LOOPFM: LEARNING FROM HISTORICAL REPRESENTATIONS OF FOUNDATION MODEL FOR RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation (KD) transfers a single scalar prediction from a large foundation model (FM) to compact vertical models (VMs), suffering from diminishing transfer ratio—the fraction of FM improvement captured by the VM—as a single scalar cannot convey the rich intermediate knowledge that larger FMs learn. To address this bottleneck, we propose **LoopFM** (**L**earning fr**O**m Hist**O**rical Re**P**resentations of **FM**), a framework that opens a high-bandwidth transfer channel by structuring FM intermediate embeddings as *input features* (e.g., user history sequence) for downstream VMs, without requiring real-time FM inference at serving and architectural coupling between FM and VM. We provide a theoretical framework for LoopFM with a gain decomposition and transfer-ratio analysis. On three public benchmarks, LoopFM demonstrates strong AUC improvements (e.g., 6%+ on TaobaoAd) and complementary knowledge transfer capability with KD. On industrial-scale systems (billions of examples, trillion-parameter FMs), LoopFM approximately **doubles** the knowledge transfer ratio on top of KD, delivering a **+0.5%** conversion improvement in the first half after its initial launch, and **+1.03%** and **+1.22%** conversion improvement from two individual launches in the subsequent half. Through systematic experiments, LoopFM demonstrates a scaling law in sequence length, embedding dimension, and upstream FM size.

1 INTRODUCTION

Industrial recommendation relies on a two-tier architecture: a large foundation model (FM) with up to trillions of parameters learns rich representations offline, while compact vertical models (VMs) serve predictions under strict latency constraints (He et al., 2014; Liang et al., 2025; Anil et al., 2022). Knowledge distillation (KD) (Hinton et al., 2015) bridges the two by transferring the FM's scalar predictions as soft labels to supervise VM training. However, as FMs scale to multi-trillion parameters, we observe that the *transfer ratio* (TR)—the fraction of FM improvement captured by the VM ($\text{TR} = \Delta\text{NE}_{\text{VM}}/\Delta\text{NE}_{\text{FM}}$)—continues to deteriorate, corroborating the findings that KD degrades under large teacher-student capacity gaps (Cho & Hariharan, 2019).

We hypothesize that this deterioration stems from a *bandwidth bottleneck*: a single scalar prediction compresses all of the FM's learned knowledge—rich cross-domain features, multi-level interaction patterns, contextual signals—into one number. As FMs grow more capable, the gap between what they learn and what a scalar can convey widens. Furthermore, FMs are typically trained on richer cross-domain features than VMs, creating a *feature gap* that scalar KD cannot bridge.

We propose **LoopFM**, a novel knowledge transfer framework to address this bottleneck by *materializing* the FM's intermediate representations as structured input features for VM consumption (Figure 1). LoopFM defines three modular stages—extraction, compression, and structuring—each independently configurable. The structuring stage groups compressed embeddings by a *grouping key* (e.g., user ID, item ID), yielding, e.g., sequences or graphs, as VM input features. In this paper we focus on user-keyed temporal sequences. Crucially, only *historical* embeddings are used—enabling VM serving without requiring real-time FM inference.

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.

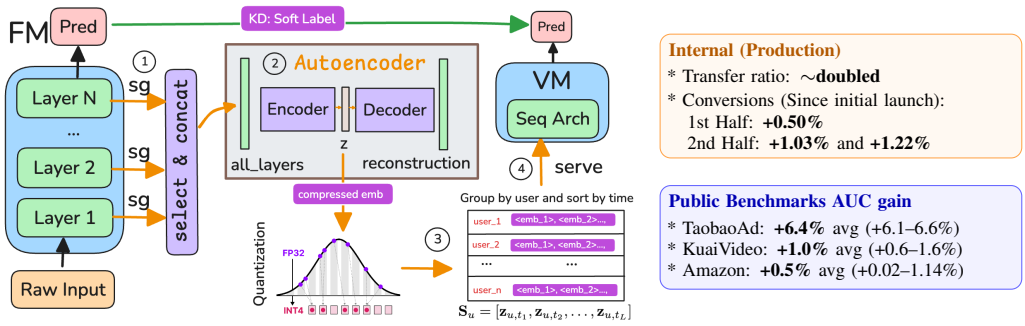


Figure 1: LoopFM overview. **Left:** KD transfers via a soft label (top). LoopFM adds a high-bandwidth embedding channel: ① extract intermediate FM embeddings; ② compress (e.g., autoencoder) and quantize (e.g., INT4) for storage efficiency; ③ group by key and structure into temporal sequences $\mathbf{S}_k$ (here, user-keyed: $k{=}u$); ④ serve as input features to the VM's encoder. No real-time FM inference is required. **Right:** Key results on internal production (top) and public benchmarks (bottom).

We provide a theoretical analysis (Section 4) showing that LoopFM's information gain decomposes into temporal history and cross-feature components minus the compression cost, and derive a lower bound on the transfer ratio that increases monotonically with the FM's feature gap over the VM.

We validate LoopFM on public benchmarks (TaobaoAd, KuaiVideo, Amazon Electronics) and industrial-scale systems with trillion-parameter FMs, where it doubles the knowledge transfer ratio and has delivered significant ad conversion improvement in production. Our **contributions** are:

1. A modular framework that transfers FM intermediate representations as structured VM input features, requiring no architectural coupling or real-time FM inference.
2. Theoretical analysis decomposing LoopFM's information gain into temporal, cross-feature, and compression-loss components, with a transfer-ratio lower bound that grows with the FM-VM feature gap (Corollary 4) and tightens with better compression (Theorem 2), and an information gain that non-decreases with sequence length L (Theorem 5).
3. Extensive experiments on three public benchmarks and internal trillion-parameter systems, including ablations on layer selection, checkpoint freshness, sequence length, embedding dimension etc. In production, LoopFM delivered +0.5% conversion improvement in the first deployment half, and +1.03% and +1.22% conversion improvements from two individual launches in the subsequent half.
4. Knowledge-transfer scaling: LoopFM's gains increase with sequence length and FM scale, and with embedding dimension when the downstream encoder has sufficient capacity, transferring improvements from richer FM features and greater FM capacity to the VM.

2 Related Work

Knowledge distillation in recommendation. KD (Hinton et al., 2015) is widely used to transfer knowledge from large teachers to compact students in recommendation (Kang et al., 2024; 2023; Chen et al., 2023; Tang & Wang, 2018). External distillation (Liang et al., 2025; Khani et al., 2024) separates teacher and student training, enabling the teacher to serve as a foundation model. Beyond scalar KD, embedding-based methods transfer intermediate representations: FitNets (Romero et al., 2015) matches hidden activations, CRD (Tian et al., 2020) maximizes representation mutual information, and privileged features distillation (Xu et al., 2020; Yang et al., 2022) leverages training-time-only signals. Cui et al. (2024) distill LLM embeddings to lightweight sequential models. All these methods either transfer scalars or match *current-sample* representations via auxiliary losses. LoopFM instead materializes *historical* FM embeddings as structured input features.

Sequence modeling and FM representations. User behavior sequence modeling is central to modern recommendation: DIN (Zhou et al., 2018) and DIEN (Zhou et al., 2019) use attention over action histories, SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019) apply

Transformers, and SIM (Pi et al., 2020) handles lifelong sequences. These model raw behavioral signals (clicked IDs, categories). Separately, foundation model research (Hou et al., 2026; Zhang et al., 2024a; Anil et al., 2022) has explored transferable entity embeddings: PinSage (Ying et al., 2018) for items, entity-level user embeddings (Zhang et al., 2024b; Li et al., 2023) for users, which miss interaction-level information. Recent industrial systems (Xiong et al., 2026; Chen et al., 2025) have adopted asynchronous caching of sequence model representations for serving efficiency. Concurrent and independent of our work, IAT (Li et al., 2026) proposes a similar compress-and-sequence pipeline for historical interaction representations from a sequence feature engineering perspective. While the high-level idea is shared, LoopFM is a general framework motivated by FM-to-VM transfer ratio improvement, with theoretical insights.

3 METHOD

Overview and problem formulation. We consider a standard industrial recommendation setup: a large foundation model (FM) with trillions of parameters and multiple compact vertical models (VMs) with millions of parameters for serving. The FM is trained on aggregated cross-domain data while each VM handles a specific ranking stage or domain. In standard external KD (Liang et al., 2025), the FM generates scalar predictions $\hat{y}^F$ as soft labels for VM training. LoopFM opens a *high-bandwidth embedding channel* alongside this scalar channel, through three modular stages—*extraction*, *compression*, and *structuring*—each admitting different algorithmic choices.

Stage 1: Embedding extraction. Given an FM with layers $l_1, \ldots, l_M$, we select a subset of K layers and concatenate their activations to form a raw embedding $\mathbf{e}^{(i)} = [\mathbf{h}_{l_1}^{(i)}; \ldots; \mathbf{h}_{l_K}^{(i)}] \in \mathbb{R}^D$ for each example (*e.g.*, an item impression a for a user u: (u, a)), where $D = \sum_{k=1}^{K} d_{l_k}$ (D could be $O(10^5)$–$O(10^6)$ for industrial FMs). One principle is that the selected layers should be shallow enough to retain rich input information (Tishby & Zaslavsky, 2015), but also deep enough to capture well-learned interactions. Extraction shares inference pass with KD, adding negligible overhead.

Stage 2: Compression. The raw embeddings $\mathbf{e}^{(i)} \in \mathbb{R}^D$ are too high-dimensional for direct storage. We compress them to $\mathbf{z}^{(i)} \in \mathbb{R}^d$ ($d \ll D$) using an autoencoder co-trained with the FM, though any dimensionality reduction method (PCA, random projection, learned codebook) could serve this role:

$$\mathbf{z}^{(i)} = f_{\text{enc}}(\mathbf{e}^{(i)}), \quad \hat{\mathbf{e}}^{(i)} = f_{\text{dec}}(\mathbf{z}^{(i)}), \quad \mathcal{L}_{\text{AE}} = \|\mathbf{e}^{(i)} - \hat{\mathbf{e}}^{(i)}\|_2^2. \tag{1}$$

A stop-gradient ensures no autoencoder gradients flow into the FM backbone. The encoder's last layer uses $\tanh$ activation to bound outputs in $[-1, 1]$, facilitating INT4 quantization ($z_{\text{quant}} = \text{round}(z \cdot 8).\text{clamp}(-8, 7)/8$), which achieves $4\times$ storage reduction from FP16.

Matryoshka compression. To enable flexible embedding dimension tuning for deployment allowing trade-off of feature value vs. storage/latency cost, we use Matryoshka-style autoencoders (Kusupati et al., 2022) so that any prefix $\mathbf{z}_{1:d'}$ is a valid representation, by summing losses over target dimensions $\mathcal{D}$: $\mathcal{L}_{\text{MAE}} = \sum_{d' \in \mathcal{D}} \|\mathbf{e} - f_{\text{dec}}^{(d')}(\mathbf{z}_{1:d'})\|_2^2$.

Stage 3: Structuring. Historical compressed embeddings within a configurable time window are first grouped by a *key* k (e.g., user ID, item ID, or other semantic IDs), and then stored in appropriate structures, such as temporal sequences:

$$\mathbf{S}_k = [\mathbf{z}_{k,t_1}, \mathbf{z}_{k,t_2}, \ldots, \mathbf{z}_{k,t_L}], \quad t_L < \cdots < t_2 < t_1 < t_{cur}, \tag{2}$$

where L is the maximum sequence length, t_i is the timestamp of event i, and t_{cur} is current timestamp when the user is being served. Note we *exclude* the current sample being served to avoid requiring real-time FM inference at serving. More generally, the grouped embeddings can be structured as a graph (e.g., a user-item bipartite graph with FM embeddings on edges). In this paper, we focus on user-keyed sequences ($k = u$): for each user u, we collect their historical FM embeddings within a retention window (e.g., 30 days) and truncate to the most recent L entries (e.g., 200) to form $\mathbf{S}_u$.

VM-side integration. The VM processes $\mathbf{S}_u$ via a sequence encoder (denoted "Seq Arch" in Figure 1), such as mean/sum pooling or attention, and concatenates the pooled representation with other feature embeddings before the interaction layers. The training loss combines task loss and KD: $\mathcal{L} = \mathcal{L}_{\text{task}}(\hat{y}^V, y) + \lambda \cdot \mathcal{L}_{\text{KD}}(\hat{y}^V, \hat{y}^F)$, where $\hat{y}^V = g(\mathbf{x}_{\text{VM}}, \mathbf{S}_u; \Theta_V)$.

4 THEORETICAL ANALYSIS

We formalize the sources of LoopFM gain and analyze how FM improvement transfers to the VM.

Setup. Although the LoopFM framework applies to any grouping key (Section 3), we present the theoretical analysis using user ID key without loss of generality—all results hold verbatim by replacing "user" with any key type. Consider predicting label y given current VM features $\mathbf{x}_{\text{VM}}^{(t)}$ for a user u with L past interactions. The FM observes a superset $\mathbf{x}_{\text{FM}} = (\mathbf{x}_{\text{VM}}, \mathbf{x}_{\text{extra}})$ including cross-domain signals $\mathbf{x}_{\text{extra}}$ unavailable to the VM. We denote the user's raw VM-side history (past ad IDs, categories, etc.) as $\mathbf{H}_u := (\mathbf{x}_{\text{VM}}^{(t_1)}, \ldots, \mathbf{x}_{\text{VM}}^{(t_L)})$, and the LoopFM sequence of compressed FM_k embeddings as $\mathbf{S}_u^{(k)} := [\mathbf{z}_{u,t_1}^{(k)}, \ldots, \mathbf{z}_{u,t_L}^{(k)}]$, where each entry encodes the FM's processing of *all* features at that timestep. We compare three configurations: (i) **FM** (full features), (ii) **Baseline VM** (current features $\mathbf{x}_{\text{VM}}^{(t)}$ only), and (iii) **LoopFM VM** (augmented with $\mathbf{S}_u^{(k)}$). For each, the Bayes risk is $\mathcal{R}^*(\cdot) = H(y \mid \cdot)$ and the achieved risk is $R_{\text{ach}}(\cdot)$.

4.1 GAIN DECOMPOSITION

Theorem 1 (Gain decomposition)**.** *Let $I(\cdot\,;\,\cdot|\,\cdot)$ denote the conditional mutual information. The LoopFM information gain $\mathcal{I}_{LoopFM}(FM_k) := \mathcal{R}^{*(VM)} - \mathcal{R}^{*(LoopFM_k)}$ decomposes exactly as:*

$$\mathcal{I}_{LoopFM}(FM_k) \;=\; \underbrace{\mathcal{I}_{\text{temporal}}}_{\substack{\textit{user event}\\ \textit{interactions}}} \;+\; \underbrace{\mathcal{I}_{\text{cross},k}}_{\substack{\textit{feature-gap}\\ \textit{gain}}} \;-\; \underbrace{\mathcal{I}_{\text{residual},k}}_{\substack{\textit{compression}\\ \textit{loss}}}, \tag{3}$$

where all three terms are non-negative conditional mutual information:

- *(i) $\mathcal{I}_{\text{temporal}} := I(\mathbf{H}_u; y \mid \mathbf{x}_{VM}^{(t)})$: predictive information in user history beyond current features—how much past behavior helps predict y, independent of the FM;*
- *(ii) $\mathcal{I}_{\text{cross},k} := I(\mathbf{S}_u^{(k)}; y \mid \mathbf{x}_{VM}^{(t)}, \mathbf{H}_u)$: additional information from the FM's extra features $\mathbf{x}_{extra}$ as encoded in the LoopFM embeddings;*
- *(iii) $\mathcal{I}_{\text{residual},k} := I(\mathbf{H}_u; y \mid \mathbf{x}_{VM}^{(t)}, \mathbf{S}_u^{(k)})$: information in $\mathbf{H}_u$ not captured by $\mathbf{S}_u^{(k)}$—the price of compression: both DNN and autoencoder lead to loss of mutual info with input.*

LoopFM's gain thus comes from two sources—temporal history and cross-feature information—minus the compression cost. By the data processing inequality,

$$\mathcal{I}_{\text{cross},k} \le I(\mathbf{x}_{extra,k}^{(t_1:t_L)}; y \mid \mathbf{x}_{VM}^{(t)}, \mathbf{H}_u) =: \mathcal{I}_{\text{feature-raw},k}.$$

See Appendix F.1 for the formal theorem statement and proof.

Pipeline loss fractions. The compression loss $\mathcal{I}_{\text{residual},k}$ further decomposes along the pipeline (FM $\to$ autoencoder $\to$ quantization) into three non-negative terms (Proposition 7 in Appendix F.2), defining two retention parameters:

- $\tau_k \ge 0$: the *temporal* pipeline loss fraction, satisfying $\mathcal{I}_{\text{residual},k} \le \tau_k \cdot \mathcal{I}_{\text{temporal}}$;
- $\eta_k \in [0, 1]$: the *cross-platform* pipeline loss fraction, satisfying $\mathcal{I}_{\text{cross},k} = (1-\eta_k) \cdot \mathcal{I}_{\text{feature-raw},k}$.

Both decrease with larger FM (reducing representation loss), larger autoencoder dimension d (reducing compression loss), and finer quantization; see Appendix F.2 for detailed notation and the full pipeline decomposition. Substituting into equation 3 yields a two-sided "gain sandwich" (Corollary 10 in Appendix F.2) that bounds $\mathcal{I}_{\text{LoopFM}}$ in terms of τ_k and η_k.

