 (0.027 vs. 0.021), suggesting text semantics help when collaborative signal is thin.

Cold-start. The primary controlled comparison is Adaptive E5 vs. Frozen E5: same architecture, same data, same cold pool—only the fine-tuning procedure differs. Adaptive E5 improves cold-start R@10 by $3.1\times$ on Yelp (0.030 ± 0.001 vs. 0.010 ± 0.002), +42% on Amazon (0.026 vs. 0.018), and +18% on MovieLens (0.041 vs. 0.035). In absolute terms, Adaptive E5 ranks third on cold R@10 behind SASRec and ID-only across all three datasets; both leverage training-time interaction patterns more directly than our mean-pooling cold-user representation. Adaptive E5 does achieve second-highest cold N@10 on MovieLens-1M (0.021, behind SASRec’s 0.026).

Variance. Across three seeds, Adaptive E5 cold R@10 standard deviations are small (0.001, 0.002, 0.002 for Yelp/Amazon/ML-1M), indicating stable learning dynamics. ML-1M warm performance shows higher seed variation (± 0.006), likely reflecting sensitivity to user-embedding initialization on the larger user set.

6 Ablation

Table 2 ablates each component of Adaptive E5 on Yelp—the dataset with both the largest controlled improvement over Frozen E5 (Table 1) and the richest item text among the three we evaluate—using a 12-epoch budget shared across all rows. We concentrate the ablation here because a larger overall effect and a stronger text signal give the decomposition the best chance of separating real component-level differences from seed noise; on Amazon or MovieLens, where the overall gain is smaller, the same decomposition would likely be dominated by the variance we already observe in Table 2.

Negative findings from the ablation

Any fine-tuning $\gg$ frozen. The most striking result is that any fine-tuning yields at least a $3.0\times$ cold R@10 lift over Frozen E5 (0.030–0.033 vs. 0.010 across all four fine-tuned variants). The com-

Table 1: Recall@10 (R) and NDCG@10 (N) under warm (full-corpus, seen-item masked) and inductive cold-start ($K=3$) evaluation. Stochastic entries are mean $\pm$ std over $n=3$ seeds; BPR-MF (RecBole) is deterministic. **Bold**: best per column; underline: second-best. $\dagger$ BPR-MF does not expose item embeddings for cold-start.

Method	Yelp				Amazon Video Games				MovieLens-1M			
	Warm		Cold		Warm		Cold		Warm		Cold	
	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
BPR-MF (Zhao et al., 2021) $\dagger$	0.064	0.033	—	—	0.048	0.030	—	—	0.063	0.032	—	—
SASRec (Kang and McAuley, 2018)	0.091$\pm$.001	0.046$\pm$.001	0.052$\pm$.003	0.026$\pm$.003	0.109$\pm$.002	0.060$\pm$.001	0.050$\pm$.002	0.026$\pm$.001	0.250$\pm$.003	0.141$\pm$.001	0.053$\pm$.003	0.026$\pm$.002
Two-Tower (ID)	0.045 $\pm$.001	0.021 $\pm$.001	0.038 $\pm$.004	0.016 $\pm$.001	0.021 $\pm$.001	0.010 $\pm$.001	0.028 $\pm$.002	0.014 $\pm$.001	0.043 $\pm$.002	0.020 $\pm$.001	0.043 $\pm$.003	0.020 $\pm$.001
Frozen E5	0.017 $\pm$.001	0.008 $\pm$.001	0.010 $\pm$.002	0.004 $\pm$.001	0.025 $\pm$.001	0.012 $\pm$.001	0.018 $\pm$.001	0.009 $\pm$.001	0.021 $\pm$.001	0.010 $\pm$.001	0.035 $\pm$.002	0.017 $\pm$.001
Adaptive E5 (ours)	0.038 $\pm$.001	0.018 $\pm$.001	0.030 $\pm$.001	0.013 $\pm$.002	0.027 $\pm$.002	0.013 $\pm$.001	0.026 $\pm$.002	0.013 $\pm$.001	0.039 $\pm$.006	0.018 $\pm$.002	0.041 $\pm$.002	<u>0.021$\pm$.001</u>

Table 2: Ablation on Yelp. Mean $\pm$ std over the same $n=3$ seeds (42, 123, 456) as Table 1. The Frozen E5 and Full-method rows are the identical Yelp entries from Table 1, reused here rather than re-derived from a separate run. **Bold**: best among fine-tuned variants (A–D) on Warm, where std bands are separated; entries within 0.001 are tied. On Cold, (A)–(D) overlap within one seed-std of each other and are not reliably distinguishable at $n=3$, so none is bolded.

