Competition-Conditioned Softmax for Implicit Recommender Systems

Anonymous Author(s)

Abstract

In implicit-feedback recommender systems (RS), the behaviour of Softmax Loss (SL) depends strongly on the sampled negative set. Some sampled sets contain weak competitors, while others contain highly relevant or informative ones, so treating all training instances with a single global temperature and uniformly weighted negatives can make optimisation brittle. A sharpness level that is appropriate for one user-item interaction may therefore be poorly matched to another because the sampled competition is different. We address this with Dual-scale Softmax Loss, a variant of SL that conditions the loss on the competition induced by the sampled negatives themselves. Dual-scale Softmax Loss operates at two levels. Within each instance, it reshapes competition across negatives using hardness and item-item similarity. Across instances, it adjusts the effective temperature according to the competition intensity of a constructed competitor slate. This retains the log-sum-exp structure of SL, but allows the competition distribution to adapt both locally and at the example level.

Across representative datasets and backbones, Dual-scale Softmax Loss consistently improves upon strong baselines. Relative to SL, gains exceed 10% in several settings and average 6.22% across datasets, metrics, and backbones. Under out-of-distribution (OOD) popularity shift, the gains are larger, with an average improvement of 9.31% over SL. We further present a distributionally robust optimisation (DRO) analysis showing that Dual-scale Softmax Loss changes both the robust payoff and the Kullback-Leibler (KL) deviation associated with ambiguous instances, providing theoretical support for the observed improvements in accuracy and robustness.

CCS Concepts

• **Information systems** → *Personalization*; **Recommender systems**; *Similarity measures.*

Keywords

Recommender systems, loss function, softmax, competitor aware

ACM Reference Format:

Anonymous Author(s). 2026. Competition-Conditioned Softmax for Implicit Recommender Systems. In *Proceedings of Proceedings of the International ACM Conference on Knowledge and Information Management (CIKM 2026) (CIKM'26).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommender Systems (RS) are at the core of personalised digital platforms [22]. Recent advancements have focused on innovations in model architecture. Latent factor models, deep sequential models using transformer self-attention, and graph-based neural network (GNN) recommender systems have been developed to understand and analyse user preferences and relationships more accurately [5, 6, 9, 10]. On the other hand, the learning objective is often treated as a largely interchangeable component, with many systems defaulting to pointwise or pairwise losses. Recent analyses argue that loss functions in RS can substantially affect robustness, bias, and ranking quality, and have remained comparatively underexplored relative to architectures [21, 22]. RS objectives are commonly grouped into three categories: *pointwise* losses that score user–item interactions independently, promoting alignment between models predictions and labels (e.g., regression or binary classification), *pairwise* losses that encourage a higher margin between the positive item and sampled negatives (e.g., Bayesian Personalized Ranking (BPR)), and *softmax* losses that normalise over item predictions, creating a multinomial distribution and optimising the probability of positives versus negatives [22]. The softmax loss (SL) becomes especially attractive at web scale via sampled softmax (SSM), which approximates the full softmax over a large collection of items by contrasting positives against a sampled negative set. This approach is used in industrial recommendation pipelines and is closely related to contrastive formulations such as InfoNCE [15].

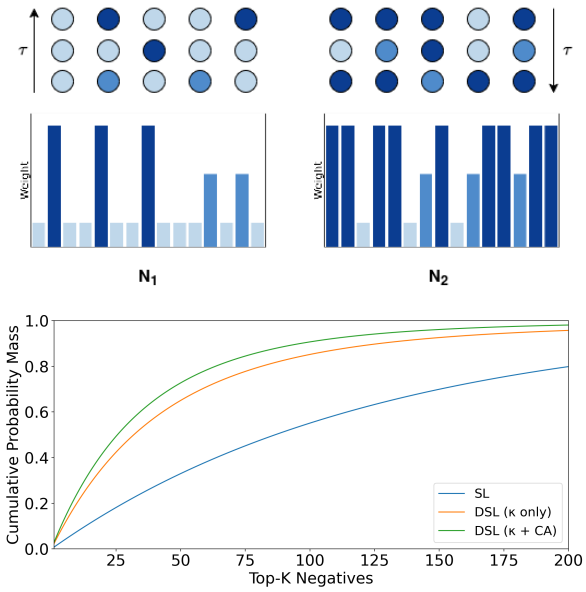

Figure 1: Simplified diagram illustrating how DSL reshapes SL competition over two slates. N_1 and N_2 are two different sets of 15 randomly sampled negatives. N_1 has few strong sampled competitors so the per-example τ (shown with arrows) smoothens the loss. N_2 has high competition so the per-example τ sharpens DSL. The graph shows that within each set, competitive items receive larger relative weightings.

CIKM'26, Rome, Italy
2026. https://doi.org/10.1145/nnnnnnn.nnnnnnn

The SL is highly sensitive to the temperature, τ, which plays a direct role in controlling how sharply the loss concentrates probability over the negatives and therefore the gradient mass assigned to competitors. Too small a τ focuses training on fewer negatives and can destabilise training, whereas too high a τ could wash out informative competition. Recent analyses discuss the conceptual advantages of SL, such as better hard-negative mining and popularity bias mitigation, but also highlight the dependence of SL performance on how negatives are sampled and weighted [21, 22]. SL assumes that randomly drawn negatives form a meaningful choice set and treats them equally. Since most sampled items are not plausible substitutes, this diverts gradients away from strong and relevant alternatives. While τ controls sharpness over the sampled set, it cannot make the pool itself align with true competitors. These observations suggest that a single global τ may not be an effective/robust choice in practice, especially when uniformly sampling a set of negatives from a large item collection under implicit feedback data, where non-interactions are confounded by exposure and can contain false negatives [12, 17, 18].

Our proposed Dual-scale Softmax Loss (DSL) addresses this gap by treating the effective temperature not as a single hyperparameter, but as an inferred quantity which is shaped both by the negatives sampled during training as well as how sharply the loss concentrates gradients on the competitors. To this end, DSL implements two light-weight adaptations into the SL backbone: *(i)* a branch that reweights sampled negatives per example to emphasise meaningful competitors and *(ii)* a competitor-aware (CA) branch that forms a pseudoslate of top competitors and adapts the sharpness per example by deriving a temperature multiplier from how strongly hardness aligns with relevance, such that effective sharpness is shaped by the competition pool. The two branches are implemented with appropriate normalisation to avoid rescaling logits globally i.e., they redistribute weights within- and per-example. Together, these adaptations attempt to correct the loss's default assumption that all sampled negatives and all examples should be treated equally. Figure 1 illustrates this intuition and how DSL reallocates the learning signal across and within sampled negatives. This weighting behaviour also targets a more fundamental limitation of SL. When training recommender systems with implicit feedback data, the sampled softmax loss training regime can be thought of as an approximation to discrete choice models (DCMs), where a user chooses an item from a displayed set and the multinomial logit formulation turns that into choice-from-a-slate probabilities [13]. Unfortunately, the more common implicit feedback datasets lack exposure logs. As such, an absence of interaction could mean either that the user was exposed to the item but did not select it, or that the user was never shown the item. This leads to what is described in the literature as missing-not-at-random (MNAR) exposure bias [12, 17, 18]. The common solution to this is to sample negatives uniformly (or under some guided sampling). However, this approach often picks negatives that were never true competitors and were not co-exposed with the interacted item. This weakens the interpretation of the loss as a true slate conditioned choice. This mismatch motivates DSL. Without impression logs, DSL reshapes the sampled competition signal by emphasising hard and relevant alternatives and adapting the temperature across examples.