4.2 TRANSFER RATIO ANALYSIS

Two teacher setup. Consider two overparameterized foundation models: FM_1 (old) with features $\mathbf{x}_{\text{FM}_1} = (\mathbf{x}_{\text{VM}}, \mathbf{x}_{\text{extra},1}) \in \mathbb{R}^{m_1}$ and $p_1 \gg m_1$ parameters, and FM_2 (new) with features $\mathbf{x}_{\text{FM}_2} = (\mathbf{x}_{\text{VM}}, \mathbf{x}_{\text{extra},1}, \mathbf{x}_{\text{extra},2}) \in \mathbb{R}^{m_2}$ and $p_2 \ge p_1$, $p_2 \gg m_2$ parameters. The VM observes $\mathbf{x}_{\text{VM}} \in$

$\mathbb{R}^{m_s}$ with $m_s < m_1 < m_2$. Each FM's achieved risk decomposes as $R_{\text{ach}}(\text{FM}_k) = \mathcal{R}^{*(\text{FM}_k)} + \epsilon_{\text{over}}(p_k, m_k)$, where ϵ_{over} is the excess risk. The total FM improvement decomposes as:

$$\Delta_{\text{teacher}} := R_{\text{ach}}(\text{FM}_1) - R_{\text{ach}}(\text{FM}_2) = \underbrace{\mathcal{R}^{*(\text{FM}_1)} - \mathcal{R}^{*(\text{FM}_2)}}_{\Delta_{\text{feat}} \geq 0} + \underbrace{\epsilon_{\text{over}}(p_1, m_1) - \epsilon_{\text{over}}(p_2, m_2)}_{\Delta_{\text{param}}}, \quad (4)$$

where Δ_{feat} is the Bayes-risk gain from richer features and Δ_{param} captures model overparameterization effects. The LoopFM *transfer ratio* (TR) is defined as

$$\text{TR}_{\text{LoopFM}} := \frac{\Delta_{\text{LoopFM}}}{\Delta_{\text{teacher}}}. \quad (5)$$

where $\Delta_{\text{LoopFM}} := R_{\text{ach}}(\text{LoopFM}_1) - R_{\text{ach}}(\text{LoopFM}_2)$ denotes the LoopFM-driven VM gain.

Assumptions (simplified; formal versions in Appendix E). We make three assumptions:

(A1) Overparameterized FM with controlled excess risk. For each FM_k, let $\epsilon_{\text{over}}(p_k, m_k) := R_{\text{ach}}(\text{FM}_k) - \mathcal{R}^{*(\text{FM}_k)}$ denote the expected excess risk (n training samples). We assume the NTK/lazy-training regime (Jacot et al., 2018) and the benign-overfitting conditions of Bartlett et al. (2020, Theorem 1) at split index m_k, yielding a two-sided bound:

$$\underline{C}_{\text{over}}\, \sigma^2\, \xi_k \;\leq\; \epsilon_{\text{over}}(p_k, m_k) \;\leq\; \bar{C}_{\text{over}}\, \sigma^2\, \xi_k, \quad (6)$$

where $\xi_k := m_k/n + n/(p_k - m_k) \to 0$ as $p_k, n \to \infty$ with $m_k = o(n)$ and $p_k \gg n$, and $0 < \underline{C}_{\text{over}} \leq \bar{C}_{\text{over}}$ are constants; see derivation in Appendix F.7.

(A2) Informative new features. For additional FM_2 features $\mathbf{x}_{\text{extra},2} = (u_{m_1+1}, \ldots, u_{m_2})$, there exist constants $0 < \underline{\kappa}_{\text{gap}} \leq \bar{\kappa}_{\text{gap}}$ and $0 < \underline{\kappa}^{\text{hist}}_{\text{gap}} \leq \bar{\kappa}^{\text{hist}}_{\text{gap}}$ such that for each $j \in \{m_1{+}1, \ldots, m_2\}$, we have the following information bounds:

(a) *current information:* $\underline{\kappa}_{\text{gap}} \leq I\left(u_j^{(t)}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{x}_{\text{extra},1}^{(t)}, u_{m_1+1:j-1}^{(t)}\right) \leq \bar{\kappa}_{\text{gap}}$;

(b) *historical information:* $\underline{\kappa}^{\text{hist}}_{\text{gap}} \leq I\left(u_j^{(t_1:t_L)}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u, \mathbf{x}_{\text{extra},1}^{(t_1:t_L)}, u_{m_1+1:j-1}^{(t_1:t_L)}\right) \leq \bar{\kappa}^{\text{hist}}_{\text{gap}}$.

(A3) Pipeline quality. The total cross-platform information lost by FM_2's pipeline (representation + autoencoder + quantization) is no worse than FM_1's: $\ell^{\text{cross}}_{\text{repr},2} + \ell^{\text{cross}}_{\text{AE},2} + \ell^{\text{cross}}_{\text{Q},2} \leq \ell^{\text{cross}}_{\text{repr},1} + \ell^{\text{cross}}_{\text{AE},1} + \ell^{\text{cross}}_{\text{Q},1}$, where $\ell^{\text{cross}}_{\text{repr},k}$, $\ell^{\text{cross}}_{\text{AE},k}$, and $\ell^{\text{cross}}_{\text{Q},k}$ are the MI losses from FM representation, autoencoder, and quantization on the cross-platform channel (Appendix F.2). It holds when $p_2 \geq p_1$ (FM capacity), $d_2 \geq d_1$ (AE bottleneck), and $b_2 \geq b_1$ (quantization bits).

Theorem 2 (Transfer-ratio bound)**.** *Assume* (A1)–(A2)*, well-trained students ($\epsilon_{\text{est},k} \to 0$), and $\Delta_{teacher} > 0$. Then it holds that*

(0) Initial launch. *When the VM has no prior LoopFM ($R_{ach}(LoopFM_1) = R_{ach}(VM)$), $\text{TR}_{LoopFM} \geq 0$ since adding $\mathbf{S}_u^{(2)}$ can only reduce Bayes risk. Corollary 10 gives a quantitative bound:*

$$\text{TR}_{LoopFM} \;\geq\; \frac{(1-\tau_2)\,\mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2}}{\Delta_{teacher}}. \quad (7)$$

(1) Negative transfer (without A3). *If the new FM's pipeline is worse ($\eta_2 > \eta_1$, violating A3), negative transfer can occur: specifically when $\tau_1 \mathcal{I}_{\text{temporal}} + (1{-}\eta_2)\mathcal{I}_{\text{feature-raw},2} - (1{-}\eta_1)\mathcal{I}_{\text{feature-raw},1} < 0$. Under* (A3)*, this condition is never satisfied (Appendix F.4).*

(2) Positive transfer (with A3). *Let $\delta := m_2 - m_1$ denote the feature gap. Under (A3), Lemma 11 gives $\eta_2 \leq \eta_1$. Then it holds that*

$$\text{TR}_{LoopFM} \;\geq\; \frac{-\tau_2\,\mathcal{I}_{\text{temporal}} + (1-\eta_1)\,\underline{\kappa}^{\text{hist}}_{\text{gap}}\,\delta}{\bar{\kappa}_{\text{gap}}\,\delta + \bar{\kappa}_{\text{over}}\,\xi_1 - \underline{\kappa}_{\text{over}}\,\xi_2} =: \text{TR}_{LB}(\delta), \quad (8)$$

where $\bar{\kappa}_{\text{over}} := \bar{C}_{\text{over}}\sigma^2$ and $\underline{\kappa}_{\text{over}} := \underline{C}_{\text{over}}\sigma^2$ arise from the two-sided (A1) envelope, and $\xi_k := m_k/n + n/(p_k - m_k)$; see detailed proof in Appendix F.4.

Remark 3 (Relaxing A1 covariance structure assumption)**.** *The bound equation 8 uses $\xi_k = m_k/n + n/(p_k - m_k)$, which assumes a bi-level covariance structure with tail effective rank $R_{m_k} = p_k - m_k$. This can be relaxed: Theorem 17 (Appendix F.8) gives the same bound with $\xi_k = d_k/n + n/R_{d_k}(\Sigma_k)$ for a general split index d_k and tail effective rank $R_{d_k}(\Sigma_k)$, valid under any benign covariance structure (Bartlett et al., 2020, Definition 4).*

Corollary 4 (Monotonicity in feature gap)**.** *Under the conditions of Theorem 2(2),* $\mathrm{TR}_{LB}(\delta)$ *is monotonically increasing in* δ *for all* $\delta \geq 0$*, and converges to* $(1-\eta_1)\,\underline{\kappa}_{\mathrm{gap}}^{\mathrm{hist}}/\bar{\kappa}_{\mathrm{gap}} > 0$ *as* $\delta \to \infty$*.*

Theorem 5 (Sequence length; informal)**.** *The LoopFM information gain* $\mathcal{I}_{LoopFM,k}(L) := I(\mathbf{S}_u^{(k,L)}; y \mid \mathbf{x}_{VM}^{(t)})$ *is non-decreasing in* L *and converges to a constant* $\mathcal{I}^*_{LoopFM,k} \leq H(y \mid \mathbf{x}_{VM}^{(t)})$*.*

The formal statement and proof appear in Appendix F.6 (Proposition 13).

Implications. **(1) Pipeline quality controls transfer:** reducing τ_k and η_k—via larger AE dimension d, finer quantization b, or stronger FM—directly raises the transfer-ratio bound equation 8. When pipeline losses dominate, the TR becomes smaller or even turns negative: the student worsens despite a better teacher. This motivates careful autoencoder design and the Matryoshka multi-granularity approach. **(2) Impact of feature gap δ:** Corollary 4 shows a larger feature gap $\delta = m_2 - m_1$ between FM generations monotonically increases the transfer ratio lower bound, with diminishing marginal returns as $\mathrm{TR}_{\mathrm{LB}}$ saturates at $(1-\eta_1)\underline{\kappa}_{\mathrm{gap}}^{\mathrm{hist}}/\bar{\kappa}_{\mathrm{gap}}$. This means LoopFM benefits most from FM upgrades that introduce many new features, and the pipeline loss η_1 is the dominant bottleneck limiting how much of this benefit is realized—motivating the *Matryoshka autoencoder* for flexible compression. **(3) Diminishing returns in sequence length:** Theorem 5 guarantees monotone gains in L with diminishing marginal returns (since the total is bounded by $H(y)$), consistent with Table 5.

5 Experiments

5.1 Experiment Setup

Public datasets. We use three public datasets: **TaobaoAd** (Tianchi, 2018) (25M samples, 22 features), **KuaiVideo** (Kuaishou, 2018) (13.7M samples, 9 features), and **Amazon Electronics** (He & McAuley, 2016) (3M samples, 6 features). We present main results and ablations on TaobaoAd (the richest feature set); results on KuaiVideo and Amazon Electronics are in Appendix B and C. TaobaoAd is partitioned into 8 calendar days, and we follow a temporal streaming protocol by Liang et al. (2025) and a typical VM training setup for Ads CTR models: FM (DMIN (Xiao et al., 2020)) is trained on days 1–4 and evaluated on days 5–8 to log labels and embeddings for KD and LoopFM sequence building; VM uses *one-pass* streaming (no shuffle) training on days 5–7 (no validation for early stopping needed), and we report test metrics on day 8.

Models. The FM is DMIN (Xiao et al., 2020) with BARS-optimal hyperparameters (embedding dim 32, DNN [512, 256, 128]) and a co-trained autoencoder (d=32). We evaluate six VMs: non-sequential (FM, FmFM, DeepFM (Guo et al., 2017)) and sequential (DIEN (Zhou et al., 2019), DMR (Lyu et al., 2020), DMIN), each extended (if needed) with a DMIN-style sequence module for LoopFM (d=32, L=50). KD uses the auxiliary head framework from Liang et al. (2025) (α=5, γ=10, β=5 on TaobaoAd; tuned per-dataset in appendix). Internal experiments use trillion-parameter FMs with billions of daily training examples and INT4 quantization for storage efficiency; public benchmark experiments use FP32 embeddings without quantization. Our public benchmark implementation is built on FuxiCTR (Zhu et al., 2022; 2021).

Metrics. **NE** (Normalized Entropy) (He et al., 2014): normalized LogLoss (lower is better). **AUC** (higher is better). **Transfer ratio**: TR $= \Delta\mathrm{NE}_{\mathrm{VM}}/\Delta\mathrm{NE}_{\mathrm{FM}}$. "KD" refers to auxiliary-head-based external KD (Liang et al., 2025) throughout.

5.2 Research Questions

- **RQ1** Does LoopFM improve over scalar KD? (Section 5.3.1)
- **RQ2** How does LoopFM compare to other embedding-based transfer methods? (Section 5.3.2)
- **RQ3** Does sequence structure matter, or is pooling sufficient? (Section 5.3.3)
- **RQ4** Are FM-learned sequences better than hand-crafted raw ID sequences? (Section 5.3.4)
- **RQ5** How do layer selection, checkpoint frequency, sequence length, and embedding dimension affect performance? (Section 5.3.5)
- **RQ6** How does FM capacity affect LoopFM's benefit? (Appendix G)

5.3 PUBLIC DATASET RESULTS

5.3.1 RQ1: DOES LOOPFM IMPROVE OVER KD?

Table 1 presents results on TaobaoAd across six VMs (we underline FM for factorization machine to distinguish from FM). LoopFM consistently improves over KD across all architectures. When the VM architecture is identical to the FM (DMIN), KD shows little gain as expected, yet LoopFM still brings 6.52% AUC gain—strong evidence of complementary value. A variance study with 5 seeds on DeepFM (Appendix D) confirms the gains are statistically significant ($p < 0.001$, paired t-test); KD+LoopFM achieves 0.6342 ± 0.0001 AUC with the lowest variance across all methods.

Table 1: LoopFM on TaobaoAd across six VMs (FM: DMIN). Non-sequential (FM, FmFM, DeepFM) and sequential (DIEN, DMR, DMIN) architectures, each extended with a DMIN-style module for LoopFM sequence.

	FM		FmFM		DeepFM		DIEN		DMR		DMIN	
Method	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss
w/o distill	0.5790	0.1979	0.5833	0.1977	0.5886	0.1979	0.5945	0.1963	0.5967	0.1966	0.5932	0.1966
KD	0.5855	0.1979	0.5885	0.1990	0.5980	0.1964	0.5958	0.1966	0.6002	0.1962	0.6002	0.1961
LoopFM-only	0.6203	0.1953	0.6066	0.2020	0.6245	0.1955	0.6335	0.1951	**0.6361**	0.1935	0.6319	0.1950
KD + LoopFM	**0.6205**	**0.1950**	**0.6106**	**0.1967**	**0.6344**	**0.1940**	**0.6354**	**0.1934**	0.6356	**0.1934**	**0.6346**	**0.1935**

5.3.2 RQ2: COMPARISON WITH EMBEDDING-BASED TRANSFER METHODS

Table 2 compares LoopFM against three alternative embedding-based transfer methods: using the FM's current-sample embedding as a VM input feature (requiring real-time FM inference), matching representations via a FitNets-style (Romero et al., 2015) auxiliary loss, and using user-only embeddings without interaction information. All three provide negligible improvement (+0.0003–0.0005 AUC), while LoopFM achieves +0.0364 (+6.09% relative), demonstrating the unique value of *historical embedding sequences*.

Table 2: Embedding transfer methods (TaobaoAd, FM: DMIN, VM: DeepFM, all build on KD).

Method	AUC	LogLoss
KD (baseline)	0.5980	0.1964
+ Current-Emb-as-Feature	0.5985	0.1964
+ Current-Emb-Loss (FitNets)	0.5984	0.1965
+ Entity-Only Embedding	0.5983	0.1965
+ LoopFM	**0.6344**	**0.1940**

5.3.3 RQ3: SEQUENCE STRUCTURE VS. POOLED EMBEDDINGS

Table 3 compares four aggregation strategies. All methods, including simple mean-pool, show significant gain over the KD baseline (0.5980). Sum-pool is surprisingly strong, suggesting that even storing sum-pooled historical embeddings as a feature (much lower storage/modeling cost) could already capture substantial gain. DMIN attention brings further significant improvement over sum-pool. We also observe internally that LoopFM gain is more pronounced as we scale up the sequence modeling module (Appendix A).

Table 3: Sequence aggregation ablation (TaobaoAd, FM: DMIN, VM: DeepFM, all build on KD).

Method	AUC	LogLoss
+ Mean-Pool Historical Emb	0.6118	0.1955
+ Sum-Pool Historical Emb	0.6300	0.1941
+ LoopFM (DIN attention)	0.6112	0.1954
+ LoopFM (DMIN attention)	**0.6344**	**0.1940**

5.3.4 RQ4: FM-LEARNED EMBEDDINGS VS. RAW SEQUENCES

Table 4 compares LoopFM sequences against custom-built raw user behavior sequences (all four ad-side ID features: ad group, campaign, advertiser, bucketized price) using DMIN as VM at two capacities: matching the FM (emb=32) and 16× smaller (emb=2).

When the VM is identical to the FM (left), LoopFM only slightly outperforms raw sequences

Table 4: Raw sequences vs. LoopFM at two VM capacity levels (TaobaoAd, FM: DMIN, no KD).

	VM = FM (emb=32)		Small VM (emb=2)	
Method	AUC	LogLoss	AUC	LogLoss
DMIN Baseline	0.5932	0.1966	0.5843	0.1967
+ Raw User Seq	0.6302	0.1942	0.6186	0.1950
+ LoopFM	0.6319	0.1950	0.6233	0.1955
+ Both	**0.6362**	**0.1942**	**0.6305**	**0.1941**

(+0.0017 AUC, +0.27% relative)—as expected, since the VM can learn equally effective embeddings directly from raw IDs. Combining both still yields the best result (0.6362). When the VM is much smaller (right), LoopFM's advantage widens to +0.0047 AUC (+0.76% relative) over raw sequences: a capacity-limited VM cannot effectively learn from high-cardinality IDs with tiny embeddings, whereas LoopFM's well-learned representations are directly consumable regardless of VM capacity. LoopFM's marginal gain on top of raw sequences is larger for the small VM (+0.0119) than for the large VM (+0.0060), confirming that LoopFM's value increases with the FM-VM capacity gap—a typical production setting.

5.3.5 RQ5: HYPERPARAMETER SENSITIVITY

Table 5 ablates key hyperparameters.

FM layer selection. We test the embedding layer (concatenated feature lookups before any DNN processing), each of the FM's three DNN hidden layers, joint compression of all layers, soft-label only (FM prediction score as a 1-d sequence), and item-only embeddings (raw item-side feature lookups without user/context features), compressed via separately trained AEs.

Shallower layers tend to transfer better (Emb. layer 0.6338, Hidden-0 0.6340 vs. Deep 0.6326), consistent with the information bottleneck view that deeper layers progressively discard input details (Tishby & Zaslavsky, 2015). However, the embedding layer—which contains no learned cross-feature interactions—performs slightly below Hidden-0, suggesting that interaction learning adds value. Despite only minor differences across layers, the overall pattern perfectly aligns with the layer selection principle mentioned earlier. The soft-label alone as a 1-d sequence (0.6286) still provides strong gains, indicating that even the *temporal structure* of FM predictions carries significant information. Joint compression of all layers (0.6341) achieves the best AUC; internally, we use this approach over a heuristically chosen set spanning shallow to deep layers.