Configuration	Warm		Cold	
	R@10	N@10	R@10	N@10
Frozen E5 (ref.)	0.017 $\pm$.001	0.008 $\pm$.001	0.010 $\pm$.002	0.004 $\pm$.001
(A) Vanilla fine-tune	0.047$\pm$.001	0.022$\pm$.001	0.032 $\pm$.005	0.014 $\pm$.002
(B) Gradual unfreeze	0.044 $\pm$.002	0.020 $\pm$.001	0.033 $\pm$.002	0.016 $\pm$.001
(C) Fixed $\lambda=0.1$	0.047$\pm$.000	0.022$\pm$.001	0.032 $\pm$.004	0.015 $\pm$.001
(D) Full method (ours)	0.038 $\pm$.001	0.018 $\pm$.001	0.030 $\pm$.001	0.013 $\pm$.002

mon practice of keeping the item tower frozen is strongly counterproductive.

Gradual unfreezing hurts warm, with no measured cold-start benefit. Contrary to intuition from ULMFiT, gradual unfreezing (B) reduces warm performance compared to vanilla (A) ($R@10$ 0.047 $\rightarrow$ 0.044). Cold $R@10$ shows no compensating benefit: (A), (B), and (C) are statistically indistinguishable given their seed-to-seed variance (Table 2). With a 12-epoch budget, most layers remain frozen for most of training, delaying adaptation.

No tested component reliably beats vanilla fine-tuning. Fixed cosine regularization (C) is statistically indistinguishable from vanilla (A) on both warm and cold $R@10$. Our full Adaptive E5 (D) is nominally the lowest of the four variants on cold $R@10$ (0.030), though the gap to (A)/(B)/(C) is modest given their wide cold-start variance; on warm, (D) is clearly below both (A) and (C). Combining gradual unfreezing with adaptive regularization does not recover, and may compound, each component’s individual warm-side cost.

Takeaway for practitioners. Vanilla fine-tuning is a strong, low-complexity default. At this budget, gradual unfreezing costs warm performance with

no compensating benefit we could measure, and added regularization did not reliably improve on vanilla either.

7 Conclusion

We tested controlled fine-tuning for two-tower recommendation and found a mixed picture that complicates assumptions inherited from language-model fine-tuning. First, freezing the pretrained item encoder is strongly counterproductive: any fine-tuning yields at least a $3.0\times$ cold $R@10$ lift. Second, gradual layer-wise unfreezing—a technique many practitioners would reasonably try—lowers warm-setting performance under realistic training budgets (12 epochs), with no measurable cold-start benefit. Third, no tested component of our adaptive recipe reliably outperforms vanilla fine-tuning; the full adaptive recipe underperforms simpler alternatives.

Thus, for two-tower retrieval, vanilla fine-tuning is a strong, low-complexity default. Gradual unfreezing costs warm performance without a measurable compensating benefit, and added regularization did not reliably improve on vanilla fine-tuning either. This negative insight refines the conventional wisdom from ULMFiT and suggests that for retrieval-style tasks, simpler fine-tuning suffices. Whether a longer training budget (e.g., 50+ epochs) would let gradual unfreezing fully open the encoder and close its warm-side gap remains an open question we leave to future work.

References

- Avinash Balakrishnan, Djallel Bouneffouf, Nicholas Mattei, and Francesca Rossi. Using contextual bandits with behavioral constraints for constrained online movie recommendation.
- Djallel Bouneffouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. 2017. Context attentive bandits: Contextual bandit with restricted context. *arXiv preprint arXiv:1705.03821*.

- Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiushi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.
- Sein Kim, Hongseok Kang, Seungyeon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round LLM-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.
- Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1748–1757.
- Baihan Lin, Djallel Bouneffouf, Guillermo A Cecchi, and Irina Rish. 2018. Contextual bandit with adaptive feature extraction. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 937–944. IEEE.
- Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? ID- vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2639–2649.
- Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*.