We summarise our contributions as follows:

- We introduce per-negative reweighting, driven by hardness and relevance proxies, with normalisation per example.
- We incorporate per-example temperature adaptation over pseudoslates of top sampled competitors with normalisation across examples.
- We conduct experiments on several model backbones and real-world datasets demonstrating the broad applicability of DSL.
- We conduct ablations to isolate the contributions of the two branches, sensitivity analyses over hyperparameters and an out-of-distribution evaluation to explore behaviour under popularity-distribution shifts.

2 Related Work

2.1 Pointwise and Pairwise Objectives

Pointwise and pairwise losses are the most common objectives used to train implicit-feedback recommender systems [22]. Pointwise losses treat the recommendation as a classification or regression task, using losses such as mean squared error (MSE) or binary-cross entropy (BCE) applied to each positive and negative instance separately [6, 7]. Most pointwise formulations take into account instance confidence, weighting observed positives and missing negatives differently [6, 7]. Many deep collaborative models adopt this perspective (observed vs. unobserved), changing the scoring function while keeping the same supervision signal [6].

Pairwise losses optimise relative preferences among items, attempting to rank positive items above negative items by increasing the gap between their predicted scores, which determines the ordering. Bayesian Personalized Ranking (BPR) formalises this idea via a probabilistic pairwise criterion (often referred to as BPR-Opt) [16]. This pairwise view aligns more clearly than pointwise classification, with the desired ranking behaviour of common recommender systems [16, 22].

2.2 Softmax-style Objectives

SL and its scalable approximation sampled SL have garnered more attention recently in recommender systems [22]. As the name suggests, SL applies a softmax over predicted scores to obtain a multinomial distribution and maximises the probability of positives relative to negatives. It has a direct link with contrastive objectives, and popular Information Noise-Contrastive Estimation (InfoNCE) can be viewed as a variant of sampled softmax [15]. The temperature, τ governs how much the gradient update is concentrated on the hardest negatives as opposed to being spread evenly across the negatives of all difficulties. We go more into further depth on SL's formulation in Section 3.2.

In a recent analysis of SL, Wu et al. confirmed the benefits of sampled SL in the realm of item recommendation, attributing those benefits to stronger hard-negative emphasis and closer alignment with ranking quality metrics (like NDCG) [8, 15]. They also highlight failure modes tied to representation scaling choices, backbone choices and loss interactions. They symmetrise SL robustness to negatives through additional positive regularisation. This extends the log-mean-exponent structure, thus reducing sensitivity to noisy positives [21]. Analysing from a pairwise perspective, Yang et al. introduced PSL, which modifies the surrogate activation function to

tighten SL's interpretation as a DCG surrogate loss [23]. The choice of activation (ReLU or Tanh) is highlighted as improving robustness by mitigating the impact of false negatives. Finally, to address top-K truncation, Yang et al. propose SoftmaxLoss@K (SL@K), which augments SL with a top-K quantile weighting for each user. This acts as a threshold that separates the top-K positive items from the rest. A Monte Carlo-based quantile estimation strategy is developed to achieve computational efficiency. The SL@K loss can be thought of as extending the idea of SL as a DCG surrogate, to being a DCG@K surrogate, with closer alignment to NDCG@K metrics [8, 24].

All these variants show a clear trend of treating SL as a flexible backbone loss, and then shaping its behaviour through symmetry, pairwise surrogacy or top-K extensions, addressing the challenges of implicit-feedback recommendation. However, one of the main limitations of these objectives is the assumption of a single global temperature and a uniform treatment of the sampled negatives, even though their quality and competitiveness may vary greatly between different user-positive pairs. Additionally, without impression logs (i.e. when true exposure is unknown), uniform sampling would include items that were never plausible co-exposed competitors [12]. The presence of such negatives weakens SL's interpretation as a DCM [13]. With this limitation in mind, DSL is designed to reshape the effective temperature from the sampled competition itself, bridging the gap between the true slate-conditioned choice intuition and the reality of training under an implicit-feedback setting.

3 Preliminaries

3.1 Task Formulation

Our discussion is carried out within the collaborative filtering (CF) [16] setting in which supervision is obtained from user–item interaction logs (implicit-feedback). $\mathcal{U}$ and $\mathcal{I}$ denote the user set and item set, respectively. A CF dataset $\mathcal{D} \subseteq \mathcal{U} \times \mathcal{I}$ is a collection of observed interactions, where each instance $(u, i) \in \mathcal{D}$ indicates that user u has interacted with item i. For a user u, the set of observed (positive) items is defined as

$$\mathcal{P}_u = \{\, i \in \mathcal{I} : (u, i) \in \mathcal{D} \,\}, \tag{1}$$

while the negatives for training are unobserved items $\mathcal{I} \setminus \mathcal{P}_u$. For each observed pair (u, i), we sample a set of N negatives

$$\mathcal{N}_u \subset \mathcal{I} \setminus \mathcal{P}_u, \qquad |\mathcal{N}_u| = N, \tag{2}$$

where we use j to denote negative items (i.e., $j \in \mathcal{N}_u$).

The objective is to learn a scoring function $f(u, i) : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ that quantifies the preference of user u for item i. An embedding-based approach is commonly used in modern recommender systems (RS) [10]. Here, we represent each user u and item i using d-dimensional embeddings $\mathbf{u}_u, \mathbf{v}_i \in \mathbb{R}^d$. We then compute the preference score from their embedding similarity. We use cosine similarity:

$$f(u, i) = \frac{\mathbf{u}_u^\top \mathbf{v}_i}{\|\mathbf{u}_u\|_2 \, \|\mathbf{v}_i\|_2}. \tag{3}$$

The scores $f(u, i)$ are then used to rank items for generating recommendations.

Additionally, we define the item–item similarity between the positive item i and a negative item j as

$$s_{ij} = \cos(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i^\top \mathbf{v}_j}{\|\mathbf{v}_i\|_2 \, \|\mathbf{v}_j\|_2} \tag{4}$$

3.2 Softmax Loss

The Softmax Loss (SL) has the log-sum-exp form given by [22]:

$$\mathcal{L}_{\mathrm{SL}} = \sum_{(u,i) \in \mathcal{B}} \log \sum_{j \in \mathcal{N}_u} \exp\left(\frac{f(u, j) - f(u, i)}{\tau} \right), \tag{5}$$

where $f(u, i)$ and $f(u, j)$ denote user scores for positive item i and negative item j.