Item-only embeddings (0.6291) capture 86% of the full interaction embedding's gain over the KD baseline, despite containing no user or context information. On TaobaoAd's shallow FM, item identity features (adgroup_id, campaign_id) are already highly predictive, so the DNN's cross-feature interactions add modest value. The gap should widen with deeper industrial FMs where the DNN contributes a larger fraction of the representation's information. Finding the sweet spot—deep enough to capture rich interactions, but not so deep that compression discards too much—remains an open question (see Appendix H).

Table 5: Hyperparameter ablation (TaobaoAd, FM: DMIN, VM: DeepFM, KD+LoopFM). †Ckpt frequency uses a separately trained fresh FM per split. ‡Item-side feature embeddings only (6 ad features, no user/context, no DNN), compressed via same AE architecture.

Setting	AUC	LL
FM layer		
Emb. layer	.6338	**.1936**
Hidden-0 (default)	.6340	.1938
Hidden-1	.6328	.1939
Deep	.6326	.1938
All (joint, d=64)	**.6341**	.1939
Soft-label (d=1)	.6286	.1940
Item-only emb‡	.6291	.1942
FM ckpt frequency†		
Fixed (default)	**.6344**	**.1940**
Every split	.6311	.1945
Sequence length L		
L=10	.6327	.1944
L=25	.6338	.1941
L=50 (default)	.6346	.1940
L=75	.6347	**.1937**
L=100	**.6352**	.1938
Emb. dim d (Matryoshka AE)		
d=8	.6339	.1941
d=16	.6337	.1937
d=32 (default)	**.6345**	**.1937**
d=64	.6341	.1939
d=128	.6342	.1941

FM checkpoint frequency. Counterintuitively, updating the FM checkpoint every split slightly *degrades* performance (AUC 0.6311 vs. 0.6344 for a fixed checkpoint), despite better soft-labels (Liang et al., 2025). We hypothesize this is due to embedding drift: with a fixed checkpoint, embedding centroid drift is near-zero (0.000006–0.000017), so embeddings from day 5 and day 8 live in the same space and the sequence encoder can learn consistent patterns; with updated checkpoints, centroid drift is $\sim$1000$\times$ larger (0.013–0.021), so embeddings from different days live in drifting spaces, making it more challenging for the sequence encoder to learn. On the other hand, embeddings are expected to drift with natural data distribution—identifying the staleness threshold beyond which freshness outweighs consistency remains an open question.

Effect of sequence length L. Performance increases monotonically with L but with diminishing returns (Table 5), corroborating Theorem 5: AUC improves from 0.6327 (L=10) to 0.6352 (L=100), with most gain captured by L=50 (0.6346).

Embedding dimension d. Using Matryoshka autoencoders (Kusupati et al., 2022), performance is stable across dimensions (AUC spread 0.0008 from d=8 to d=128). Internally, where FMs are much larger, higher dimensions unlock significant additional gain when paired with a wider sequence encoder (Appendix A).

5.4 Internal Results: Validation at Scale

We validate LoopFM at industrial systems with trillion-parameter FMs and million-parameter VMs.

Transfer ratio. Across multiple FM-VM configurations and FM updates, adding LoopFM approximately doubles the transfer ratio of KD alone, with the combined gain consistently exceeding either channel alone—confirming that KD and LoopFM transfer largely orthogonal knowledge. Additional ablations on embedding dimensions are in Appendix A.

Quantization. INT8 quantization is near-lossless; vanilla INT4 introduces a 0.03–0.07% NE gap (where 0.02% is considered significant in production). K-means INT4—learning 16 non-uniform cluster centers adapted to the $\tanh$-shaped embedding distribution—reduces this gap to 0.01–0.02%, achieving 4× compression with minimal quality loss.

Online results. LoopFM v1 was first launched to production on a single surface in Y1H1. Later, LoopFM v2 expanded to three additional surfaces with 4× storage cost reduction via INT4 quantization while LoopFM v3 uses the latest FM. LoopFM has improved ad conversion by +0.5% in the first half after its initial launch, and by +1.03% and +1.22% from two individual launches in the subsequent half.

FM compute scaling. Figure 2 plots the relative NE gain of LoopFM versus FM compute budget (measured in FLOPs) on a logarithmic scale. The results show that LoopFM performance follows a predictable power-law relationship with FM compute.

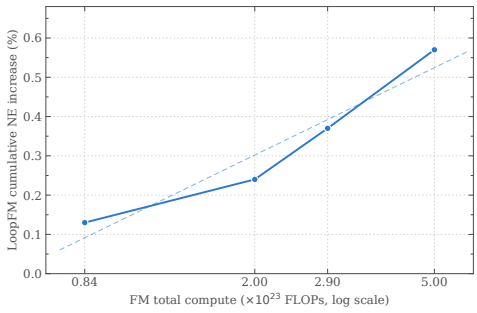


Figure 2: Relative NE gain (%) versus FM training compute (FLOPs, log scale).

6 Conclusion

We presented LoopFM, a framework that complements scalar KD with a high-bandwidth embedding channel—extracting, compressing, and structuring historical FM representations as direct VM input features. The key insight is that *what* knowledge is transferred matters as much as *how*: while KD captures the FM's current prediction, LoopFM provides a general framework of structuring users' historical embeddings, encoding cross-feature patterns that a single scalar cannot convey. Our theoretical analysis formalizes this via a gain decomposition into temporal, cross-feature and compression-loss terms, with a transfer-ratio bound that tightens as compression improves.

Empirical findings have practical implications beyond LoopFM itself. First, our embedding-based and scalar-based transfer are largely orthogonal—combining them outperforms either alone in most settings, suggesting that multi-channel knowledge transfer should be a default design pattern in FM-VM systems. Second, shallower FM layers transfer better than deeper ones, revealing a depth-vs-compression tradeoff: shallower representations are information-richer but contain less interaction learning, and finding the optimal extraction point remains an open problem. Third, within the timescales we tested (2–3 days), embedding consistency within a user's sequence matters more than individual embedding freshness—though identifying the staleness threshold where this reverses remains open. Fourth, LoopFM demonstrates a predictable scaling law.

LoopFM's three-stage design is modular: each stage can evolve independently as better compression methods, richer structuring (e.g., graph-based), or new grouping keys emerge. Intriguing future directions include *self-LoopFM*—having the FM consume its own historical embeddings to create a self-improving loop (Appendix J).

AI USE STATEMENT

We used generative AI tools (large language models and AI coding assistants) for four purposes. **Writing:** polishing prose and formatting LaTeX. **Code:** drafting and debugging experiment, data-processing, and plotting scripts; the authors inspected and re-ran all such code, and no result was generated or altered by an AI tool. **Literature:** surfacing and summarizing candidate related work; the authors read every cited paper in the original and confirmed no reference was fabricated. **Theory:** drafting and error-checking the formal proofs in Appendix F; the assumptions (Appendix E) and proof strategies are the authors' own, and every step was verified by the authors. We did not use AI for the research idea and the LoopFM design.

REFERENCES

Rohan Anil, Sandra Gadanho, Da Huang, Nijith Jacob, Zhuoshu Li, Dong Lin, Todd Phillips, Cristina Pop, Kevin Regan, Gil I Shamir, et al. On the factory floor: Ml engineering for industrial-scale ads recommendation models. In *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. Unbiased knowledge distillation for recommendation. In *WSDM*, pp. 976–984, 2023.

Zhimin Chen, Chenyu Zhao, Ka Chun Mo, Yunjiang Jiang, Jane H. Lee, Khushhall Chandra Mahajan, Ning Jiang, Kai Ren, Jinhui Li, and Wen-Yun Yang. Massive memorization with hundreds of trillions of parameters for sequential transducer generative recommenders. *arXiv preprint arXiv:2510.22049*, 2025.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4794–4802, 2019.

Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. Distillation matters: Empowering sequential recommenders to match the performance of large language model. *arXiv preprint arXiv:2405.00338*, 2024.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1725–1731, 2017.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international conference on world wide web*, pp. 507–517, 2016.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at Facebook. In *Proceedings of the eighth international workshop on data mining for online advertising*, pp. 1–9, 2014.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Bojian Hou, Xiaolong Liu, Xiaoyi Liu, Jiaqi Xu, Yasmine Badr, Mengyue Hang, Sudhanshu Chanpuriya, Junqing Zhou, Yuhang Yang, Han Xu, Qiuling Suo, Laming Chen, Yuxi Hu, Jiasheng Zhang, Huaqing Xiong, Yuzhen Huang, Chao Chen, Yue Dong, Yi Yang, Shuo Chang, Xiaorui Gan, Wenlin Chen, Santanu Kolay, Darren Liu, Jade Nie, Chunzhi Yang, Ellie Wen, Jiyan Yang, and Huayu Li. Kunlun: Establishing scaling laws for massive-scale recommendation systems through unified architecture design. *arXiv preprint arXiv:2602.10016*, 2026.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

SeongKu Kang, Wonbin Kweon, Dongha Lee, Jianxun Lian, Xing Xie, and Hwanjo Yu. Distillation from heterogeneous models for top-k recommendation. In *Proceedings of the ACM Web Conference 2023*, pp. 801–811, 2023.

SeongKu Kang, Wonbin Kweon, Dongha Lee, Jianxun Lian, Xing Xie, and Hwanjo Yu. Unbiased, effective, and efficient distillation from heterogeneous models for recommender systems. *ACM Transactions on Recommender Systems*, 2024.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018.

Nikhil Khani, Li Wei, Aniruddh Nath, Shawn Andrews, Shuo Yang, Yang Liu, Pendo Abbo, Maciej Kula, Jarrod Kahn, Zhe Zhao, et al. Bridging the gap: Unpacking the hidden challenges in knowledge distillation for online ranking systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 758–761, 2024.

Kuaishou. Kuaivideo. `https://www.kuaishou.com/activity/uimc`, 2018.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Chenglin Li, Yuanzhen Xie, Chenyun Yu, Bo Hu, Zang Li, Guoqiang Shu, Xiaohu Qie, and Di Niu. One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 366–374, 2023.

Xinchun Li, Ning Zhang, Qianqian Yang, Fei Teng, Wenlin Zhao, Huizhi Yang, Heng Shi, Linlan Chen, Yixin Wu, Zhen Wang, et al. Iat: Instance-as-token compression for historical user sequence modeling in industrial recommender systems. *arXiv preprint arXiv:2604.08933*, 2026.

Mingfu Liang, Xi Liu, Rong Jin, Boyang Liu, Qiuling Suo, Qinghai Zhou, Song Zhou, Laming Chen, Hua Zheng, Zhiyuan Li, et al. External large foundation model: How to efficiently serve trillions of parameters for online ads recommendation. *arXiv preprint arXiv:2502.17494*, 2025.

Ze Lyu, Yu Dong, Chengfu Huo, and Weijun Ren. Deep match to rank model for personalized click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 156–163, 2020.

Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2685–2692, 2020.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Karim, Antoine Chassang, Carlo Gatti, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformers. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019.

Jiaxi Tang and Ke Wang. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *SIGKDD*, pp. 2289–2298, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

Tianchi. Ad display/click data on taobao.com. https://tianchi.aliyun.com/dataset/56, 2018.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *arXiv preprint arXiv:1503.02406*, 2015.

Ke Wang, Vidya Muthukumar, and Christos Thrampoulidis. Benign overfitting in multiclass classification: All roads lead to interpolation. *Advances in Neural Information Processing Systems*, 36, 2023.

Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. Deep multi-interest network for click-through rate prediction. In *CIKM*, pp. 2265–2268, 2020.

Lee Xiong, Zhirong Chen, Rahul Mayuranath, Shangran Qiu, Arda Ozdemir, Lu Li, Yang Hu, Dave Li, Jingtao Ren, Howard Cheng, et al. LLaTTE: Scaling laws for multi-stage sequence modeling in large-scale ads recommendation. *arXiv preprint arXiv:2601.20083*, 2026.

Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. Privileged features distillation at Taobao recommendations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2590–2598, 2020.

Shuo Yang, Sujay Sanghavi, Holakou Rahmanian, Jan Bakus, and S.V.N. Vishwanathan. Toward understanding privileged features distillation in learning-to-rank. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

Buyun Zhang, Liang Luo, Yuxin Chen, Jade Nie, Xi Liu, Shen Li, Yanli Zhao, Yuchen Hao, Yantao Yao, Ellie Dingqiao Wen, et al. Wukong: Towards a scaling law for large-scale recommendation. In *Forty-first International Conference on Machine Learning*, 2024a.

Wei Zhang, Dai Li, Chen Liang, Fang Zhou, Zhongke Zhang, Xuewei Wang, Ru Li, Yi Zhou, Yaning Huang, Dong Liang, et al. Scaling user modeling: Large-scale online user representations for ads personalization in Meta. In *Companion Proceedings of the ACM on Web Conference 2024*, pp. 47–55, 2024b.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1059–1068, 2018.

Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5941–5948, 2019.

Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2759–2769, 2021.

Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. BARS: towards open benchmarking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2912–2923, 2022. https://github.com/reczoo/BARS.

A Additional Internal Experiments

We ablate the LoopFM embedding dimension and the sequence encoder's hidden width (d_{model}, i.e., the model dimension of the self-attention-based sequence encoder that processes LoopFM features) on an internal downstream model. NE gains range from 0.09% to 0.14% depending on the configuration, with three key findings:

1. **A small d_{model} bottlenecks transfer.** With d_{model} fixed at the base width, increasing the embedding dimension yields no additional NE gain (all achieve 0.09%), indicating that the sequence architecture must be wide enough to utilize the richer LoopFM information.
2. **Embedding dimension bounds the information content.** Conversely, when the embedding dimension is small and fixed, enlarging d_{model} provides very limited improvement (0.09% $\rightarrow$ 0.10%), because the LoopFM feature itself carries limited information.
3. **Both dimensions must scale together.** The largest gains (0.13–0.14%) are realized when a high embedding dimension is paired with a sufficiently expressive sequence architecture. Increasing d_{model} from the base width to $\sim 3\times$ improves NE gain from 0.09% to 0.13% at the highest embedding dimension.

These results are consistent with the theoretical analysis in Section 4: the embedding dimension controls the total information available in the LoopFM feature (analogous to the AE dimension d in Theorem 2), while d_{model} governs the downstream model's capacity to absorb that information. Corollary 4 predicts that enlarging the feature set alone yields diminishing returns once the pipeline bottleneck (here, d_{model}) saturates—precisely the plateau observed when only the embedding dimension is scaled.

B KuaiVideo Results

Table 6: LoopFM on KuaiVideo (FM: DMIN, BARS-tuned hyperparameters (Zhu et al., 2022)) across four VM architectures. Features: user_id, item_id, item_emb (64-dim visual embedding), pos_items/neg_items (ID sequences), pos_items_emb/neg_items_emb (visual embedding sequences). LoopFM uses watched-only sequences (label=1), d=32, L=100.

	DeepFM		DIEN		DMR		DMIN	
Method	AUC	LL	AUC	LL	AUC	LL	AUC	LL
w/o distill	.6877	.5419	.7115	.4767	.7158	.4763	.7176	.4724
KD	**.6998**	**.5125**	.7223	.4568	.7261	.4528	.7233	.4611
LoopFM-only (mean)	.6919	.5558	.7158	.4565	.7271	.4516	.7261	.4553
LoopFM-only (dmin)	.6898	.5137	.7212	.4542	.7244	.4617	.7244	.4600
KD + LoopFM (mean)	.6965	.5204	.7217	**.4531**	**.7280**	.4527	**.7265**	**.4527**
KD + LoopFM (dmin)	.6847	.5334	**.7252**	.4550	.7267	**.4512**	.7257	.4612

Table 6 presents results on KuaiVideo across four DNN-based VMs with both mean pooling and DMIN attention for the LoopFM sequence encoder. We exclude FM and FmFM as KuaiVideo's features are dominated by behavioral and visual sequences that these non-DNN models cannot process well. KuaiVideo features include user_id, item_id, pretrained visual embeddings (item_emb, 64-dim), behavioral ID sequences (pos_items, neg_items), and their corresponding visual embedding sequences (pos_items_emb, neg_items_emb). Mean pooling often outperforms DMIN attention on this dataset, in contrast to TaobaoAd where DMIN attention strongly outperforms mean pooling (Table 3). We hypothesize that the pretrained visual embeddings dominate the LoopFM embedding space on KuaiVideo, making attention-based aggregation less effective. We also use watched-only sequences (label=1 only), as KuaiVideo's high interaction density ($\sim$215 per user per day) means most events are skips; including them dilutes the user-interest signal with item-specific noise. Adding a separate unwatched sequence (analogous to neg_items_emb) could provide complementary negative-interest signals, but this is not the main focus of this paper. LoopFM provides consistent AUC gains across all VMs with both pooling modes. The best overall result is DMR KD+LoopFM (mean) at 0.7280 (+1.7% over baseline). For DeepFM, KD alone outperforms all KD+LoopFM variants (notably

KD+LoopFM dmin drops to 0.6847, a case of negative transfer); KD+LoopFM (dmin) achieves the best AUC for DIEN (0.7252). We note that DeepFM is not well-suited for KuaiVideo, where behavioral sequences are the dominant features—its baseline *without* sequence features (0.7056, Table 7) actually outperforms the baseline *with* sequences (0.6877), indicating that DeepFM struggles to leverage raw sequence features on this dataset.

Compared to TaobaoAd, LoopFM gains are smaller on KuaiVideo (+0.6–1.6% vs. +6.1–6.6%). We attribute this primarily to lack of *feature richness*: TaobaoAd has 22 features spanning user demographics, ad properties, and context, so LoopFM embeddings encode rich cross-feature interactions beyond what the raw ID-based sequences capture. KuaiVideo has far fewer features (mostly IDs and visual embeddings), and its existing behavioral sequences already cover most of the available signal—leaving less incremental value for LoopFM to add. This is consistent with the theoretical gain decomposition (Theorem 1), where the cross-feature term $\mathcal{I}_{\text{cross}}$ is larger when the FM processes richer features than the VM's sequences alone, as demonstrated below on no-sequence baseline. In typical production settings, FMs train on hundreds to thousands of cross-domain features while VMs are constrained to a small feature subset due to latency requirements, creating a large feature gap that LoopFM is well-positioned to bridge.

B.1 No-Sequence Baseline

Table 7: LoopFM on KuaiVideo with no-sequence baseline (user_id + item_id + item_emb only, VM: DeepFM). Same FM, LoopFM setup (watched-only, mean pooling, d=32, L=100) as Table 6.

Method	AUC	LogLoss
w/o distill	0.7056	0.4710
KD	0.7071	0.4634
LoopFM-only (mean)	**0.7167**	**0.4584**
KD + LoopFM (mean)	0.7146	**0.4584**

Table 7 presents results on a no-sequence baseline where the VM (DeepFM) has access to only user_id, item_id, and item_emb—no behavioral sequence features. Note that without sequence features, the sequence-processing components of DMIN, DIEN, and DMR are vacuous, so we use DeepFM as the base model and report only DeepFM here. LoopFM-only achieves +1.6% AUC gain, substantially larger than the +0.6% on the baseline with sequences (Table 6), confirming that LoopFM's value is greatest when the VM lacks its own sequence feature inputs. KD+LoopFM (0.7146) slightly underperforms LoopFM-only (0.7167) here, likely because KD parameters are not re-tuned on this no-sequence baseline. Notably, DeepFM's no-sequence baseline (0.7056) is even higher than its baseline with sequences (0.6877 in Table 6), as discussed above.

C Amazon Electronics Results

Amazon Electronics (He & McAuley, 2016) contains user reviews on electronics products with features: user_id, item_id, cate_id, and pre-built behavioral sequences (item_history, cate_history). Unlike TaobaoAd, this dataset does not contain timestamps; we use the provided sequential ordering and split into 8 temporal chunks following the same streaming protocol. The FM and VMs use BARS-tuned hyperparameters (emb_dim=64, hidden=[1024, 512, 256]).

Table 8: LoopFM on Amazon Electronics (FM: DMIN) across four DNN VM architectures. KD: α=5, γ=10, β=5.

	DeepFM		DIEN		DMR		DMIN	
Method	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss
w/o distill	0.8035	0.6327	0.8490	0.4948	0.8456	0.4882	0.8386	0.4947
KD	0.8119	0.6086	0.8640	**0.4627**	0.8631	0.4622	0.8551	**0.4751**
LoopFM-only	0.8127	0.6110	0.8537	0.4772	0.8475	0.4866	0.8388	0.4995
KD + LoopFM	**0.8179**	**0.5710**	**0.8645**	0.4646	**0.8650**	**0.4608**	**0.8554**	0.4809

Table 8 presents results on Amazon Electronics across four DNN-based VMs. Similar to KuaiVideo, we exclude FM and FmFM because Amazon's dominant features are behavioral sequences (item_history, cate_history), which these non-sequential models cannot process well. KD + LoopFM achieves the best AUC for all four VMs: DeepFM (0.8179), DIEN (0.8645), DMR (0.8650), and DMIN (0.8554). Note that while AUC consistently improves, LogLoss slightly degrades for DIEN and DMIN when adding LoopFM to KD, suggesting that the additional sequence features can affect calibration even when ranking quality improves.

Compared to TaobaoAd, KD is the dominant signal on Amazon (+1.8–2.1% AUC for sequential VMs), while LoopFM-only provides modest gains (+0.02–1.1%). Similar to KuaiVideo, we attribute this to the limited feature richness—and even more so here: Amazon has only 6 features (user_id, item_id, cate_id, and their history sequences), so the existing sequences already capture most of the available signal, leaving minimal cross-feature interactions for LoopFM to add values. KD + LoopFM nonetheless consistently outperforms KD alone for all four VMs, with DMR achieving the best overall AUC of 0.8650 (+2.29% relative over baseline).

D Seed Variance Study

To verify statistical significance of single-run results, we repeat the four main configurations (Baseline, KD, LoopFM-only, KD+LoopFM) on TaobaoAd with 5 random seeds using DeepFM as VM with the same setup as Table 1 (no shuffle, streaming, 1 epoch). Table 9 reports the results. All gains are highly statistically significant under paired t-tests ($p < 0.001$): LoopFM-only achieves $+0.0347 \pm 0.0004$ AUC over baseline (t=189.5), and KD+LoopFM achieves $+0.0458 \pm 0.0004$ (t=265.2). KD+LoopFM has the lowest variance (std $= 0.0001$), suggesting the two transfer channels stabilize each other.

Table 9: Full pipeline variance study on TaobaoAd (VM: DeepFM, FM: DMIN, 5 seeds). All gains over baseline are significant at $p < 0.001$ (paired t-test, df=4).

Method	AUC (mean ± std)	Range	ΔAUC vs. baseline
Baseline	0.5884 ± 0.0004	0.0011	—
KD	0.5978 ± 0.0004	0.0011	+0.0094 (t=32.6)
LoopFM-only	0.6230 ± 0.0005	0.0015	+0.0347 (t=189.5)
KD + LoopFM	$\mathbf{0.6342 \pm 0.0001}$	0.0003	+0.0458 (t=265.2)

E Formal Assumptions for Theoretical Analysis

We state the assumptions used in Theorem 2. All notation follows Sections 3 and 4.

(A1) NTK-linearized benign-overfitting assumptions. For each FM_k, define the expected excess risk $\epsilon_{\text{over}}(p_k, m_k) := R_{\text{ach}}(\text{FM}_k) - \mathcal{R}^{*(\text{FM}_k)}$ with n training samples. Let $\phi_k(u, a) \in \mathbb{R}^{p_k}$ denote the local linearized feature map and $\Sigma_k := \mathbb{E}[\phi_k \phi_k^\top]$ with eigenvalues $\lambda_{k,1} \geq \cdots \geq \lambda_{k,p_k} \geq 0$. Here m_k denotes the Bartlett split index, with $1 \leq m_k < p_k$. We assume:

- (A1.i) *NTK/lazy-training linearization:* training remains in a local kernel regime so FM_k is well-approximated by a linear predictor on ϕ_k (Jacot et al., 2018).
- (A1.ii) *Bartlett benign-overfitting conditions:* for split index $m_k < p_k$, the tail functionals $r_{m_k}(\Sigma_k)$ and $R_{m_k}(\Sigma_k)$ satisfy the finite-sample conditions in Bartlett et al. (2020, Theorem 1), with $p_k \gg n$ and $m_k = o(n)$.

Then Appendix F.7 yields the two-sided envelope:

$$\underline{C}_{\text{over}}\, \sigma^2\, \xi_k \leq \epsilon_{\text{over}}(p_k, m_k) \leq \bar{C}_{\text{over}}\, \sigma^2\, \xi_k, \tag{9}$$

where $\xi_k := m_k/n + n/(p_k - m_k) \to 0$ as $p_k, n \to \infty$, and $0 < \underline{C}_{\text{over}} \leq \bar{C}_{\text{over}}$ are shared constants.

Justification for (A1). The specialization $\xi_k = m_k/n + n/(p_k - m_k)$ assumes a bi-level (spiked) covariance where the $p_k - m_k$ tail eigenvalues are approximately equal, giving tail effective rank $R_{m_k} = p_k - m_k$ (see Corollary 16 for the derivation). This is natural for overparameterized FMs: the NTK covariance has m_k signal directions aligned with input features, and $p_k - m_k \gg n$ near-uniform tail directions from overparameterization, matching the bi-level structure of Bartlett et al. (2020, Theorem 2.2). For more general covariance structures, Theorem 17 (Appendix F.8) gives the same TR bound with $\xi_k = m_k/n + n/R_{m_k}(\Sigma_k)$, valid under any benign covariance satisfying Bartlett et al. (2020, Definition 4).

(A2) Per-feature mutual information bound. Write the additional FM_2 feature block as $\mathbf{x}_{\text{extra},2} = (u_{m_1+1}, \ldots, u_{m_2})$. For each $j \in \{m_1+1, \ldots, m_2\}$:

(a) *Current-step:* there exist constants $0 < \underline{\kappa}_{\text{gap}} \le \bar{\kappa}_{\text{gap}}$ such that

$$\underline{\kappa}_{\text{gap}} \le I\Big(u_j^{(t)}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{x}_{\text{extra},1}^{(t)}, u_{m_1+1:j-1}^{(t)}\Big) \le \bar{\kappa}_{\text{gap}}. \tag{10}$$

(b) *Historical:* there exist constants $0 < \underline{\kappa}_{\text{gap}}^{\text{hist}} \le \bar{\kappa}_{\text{gap}}^{\text{hist}}$ such that

$$\underline{\kappa}_{\text{gap}}^{\text{hist}} \le I\Big(u_j^{(t_1:t_L)}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u, \mathbf{x}_{\text{extra},1}^{(t_1:t_L)}, u_{m_1+1:j-1}^{(t_1:t_L)}\Big) \le \bar{\kappa}_{\text{gap}}^{\text{hist}}. \tag{11}$$

Justification for (A2). (A2) states that each additional feature group in FM_2 contributes bounded conditional MI to the label. The lower bound $\underline{\kappa}_{\text{gap}} > 0$ encodes that each new feature carries non-trivial predictive signal—this is the premise of adding features to FM_2. The upper bound $\bar{\kappa}_{\text{gap}}$ reflects diminishing returns: conditioned on prior features, each additional feature's marginal contribution is bounded (a consequence of the label having finite entropy $H(y) \le \log 2$). The historical variant (b) bounds the same quantity over the full event history, with potentially different constants since L timesteps of a feature carry more information than a single timestep.

(A3) Non-worsening cross-platform pipeline quality. FM_2 uses at least as many parameters ($p_2 \ge p_1$), the same or larger autoencoder bottleneck ($d_2 \ge d_1$), and the same or finer quantization ($b_2 \ge b_1$). We require:

$$\ell_{\text{repr},2}^{\text{cross}} + \ell_{\text{AE},2}^{\text{cross}} + \ell_{\text{Q},2}^{\text{cross}} \;\le\; \ell_{\text{repr},1}^{\text{cross}} + \ell_{\text{AE},1}^{\text{cross}} + \ell_{\text{Q},1}^{\text{cross}}, \tag{12}$$

where $\ell_{\text{repr},k}^{\text{cross}}$, $\ell_{\text{AE},k}^{\text{cross}}$, and $\ell_{\text{Q},k}^{\text{cross}}$ are the cross-platform representation, autoencoder, and quantization losses defined in equation 28–equation 30.

Justification for including $\ell_{\text{repr},k}^{\text{cross}}$ in (A3). The AE and quantization components are directly controlled by design ($d_2 \ge d_1$, $b_2 \ge b_1$). The representation loss $\ell_{\text{repr},k}^{\text{cross}}$ measures how much cross-platform information the FM's embedding $\mathbf{E}_u^{(k)}$ fails to capture from the raw extra features $\mathbf{x}_{\text{extra},k}^{(t_1:t_L)}$. A larger FM ($p_2 \ge p_1$) with lower excess risk produces richer intermediate representations at each time step, reducing the information gap between raw features and embeddings. Therefore, the total cross-platform pipeline loss (representation + AE + quantization) for FM_2 should not exceed that of FM_1 when $p_2 \ge p_1$, $d_2 \ge d_1$, and $b_2 \ge b_1$, making (A3) a natural and mild assumption.

F Proofs for Theoretical Analysis

F.1 Formal Statement and Proof of Theorem 1 (Gain Decomposition)

We first restate Theorem 1 with full detail, including the *Temporally-Privileged VM* (Temp-Priv) configuration used in the formal definitions.

Additional configuration. **Temp-Priv VM**: a hypothetical VM with access to both current features $\mathbf{x}_{\text{VM}}^{(t)}$ and the full raw VM-side history $\mathbf{H}_u = (\mathbf{x}_{\text{VM}}^{(t_1)}, \ldots, \mathbf{x}_{\text{VM}}^{(t_L)})$. This serves as a theoretical upper bound for Self-LoopFM, since raw features contain strictly more information than any compressed embedding derived from them (by the data processing inequality).

Theorem 6 (Gain decomposition; formal restatement of Theorem 1)**.** *Let $\mathcal{R}^*(\cdot) = H(y \mid \cdot)$ denote the Bayes risk under binary cross-entropy. The LoopFM information gain $\mathcal{I}_{LoopFM}(FM_k) := \mathcal{R}^{*(VM)} - \mathcal{R}^{*(LoopFM_k)}$ decomposes exactly as:*

$$\mathcal{I}_{LoopFM}(FM_k) \;=\; \mathcal{I}_{\text{temporal}} \;+\; \mathcal{I}_{\text{cross},k} \;-\; \mathcal{I}_{\text{residual},k}, \tag{13}$$

where all three terms are non-negative mutual informations:

(i) ***Temporal information:*** *$\mathcal{I}_{\text{temporal}} := I(\mathbf{H}_u; y \mid \mathbf{x}_{VM}^{(t)}) = \mathcal{R}^{*(VM)} - \mathcal{R}^{*(Temp\text{-}Priv)} \geq 0$. This is the predictive information in the user's raw feature history beyond what the current features reveal. It is a property of the data distribution alone, independent of any FM.*

(ii) ***Cross-feature information:*** *$\mathcal{I}_{\text{cross},k} := I(\mathbf{S}_u^{(k)}; y \mid \mathbf{x}_{VM}^{(t)}, \mathbf{H}_u) \geq 0$. This is the additional predictive information that the FM's extra features $\mathbf{x}_{extra}$ contribute, as encoded in the LoopFM embeddings, beyond what the raw VM history already provides. For Self-LoopFM (where FM_k = VM), this term equals zero since the VM's own embeddings cannot introduce features the VM didn't have.*

(iii) ***Compression residual:*** *$\mathcal{I}_{\text{residual},k} := I(\mathbf{H}_u; y \mid \mathbf{x}_{VM}^{(t)}, \mathbf{S}_u^{(k)}) \geq 0$. This is the predictive information in $\mathbf{H}_u$ that is* not *captured by the LoopFM sequence $\mathbf{S}_u^{(k)}$, representing the cost of the compression pipeline (FM representation → autoencoder → quantization).*

The following special cases and bounds hold:

(a) ***Self-LoopFM:*** *$\mathcal{I}_{LoopFM}(Self) = \mathcal{I}_{\text{temporal}} - \mathcal{I}_{\text{residual,self}}$. LoopFM dominates Self-LoopFM whenever $\mathcal{I}_{\text{residual},k} \leq \mathcal{I}_{\text{residual,self}}$.*

(b) ***Temporal upper bound:*** *$\mathcal{I}_{LoopFM}(Self) \leq \mathcal{I}_{\text{temporal}}$, with equality iff $\mathbf{S}_u^{self}$ is a sufficient statistic for $\mathbf{H}_u$ w.r.t. y given $\mathbf{x}_{VM}^{(t)}$.*

(c) ***Cross-feature upper bound:*** *By the data processing inequality, $\mathcal{I}_{\text{cross},k} \leq I(\mathbf{x}_{extra,k}^{(t_1:t_L)}; y \mid \mathbf{x}_{VM}^{(t)}, \mathbf{H}_u) =: \mathcal{I}_{\text{feature-raw},k}$.*

Proof. The proof applies the chain rule of mutual information to the pair $(\mathbf{S}_u^{(k)}, \mathbf{H}_u)$ in two orderings, where $\mathbf{H}_u := (\mathbf{x}_{\text{VM}}^{(t_1)}, \ldots, \mathbf{x}_{\text{VM}}^{(t_L)})$ is the raw VM feature history.

Step 1: Chain rule in two orderings. Applying the chain rule to the joint mutual information $I(\mathbf{S}_u^{(k)}, \mathbf{H}_u; y \mid \mathbf{x}_{\text{VM}}^{(t)})$:

Ordering 1 ($\mathbf{H}_u$ first):

$$I(\mathbf{S}_u^{(k)}, \mathbf{H}_u;\; y \mid \mathbf{x}_{\text{VM}}^{(t)}) = \underbrace{I(\mathbf{H}_u;\; y \mid \mathbf{x}_{\text{VM}}^{(t)})}_{\mathcal{I}_{\text{temporal}}} + \underbrace{I(\mathbf{S}_u^{(k)};\; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)}_{\mathcal{I}_{\text{cross},k}}. \tag{14}$$

Conditioning on $\mathbf{H}_u$ first captures the temporal information; any remaining information in $\mathbf{S}_u^{(k)}$ about y given $(\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)$ is the cross-platform contribution.

Ordering 2 ($\mathbf{S}_u^{(k)}$ first):

$$I(\mathbf{S}_u^{(k)}, \mathbf{H}_u;\; y \mid \mathbf{x}_{\text{VM}}^{(t)}) = \underbrace{I(\mathbf{S}_u^{(k)};\; y \mid \mathbf{x}_{\text{VM}}^{(t)})}_{\mathcal{I}_{\text{LoopFM}}(\text{FM}_k)} + \underbrace{I(\mathbf{H}_u;\; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{S}_u^{(k)})}_{\mathcal{I}_{\text{residual},k}}. \tag{15}$$

Step 2: Equate and rearrange. Since both orderings compute the same joint mutual information, equating equation 14 and equation 15:

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_k) + \mathcal{I}_{\text{residual},k} = \mathcal{I}_{\text{temporal}} + \mathcal{I}_{\text{cross},k}. \tag{16}$$

Rearranging:

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_k) = \mathcal{I}_{\text{temporal}} + \mathcal{I}_{\text{cross},k} - \mathcal{I}_{\text{residual},k}. \tag{17}$$

Non-negativity of all three terms follows from the non-negativity of mutual information. The identification $\mathcal{I}_{\text{temporal}} = \mathcal{R}^{*(\text{VM})} - \mathcal{R}^{*(\text{Temp-Priv})}$ follows from the conditional MI identity $I(X;Y \mid Z) = H(Y \mid Z) - H(Y \mid X, Z)$ and the BCE Bayes-risk identity $\mathcal{R}^*(\cdot) = H(y \mid \cdot)$.