SL is a smooth surrogate for Discounted Cumulative Gain (DCG) [2]. Let $r_u(i) \in \{1, 2, \dots\}$ denote the (1-indexed) ranking position of item i under scores $f(u, \cdot)$. For binary relevance, the user-level DCG can be written as

$$\mathrm{DCG}(u) = \sum_{i \in \mathcal{P}_u} \frac{1}{\log_2(1 + r_u(i))}, \tag{6}$$

where $\mathcal{P}_u$ is the set of observed positives for user u. A standard relaxation uses $\log_2(1 + \pi_u(i)) \le \pi_u(i)$ and Jensen's inequality to obtain

$$-\log \mathrm{DCG}(u) + \log |\mathcal{P}_u| \;\le\; \frac{1}{|\mathcal{P}_u|} \sum_{i \in \mathcal{P}_u} \log r_u(i). \tag{7}$$

Expressing the rank position through pairwise comparisons and writing the margin $d_{uij} = f(u, j) - f(u, i)$, we get:

$$r_u(i) = 1 + \sum_{j \in \mathcal{I} \setminus \{i\}} \mathbb{I}(f(u, j) \ge f(u, i)) = 1 + \sum_{j \in \mathcal{I} \setminus \{i\}} \delta(d_{uij}), \tag{8}$$

where $\delta(\cdot)$ is the Heaviside step function. For any $\tau > 0$ we get $\delta(d) \le \exp(d/\tau)$, and we can define the smooth upper bound:

$$\log \pi_u(i) = \log \sum_{j \in \mathcal{I}} \delta(d_{uij}) \;\le\; \log \sum_{j \in \mathcal{I}} \exp\left(\frac{d_{uij}}{\tau} \right). \tag{9}$$

Substituting (9) into (7) yields that SL (with $\mathcal{I}$ or a sampled approximation of $\mathcal{I}$) serves as a smooth upper bound on the negative log-DCG, providing a principled surrogate objective for ranking quality (and similarly for mean reciprocal rank (MRR)).[1]

4 Methodology

4.1 Within Example Weighted Competition (κ Branch)

The sampled Softmax Loss (SL) in Eq. (5) treats all sampled negatives $j \in \mathcal{N}_u$ symmetrically within each training instance (u, i). However, in implicit-feedback collaborative filtering (CF), some negatives are more *competitive* than others—e.g., they may already score highly for u (hard negatives), or be highly similar to the positive item i (near-miss alternatives). To model this heterogeneity without changing the sampler, we introduce a per-negative *competition weight* $\kappa_{uij} > 0$ that scales each logit difference inside the softmax.

[1] In practice $\mathcal{I}$ is approximated by $\mathcal{N}_u$ in (5).

4.1.1 Constructing κ_{uij} from hardness and item–item similarity: The negative score $f(u, j)$ is used as a hardness proxy. High $f(u, j)$ indicates that the model estimates negative j as highly relevant to user u at this point in training. Additionally, item–item similarity between the positive item i and negative item j is used as a proxy for substitutability (nearest-neighbour alternatives). To ensure the similarity range is $s_{ij} \in [0, 1]$, avoiding negative s_{ij} for numerical stability, we use the shifted form of Eq.4:

$$\bar{s}_{ij} = \frac{s_{ij} + 1}{2} \in [0, 1]. \tag{10}$$

High s_{ij} means j and i are alike in representation space, making the negative j a valid alternative that user u could accept. These high s_{ij} are very informative as they represent confusable items. High s_{ij} also suggests that j lies in the same choice set as i. Without impression logs, it is a reasonable proxy for co-exposure i.e., the competition that could be modeled with impression logs.

We then form a per-negative *logit* ℓ_{uij} by combining hardness and similarity:

$$\ell_{uij} = f(u, j) + \bar{s}_{ij}. \tag{11}$$

We convert ℓ_{uij} into positive, mean-one competition weights κ_{uij} via a single per-instance exponential normalization:

$$\kappa_{uij} = \frac{\exp(\ell_{uij})}{\frac{1}{N}\sum_{j' \in \mathcal{N}_u} \exp(\ell_{uij'})}, \qquad \Rightarrow \qquad \frac{1}{N}\sum_{j \in \mathcal{N}_u} \kappa_{uij} = 1. \tag{12}$$

To control the reweighting strength, we introduce the hyperparameter $\beta \geq 0$:

$$\kappa_{uij} = 1 + \beta\left(\kappa_{uij} - 1\right), \tag{13}$$

which preserves the mean of one property within each example (u, i).

4.1.2 Per-negative logit scaling: With κ_{uij} defined, we revisit the softmax loss definition. To aid brevity, we define the score difference for the training triplet (u, i, j) with $j \in \mathcal{N}_u$

$$d_{uij} = f(u, j) - f(u, i) \tag{14}$$

We then augment the softmax loss as $\kappa_{uij}\, d_{uij}$ inside the log-sum-exp, giving the κ-augmented SL

$$\mathcal{L}_{\kappa\text{-SL}} = \sum_{(u,i) \in \mathcal{B}} \log \sum_{j \in \mathcal{N}_u} \exp\left(\frac{\kappa_{uij}\, d_{uij}}{\tau}\right). \tag{15}$$

Effectively, a per-negative effective temperature $\tau_{uij} = \tau / \kappa_{uij}$ is induced by κ_{uij}, such that a bigger κ_{uij} sharpens the loss around that negative increasing its gradient contribution and vice versa.

4.2 Competition-Aware Temperature (CA Branch)

The κ-branch in Sec. 4.1 models *within-example* heterogeneity by reweighting individual negatives $j \in \mathcal{N}_u$ for a fixed global temperature τ. Complementary to the κ-branch's within-example reweighting, we introduce a *competition-aware (CA)* temperature that changes *across positive training instances* (u, i). Because uniform sampling does not guarantee an equal number of strong, substitutable negatives, some user–positive pairs are intrinsically more or less ambiguous. In general, more competitive instances would benefit from a sharper softmax, whereas less competitive ones would benefit from a smoother softmax for stability.

4.2.1 Competitor slate: For each observed pair (u, i) with N sampled negatives $\mathcal{N}_u$, we form a small competitor slate

$$C_{ui} \subseteq \mathcal{N}_u, \qquad |C_{ui}| = K, \tag{16}$$

by selecting the top-K negatives with the highest current scores $f(u, j)$. This focuses the CA statistics on the most competitive portion of the sampled set while reducing noise from easy negatives.

4.2.2 Hardness-induced competitor distribution: We convert scores on the competitor slate into a probability distribution over negatives:

$$p_{uij} = \frac{\exp\left(\frac{f(u,j)}{\tau}\right)}{\sum_{k \in C_{ui}} \exp\left(\frac{f(u,k)}{\tau}\right)}, \qquad j \in C_{ui}, \tag{17}$$

Intuitively, p_{uij} concentrates mass on the hardest negatives, approximating a local choice set using the model's current scoring mechanism, without impression logs.

4.2.3 Competition intensity from (hardness × similarity): Using the shifted similarity $\bar{s}_{ij} \in [0, 1]$ defined in Eq. 10, we quantify the resemblance of the competitor slate to the positive item under the hardness distribution using the following bounded *log-partition gap* statistic:

$$c_{ui} = \log \sum_{j \in C_{ui}} \exp\left(\frac{f(u,j)}{\tau} + \bar{s}_{ij}\right) - \log \sum_{j \in C_{ui}} \exp\left(\frac{f(u,j)}{\tau}\right) \tag{18}$$

$$= \log\, \mathbb{E}_{j \sim p_{ui}}\left[\exp(\bar{s}_{ij})\right]. \tag{19}$$

Because $\bar{s}_{ij} \in [0, 1]$, we have $\exp(\bar{s}_{ij}) \in [1, e]$ and thus $c_{ui} \in [0, 1]$, making the signal well-scaled and stable.

4.2.4 From competition intensity to per-example temperature: We map c_{ui} to a positive *temperature multiplier* $m_{ui} > 0$ using exponential tilting followed by mean normalization across the mini-batch $\mathcal{B}$, ensuring the average temperature remains calibrated:

$$x_{ui} = \alpha c_{ui} \tag{20}$$

$$m_{ui} = \frac{\exp(x_{ui})}{\frac{1}{|\mathcal{B}|}\sum_{(u',i') \in \mathcal{B}} \exp(x_{u'i'})}, \qquad \Rightarrow \qquad \frac{1}{|\mathcal{B}|}\sum_{(u,i) \in \mathcal{B}} m_{ui} = 1, \tag{21}$$

where $\alpha \geq 0$ controls the modulation strength. The mean normalization in Eq. (21) is designed to prevent global temperature drift, ensuring results remain comparable between techniques.