Step 3: Self-LoopFM special case (part (i)). When FM_k = VM, the embedding $\mathbf{z}_k^{(t_j)} = \psi(\mathbf{x}_{\text{VM}}^{(t_j)})$ is a deterministic function of $\mathbf{x}_{\text{VM}}^{(t_j)}$. Therefore $\mathbf{S}_u^{\text{self}} = [\psi(\mathbf{x}_{\text{VM}}^{(t_1)}), \ldots, \psi(\mathbf{x}_{\text{VM}}^{(t_L)})]$ is a function of $\mathbf{H}_u$. By the data processing inequality, $I(\mathbf{S}_u^{\text{self}}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u) = 0$ since $\mathbf{S}_u^{\text{self}}$ is $\sigma(\mathbf{H}_u)$-measurable. Hence $\mathcal{I}_{\text{cross,self}} = 0$, giving $\mathcal{I}_{\text{LoopFM}}(\text{Self}) = \mathcal{I}_{\text{temporal}} - \mathcal{I}_{\text{residual,self}}$. Note that LoopFM similarly accesses temporal information through its FM-derived sequence $\mathbf{S}_u^{(k)}$; the only difference is the source of event embeddings (FM vs. VM). If the retention-dominance condition $\mathcal{I}_{\text{residual},k} \leq \mathcal{I}_{\text{residual,self}}$ holds, LoopFM is no worse than Self-LoopFM.

Step 4: Temporal upper bound (part (ii)). Since $\mathcal{I}_{\text{residual,self}} \geq 0$, we have $\mathcal{I}_{\text{LoopFM}}(\text{Self}) \leq \mathcal{I}_{\text{temporal}}$. Equality holds when $\mathcal{I}_{\text{residual,self}} = I(\mathbf{H}_u; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{S}_u^{\text{self}}) = 0$, i.e., when $\mathbf{S}_u^{\text{self}}$ is a sufficient statistic for $\mathbf{H}_u$ with respect to y given $\mathbf{x}_{\text{VM}}^{(t)}$.

Step 5: Cross-platform upper bound (part (iii)). The FM_k embedding at time t_j is computed as $\mathbf{z}_{k,t_j} = f_{\text{enc}}(f_{\text{FM}_k}(\mathbf{x}_{\text{FM}_k}^{(t_j)}))$, where $\mathbf{x}_{\text{FM}_k}^{(t_j)} = (\mathbf{x}_{\text{VM}}^{(t_j)}, \mathbf{x}_{\text{extra},k}^{(t_j)})$. Since both f_{FM_k} and f_{enc} are deterministic functions, $\mathbf{S}_u^{(k)}$ is a deterministic function of $(\mathbf{H}_u, \mathbf{x}_{\text{extra},k}^{(t_1:t_L)})$. Conditioning on $\mathbf{H}_u$ makes $\mathbf{S}_u^{(k)}$ a function of $\mathbf{x}_{\text{extra},k}^{(t_1:t_L)}$ alone. By the data processing inequality applied to the Markov chain $\mathbf{x}_{\text{extra},k}^{(t_1:t_L)} \to \mathbf{S}_u^{(k)} \to y$ (conditioned on $(\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)$):

$$\mathcal{I}_{\text{cross},k} = I(\mathbf{S}_u^{(k)}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u) \leq I(\mathbf{x}_{\text{extra},k}^{(t_1:t_L)}; y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u) = \mathcal{I}_{\text{feature-raw},k}. \tag{18}$$

□

F.2 Pipeline Decomposition of Compression Loss

Proposition 7 (Pipeline decomposition)**.** *The residual $\mathcal{I}_{\text{residual},k}$ decomposes along the LoopFM pipeline: raw FM embeddings $\mathbf{E}_u^{(k)} \in \mathbb{R}^{L \times D}$ (L timesteps, D-dim per step) → autoencoder-compressed $\mathbf{Z}_u^{(k)} \in \mathbb{R}^{L \times d}$ ($d \ll D$) → quantized $\mathbf{S}_u^{(k)}$ (b_k bits). Since $\sigma(\mathbf{S}_u^{(k)}) \subseteq \sigma(\mathbf{Z}_u^{(k)}) \subseteq \sigma(\mathbf{E}_u^{(k)})$:*

$$\mathcal{I}_{\text{residual},k} \leq \underbrace{\ell_{\text{repr},k}(p_k)}_{\substack{\textit{FM repr.}\\ \textit{residual}}} + \underbrace{\ell_{\text{AE},k}(d)}_{\substack{\textit{AE compr.}\\ \textit{loss}}} + \underbrace{\ell_{\text{Q},k}(b_k)}_{\substack{\textit{quantization}\\ \textit{loss}}}, \tag{19}$$

where:

$$\ell_{\text{repr},k}(p_k) := I\Big(\mathbf{H}_u;\ y \;\Big|\; \mathbf{x}_{VM}^{(t)},\ \mathbf{E}_u^{(k)}\Big) \geq 0, \tag{20}$$

$$\ell_{\text{AE},k}(d) := I\Big(\mathbf{H}_u;\ \mathbf{E}_u^{(k)} \;\Big|\; \mathbf{x}_{VM}^{(t)}\Big) - I\Big(\mathbf{H}_u;\ \mathbf{Z}_u^{(k)} \;\Big|\; \mathbf{x}_{VM}^{(t)}\Big) \geq 0, \tag{21}$$

$$\ell_{\text{Q},k}(b_k) := I\Big(\mathbf{H}_u;\ \mathbf{Z}_u^{(k)} \;\Big|\; \mathbf{x}_{VM}^{(t)}\Big) - I\Big(\mathbf{H}_u;\ \mathbf{S}_u^{(k)} \;\Big|\; \mathbf{x}_{VM}^{(t)}\Big) \geq 0. \tag{22}$$

The three terms correspond to successive stages of the pipeline: $\ell_{\text{repr},k}$ is the representation residual*—the predictive information in user history $\mathbf{H}_u$ about y that the FM's raw embeddings $\mathbf{E}_u^{(k)}$ fail to retain. A larger FM (more parameters p_k) produces richer intermediate representations that retain more of this information, driving $\ell_{\text{repr},k} \to 0$. $\ell_{\text{AE},k}$ is the* autoencoder compression loss*—the information about $\mathbf{H}_u$ lost when compressing $\mathbf{E}_u^{(k)} \in \mathbb{R}^{L \times D}$ to $\mathbf{Z}_u^{(k)} \in \mathbb{R}^{L \times d}$; decreases with larger bottleneck dimension d. $\ell_{\text{Q},k}$ is the* quantization loss*—the information lost when discretizing $\mathbf{Z}_u^{(k)}$ to b_k bits; decreases with finer quantization.*

Proof. Let

$$X := \mathbf{x}_{\text{VM}}^{(t)}, \quad H := \mathbf{H}_u, \quad E_p := \mathbf{E}_u^{(k)}(p_k), \quad Z_{p,d} := \mathbf{Z}_u^{(k)}(p_k, d), \quad S_{p,d,b} := \mathbf{S}_u^{(k)}(p_k, d, b_k).$$

Step 1: Decompose the residual into representation and downstream losses. Starting from

$$\mathcal{I}_{\text{residual},k} = I(H; y \mid X, S_{p,d,b}),$$

add and subtract $I(H; y \mid X, E_p)$:

$$\mathcal{I}_{\text{residual},k} = \underbrace{I(H; y \mid X, E_p)}_{\ell_{\text{repr},k}(p_k)} + \big[I(H; y \mid X, S_{p,d,b}) - I(H; y \mid X, E_p)\big]. \tag{23}$$

Using the chain-rule identity

$$I(A; B \mid C) - I(A; B \mid C, D) = I(A; D \mid C) - I(A; D \mid B, C),$$

with $(A, B, C, D) = (H, y, (X, S_{p,d,b}), E_p)$, we get

$$\begin{aligned} I(H; y \mid X, S_{p,d,b}) - I(H; y \mid X, E_p) &= I(H; E_p \mid X, S_{p,d,b}) - I(H; E_p \mid X, S_{p,d,b}, y) \\ &\le I(H; E_p \mid X, S_{p,d,b}). \end{aligned} \tag{24}$$

Step 2: Telescope through AE and quantization. Since $\sigma(S_{p,d,b}) \subseteq \sigma(Z_{p,d}) \subseteq \sigma(E_p)$, the chain rule gives

$$\begin{aligned} I(H; E_p \mid X, S_{p,d,b}) &= I(H; E_p \mid X) - I(H; S_{p,d,b} \mid X) \\ &= \big[I(H; E_p \mid X) - I(H; Z_{p,d} \mid X)\big] \\ &\quad + \big[I(H; Z_{p,d} \mid X) - I(H; S_{p,d,b} \mid X)\big] \\ &= \ell_{\text{AE},k}(d) + \ell_{\text{Q},k}(b_k). \end{aligned} \tag{25}$$

Combining equation 23, equation 24, and equation 25 yields

$$\mathcal{I}_{\text{residual},k} \le \ell_{\text{repr},k}(p_k) + \ell_{\text{AE},k}(d) + \ell_{\text{Q},k}(b_k),$$

which is Eq. equation 19.

Step 3: Non-negativity of all three losses.

- $\ell_{\text{repr},k}(p_k) = I(H; y \mid X, E_p) \ge 0$ by non-negativity of conditional MI.
- $\ell_{\text{AE},k}(d) = I(H; E_p \mid X) - I(H; Z_{p,d} \mid X) \ge 0$ by DPI under $H \to E_p \to Z_{p,d}$ (given X).
- $\ell_{\text{Q},k}(b_k) = I(H; Z_{p,d} \mid X) - I(H; S_{p,d,b} \mid X) \ge 0$ by DPI under $H \to Z_{p,d} \to S_{p,d,b}$ (given X).

Step 4: Connection between $\ell_{\text{AE},k}(d)$ and AE MSE. Let $\hat{E}_{p,d} := f_{\text{dec}}(Z_{p,d})$ and

$$D_{\text{AE},k}(d) := \frac{1}{LD}\,\mathbb{E}\Big[\|E_p - \hat{E}_{p,d}\|_2^2\Big].$$

Because $\hat{E}_{p,d}$ is a deterministic function of $Z_{p,d}$, DPI gives

$$\ell_{\text{AE},k}(d) = I(H; E_p \mid X) - I(H; Z_{p,d} \mid X) \le I(H; E_p \mid X) - I(H; \hat{E}_{p,d} \mid X). \tag{26}$$

Now impose the linear-Gaussian approximation conditioned on X:

$$E_p = A_k H + \varepsilon_{\text{repr}}, \qquad \hat{E}_{p,d} = E_p + \varepsilon_{\text{AE}},$$

with $\varepsilon_{\text{repr}} \sim \mathcal{N}(0, \Sigma_{\text{repr},k})$, $\varepsilon_{\text{AE}} \sim \mathcal{N}(0, q_k(d)I)$, independent of each other and of H, and $q_k(d) = D_{\text{AE},k}(d)$. If $\Sigma_{\text{signal},k} := A_k \Sigma_{H|X} A_k^\top$ and $\Sigma_{\text{repr},k}$ are simultaneously diagonalizable with eigenvalues $(\lambda_{k,j}, \rho_{k,j})_{j=1}^{r_k}$, then

$$\begin{aligned} I(H; E_p \mid X) - I(H; \hat{E}_{p,d} \mid X) &= \frac{1}{2}\sum_{j=1}^{r_k} \log \frac{1 + \lambda_{k,j}/\rho_{k,j}}{1 + \lambda_{k,j}/(\rho_{k,j} + q_k(d))} \\ &\le \frac{1}{2}\sum_{j=1}^{r_k} \log\left(1 + \frac{q_k(d)}{\rho_{k,j}}\right) \\ &\le \frac{r_k}{2}\log\left(1 + \frac{q_k(d)}{\sigma^2_{\text{repr},k,\min}}\right), \end{aligned} \tag{27}$$

where $\sigma^2_{\text{repr},k,\min} := \min_j \rho_{k,j}$. Combining equation 26 and equation 27 with $q_k(d) = D_{\text{AE},k}(d)$ gives the claimed AE-MSE control for $\ell_{\text{AE},k}(d)$. □

Cross-platform pipeline losses. Conditioned on $(\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)$, the Markov chain $\mathbf{x}_{\text{extra},k}^{(t_1:t_L)} \to \mathbf{E}_u^{(k)} \to \mathbf{Z}_u^{(k)} \to \mathbf{S}_u^{(k)}$ yields cross-platform pipeline losses:

$$\ell_{\text{repr},k}^{\text{cross}} := I\Big(\mathbf{x}_{\text{extra},k}^{(t_1:t_L)};\ y \;\Big|\; \mathbf{x}_{\text{VM}}^{(t)},\ \mathbf{H}_u\Big) - I\Big(\mathbf{E}_u^{(k)};\ y \;\Big|\; \mathbf{x}_{\text{VM}}^{(t)},\ \mathbf{H}_u\Big) \geq 0, \tag{28}$$

$$\ell_{\text{AE},k}^{\text{cross}} := I\Big(\mathbf{E}_u^{(k)};\ y \;\Big|\; \mathbf{x}_{\text{VM}}^{(t)},\ \mathbf{H}_u\Big) - I\Big(\mathbf{Z}_u^{(k)};\ y \;\Big|\; \mathbf{x}_{\text{VM}}^{(t)},\ \mathbf{H}_u\Big) \geq 0, \tag{29}$$

$$\ell_{\text{Q},k}^{\text{cross}} := I\Big(\mathbf{Z}_u^{(k)};\ y \;\Big|\; \mathbf{x}_{\text{VM}}^{(t)},\ \mathbf{H}_u\Big) - I\Big(\mathbf{S}_u^{(k)};\ y \;\Big|\; \mathbf{x}_{\text{VM}}^{(t)},\ \mathbf{H}_u\Big) \geq 0. \tag{30}$$

Unlike the temporal losses (which measure information about $\mathbf{H}_u$), the cross-platform losses measure information about y from the FM-VM feature gap, conditioned on $(\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)$: $\ell_{\text{repr},k}^{\text{cross}}$ is the predictive signal from the raw extra features $\mathbf{x}_{\text{extra},k}^{(t_1:t_L)}$ that the FM's embeddings $\mathbf{E}_u^{(k)}$ fail to retain—the FM-VM feature gap information lost during the FM's forward pass; $\ell_{\text{AE},k}^{\text{cross}}$ is the feature-gap information lost during autoencoder compression ($\mathbf{E} \to \mathbf{Z}$); $\ell_{\text{Q},k}^{\text{cross}}$ is the feature-gap information lost during quantization ($\mathbf{Z} \to \mathbf{S}$). Their sum determines η_k: the fraction of raw cross-platform information destroyed by the pipeline.

Lemma 8 (Cross-platform retention)**.** $\mathcal{I}_{\text{feature-raw},k} - \mathcal{I}_{\text{cross},k} = \ell_{\text{repr},k}^{\text{cross}} + \ell_{\text{AE},k}^{\text{cross}} + \ell_{\text{Q},k}^{\text{cross}}$. *Equivalently,* $\mathcal{I}_{\text{cross},k} = (1 - \eta_k) \cdot \mathcal{I}_{\text{feature-raw},k}$ *where* $\eta_k := (\ell_{\text{repr},k}^{\text{cross}} + \ell_{\text{AE},k}^{\text{cross}} + \ell_{\text{Q},k}^{\text{cross}})/\mathcal{I}_{\text{feature-raw},k}$.

Proof. Conditioned on $(\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)$, the embedding $\mathbf{S}_u^{(k)}$ is a deterministic function of $\mathbf{x}_{\text{extra},k}^{(t_1:t_L)}$ alone (since the FM and autoencoder are fixed functions and $\mathbf{H}_u$ supplies the VM features at each time step). This gives the Markov chain:

$$\mathbf{x}_{\text{extra},k}^{(t_1:t_L)} \ \to\ \mathbf{E}_u^{(k)} \ \to\ \mathbf{Z}_u^{(k)} \ \to\ \mathbf{S}_u^{(k)} \qquad (\text{given } (\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)). \tag{31}$$

Telescope the mutual information with y:

$$\begin{aligned}\mathcal{I}_{\text{feature-raw},k} - \mathcal{I}_{\text{cross},k} &= I(\mathbf{x}_{\text{extra},k}^{(t_1:t_L)}; y \mid \mathbf{x}_{\text{VM}}, \mathbf{H}_u) - I(\mathbf{S}_u^{(k)}; y \mid \mathbf{x}_{\text{VM}}, \mathbf{H}_u) \\ &= \underbrace{\big[I(\mathbf{x}_{\text{extra}}; y \mid \cdot) - I(\mathbf{E}; y \mid \cdot)\big]}_{\ell_{\text{repr},k}^{\text{cross}}} + \underbrace{\big[I(\mathbf{E}; y \mid \cdot) - I(\mathbf{Z}; y \mid \cdot)\big]}_{\ell_{\text{AE},k}^{\text{cross}}} + \underbrace{\big[I(\mathbf{Z}; y \mid \cdot) - I(\mathbf{S}; y \mid \cdot)\big]}_{\ell_{\text{Q},k}^{\text{cross}}},\end{aligned} \tag{32}$$

where $(\cdot) = (\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{H}_u)$ throughout. Each bracket is non-negative by the data processing inequality: since $\sigma(\mathbf{S}) \subseteq \sigma(\mathbf{Z}) \subseteq \sigma(\mathbf{E}) \subseteq \sigma(\mathbf{x}_{\text{extra}})$ (given $(\mathbf{x}_{\text{VM}}, \mathbf{H}_u)$), we have $I(W; y \mid C) \geq I(f(W); y \mid C)$ for any deterministic f and conditioning C.

The cross-platform pipeline identity is exactly equation 32. Dividing by $\mathcal{I}_{\text{feature-raw},k}$ (assumed > 0) gives $\eta_k = (\ell_{\text{repr},k}^{\text{cross}} + \ell_{\text{AE},k}^{\text{cross}} + \ell_{\text{Q},k}^{\text{cross}})/\mathcal{I}_{\text{feature-raw},k}$. □

Lemma 9 (Temporal retention)**.** $0 \leq \mathcal{I}_{\text{residual},k} \leq \tau_k \cdot \mathcal{I}_{\text{temporal}}$ *where* $\tau_k := (\ell_{\text{repr},k} + \ell_{\text{AE},k} + \ell_{\text{Q},k})/\mathcal{I}_{\text{temporal}}$.

Proof. This is a restatement of Proposition 7 with τ_k defined as the normalized bound. From equation 19:

$$\mathcal{I}_{\text{residual},k} \ \leq\ \ell_{\text{repr},k} + \ell_{\text{AE},k} + \ell_{\text{Q},k}. \tag{33}$$

Dividing by $\mathcal{I}_{\text{temporal}}$ (assumed > 0) gives $\tau_k := (\ell_{\text{repr},k} + \ell_{\text{AE},k} + \ell_{\text{Q},k})/\mathcal{I}_{\text{temporal}}$, so $\mathcal{I}_{\text{residual},k} \leq \tau_k \cdot \mathcal{I}_{\text{temporal}}$. □

Corollary 10 (LoopFM gain sandwich)**.** $(1 - \tau_k)\,\mathcal{I}_{\text{temporal}} + (1 - \eta_k)\,\mathcal{I}_{\text{feature-raw},k} \leq \mathcal{I}_{\textit{LoopFM}}(\textit{FM}_k) \leq \mathcal{I}_{\text{temporal}} + (1 - \eta_k)\,\mathcal{I}_{\text{feature-raw},k}$.

Proof. From Theorem 1: $\mathcal{I}_{\text{LoopFM}}(\text{FM}_k) = \mathcal{I}_{\text{temporal}} + \mathcal{I}_{\text{cross},k} - \mathcal{I}_{\text{residual},k}$.

By Lemma 8: $\mathcal{I}_{\text{cross},k} = (1 - \eta_k)\,\mathcal{I}_{\text{feature-raw},k} \geq (1 - \eta_k)\,\mathcal{I}_{\text{feature-raw},k}$.

By Lemma 9: $\mathcal{I}_{\text{residual},k} \leq \tau_k\,\mathcal{I}_{\text{temporal}}$, so $-\mathcal{I}_{\text{residual},k} \geq -\tau_k\,\mathcal{I}_{\text{temporal}}$.

Combining:

$$\begin{aligned}\mathcal{I}_{\text{LoopFM}}(\text{FM}_k) &\geq \mathcal{I}_{\text{temporal}} - \tau_k\,\mathcal{I}_{\text{temporal}} + (1 - \eta_k)\,\mathcal{I}_{\text{feature-raw},k} \\ &= (1 - \tau_k)\,\mathcal{I}_{\text{temporal}} + (1 - \eta_k)\,\mathcal{I}_{\text{feature-raw},k}.\end{aligned}$$ □

Lemma 11 (Non-worsening pipeline quality)**.** *Under* (A2)–(A3)*,* $\eta_2 \le \eta_1$.

Proof. By Lemma 8, $\eta_k = (\ell^{\text{cross}}_{\text{repr},k} + \ell^{\text{cross}}_{\text{AE},k} + \ell^{\text{cross}}_{\text{Q},k})/\mathcal{I}_{\text{feature-raw},k}$. By Assumption (A3), Eq. equation 12, the numerator for k=2 is at most the numerator for k=1. By Assumption (A2) and Lemma 12, $\mathcal{I}_{\text{feature-raw},2} \ge \mathcal{I}_{\text{feature-raw},1} > 0$. Therefore

$$\eta_2 = \frac{\ell^{\text{cross}}_{\text{repr},2} + \ell^{\text{cross}}_{\text{AE},2} + \ell^{\text{cross}}_{\text{Q},2}}{\mathcal{I}_{\text{feature-raw},2}} \le \frac{\ell^{\text{cross}}_{\text{repr},1} + \ell^{\text{cross}}_{\text{AE},1} + \ell^{\text{cross}}_{\text{Q},1}}{\mathcal{I}_{\text{feature-raw},2}} \le \frac{\ell^{\text{cross}}_{\text{repr},1} + \ell^{\text{cross}}_{\text{AE},1} + \ell^{\text{cross}}_{\text{Q},1}}{\mathcal{I}_{\text{feature-raw},1}} = \eta_1. \quad \square$$

F.3 Supporting Lemma and Proof for the Transfer-Ratio Theorem

Proof of the achieved-risk decomposition. Let $X := \mathbf{x}^{(t)}_{\text{VM}}$ and $\mathbf{S}_k := \mathbf{S}^{(k)}_u$. Under BCE, the Bayes-optimal risk equals conditional entropy, so

$$\mathcal{R}^{*(\text{LoopFM}_k)} = H(y \mid X, \mathbf{S}_k) = H(y \mid X) - I(\mathbf{S}_k; y \mid X) = \mathcal{R}^{*(\text{VM})} - \mathcal{I}_{\text{LoopFM}}(\text{FM}_k).$$

By definition, $R_{\text{ach}}(\text{LoopFM}_k) = \mathcal{R}^{*(\text{LoopFM}_k)} + \epsilon_{\text{est},k}$ with $\epsilon_{\text{est},k} := \epsilon_{\text{opt},k} + \epsilon_{\text{gen},k} \ge 0$. Subtracting for $k = 1$ and $k = 2$:

$$\Delta_{\text{student}} = \Big(\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) - \mathcal{I}_{\text{LoopFM}}(\text{FM}_1)\Big) + \Delta_{\text{est}}.$$

Lemma 12 (Feature-gap bounds)**.** *Assume* (A2)*. The Bayes-risk gap* $\Delta_{feat} := \mathcal{R}^{*(FM_1)} - \mathcal{R}^{*(FM_2)}$ *and the raw-channel gap* $\Delta_{\text{feat-raw}} := \mathcal{I}_{\text{feature-raw},2} - \mathcal{I}_{\text{feature-raw},1}$ *both satisfy*

$$\underline{\kappa}_{\text{gap}}\,(m_2 - m_1) \le \Delta_{feat} \le \bar{\kappa}_{\text{gap}}\,(m_2 - m_1), \qquad \underline{\kappa}^{\text{hist}}_{\text{gap}}\,(m_2 - m_1) \le \Delta_{\text{feat-raw}} \le \bar{\kappa}^{\text{hist}}_{\text{gap}}\,(m_2 - m_1). \tag{34}$$

Note that Δ_{feat} *involves current-step features (via the FM's per-sample Bayes risk), while* $\Delta_{\text{feat-raw}}$ *involves historical feature sequences (via the Theorem 1 definition* $\mathcal{I}_{\text{feature-raw},k} = I(\mathbf{x}^{(t_1:t_L)}_{extra,k}; y \mid \mathbf{x}^{(t)}_{VM}, \mathbf{H}_u)$*); these are distinct quantities that admit the same chain-rule structure.*

Proof. Both quantities decompose via the MI chain rule over feature groups $j \in \{m_1{+}1, \ldots, m_2\}$.

For the Bayes-risk gap (current-step features):

$$\Delta_{\text{feat}} = H(y \mid \mathbf{x}_{\text{VM}}, \mathbf{x}_{\text{extra},1}) - H(y \mid \mathbf{x}_{\text{VM}}, \mathbf{x}_{\text{extra},1}, \mathbf{x}_{\text{extra},2}) = \sum_{j=m_1+1}^{m_2} I(u_j; y \mid \mathbf{x}_{\text{VM}}, \mathbf{x}_{\text{extra},1}, u_{m_1+1:j-1})\,.$$

For the raw-channel gap (historical sequences, using the Theorem 1 definition):

$$\Delta_{\text{feat-raw}} = I\Big(\mathbf{x}^{(t_1:t_L)}_{\text{extra},2}; y \mid \mathbf{x}^{(t)}_{\text{VM}}, \mathbf{H}_u, \mathbf{x}^{(t_1:t_L)}_{\text{extra},1}\Big) = \sum_{j=m_1+1}^{m_2} I\Big(u^{(t_1:t_L)}_j; y \mid \mathbf{x}^{(t)}_{\text{VM}}, \mathbf{H}_u, \mathbf{x}^{(t_1:t_L)}_{\text{extra},1}, u^{(t_1:t_L)}_{m_1+1:j-1}\Big)\,.$$

Applying (A2) to each per-feature-group term in both expansions and summing gives equation 34. In particular, $\mathcal{I}_{\text{feature-raw},2} \ge \mathcal{I}_{\text{feature-raw},1} + \underline{\kappa}^{\text{hist}}_{\text{gap}}(m_2 - m_1) > \mathcal{I}_{\text{feature-raw},1}$. $\square$

F.4 Proof of Theorem 2 (Transfer-Ratio Bound)

Proof. Let

$$\Delta_{\text{LoopFM}} := R_{\text{ach}}(\text{LoopFM}_1) - R_{\text{ach}}(\text{LoopFM}_2), \qquad \Delta_{\text{teacher}} := R_{\text{ach}}(\text{FM}_1) - R_{\text{ach}}(\text{FM}_2).$$

By Eq. equation 5,

$$\text{TR}_{\text{LoopFM}} = \frac{\Delta_{\text{LoopFM}}}{\Delta_{\text{teacher}}}.$$

By the achieved-risk decomposition,

$$\Delta_{\text{LoopFM}} = \Big(\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) - \mathcal{I}_{\text{LoopFM}}(\text{FM}_1)\Big) + \Delta_{\text{est}}.$$

Under the well-trained assumption $\epsilon_{\text{est},k} \to 0$, $\Delta_{\text{est}} \to 0$, so

$$\Delta_{\text{LoopFM}} = \mathcal{I}_{\text{LoopFM}}(\text{FM}_2) - \mathcal{I}_{\text{LoopFM}}(\text{FM}_1).$$

By Eq. equation 4,

$$\Delta_{\text{teacher}} = \Delta_{\text{feat}} + \Delta_{\text{param}} = \mathcal{R}^{*(\text{FM}_1)} - \mathcal{R}^{*(\text{FM}_2)} + \epsilon_{\text{over}}(p_1, m_1) - \epsilon_{\text{over}}(p_2, m_2).$$

Substituting into Eq. equation 5 gives the transfer-ratio expression.

Assume now $\Delta_{\text{teacher}} > 0$.

(0) Initial launch. When the VM has no prior LoopFM, $R_{\text{ach}}(\text{LoopFM}_1) = R_{\text{ach}}(\text{VM})$ and $\mathcal{I}_{\text{LoopFM}}(\text{FM}_1) = 0$ (the baseline VM receives no embedding sequence). The numerator becomes $\mathcal{I}_{\text{LoopFM}}(\text{FM}_2)$. This quantity is non-negative by conditioning reduces entropy:

$$\mathcal{R}^{*(\text{LoopFM}_2)} = H(y \mid \mathbf{x}_{\text{VM}}^{(t)}, \mathbf{S}_u^{(2)}) \;\le\; H(y \mid \mathbf{x}_{\text{VM}}^{(t)}) = \mathcal{R}^{*(\text{VM})},$$

since the LoopFM$_2$ model observes $(\mathbf{x}_{\text{VM}}^{(t)}, \mathbf{S}_u^{(2)})$, a strict superset of the baseline VM's input $\mathbf{x}_{\text{VM}}^{(t)}$. Therefore $\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) = \mathcal{R}^{*(\text{VM})} - \mathcal{R}^{*(\text{LoopFM}_2)} \ge 0$. Since $\Delta_{\text{teacher}} > 0$ by assumption, $\text{TR}_{\text{LoopFM}} \ge 0$.

For the tighter bound, substitute the gain sandwich lower bound (Corollary 10):

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) \;\ge\; (1-\tau_2)\,\mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2}.$$

Dividing by $\Delta_{\text{teacher}} > 0$ gives equation 7. Note that $\text{TR}_{\text{LoopFM}} \ge 0$ follows independently from the Bayes-risk argument above, not from the sign of this bound.

(1) Negative transfer (without A3). If $\Delta_{\text{LoopFM}} < 0$, then from $\text{TR}_{\text{LoopFM}} = \Delta_{\text{LoopFM}}/\Delta_{\text{teacher}}$ we immediately get $\text{TR}_{\text{LoopFM}} < 0$.

For a sufficient condition, Corollary 10 gives

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) \le \mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2},$$

and

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_1) \ge (1-\tau_1)\,\mathcal{I}_{\text{temporal}} + (1-\eta_1)\,\mathcal{I}_{\text{feature-raw},1}.$$

Therefore,

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) - \mathcal{I}_{\text{LoopFM}}(\text{FM}_1) \le \tau_1\,\mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2} - (1-\eta_1)\,\mathcal{I}_{\text{feature-raw},1}. \tag{35}$$

So if the right-hand side is negative, then $\Delta_{\text{LoopFM}} < 0$ and $\text{TR}_{\text{LoopFM}} < 0$.

Why this requires violating (A3). Under (A3), Lemma 11 gives $\eta_2 \le \eta_1$, so $(1-\eta_2) \ge (1-\eta_1)$. Combined with (A2) ($\mathcal{I}_{\text{feature-raw},2} \ge \mathcal{I}_{\text{feature-raw},1}$), the right-hand side of equation 35 satisfies

$$\tau_1\,\mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2} - (1-\eta_1)\,\mathcal{I}_{\text{feature-raw},1} \;\ge\; \tau_1\,\mathcal{I}_{\text{temporal}} \;\ge\; 0,$$

so the negative-transfer condition is never met. Therefore, negative transfer requires $\eta_2 > \eta_1$ (i.e., violating A3).

(2) Positive transfer (with A3). By Assumption (A3) and Lemma 11, $\eta_2 \le \eta_1$. For the numerator lower bound, Corollary 10 implies

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) \ge (1-\tau_2)\,\mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2},$$

and

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_1) \le \mathcal{I}_{\text{temporal}} + (1-\eta_1)\,\mathcal{I}_{\text{feature-raw},1}.$$

Therefore

$$\mathcal{I}_{\text{LoopFM}}(\text{FM}_2) - \mathcal{I}_{\text{LoopFM}}(\text{FM}_1) \ge -\tau_2\,\mathcal{I}_{\text{temporal}} + (1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2} - (1-\eta_1)\,\mathcal{I}_{\text{feature-raw},1}. \tag{36}$$

Under $\eta_2 \le \eta_1$,

$$\begin{aligned}
&(1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2} - (1-\eta_1)\,\mathcal{I}_{\text{feature-raw},1} \\
&\qquad = (1-\eta_1)\big(\mathcal{I}_{\text{feature-raw},2} - \mathcal{I}_{\text{feature-raw},1}\big) + (\eta_1-\eta_2)\,\mathcal{I}_{\text{feature-raw},2} \\
&\qquad \ge (1-\eta_1)\big(\mathcal{I}_{\text{feature-raw},2} - \mathcal{I}_{\text{feature-raw},1}\big) \\
&\qquad = (1-\eta_1)\,\Delta_{\text{feat-raw}}.
\end{aligned} \tag{37}$$

By Lemma 12,

$$\Delta_{\text{feat-raw}} = \mathcal{I}_{\text{feature-raw},2} - \mathcal{I}_{\text{feature-raw},1} \geq \underline{\kappa}_{\text{gap}}^{\text{hist}}(m_2 - m_1).$$

Combining with equation 36–equation 37,

$$\Delta_{\text{LoopFM}} \geq -\tau_2 \mathcal{I}_{\text{temporal}} + (1-\eta_1)\,\underline{\kappa}_{\text{gap}}^{\text{hist}}(m_2 - m_1).$$

For the denominator upper bound, by Lemma 12:

$$\Delta_{\text{feat}} \leq \bar{\kappa}_{\text{gap}}(m_2 - m_1).$$

By the two-sided Assumption (A1), Eq. equation 9:

$$\underline{C}_{\text{over}}\,\sigma^2\,\xi_k \;\leq\; \epsilon_{\text{over}}(p_k, m_k) \;\leq\; \bar{C}_{\text{over}}\,\sigma^2\,\xi_k, \quad k \in \{1,2\}.$$

To rigorously upper-bound the difference $\Delta_{\text{param}} = \epsilon_{\text{over}}(p_1, m_1) - \epsilon_{\text{over}}(p_2, m_2)$, we apply the *upper* envelope to $\epsilon_{\text{over}}(p_1, m_1)$ and the *lower* envelope to $\epsilon_{\text{over}}(p_2, m_2)$:

$$\Delta_{\text{param}} \;\leq\; \bar{C}_{\text{over}}\,\sigma^2\,\xi_1 - \underline{C}_{\text{over}}\,\sigma^2\,\xi_2.$$

With $\bar{\kappa}_{\text{over}} := \bar{C}_{\text{over}}\,\sigma^2$ and $\underline{\kappa}_{\text{over}} := \underline{C}_{\text{over}}\,\sigma^2$:

$$\Delta_{\text{param}} \;\leq\; \bar{\kappa}_{\text{over}}\,\xi_1 - \underline{\kappa}_{\text{over}}\,\xi_2.$$

Hence

$$\Delta_{\text{feat}} + \Delta_{\text{param}} \leq \bar{\kappa}_{\text{gap}}(m_2 - m_1) + \bar{\kappa}_{\text{over}}\,\xi_1 - \underline{\kappa}_{\text{over}}\,\xi_2.$$

If additionally

$$-\tau_2 \mathcal{I}_{\text{temporal}} + (1-\eta_1)\underline{\kappa}_{\text{gap}}^{\text{hist}}(m_2 - m_1) \geq 0,$$

then

$$\text{TR}_{\text{LoopFM}} = \frac{\Delta_{\text{LoopFM}}}{\Delta_{\text{feat}} + \Delta_{\text{param}}} \geq \frac{-\tau_2 \mathcal{I}_{\text{temporal}} + (1-\eta_1)\underline{\kappa}_{\text{gap}}^{\text{hist}}(m_2 - m_1)}{\bar{\kappa}_{\text{gap}}(m_2 - m_1) + \bar{\kappa}_{\text{over}}\,\xi_1 - \underline{\kappa}_{\text{over}}\,\xi_2},$$

which is Eq. equation 8. □

F.5 Proof of Corollary 4 (Monotonicity in Feature Gap)

Proof. Write $\delta := m_2 - m_1$ and define

$$N(\delta) := a\delta + b, \qquad D(\delta) := \bar{\kappa}_{\text{gap}}\,\delta + \underbrace{\bar{\kappa}_{\text{over}}\,\xi_1 - \underline{\kappa}_{\text{over}}\,\xi_2(\delta)}_{=:\,\epsilon(\delta)},$$

where $a := (1-\eta_1)\underline{\kappa}_{\text{gap}}^{\text{hist}} > 0$, $b := -\tau_2 \mathcal{I}_{\text{temporal}} \leq 0$, and $\xi_2(\delta) := (m_1+\delta)/n + n/(p_2 - m_1 - \delta)$. Then $\text{TR}_{\text{LB}}(\delta) = N(\delta)/D(\delta)$.