We then define a *per-example* temperature

$$\tau_{ui} = \frac{\tau}{m_{ui}}, \qquad \text{equivalently} \qquad \frac{1}{\tau_{ui}} = \frac{m_{ui}}{\tau}. \tag{22}$$

Instances with higher competition intensity (larger c_{ui}) tend to yield larger m_{ui} and therefore smaller τ_{ui}, sharpening the softmax around these competitive examples.

4.2.5 CA-softmax objective: Let $d_{uij} = f(u, j) - f(u, i)$ as in Sec. 4.1. With CA only (i.e., $\kappa_{uij} \equiv 1$), we obtain

$$\mathcal{L}_{\text{CA-SL}} = \sum_{(u,i) \in \mathcal{B}} \log \sum_{j \in \mathcal{N}_u} \exp\left(\frac{d_{uij}}{\tau_{ui}}\right). \tag{23}$$

4.3 Combined DSL Objective

When combining both the κ and CA branches, the per-negative logit scaling and the per-example temperature act multiplicatively:

$$\mathcal{L}_{\mathrm{DSL}} = \sum_{(u,i)\in\mathcal{B}} \log \sum_{j\in\mathcal{N}_u} \exp\left(\frac{\kappa_{uij}\, d_{uij}}{\tau_{ui}}\right), \tag{24}$$

which corresponds to an effective per-negative temperature $\tau_{uij} = \tau_{ui}/\kappa_{uij}$.

4.3.1 Final τ drift control: Although the row-wise mean-one constraint $\mathbb{E}_{j\in\mathcal{N}_u}[\kappa_{uij}] = 1$ is enforced, this does not control the average inverse weight, and in general $\mathbb{E}_j[1/\kappa_{uij}] \neq 1$. By Jensen's inequality, for non-constant $\kappa_{uij} > 0$ we have $\mathbb{E}[1/\kappa_{uij}] \geq 1/\mathbb{E}[\kappa_{uij}] = 1$. We therefore rescale by $\mathbb{E}_j[1/\kappa_{uij}]$ to prevent unintended logit-scale (temperature) drift when combining κ with CA.

5 DSL Theoretical Analysis

5.1 Variational view of sampled Softmax and KL-DRO

The standard change-of-measure (Donsker–Varadhan / Gibbs) variational identity connects log-sum-exp to KL-based Distributionally Robust Optimization (KL-DRO). [4, 14].

The uniform reference distribution over negatives is $\pi_u(j) = 1/N$ for $j \in \mathcal{N}_u$. The (normalised) free-energy form is more convenient to work with for analysis purposes:

$$\tilde{\ell}^{\mathrm{SL}}_{ui}(\tau) = \tau \log\left(\mathbb{E}_{j\sim\pi_u}\left[\exp\left(\tfrac{d_{uij}}{\tau}\right)\right]\right) = \tau \log\left(\frac{1}{N}\sum_{j\in\mathcal{N}_u}\exp\left(\tfrac{d_{uij}}{\tau}\right)\right), \tag{25}$$

which differs from the sampled Softmax loss ($\log\sum_{j\in\mathcal{N}_u}\exp(d_{uij}/\tau)$) only by the constant $\tau\log N$.

The variational identity states that for any base measure π and any measurable h,

$$\log\, \mathbb{E}_{j\sim\pi}\big[\exp(h(j))\big] = \sup_{q\in\Delta}\Big\{\mathbb{E}_{j\sim q}[h(j)] - D_{\mathrm{KL}}(q\,\|\,\pi)\Big\}, \tag{26}$$

where Δ is the probability simplex and $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback–Leibler (KL) divergence.

Applying (26) to $h(j) = d_{uij}/\tau$ with $\pi = \pi_u$ yields the KL-DRO form of sampled Softmax:

$$\tilde{\ell}^{\mathrm{SL}}_{ui}(\tau) = \sup_{q_{ui}\in\Delta(\mathcal{N}_u)}\Big\{\mathbb{E}_{j\sim q_{ui}}[d_{uij}] - \tau\, D_{\mathrm{KL}}(q_{ui}\,\|\,\pi_u)\Big\}. \tag{27}$$

The optimiser is the Gibbs/softmax reweighting over negatives,

$$q^{\star}_{uij}(\tau) = \frac{\pi_u(j)\exp\left(\frac{d_{uij}}{\tau}\right)}{\mathbb{E}_{k\sim\pi_u}\left[\exp\left(\frac{d_{uik}}{\tau}\right)\right]} = \frac{\exp\left(\frac{d_{uij}}{\tau}\right)}{\sum_{k\in\mathcal{N}_u}\exp\left(\frac{d_{uik}}{\tau}\right)}. \tag{28}$$

Equation (27) demonstrates the direct role of temperature, such that τ is the penalty weight on deviating from the reference π_u. Smaller τ permits more deviation (a sharper $q^{\star}$ concentrating on the hardest negatives), while larger τ enforces $q^{\star}$ being closer to π_u, which is a smoother and less robust reweighting. This is also seen from the classical log-sum-exp smoothing bound [1]:

$$\max_{j\in\mathcal{N}_u} d_{uij} - \tau\log N \;\leq\; \tilde{\ell}^{\mathrm{SL}}_{ui}(\tau) \;\leq\; \max_{j\in\mathcal{N}_u} d_{uij}, \tag{29}$$

so the approximation gap to the hard maximum is controlled by $\tau\log N$.

5.2 Implications of κ and CA for robustness (KL-DRO view)

Let π_u be uniform on $\mathcal{N}_u$ ($\pi_u(j) = 1/N$). The DSL payoff is defined as $a_{uij} = \kappa_{uij}\, d_{uij}$ and the CA temperature is $\tau_{ui} = \tau/m_{ui}$. Unlike Eq. (24), this expression includes an extra term $\tau_{ui}\log N$

$$\tilde{\ell}^{\mathrm{DSL}}_{ui} = \tau_{ui}\log\left(\mathbb{E}_{j\sim\pi_u}\left[\exp\left(\frac{a_{uij}}{\tau_{ui}}\right)\right]\right) = \tau_{ui}\log\left(\frac{1}{N}\sum_{j\in\mathcal{N}_u}\exp\left(\frac{\kappa_{uij}d_{uij}}{\tau_{ui}}\right)\right). \tag{30}$$

Using the standard Gibbs / Donsker–Varadhan variational identity, $\tilde{\ell}^{\mathrm{DSL}}_{ui}$ yields the KL-DRO form [4]:

$$\tilde{\ell}^{\mathrm{DSL}}_{ui} = \sup_{q_{ui}\in\Delta(\mathcal{N}_u)}\Big\{\mathbb{E}_{j\sim q_{ui}}[a_{uij}] - \tau_{ui}\, D_{\mathrm{KL}}(q_{ui}\,\|\,\pi_u)\Big\}, \tag{31}$$

with optimizer

$$q^{\star}_{uij} = \frac{\exp(a_{uij}/\tau_{ui})}{\sum_{k\in\mathcal{N}_u}\exp(a_{uik}/\tau_{ui})} = \frac{\exp(\kappa_{uij}d_{uij}/\tau_{ui})}{\sum_{k\in\mathcal{N}_u}\exp(\kappa_{uik}d_{uik}/\tau_{ui})}. \tag{32}$$