Step 1: Sign of $D'(\delta)$. $D'(\delta) = \bar{\kappa}_{\text{gap}} + \epsilon'(\delta) = \bar{\kappa}_{\text{gap}} - \underline{\kappa}_{\text{over}}\xi_2'(\delta)$, where $\xi_2'(\delta) = 1/n + n/(p_2 - m_1 - \delta)^2 > 0$. In the overparameterized regime ($p_2 - m_1 - \delta \gg \sqrt{n}$), $\underline{\kappa}_{\text{over}}\xi_2'(\delta) \ll \bar{\kappa}_{\text{gap}}$, so $D'(\delta) > 0$.

Step 2: Decomposition of $g(\delta) := aD(\delta) - N(\delta)D'(\delta)$. Expanding:

$$\begin{aligned} g(\delta) &= a[\bar{\kappa}_{\text{gap}}\delta + \epsilon(\delta)] - (a\delta + b)[\bar{\kappa}_{\text{gap}} + \epsilon'(\delta)] \\ &= a[\epsilon(\delta) - \delta\epsilon'(\delta)] + |b|[\bar{\kappa}_{\text{gap}} + \epsilon'(\delta)] \\ &= a[\epsilon(\delta) - \delta\epsilon'(\delta)] + |b|\,D'(\delta). \end{aligned} \tag{38}$$

Step 3: The second term $|b|\,D'(\delta) \geq 0$. Immediate from Step 1 ($D' > 0$) and $|b| \geq 0$.

Step 4: The first term $a[\epsilon(\delta) - \delta\epsilon'(\delta)] \geq 0$. Since $\epsilon'(\delta) = -\underline{\kappa}_{\text{over}}\xi_2'(\delta) < 0$, we have $-\delta\epsilon'(\delta) > 0$, so it suffices to show $\epsilon(\delta) - \delta\epsilon'(\delta) \geq 0$. Consider the function $h(\delta) := \xi_2(\delta) - \delta\xi_2'(\delta)$. Then $\epsilon - \delta\epsilon' = \bar{\kappa}_{\text{over}}\xi_1 - \underline{\kappa}_{\text{over}}h(\delta)$, so we need $\bar{\kappa}_{\text{over}}\xi_1 \geq \underline{\kappa}_{\text{over}}h(\delta)$. Differentiating: $h'(\delta) = -\delta\xi_2''(\delta)$, and $\xi_2''(\delta) = 2n/(p_2 - m_1 - \delta)^3 > 0$, so $h'(\delta) = -\delta\xi_2''(\delta) \leq 0$. Hence $h(\delta) \leq h(0) = \xi_2(0) = m_1/n + n/(p_2 - m_1)$ for all $\delta \geq 0$. Since $p_2 \geq p_1$: $\xi_2(0) = m_1/n + n/(p_2 - m_1) \leq m_1/n + n/(p_1 - m_1) = \xi_1$. Since $\bar{\kappa}_{\text{over}} = \bar{C}_{\text{over}}\sigma^2 \geq \underline{C}_{\text{over}}\sigma^2 = \underline{\kappa}_{\text{over}}$:

$$\underline{\kappa}_{\text{over}}h(\delta) \leq \underline{\kappa}_{\text{over}}\xi_2(0) \leq \underline{\kappa}_{\text{over}}\xi_1 \leq \bar{\kappa}_{\text{over}}\xi_1.$$

Therefore $\epsilon(\delta) - \delta\epsilon'(\delta) \geq 0$.

Step 5: Conclusion. From equation 38, $g(\delta) \geq 0$ for all $\delta \geq 0$, with strict inequality whenever $b < 0$ or $\epsilon(\delta) > 0$. Since $\mathrm{TR}'_{\mathrm{LB}}(\delta) = g(\delta)/D(\delta)^2 \geq 0$, $\mathrm{TR}_{\mathrm{LB}}$ is non-decreasing (and strictly increasing under mild non-degeneracy).

Asymptote. As $\delta \to \infty$ (with $p_2 \gg m_1 + \delta$ maintained so that $\xi_2(\delta)/\delta \to 0$), $\epsilon(\delta)/\delta \to 0$, so $D(\delta) \sim \bar{\kappa}_{\mathrm{gap}}\,\delta$ and

$$\mathrm{TR}_{\mathrm{LB}}(\delta) \;\to\; \frac{a}{\bar{\kappa}_{\mathrm{gap}}} = \frac{(1-\eta_1)\underline{\kappa}^{\mathrm{hist}}_{\mathrm{gap}}}{\bar{\kappa}_{\mathrm{gap}}} \;>\; 0.$$

Convergence is from below since $\mathrm{TR}_{\mathrm{LB}}$ is increasing. □

F.6 Sequence Length Analysis

We now make the dependence on the sequence length L explicit and show that longer sequences monotonically increase the LoopFM information gain. Write $\mathbf{H}_u^{(L)} := (\mathbf{x}_{\mathrm{VM}}^{(t_1)}, \ldots, \mathbf{x}_{\mathrm{VM}}^{(t_L)})$ and $\mathbf{S}_u^{(k,L)} := [\mathbf{z}_{k,t_1}, \ldots, \mathbf{z}_{k,t_L}]$ with the three information quantities now L-dependent: $\mathcal{I}_{\mathrm{temporal}}(L)$, $\mathcal{I}_{\mathrm{cross},k}(L)$, $\mathcal{I}_{\mathrm{residual},k}(L)$.

Proposition 13 (Monotonicity in sequence length)**.** *For all $L \geq 1$:*

(i) ***Temporal information is non-decreasing:***

$$\mathcal{I}_{\mathrm{temporal}}(L+1) = \mathcal{I}_{\mathrm{temporal}}(L) + \underbrace{I\Big(\mathbf{x}_{VM}^{(t_{L+1})};\; y \;\Big|\; \mathbf{x}_{VM}^{(t)},\, \mathbf{H}_u^{(L)}\Big)}_{\delta_{\mathrm{temp}}(L) \geq 0} \geq \mathcal{I}_{\mathrm{temporal}}(L). \quad (39)$$

(ii) ***Total LoopFM gain is non-decreasing:***

$$\mathcal{I}_{LoopFM,k}(L+1) = \mathcal{I}_{LoopFM,k}(L) + \underbrace{I\Big(\mathbf{z}_{k,t_{L+1}};\; y \;\Big|\; \mathbf{x}_{VM}^{(t)},\, \mathbf{S}_u^{(k,L)}\Big)}_{\delta_{LoopFM,k}(L) \geq 0} \geq \mathcal{I}_{LoopFM,k}(L), \quad (40)$$

where $\mathcal{I}_{LoopFM,k}(L) := I(\mathbf{S}_u^{(k,L)}; y \mid \mathbf{x}_{VM}^{(t)})$.

(iii) ***Convergence:*** *Both sequences converge: $\mathcal{I}_{\mathrm{temporal}}(L) \nearrow \mathcal{I}^*_{\mathrm{temporal}}$ and $\mathcal{I}_{LoopFM,k}(L) \nearrow \mathcal{I}^*_{LoopFM,k}$ as $L \to \infty$, since they are non-decreasing and bounded above by $H(y \mid \mathbf{x}_{VM}^{(t)}) \leq \log 2$.*

Both monotonicity results follow from the chain rule of MI, which decomposes the information in $L{+}1$ events into the first L events plus a non-negative marginal contribution from the $(L{+}1)$-th event.

Remark 14 (Sequence length and achieved risk)**.** *At the population level, Proposition 13 guarantees that longer sequences always improve LoopFM. Since the total gain is bounded by $H(y)$, the marginal contribution $\delta_{LoopFM,k}(L) \to 0$, yielding diminishing returns. For achieved risk, there is a bias–variance trade-off: larger L increases information gain but can increase estimation cost, so there exists an optimal L^* beyond which estimation cost dominates—though in practice, Transformer-based encoders handle long sequences efficiently, making L^* large. This is consistent with Table 5, which shows continued gains up to $L = 100{+}$.*

F.7 Derivation of the A1 Excess-Risk Envelope from NTK + Benign Overfitting Assumptions

Rather than postulating the A1 envelope directly, we derive it from: (i) NTK/lazy-training linearization (Jacot et al., 2018), and (ii) the finite-sample benign-overfitting conditions in Bartlett et al. (2020, Theorem 1).

Applicability to cross-entropy training. Bartlett et al. (2020) analyzes the minimum-norm interpolating (MNI) estimator under squared loss, whereas our FMs are trained with binary cross-entropy (BCE). Wang et al. (2023) bridge this gap: in the overparameterized regime ($p_k \gg n$), gradient

descent on CE loss converges to the SVM solution, which equals the MNI solution (Wang et al., 2023, Theorem 1, Corollary 1)—the predictor $\hat{w}$ and classification error are identical regardless of training loss. Bartlett's excess-risk bounds on the MNI therefore apply directly to CE-trained FMs. Wang et al. (2023, Theorems 4–5) further establish that benign overfitting extends to classification with the same spectral conditions, confirming the applicability of our assumptions.

Notation alignment with Bartlett et al. (2020). Their linear model is $y = \langle w^\star, x\rangle + \xi$ with covariance matrix $\Sigma := \mathbb{E}[xx^\top]$ and eigenvalues $(\lambda_j)_j$. In our linearized FM surrogate, we identify:

- $x \leftrightarrow \phi_k(u, a) \in \mathbb{R}^{p_k}$ (NTK/lazy-training feature of FM_k),
- $w^\star \leftrightarrow w_k^\star$,
- $\Sigma \leftrightarrow \Sigma_k := \mathbb{E}[\phi_k \phi_k^\top]$,
- prediction error $\epsilon_{\text{over}}(p_k, m_k) \leftrightarrow$ excess risk term induced by finite-width/finite-sample interpolation.

Assumptions (Bartlett-style specialization). Let $\lambda_{k,1} \geq \cdots \geq \lambda_{k,p_k} \geq 0$ be the eigenvalues of Σ_k. We assume:

(B1) **Linearized regime:** FM_k operates in a local NTK/lazy-training regime so that prediction is modeled by a linear rule on ϕ_k (Jacot et al., 2018).

(B2) **Data model:** $y = \langle w_k^\star, \phi_k\rangle + \xi$, with $\mathbb{E}[\xi \mid \phi_k] = 0$ and $\mathbb{E}[\xi^2 \mid \phi_k] \leq \sigma^2$.

(B3) **Bartlett spectral split condition:** with split index d_k, the quantities

$$r_{d_k}(\Sigma_k) := \frac{\sum_{j>d_k} \lambda_{k,j}}{\lambda_{k,d_k+1}}, \qquad R_{d_k}(\Sigma_k) := \frac{\left(\sum_{j>d_k} \lambda_{k,j}\right)^2}{\sum_{j>d_k} \lambda_{k,j}^2}$$

satisfy the finite-sample side conditions required by Bartlett et al. (2020, Theorem 1).

Here, the *split index* $d_k \in \{1, \ldots, p_k - 1\}$ is the cutoff used in Bartlett et al. (2020) to separate the spectrum into a leading block and a tail block: indices $j \leq d_k$ are the principal/signal directions, while indices $j > d_k$ form the overparameterized tail that governs interpolation noise.

Theorem 15 (Linearized-surrogate excess-risk scaling)**.** *Under* (B1)–(B3) *and the distributional/moment conditions of Bartlett et al. (2020, Theorem 1), let n denote the number of training samples. The expected excess risk of FM_k satisfies*

$$\underline{C}_{\text{BO}}\, \sigma^2\, \Psi_k(\Sigma_k, d_k) \;\leq\; \epsilon_{over}(p_k, d_k) \;\leq\; \bar{C}_{\text{BO}}\, \sigma^2\, \Psi_k(\Sigma_k, d_k)\,, \tag{41}$$

for constants $\underline{C}_{\text{BO}}, \bar{C}_{\text{BO}} > 0$, where the noise functional is

$$\Psi_k(\Sigma_k, d_k) := \frac{d_k}{n} + \frac{n}{R_{d_k}(\Sigma_k)}, \tag{42}$$

with $\widetilde{r}_{d_k}(\Sigma_k) := \sum_{j>d_k} \lambda_{k,j}$ and $R_{d_k}(\Sigma_k)$ as defined above. The two-sided bound holds under the benign-regime condition $r_0(\Sigma_k)/n \to 0$, which ensures the signal (bias) term is dominated by the noise term.

Proof. By (B1)–(B3), the aligned surrogate pair $(w_k^\star, \Sigma_k)$ is in the linear-model setting of Bartlett et al. (2020, Theorem 1).

Lower bound (expected risk). Bartlett's Theorem 1 gives $\mathbb{E}[R(\hat{\theta})] \geq (\sigma^2/c)(d_k/n + n/R_{d_k})$, establishing the lower side of equation 41.

Upper bound (high-probability to expectation). Bartlett's Theorem 1 gives, with probability $\geq 1 - \delta$ (for $\log(1/\delta) < n/c$):

$$R(\hat{\theta}) \leq \underbrace{c\left(\|w_k^\star\|^2 \|\Sigma_k\| \max\left(\sqrt{\frac{r_0}{n}}, \frac{r_0}{n}, \sqrt{\frac{\log(1/\delta)}{n}}\right)\right)}_{=:\, S(\delta)} + c \log(1/\delta)\, \sigma^2\, \Psi_k.$$

Set $t := c\log(1/\delta)\sigma^2\Psi_k$, so $\delta = \exp(-t/(c\sigma^2\Psi_k))$, giving $P(R(\hat{\theta}) - S(\delta) > t) \le e^{-t/(c\sigma^2\Psi_k)}$. Tail-integrating:

$$\mathbb{E}[R(\hat{\theta})] \;\le\; S_0 + \int_0^\infty e^{-t/(c\sigma^2\Psi_k)}\,dt \;=\; S_0 + c\,\sigma^2\,\Psi_k,$$

where $S_0 := S(e^{-1}) = O(\|w_k^\star\|^2\|\Sigma_k\|\sqrt{r_0/n})$ is the signal term evaluated at $\delta = e^{-1}$. Under bounded signal-to-noise ratio $\|w_k^\star\|^2\|\Sigma_k\|/\sigma^2 = O(1)$, we have $S_0 = O(\sigma^2\sqrt{r_0/n}) = O(\sigma^2/\sqrt{n})$, while $\sigma^2\Psi_k \ge \sigma^2 n/R_{d_k}$. Hence $S_0/(\sigma^2\Psi_k) = O(R_{d_k}/n^{3/2}) \to 0$ as $n \to \infty$, so $S_0 = o(\sigma^2\Psi_k)$ is absorbed into the noise term, yielding the upper side of equation 41. □

Corollary 16 (Recovering the two-sided A1 envelope from Bartlett)**.** *Under* (B1)–(B2)*, Theorem 15 yields the two-sided Assumption (A1) with* $\xi_k := \Psi_k(\Sigma_k, d_k) = d_k/n + n/R_{d_k}(\Sigma_k)$*. For* $\xi_k \to 0$ *(i.e.,* $\epsilon_{over} \to 0$*), we adopt the benign covariance condition of Bartlett et al. (2020, Definition 4): the covariance sequence* Σ_k *(indexed by* n*) is* benign *if*

$$\frac{r_0(\Sigma_k)}{n} \to 0, \qquad \frac{d_k}{n} \to 0, \qquad \frac{n}{R_{d_k}(\Sigma_k)} \to 0,$$

where d_k *is the split index (corresponding to* k_n^* *in Bartlett's notation). These three conditions ensure (i) the signal term* $S_0 \to 0$ *(via* $r_0/n \to 0$*), and (ii) the noise functional* $\xi_k = d_k/n + n/R_{d_k} \to 0$*. Bartlett et al. (2020, Theorem 2.2) shows these conditions are satisfied for finite-dimensional models (*$p_k \gg n$*) with a bi-level covariance:* d_k *large eigenvalues plus* $p_k - d_k$ *small, nearly-equal eigenvalues at level* λ_{tail}*, giving* $R_{d_k} \asymp p_k - d_k \gg n$*. This corresponds to* $d_k \leftrightarrow m_k$ *in the main-text notation.*

Proof. **Step 1 (split index choice).** We set $d_k := m_k$, the input feature dimension of FM_k. In the NTK linearization, the p_k-dimensional feature map ϕ_k has m_k directions aligned with the input features (the signal subspace) and $p_k - m_k$ directions from overparameterization (the interpolation tail). This is the natural Bartlett split: the first m_k eigendirections of Σ_k capture input signal, and the remaining $p_k - m_k$ form the tail whose effective rank R_{m_k} governs the noise functional.

Step 2 (two-sided bound). With $d_k = m_k$, Theorem 15 gives $\underline{C}_{\text{BO}}\sigma^2\Psi_k \le \epsilon_{\text{over}} \le \bar{C}_{\text{BO}}\sigma^2\Psi_k$ where $\Psi_k = m_k/n + n/R_{m_k}$. Setting $\xi_k := \Psi_k$, $\underline{C}_{\text{over}} := \underline{C}_{\text{BO}}$, $\bar{C}_{\text{over}} := \bar{C}_{\text{BO}}$ recovers the A1 envelope.

Step 3 (benign regime). Following Bartlett et al. (2020, Definition 4), the covariance Σ_k is benign if three conditions hold as $p_k, n \to \infty$:

(i) $r_0(\Sigma_k)/n \to 0$: The full effective rank $r_0 = \text{tr}(\Sigma_k)/\|\Sigma_k\|$ grows slower than n. This ensures the signal (bias) term $S_0 = O(\|w_k^\star\|^2\|\Sigma_k\|\sqrt{r_0/n})$ is absorbed into the noise term (see Theorem 15 proof).

(ii) $m_k/n \to 0$: The split index (number of signal directions) is sublinear in the sample size. This ensures the first term of $\xi_k = m_k/n + n/R_{m_k}$ vanishes.

(iii) $n/R_{m_k}(\Sigma_k) \to 0$: The tail effective rank R_{m_k} exceeds the sample size. We show $R_{m_k} = p_k - m_k$ under a bi-level covariance where the $p_k - m_k$ tail eigenvalues are all equal to $\lambda_{\text{tail}} > 0$. Let $q := p_k - m_k$. Then:

$$\sum_{j>m_k} \lambda_j = q\,\lambda_{\text{tail}}, \qquad \sum_{j>m_k} \lambda_j^2 = q\,\lambda_{\text{tail}}^2.$$

By definition of the effective rank:

$$R_{m_k} = \frac{\left(\sum_{j>m_k}\lambda_j\right)^2}{\sum_{j>m_k}\lambda_j^2} = \frac{(q\,\lambda_{\text{tail}})^2}{q\,\lambda_{\text{tail}}^2} = \frac{q^2\,\lambda_{\text{tail}}^2}{q\,\lambda_{\text{tail}}^2} = q = p_k - m_k.$$

Since $p_k \gg n$: $R_{m_k} = p_k - m_k \gg n$, so $n/R_{m_k} \to 0$.