Compared to standard SL Eq. (31) exhibits two direct robustness effects: (i) κ changes the robustness payoff from d_{uij} to $\kappa_{uij}d_{uij}$ (so the "worst-case negative" is taken over *rescaled* margins), and (ii) CA changes the robustness level per example via the KL penalty weight τ_{ui}. A compact worst-case bound follows from log-sum-exp smoothing:

$$\max_{j\in\mathcal{N}_u} a_{uij} - \tau_{ui}\log N \;\leq\; \tilde{\ell}^{\mathrm{DSL}}_{ui} \;\leq\; \max_{j\in\mathcal{N}_u} a_{uij}, \tag{33}$$

so smaller τ_{ui} makes the objective more "max-like" over $\kappa_{uij}d_{uij}$ (more robust / more peaked $q^{\star}$), while larger τ_{ui} enforces $q^{\star}$ being closer to uniform.

Let $\lambda_{ui} = 1/\tau_{ui}$ and define the log-partition

$$A_{ui}(\lambda) = \log \mathbb{E}_{j\sim\pi_u}\big[\exp(\lambda a_{uij})\big]. \tag{34}$$

The KL radius actually attained by the adversary is

$$\rho_{ui}(\lambda) := D_{\mathrm{KL}}(q^{\star}_{ui}(\lambda)\,\|\,\pi_u) = \lambda A'_{ui}(\lambda) - A_{ui}(\lambda), \tag{35}$$

and it satisfies the monotonicity [20]

$$\frac{d}{d\lambda}\rho_{ui}(\lambda) = \lambda A''_{ui}(\lambda) = \lambda\,\mathrm{Var}_{j\sim q^{\star}_{ui}(\lambda)}[a_{uij}] \;\geq\; 0. \tag{36}$$

Hence, when CA produces larger m_{ui} (so larger $\lambda_{ui} = m_{ui}/\tau$) and κ assigns larger weights for harder and semantically close negatives, the adversary provably deviates further from uniform in KL, strengthening the KL-DRO robustness for that example. Robustness is therefore adaptively concentrated where it matters more. This leads us to the guarantee that CA increases the attained KL deviation when it sharpens.

A limitation that arises from this analysis is that, because sharpening concentrates mass on the tail of the sampled negatives, DSL will be more sensitive to false negatives (noise). Intuitively, because DRO assigns larger losses to harder instances, it will amplify the contribution of noisy data for optimisation. This has been demonstrated in many DRO studies [11, 25].

5.2.1 Calibration and drift: Because $\frac{1}{|\mathcal{B}|}\sum_{(u,i)\in\mathcal{B}} m_{ui} = 1$ (Eq. (21)), CA preserves the batch-average inverse temperature:

$$\frac{1}{|\mathcal{B}|}\sum_{(u,i)\in\mathcal{B}}\frac{1}{\tau_{ui}} = \frac{1}{|\mathcal{B}|}\sum_{(u,i)\in\mathcal{B}}\frac{m_{ui}}{\tau} = \frac{1}{\tau}. \tag{37}$$

When combining with κ, the effective per-negative temperature is $\tau_{uij} = \tau_{ui}/\kappa_{uij}$, so even with $\mathbb{E}_j[\kappa_{uij}] = 1$,

$$\mathbb{E}_{j\sim\pi_u}[\tau_{uij}] = \tau_{ui}\,\mathbb{E}_{j\sim\pi_u}\Big[\frac{1}{\kappa_{uij}}\Big] \;\geq\; \tau_{ui}, \tag{38}$$

motivating the inverse-mean drift control described in Section 4.3. Thus, the full technique avoids temperature drift, and results are comparable to those of other approaches using the same base temperature.

5.3 Metric-Aligned Gradient Estimation of DSL

In metric-driven learning-to-rank, LambdaLoss-style methods weight pairwise updates by the change in the target metric induced by swapping the ranks of (i, j), i.e., $\Delta\text{NDCG@}K_{uij}$ [3, 8]. DSL can be viewed as a sampled-softmax instantiation of this principle: the weight $\lambda^{\text{DSL}}_{uij}$ is a smooth, readily computable proxy that (i) increases with the likelihood that j is a Top-K rank threat (via q_{uij}), and (ii) increases for semantically substitutable competitors (via κ_{uij}), which are the swaps that tend to incur the largest $\Delta\text{NDCG@}K$. When combined with the Top-K emphasis provided by SL@K (where $g(\text{rank}_K(i))$) DSL concentrates gradient budget on the pairs with the highest expected impact on the Top-K metrics [23, 24]. This improves the efficiency and stability of the stochastic metric surrogate, when compared to treating sampled negatives symmetrically.

DSL keeps the same log-sum-exp backbone as SL, but modifies the *competition geometry* in two complementary ways: firstly through *competition-shaped* per-negative scaling (κ branch), and secondly through *competition-aware* per-example temperature modulation (CA branch). Concretely, for a training pair (u, i) and sampled negatives $\mathcal{N}_u$, we define the margin $d_{uij} = f(u, j) - f(u, i)$ and the DSL exponent

$$z_{uij} \;=\; \frac{\kappa_{uij}\, d_{uij}}{\tau_{ui}}, \qquad q_{uij} \;=\; \frac{\exp(z_{uij})}{\sum_{k\in\mathcal{N}_u}\exp(z_{uik})}. \tag{39}$$

For the per-positive loss $\ell_{ui} = \log\sum_{j\in\mathcal{N}_u}\exp(z_{uij})$, the score gradients are

$$\frac{\partial\ell_{ui}}{\partial f(u,j)} = q_{uij}\,\frac{\kappa_{uij}}{\tau_{ui}}, \qquad \frac{\partial\ell_{ui}}{\partial f(u,i)} = -\sum_{j\in\mathcal{N}_u} q_{uij}\,\frac{\kappa_{uij}}{\tau_{ui}}. \tag{40}$$

Thus, DSL induces an implicit *pairwise update weight*

$$\lambda^{\text{DSL}}_{uij} \;\triangleq\; q_{uij}\,\frac{\kappa_{uij}}{\tau_{ui}}. \tag{41}$$

The softmax factor q_{uij} is monotonically increasing in the margin d_{uij}, so it concentrates mass on the highest-scoring negatives. These are the items most likely to swap order with i inside the Top-K band, and therefore most likely to affect the ranking and change the Recall@K and NDCG@K.

5.3.1 Within-example competition shaping (κ branch): The per-negative multiplier κ_{uij} further redistributes update mass *within* each sampled set $\mathcal{N}_u$ without changing the sampler. It increases the effective slope of the comparisons deemed most *competitive* (e.g., near-substitutes or highly confusable rivals), while preserving the log-sum-exp structure. Operationally, κ_{uij} scales both the exponent that defines q_{ui} and the resulting gradient weight in (40), so confusable negatives receive a larger share of the stochastic gradient exactly where pairwise swaps are most impactful for Top-K metrics.

5.3.2 Across-example competition awareness (CA branch): Uniform sampling yields heterogeneous training instances where some (u, i) pairs have stronger, more substitutable sampled negatives in $\mathcal{N}_u$ than others. The CA branch addresses this *across positives* by constructing a small competitor slate $\mathcal{C}_{ui}$ (top-K by $f(u, j)$), estimating a bounded competition intensity c_{ui} on that slate, and mapping it to a multiplier m_{ui} with mean normalization. The resulting per-example temperature $\tau_{ui} = \tau/m_{ui}$ sharpens the softmax for intrinsically ambiguous instances (large c_{ui}) and smooths it otherwise, stabilising learning while increasing gradient concentration on examples where Top-K errors are most likely.

6 Experiments

We first conduct experiments to demonstrate the performance of DSL and DSL@K compared to SL, SL@K and other state-of-the-art losses. We then dig deeper into the properties of DSL, ablating the contributions of the κ and *CA* branches. We explore DSL's robustness to out-of-distribution (OOD) data. Finally, we compare SL and DSL performance with head and tail items.

6.1 Experimental Setup

6.1.1 Datasets: To enable fair comparisons, our experimental protocol is consistent with Yang et al. [23, 24] and Wu et al. [21]. We evaluate on four commonly used and publicly available datasets, namely Amazon-Health (Health), Amazon-Electronic (Electronic), Amazon-Movie (Movie) and Gowalla. These datasets are used to evaluate under IID settings (where the distribution of randomly split training and test data is identical). We also evaluate under OOD settings (where the item popularity distribution shifts between training and test sets). Previous work excludes Health and Movie from OOD comparisons because item popularity is not heavily skewed on these datasets, so we only consider Electronic and Gowalla here [23]. The datasets are split into 80% training and 20% test sets. Under the IID setting, we further split the training set into 90% and 10% validation sets for hyperparameter tuning. To avoid test set information leakage, no validation set is introduced under the OOD setting.

6.1.2 Metrics: Similar to Wu et al. and Zhang et al., NDCG@K and Recall@K are utilised for performance evaluation and comparison. The neighbourhood size is set to $K = 20$ as in recent works [21, 26]. However, similar results were obtained with different K choices.

6.1.3 Recommendation backbones: DSL and all baseline losses are evaluated on two representative backbones that are central to modern recommender systems, MF and LightGCN.

6.1.4 Baseline Losses: We compare against seven loss functions. These include the commonly adopted pairwise loss BPR, Partial AUC surrogate loss LLPAUC, NDCG surrogate losses, GuidedRec

and SL, and the more recent SL enhancements, AdvInfoNCE, BSL and PSL.

6.1.5 Hyperparameter Settings: The Adam optimiser is used for all training. The learning rate is varied in $\{10^{-1}, 10^{-2}, 10^{-3}\}$, except for BPR where 10^{-4} is used as recommended. The weight decay factor (wd) is searched over $\{0, 10^{-4}, 10^{-5}, 10^{-6}\}$. Batch size is set to 1024, and we train for 200 epochs. Following previous work [21, 23, 24], 1000 negative items per positive are uniformly sampled.

Interestingly, during our analysis it became apparent that the gains presented in some prior works appear inflated. To resolve this, we conduct an extensive grid search over the hyperparameter configurations of each technique, determining their individually optimal hyperparameters. See Appendix A for the optimal configurations used for our analysis.

We tune the loss-specific hyperparameters as follows.

- **BPR:** no additional hyperparameters [16].
- **LLPAUC:** following Shi et al. [19], we search α in {0.1, 0.3, 0.5, 0.7, 0.9} and β in $\{0.01, 0.1\}$.
- **Softmax Loss (SL):** we search τ in {0.005, 0.025, 0.05, 0.1, 0.2, 0.25} [23].
- **AdvInfoNCE:** we search τ same as SL, and fix the remaining hyperparameters to the defaults of Zhang et al. []. In particular, we set the negative weight to 64, perform adversarial updates every 5 epochs, and use an adversarial learning rate of 5×10^{-5} [26].
- **BSL:** the positive- and negative-term temperatures (τ_1, τ_2) are searched independently using the same candidate set as SL [21].
- **PSL:** we tune τ over the same grid as SL [23].
- **SL@K:** τ is set to best SL τ, $\tau_w \in \{0.5, 3.0\}$ with 0.25 search step and T_β in {5, 20} [24].
- For **DSL:** τ is searched same as SL and β and α are searched in $\{1.0, 1.5, 2.0, 2.5, 3.0\}$. We set $C_{ui} = 20$.

6.2 Performance Comparison

Table 1 demonstrates how the performance of DSL compares to SL and other baseline losses on MF and LightGCN backbones. These results illustrate substantial improvements with DSL over other baselines. When compared to SL, in particular, gains on Recall@20 and NDCG@20 can exceed 10%. On average, across all datasets, metrics and backbones, DSL outperforms SL by an average of **6.22%**. It also surpasses all other baselines, indicating that gains are not backbone or dataset-specific, but reflect a more effective objective.

These results align with our metric-driven analysis of DSL in Section 5.3. By applying pairwise weights and adjusting per-instance τ_{ui}, DSL allows gradients to concentrate on hard and highly similar negatives that are likely to incur the largest ΔNDCG@K and ΔRecall@K when swapped. While maintaining the log-sum-exp form, DSL reshapes competition within and across training examples, resulting in the consistent Top-K improvements observed in Table 1.

6.3 OOD Performance

Table 2 demonstrates performance under the OOD popularity shift setting. Since the trends are consistent across backbones, only the MF results are reported here for brevity. DSL exhibits high robustness to distributional shifts, as shown in the results. An average improvement of 9.31% over SL is achieved across the Electronic and Gowalla datasets. DSL outperforms all other baselines, too. We also note that gains are larger than on IID setting. For example, Recall@20 on Gowalla increases from +1.45% to +4.42% and on Electronic from +7.15% to +10.60%. Similar relative increases can also be seen with NDCG@20.

The pattern observed here is consistent with the KL-DRO analysis in Section 5.2. DSL still corresponds to a KL-regularized worst-case objective over negatives (as does SL), but the additional κ changes the robustness payoff, focusing adversarial learning on strong, highly substitutable competitors. Meanwhile, CA adapts per-example robustness level via τ_{ui} for ambiguous, shift-prone instances, which is relevant for this popularity shift evaluation. Thanks to the normalisation measures introduced to prevent global temperature drift, these OOD gains can be directly attributed to DSL's competition shaping.

6.4 Ablation Study

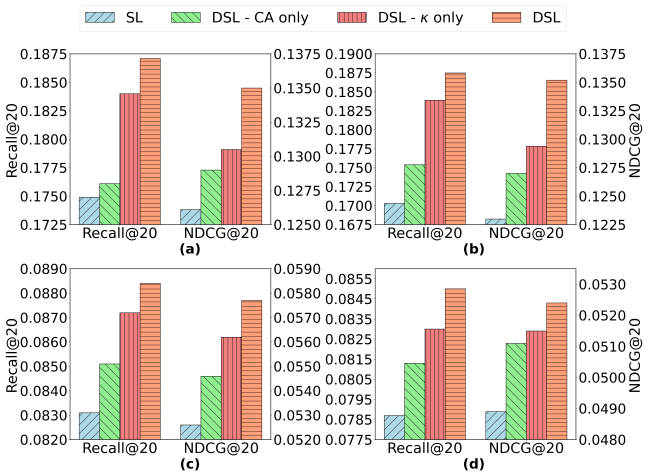

Figure 2: Health and Electronic Ablation

We conduct an ablation study to determine the κ and CA branches' contributions, separately, and how they influence the combined DSL objective. Figure 2 reports results on the Health and Electronic datasets on both MF and LightGCN backbones. The four variants compared include (i) **SL** with both components disabled, (ii) **DSL–CA only** with the per-instance temperature multiplier enabled only, (iii) **DSL–κ only** with only the per-negative κ modulation enabled, and (iv) **DSL** having both branches enabled. We can observe that enabling either component improves over the base SL, confirming that each branch contributes a useful signal beyond SL's standard log-sum-exponent optimisation. We note that **DSL–κ only** consistently yields bigger improvements than **DSL–CA only**, suggesting that per-negative reweighting is the stronger driver of gains. Finally, **full DSL** achieves the best results in all settings, highlighting the clear complementary role that the two branches play.