Together, (i)–(iii) give $\xi_k = m_k/n + n/R_{m_k} \to 0$ and the signal absorption holds, yielding $\epsilon_{\text{over}} \to 0$. □

Spectral validation. In overparameterized FMs ($p_k \gg n$), the NTK covariance Σ_k exhibits a spiked structure: a few large eigenvalues (signal directions aligned with the m_k input features) followed by a large bulk of $p_k - m_k$ small, approximately equal eigenvalues. This matches the bi-level structure required by the benign condition of Bartlett et al. (2020): the $p_k - m_k$ tail eigenvalues form an approximately flat bulk, giving $R_{m_k} \asymp p_k - m_k \gg n$.

Interpretation for FM comparisons. FM_1 and FM_2 are trained on the same pipeline/distribution, so multiplicative constants can be shared; FM-specific variation is isolated in $\xi_k = d_k/n + n/(p_k - d_k)$. Hence the parameter term in Eq. equation 4 is driven by differences in sample-to-dimension ratios (split index vs. overparameterized width), not just raw parameter count.

F.8 General Transfer-Ratio Bound via Bartlett's Effective Rank

The transfer-ratio bound in Theorem 2 uses $\xi_k = m_k/n + n/(p_k - m_k)$, which assumes a bi-level covariance with $R_{m_k}(\Sigma_k) = p_k - m_k$. This can be relaxed to hold under any covariance satisfying the benign conditions of Bartlett et al. (2020, Definition 4).

Theorem 17 (General transfer-ratio bound)**.** *Under the same conditions as Theorem 2, but replacing (A1) with the general Bartlett noise functional at split index d_k:*

$$\xi_k := \frac{d_k}{n} + \frac{n}{R_{d_k}(\Sigma_k)}, \qquad R_{d_k}(\Sigma_k) := \frac{\left(\sum_{j>d_k} \lambda_{k,j}\right)^2}{\sum_{j>d_k} \lambda_{k,j}^2},$$

the transfer-ratio lower bound becomes

$$\mathrm{TR}_{\textit{LoopFM}} \;\geq\; \frac{-\tau_2\, \mathcal{I}_{\mathrm{temporal}} + (1-\eta_1)\, \underline{\kappa}_{\mathrm{gap}}^{\mathrm{hist}}\, \delta}{\bar{\kappa}_{\mathrm{gap}}\, \delta + \bar{C}_{\mathrm{over}}\, \sigma^2\, \xi_1 - \underline{C}_{\mathrm{over}}\, \sigma^2\, \xi_2}. \tag{43}$$

Setting $d_k = m_k$ (input feature dimension) and $R_{m_k} = p_k - m_k$ (bi-level covariance) recovers Theorem 2.

Proof. The proof of Theorem 2 carries through verbatim with $\xi_k = d_k/n + n/R_{d_k}(\Sigma_k)$ in place of $m_k/n + n/(p_k - m_k)$. The only change is that the two-sided A1 envelope uses the general noise functional $\Psi_k = \xi_k$ from Theorem 15, without specializing the split index or effective rank. □

G Impact of FM Capacity

Table 10 scales DMIN across five capacity levels (1.9M–31.5M parameters). KD+LoopFM performance is *stable* on TaobaoAd: AUC ranges from 0.6243 to 0.6284 (spread 0.0041), meaning even a small FM produces embeddings rich enough for strong LoopFM gains. The ΔAUC (KD+LoopFM minus KD) remains large across all capacity levels (0.0332–0.0380), showing no systematic trend with FM size—LoopFM's embeddings provide strong complementary value regardless of FM capacity. Standalone DMIN scaling shows AUC saturates at 31–65M params and degrades beyond due to overfitting, motivating knowledge transfer over brute-force scaling. We acknowledge that this study might not be representative of real industrial FM with trillions of parameters and billions or even larger number of examples.

Table 10: LoopFM gain across FM capacity levels (TaobaoAd, FM: DMIN variants, VM: DeepFM). FM capacity is scaled by varying embedding dimensions and MLP widths/depths while keeping the architecture fixed. ΔAUC = AUC(KD+LoopFM) − AUC(KD).

Params (M)	Emb. dim	DNN hidden	AUC (KD)	AUC (LoopFM)	AUC (KD+LoopFM)	ΔAUC
1.9	2	[32, 16, 8]	0.5888	0.6159	0.6243	+0.0356
3.8	4	[64, 32, 16]	0.5900	0.6179	0.6266	+0.0366
7.7	8	[128, 64, 32]	0.5952	0.6142	**0.6284**	+0.0332
15.5	16	[256, 128, 64]	0.5887	0.6172	0.6267	+0.0380
31.5	32	[512, 256, 128]	0.5901	0.6175	0.6272	+0.0371

H Interaction Embeddings vs. Item-Only Embeddings

A natural question is whether LoopFM's gains come from the FM's learned *user-item interaction* representations, or whether simpler item-side-only embeddings—raw feature lookup table outputs without DNN processing—would suffice. We ablate this by extracting only the 6 pure item-side feature embeddings from the FM's embedding layer (adgroup_id, cate_id, campaign_id, customer, brand, price; $6 \times 32 = 192$ dimensions), compressing them to d=32 via the same autoencoder architecture used for dnn_hidden_0, and building LoopFM sequences identically to the standard pipeline.

Table 11: Interaction embeddings (dnn_hidden_0) vs. item-only embeddings on TaobaoAd (FM: DMIN, VM: DeepFM). Item-side embeddings use 6 pure ad feature lookups (no user/context features, no DNN processing), compressed via the same AE architecture as dnn_hidden_0.

Method	AUC	LogLoss
Baseline (no distill)	0.5882	0.1977
KD	0.5980	0.1964
LoopFM-only (item-side)	0.6172	0.1956
LoopFM-only (dnn_hidden_0)	0.6245	0.1955
KD + LoopFM (item-side)	0.6291	0.1942
KD + LoopFM (dnn_hidden_0)	**0.6342**	**0.1938**

Table 11 shows that item-only embeddings capture most of LoopFM's gain: KD + LoopFM (item-side) achieves AUC 0.6291, covering 86% of the full interaction embedding's improvement over the KD baseline ($+0.0311$ vs. $+0.0362$).

Why item-side embeddings work well on TaobaoAd. On TaobaoAd, the FM (DMIN) is a relatively shallow model with 3 DNN layers, and the FM and VM share identical features. In this setting, the FM's DNN layers add limited cross-feature interaction information beyond what the raw item embeddings already encode—the item identity features (adgroup_id, campaign_id) are already highly predictive.

Why the gap should widen in practice. This result is a single data point on a public benchmark with a shallow FM. In industrial FMs with much deeper and wider sequence layers, multi-task heads, and cross-domain features, the DNN (or more general interaction) layers contribute a much larger fraction of the representation's total information—fusing user context, behavioral signals, and feature interactions that raw item lookups cannot capture. A 3-layer DMIN barely transforms raw embeddings, so item-only is close to the full representation; a trillion-parameter FM produces interaction representations that are qualitatively different from feature lookups, and the gap should grow accordingly. Identifying the sweet spot—deep enough to capture rich interactions, but not so deep that compression discards too much information (cf. the layer selection ablation in Table 5)—remains an open question.

I Embedding Analysis

To understand what information LoopFM embeddings encode, we conduct three analyses on TaobaoAd (FM: DMIN, autoencoder bottleneck d=32).

Embedding structure. We analyze 50,000 compressed embeddings from Day 5 (FM trained on Days 1–4). The embeddings exhibit full effective rank: all 32 dimensions contribute meaningfully, with the top-4 and top-8 singular values explaining 28.0% and 47.3% of total variance respectively. This confirms that the autoencoder distributes information across all dimensions rather than concentrating it in a low-rank subspace. The $\tanh$ activation bounds values to $[-0.92, 1.00]$ with mean L2 norm 1.71 ± 0.14.

t-SNE visualization. Figure 3 visualizes 20,000 compressed embeddings via t-SNE, colored by FM soft-label (left) and ground-truth click label (right). The soft-label coloring reveals a smooth

gradient from low predicted CTR (blue) to high (red), confirming that the 32-dimensional compressed space preserves the FM's calibration structure. The ground-truth coloring shows partial separation between clicked (red) and non-clicked (blue) samples, consistent with the inherent noise in click prediction. The final t-SNE KL divergence of 4.31 indicates a good fit between the high-dimensional and low-dimensional affinity structures.

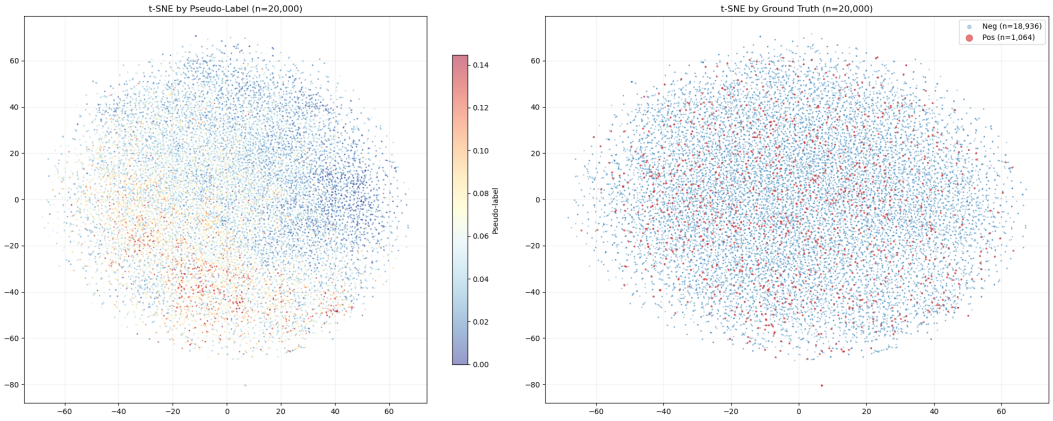


Figure 3: t-SNE visualization of compressed LoopFM embeddings (d=32, 20K samples from TaobaoAd Day 5). **Left:** Colored by FM soft-label (predicted CTR), showing smooth calibration gradients. **Right:** Colored by ground-truth click label, showing partial separation.

Probing experiment. We use the FM's soft-label predictions (scalar soft labels) as a proxy for the FM's learned knowledge. On Day 5 test data, soft-labels achieve AUC 0.6136 as a predictor of ground-truth click labels, confirming they carry substantial task-relevant signal. Figure 4 shows per-dimension Pearson correlations with soft-labels (left) and ground-truth labels (right). The correlation patterns are strongly aligned ($\rho = 0.95$), indicating that the compressed embeddings preserve the FM's ranking of feature importance almost perfectly. This validates that the autoencoder retains task-relevant information despite aggressive dimensionality reduction.

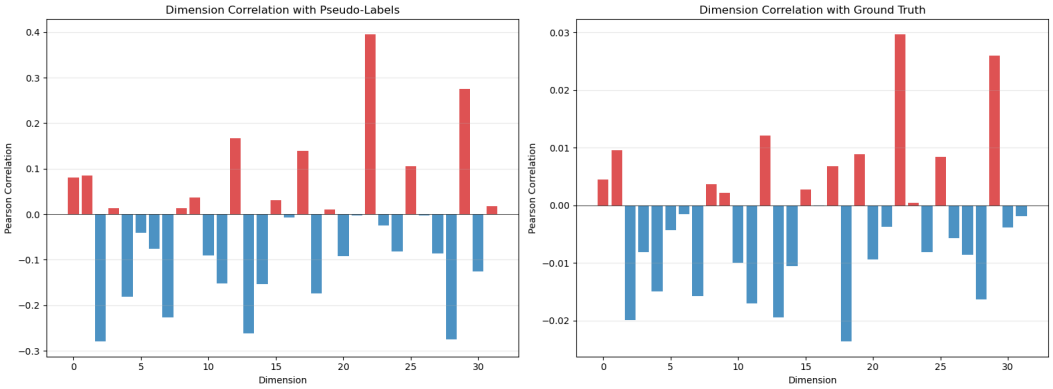


Figure 4: Per-dimension Pearson correlation of compressed embeddings with FM soft-labels (left) and ground-truth click labels (right). The correlation patterns are strongly aligned (ρ=0.95), confirming the autoencoder preserves the FM's learned feature importance structure.

Attention weight analysis. We analyze the multi-head attention weights (4 heads) in the VM's DMIN-style sequence encoder over LoopFM positions on 163,840 test samples (mean sequence length 16.7).

Figure 5 shows that attention weight decreases monotonically with temporal distance to the target interaction: entries within 10 minutes receive $\sim$3$\times$ higher attention than those 4–8 hours away,

confirming strong recency bias consistent with user interest decay. All four heads exhibit nearly identical temporal profiles, suggesting temporal recency is a universal signal rather than a head-specialized one.

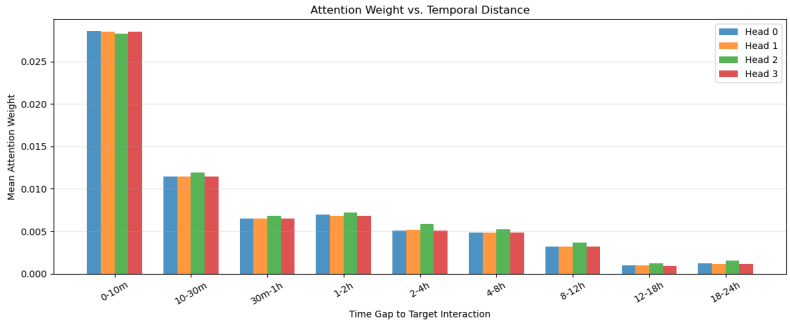


Figure 5: Attention weight vs. temporal distance to target interaction. All 4 heads show strong recency bias: recent entries ($<$10 min) receive $\sim$3$\times$ higher attention than distant ones ($>$4h).

Figure 6 examines whether attention correlates with semantic relevance by comparing weights assigned to same-category vs. different-category historical interactions. All four heads show a consistent 1.13–1.16$\times$ attention lift for same-category items, confirming that the sequence encoder learns to attend preferentially to semantically relevant history entries. Brand matching shows a weaker but consistent 1.04–1.06$\times$ lift, reflecting that category is a stronger relevance signal than brand in TaobaoAd. Interestingly, attention weights for historically clicked items are slightly *lower* than for unclicked items (0.84–0.97$\times$ ratio), suggesting the model leverages both positive and negative interaction signals.

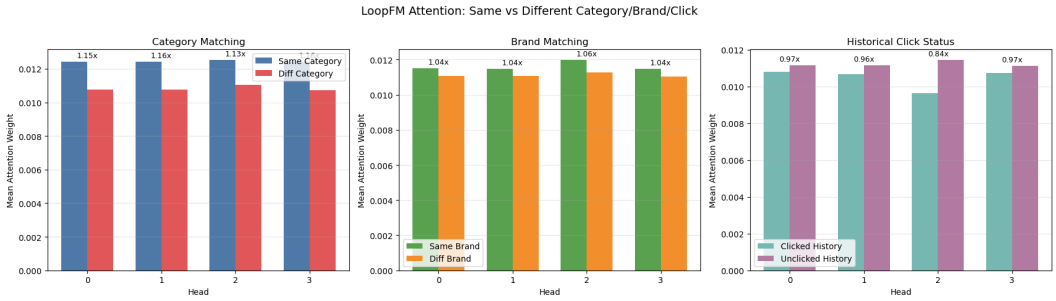


Figure 6: Attention weight comparison by semantic matching. **Left:** Same vs. different category (1.13–1.16$\times$ lift). **Center:** Same vs. different brand (1.04–1.06$\times$ lift). **Right:** Historically clicked vs. unclicked items (0.84–0.97$\times$ ratio), showing the model uses negative signals.

J Future Directions

The modularity of LoopFM's three-stage design naturally suggests extensions along each axis. We highlight several promising directions.

Alternative structuring: item sequences and graph-based features. Our experiments use user-level temporal sequences for Stage 3. An equally natural choice is *ad-side sequences*: grouping FM embeddings by item (ad) rather than user to capture how different users have interacted with the same ad over time, providing item-level demand signals. Beyond flat sequences, FM embeddings can be organized into *graph-structured neighborhoods*—for example, constructing user-item bipartite graphs where edges carry FM embeddings, enabling graph neural network encoders in the VM.

Cross-domain transfer. LoopFM's decoupled architecture naturally supports cross-domain knowledge transfer: an FM trained on one domain (e.g., organic content) can generate embeddings

consumed by VMs in a different domain (e.g., ads). Because the VM consumes FM embeddings as opaque input features, no alignment of prediction objectives or feature schemas is needed

Self-LoopFM. An intriguing direction is *self-LoopFM*: having the FM itself consume its own historical embeddings as input features in subsequent training iterations, creating a self-improving loop.

Checkpoint-aligned embeddings. Our FM checkpoint frequency ablation (Section 5.3.5) reveals that embedding consistency within a user's sequence matters more than individual embedding freshness. When the FM is updated incrementally, its embeddings drift (paired cosine similarity decays to $\sim$0.80 over 3 days), degrading sequence encoder performance. Techniques such as anchored training, projection alignment (learning a linear map from new-checkpoint space to old-checkpoint space), or EMA-based checkpoint interpolation could enable LoopFM to benefit from fresher FM knowledge without sacrificing within-sequence consistency.

Vector quantization. Currently we use scalar quantization such as INT4 for storage efficiency. While K-means can minimize INT4 quantization loss, it is intriguing to pursue quantization in vector space, with notable approaches such as RQ-VAE (Lee et al., 2022).

K LIMITATIONS

(1) **Storage**: LoopFM sequences require significant storage at scale. (2) **Cold-start**: New users lack embedding history. (3) **Compression**: High-dimensional embeddings inevitably lose information. (4) **Latency**: Sequence features incur VM inference cost. (5) **Scale gap**: Public benchmarks ($\sim$6M params) results may not fully generalize to trillion-parameter settings. (6) **Weakness of the theoretical analysis**: Our bounds are stated in terms of population-level quantities (mutual information and Bayes risk) and characterize optimal transfer under ideal learning, without accounting for finite-sample effects, optimization issues, and model capacity limitations.