Backbone	Loss	Health		Electronic		Movie		Gowalla	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
MF	BPR	0.1240	0.0815	0.0406	0.0236	0.0428	0.0272	0.0554	0.0382
	GuidedRec	0.1478	0.1002	0.0645	0.0383	0.0612	0.0405	0.1129	0.0853
	LLPAUC	0.1627	0.1185	0.0829	0.0501	0.1264	0.0880	0.1611	0.1184
	SL	0.1749	0.1261	0.0825	0.0531	0.1285	0.0924	0.2069	0.1623
	AdvInfoNCE	0.1757	0.1282	0.0826	0.0527	0.0994	0.0721	0.2069	0.1625
	BSL	0.1749	0.1261	0.0829	0.0527	0.1004	0.0742	0.2068	0.1623
	PSL	0.1713	0.1270	0.0833	0.0537	0.1296	0.0935	0.2090	0.1645
	SL@K	0.1756	**0.1355**	**0.0893**	**0.0585**	0.1155	0.0832	**0.2128**	**0.1709**
	DSL (Ours)	**0.1871**	0.1350	0.0884	0.0577	**0.1336**	**0.0954**	0.2099	0.1705
	Imp.%	+6.98%	+7.06%	+7.15%	+8.66%	+3.97%	+3.25%	+1.45%	+5.05%
LightGCN	BPR	0.1177	0.0759	0.0297	0.0173	0.0488	0.0314	0.0433	0.0283
	GuidedRec	0.1358	0.0892	0.0614	0.0364	0.0594	0.0392	0.0766	0.0533
	LLPAUC	0.1658	0.1173	0.0823	0.0501	0.1263	0.0879	0.1551	0.1134
	SL	0.1703	0.1230	0.0774	0.0485	0.1295	0.0930	0.2006	0.1549
	AdvInfoNCE	0.1704	0.1217	0.0759	0.0478	0.0998	0.0738	0.2001	0.1544
	BSL	0.1703	0.1230	0.0774	0.0485	0.1023	0.0758	0.2005	0.1549
	PSL	0.1619	0.1184	0.0766	0.0483	0.1290	0.0940	0.2026	0.1570
	SL@K	0.1723	0.1311	0.0842	**0.0539**	0.1069	0.0812	0.2004	0.1564
	DSL (Ours)	**0.1875**	**0.1357**	**0.0850**	0.0535	**0.1328**	**0.0951**	**0.2081**	**0.1654**
	Imp.%	+10.10%	+10.33%	+9.82%	+10.31%	+2.56%	+2.26%	+3.74%	+6.78%

Table 1: IID performance (Recall@20 and NDCG@20). The best loss per dataset per metric is in bold, and the second best is underlined. Imp.% is computed against SL.

Backbone	Loss	Electronic		Gowalla	
		Recall@20	NDCG@20	Recall@20	NDCG@20
MF	BPR	0.0102	0.0057	0.0256	0.0174
	GuidedRec	0.0114	0.0063	0.0296	0.0214
	LLPAUC	0.0227	0.0138	0.0727	0.0522
	SL	0.0217	0.0129	0.0972	0.0705
	AdvInfoNCE	0.0191	0.0115	0.0649	0.0463
	BSL	0.0228	0.0130	0.0975	0.0709
	PSL	0.0237	0.0146	0.1005	0.0743
	SL@K	0.0211	0.0128	0.0980	0.0734
	DSL (Ours)	**0.0240**	**0.0150**	**0.1015**	**0.0747**
	Imp.%	+10.60%	+16.28%	+4.42%	+5.96%

Table 2: Out-of-distribution (OOD) performance (Recall@20 and NDCG@20). Best loss per metric is in bold, and second best is underlined. Imp.% is computed against SL (Softmax).

6.5 Tail and Head Performance Comparison

We next examine DSL under popularity bias using a popularity-filtered evaluation on head and tail items, defined as the top and bottom 20% by training interaction count. For each bucket, we restrict the test set to positives in that bucket and average over users with at least one such test positive. Figure 3 shows that DSL consistently improves more than SL, with most gains coming from tail rather than head items. This is desirable because rare items are harder to model and more sensitive to exposure effects. It is apparent that the majority of the performance gain for DSL comes from better recommendation of rare, less popular items, with smaller accuracy improvement on the dominant items. By reshaping the gradient signal, DSL emphasises informative, competitive alternatives while reducing the influence of abundant popular negatives.

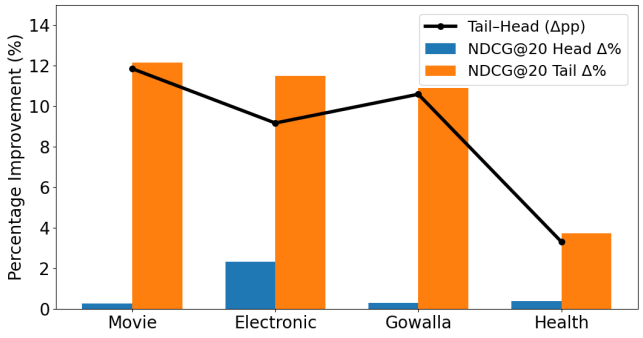

Figure 3: Percentage improvements for DSL over SL on NDCG@20 for Head and Tail item buckets, and the Tail–Head gap

7 Conclusion

In this work, we introduced Dual-scale Softmax Loss (DSL), rethinking SL from a competition-first perspective. By reshaping how negatives compete within each training instance, and adapting sharpness between instances, DSL delivered gains consistently over SL and other baseline losses for several backbones. These gains were more pronounced under OOD popularity shift, suggesting improved robustness to unknown distributional biases. Overall, DSL preserves the simplicity and scalability of SL, but makes its gradient budget more metric-relevant, aligning better with the competitors that actually matter for Top-K recommendation.

A Optimal Hyperparameters and Additional Results

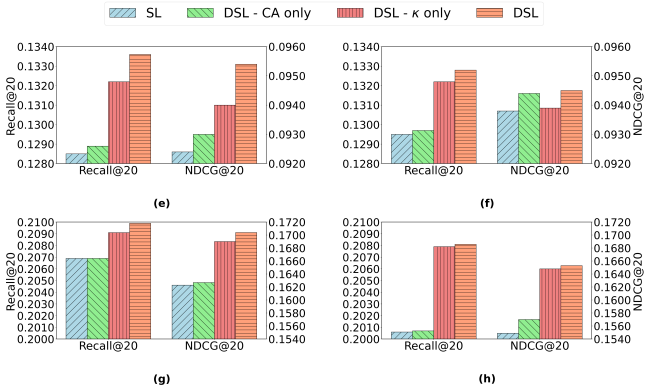

Figure 4: Movie and Gowalla Ablation

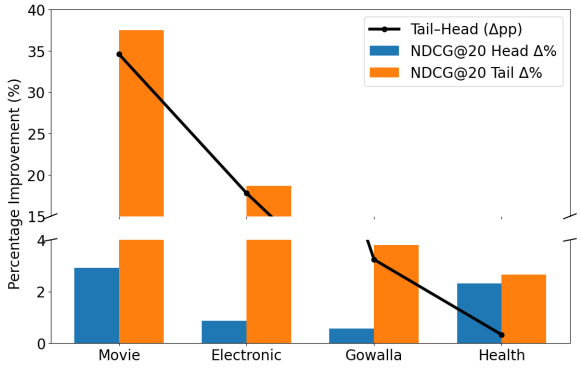

Figure 5: Percentage improvements for DSL over SL on NDCG@20 for Head and Tail item buckets on LightGCN

Model	Loss	lr	wd	others
MF	BPR	0.001	0.0001	–
	GuidedRec	0.01	0	–
	LLPAUC	0.1	0	{0.7, 0.01}
	SL	0.1	0	{0.25}
	AdvInfoNCE	0.1	0	{0.2}
	BSL	0.1	0	{0.2, 0.2}
	PSL	0.1	0	{0.1}
	SL@K	0.1	0	{0.25, 2.5, 5}
LightGCN	BPR	0.001	0.000001	–
	GuidedRec	0.01	0	–
	LLPAUC	0.1	0	{0.7, 0.1}
	SL	0.1	0	{0.2}
	AdvInfoNCE	0.1	0	{0.2}
	BSL	0.1	0	{0.05, 0.2}
	PSL	0.1	0	{0.1}
	SL@K	0.01	0	{0.2, 2.5, 20}

Model	Loss	lr	wd	others
MF	BPR	0.001	0.00001	–
	GuidedRec	0.01	0	–
	LLPAUC	0.1	0	{0.5, 0.01}
	SL	0.01	0	{0.2}
	AdvInfoNCE	0.1	0	{0.2}
	BSL	0.1	0	{0.5, 0.2}
	PSL	0.01	0	{0.1}
	SL@K	0.01	0	{0.2, 2.25, 20}
LightGCN	BPR	0.01	0.000001	–
	GuidedRec	0.01	0	–
	LLPAUC	0.1	0	{0.5, 0.01}
	SL	0.01	0	{0.2}
	AdvInfoNCE	0.01	0	{0.2}
	BSL	0.01	0	{0.2, 0.2}
	PSL	0.01	0	{0.1}
	SL@K	0.01	0	{0.2, 2.25, 5}

Table 3: IID Hyperparameters for Amazon-Health (left) and Amazon-Electronic (right). SL@K denotes SL@20.

Model	Loss	lr	wd	others
MF	BPR	0.001	0.000001	–
	GuidedRec	0.001	0	–
	LLPAUC	0.1	0	{0.7, 0.01}
	SL	0.1	0	{0.1}
	AdvInfoNCE	0.1	0	{0.1}
	BSL	0.1	0	{0.2, 0.1}
	PSL	0.1	0	{0.05}
	SL@K	0.01	0	{0.1, 1, 20}
LightGCN	BPR	0.001	0	–
	GuidedRec	0.001	0	–
	LLPAUC	0.1	0	{0.7, 0.01}
	SL	0.1	0	{0.1}
	AdvInfoNCE	0.1	0	{0.1}
	BSL	0.1	0	{0.05, 0.1}
	PSL	0.1	0	{0.05}
	SL@K	0.01	0	{0.1, 0.75, 5}

Model	Loss	lr	wd	others
MF	BPR	10^{-3}	10^{-6}	–
	GuidedRec	0.001	0	–
	LLPAUC	10^{-1}	0	{0.7, 0.01}
	AdvInfoNCE	10^{-1}	0	{0.05}
	SL	10^{-1}	0	{0.05}
	BSL	10^{-2}	0	{0.25, 0.05}
	PSL	10^{-1}	0	{0.05}
	SL@K	10^{-1}	0	{0.05, 0.75, 20}
LightGCN	BPR	10^{-3}	0	–
	GuidedRec	0.001	0	–
	LLPAUC	10^{-1}	0	{0.7, 0.01}
	AdvInfoNCE	10^{-1}	0	{0.05}
	SL	10^{-1}	0	{0.05}
	BSL	10^{-1}	0	{0.025, 0.05}
	PSL	10^{-1}	0	{0.05}
	SL@K	10^{-1}	0	{0.05, 0.75, 20}

Table 4: IID Hyperparameters for Gowalla (left) and Amazon-Movie (right). SL@K denotes SL@20.

References

[1] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[2] Sebastian Bruch, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2019. An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance. In *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*. 75–78.

[3] Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems* 19 (2006).

[4] Monroe D Donsker and SR Srinivasa Varadhan. 1975. Asymptotic evaluation of certain Markov process expectations for large time, II. *Communications on Pure and Applied Mathematics* 28, 2 (1975), 279–301.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[7] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*. Ieee, 263–272.

[8] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[9] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[11] Jiashuo Liu, Jiayun Wu, Tianyu Wang, Hao Zou, Bo Li, and Peng Cui. 2023. Geometry-calibrated dro: Combating over-pessimism with free energy implications. *arXiv preprint arXiv:2311.05054* (2023).

[12] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*. 5–12.

[13] Daniel McFadden. 1972. Conditional logit analysis of qualitative choice behavior. (1972).

[14] Hongseok Namkoong and John C Duchi. 2016. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems* 29 (2016).

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[16] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[17] Yuta Saito and Masahiro Nomura. 2019. Towards resolving propensity contradiction in offline recommender learning. *arXiv preprint arXiv:1910.07295* (2019).

[18] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[19] Wentao Shi, Chenxu Wang, Fuli Feng, Yang Zhang, Wenjie Wang, Junkang Wu, and Xiangnan He. 2024. Lower-left partial auc: An effective and efficient optimization metric for recommendation. In *Proceedings of the ACM Web Conference 2024*. 3253–3264.

[20] Martin J Wainwright, Michael I Jordan, et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1, 1–2 (2008), 1–305.

[21] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Jizhi Zhang, and Xiang Wang. 2024. Bsl: Understanding and improving softmax loss for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 816–830.

[22] Jiancan Wu, Xiang Wang, Xingyu Gao, Jiawei Chen, Hongcheng Fu, and Tianyu Qiu. 2024. On the effectiveness of sampled softmax loss for item recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 1–26.

[23] Weiqin Yang, Jiawei Chen, Xin Xin, Sheng Zhou, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. 2024. PSL: Rethinking and Improving Softmax Loss from Pairwise Perspective for Recommendation. *Advances in Neural Information Processing Systems* 37 (2024), 120974–121006.

[24] Weiqin Yang, Jiawei Chen, Shengjia Zhang, Peng Wu, Yuegang Sun, Yan Feng, Chun Chen, and Can Wang. 2025. Breaking the top-k barrier: Advancing top-k ranking metrics optimization in recommender systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 3542–3552.

[25] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. 2021. Doro: Distributional and outlier robust optimization. In *International Conference on Machine Learning*. PMLR, 12345–12355.

[26] An Zhang, Leheng Sheng, Zhibo Cai, Xiang Wang, and Tat-Seng Chua. 2023. Empowering collaborative filtering with principled adversarial contrastive loss. *Advances in Neural Information Processing Systems* 36 (2023), 6242–6266.