IS SOFTMAX LOSS ALL YOU NEED? A PRINCIPLED ANALYSIS OF SOFTMAX-FAMILY LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

The **Softmax Cross-Entropy** Loss is one of the most widely employed surrogate objectives for classification and ranking tasks. To elucidate its theoretical properties, the **Fenchel-Young** framework situates it as a canonical instance within a broad family of surrogates. Concurrently, another line of research has addressed scalability when the number of classes is exceedingly large, proposing alternatives that reduce the computational cost of the exact objective. Building on these two perspectives, we present a principled investigation of **Softmax-family** losses. We examine their consistency with task-specific classification and ranking metrics, characterize their gradient structures and score-space smoothness, and derive convergence bounds in a regularized fixed-feature setting. We also analyze approximate methods through a loss-level bias–variance decomposition and per-epoch computational costs, clarifying trade-offs between approximation fidelity and efficiency. Experiments across representative recommendation backbones reveal setting-dependent differences in predictive performance and training behavior, complementing the theoretical analysis. Together, these results establish a principled basis for comparing surrogate objectives and offer practical guidance for loss selection in large-scale machine learning.

1 INTRODUCTION

The **Softmax cross-entropy loss** is a standard training objective in modern machine learning, supporting applications such as machine translation, computer vision, and language modeling (Sutskever et al., 2014; He et al., 2016; Brown et al., 2020). Its appeal combines a probabilistic interpretation, a smooth score-space objective, and well-established consistency guarantees. In architectures such as Transformers (Vaswani et al., 2017), softmax cross-entropy remains a common choice for predicting among many output candidates. Yet its widespread use raises a practical question: *what underlines Softmax's effectiveness, and when should it be preferred over alternative objectives?*

One direction addresses the **statistical and geometric properties** of the objective. The Fenchel-Young (F-Y) framework (Blondel et al., 2020) recovers softmax cross-entropy from negative Shannon entropy and places it alongside **Sparsemax** (Martins & Astudillo, 2016) and α**-Entmax** (Peters et al., 2019). These alternatives produce sparse probability distributions and change how training signals are allocated across candidates. However, it remains underexplored whether these sparse alternatives preserve consistency properties, and how their support-selection mechanisms affect optimization.

A second direction addresses **computational scalability**. For large output spaces, evaluating the log-partition $\log Z(\boldsymbol{s}) = \log \sum_{j=1}^{C} \exp s_j$ can be costly. This phenomenon has motivated a wide range of approximate methods, such as Noise Contrastive Estimation (NCE) (Gutmann & Hyvärinen, 2010) and Sampled Softmax (SSM) (Jean et al., 2015), which trade exactness for tractability.

These developments motivate three questions: *Do alternatives preserve Softmax's consistency guarantees? How do they affect optimization? What computational savings do they offer?* We address these questions through a comparative analysis of consistency, gradient structure, approximation error, and training cost. We collectively refer to the objectives studied here as the **Softmax-family**, encompassing both F-Y losses and computational alternatives to Softmax.

This work develops a comparative analysis connecting these statistical, optimization, and computational perspectives. Our contributions are:

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.

Table 1: Full-score Softmax alternatives.

Property	**Softmax**	**Sparsemax**	**α-Entmax**	**Rankmax**
Top-k consistency	✓	✓	✓	×
P@k/DCG@k consistency	✓	✓	✓	×
Map ordering	SOP	WOP	WOP	Label-wise WOP
Convexity	✓	✓	✓	×
Smoothness Bound	$1/2$	1	1	Not smooth
Per-epoch head cost	$O(NCd)$	$O(NC(d+\log C))$	$O(NC(d+T))$	$O(NCd)$

Table 2: Softmax approximation alternatives.

Property	**SSM**	**NCE**	**HSM**	**RG**
Smoothness Bound	$1/2$	$1/4$	$1/2$	$1/C$
Bias	$-D_c/(2K)+O(K^{-2})$	B_{NCE}	$\log(p_c/H_c)$	$O(\|\boldsymbol{z}\|_2^3)$
Var	$D_c/K+O(K^{-2})$	$K\,\mathrm{Var}_Q(h_i)$	0	0
Per-epoch cost	$O(N(K+1)d)$	$O(N(K+1)d)$	$O(Nd\log C)$	$O(NCd)$

- **Task-specific consistency.** We analyze score-order preservation and consistency within the F-Y framework, distinguishing classification guarantees from metric-dependent ranking targets. A counterexample for the studied Rankmax objective shows that label-wise order preservation alone does not ensure Top-k or DCG consistency.
- **Training signals and optimization.** We characterize support-dependent gradients and local curvature to derive convergence bounds for a regularized fixed-feature setting. The analysis identifies a smaller worst-case smoothness bound for Softmax than its sparse alternatives.
- **Approximation and computational trade-offs.** We derive method-specific loss bias and variance with delta-method for sampled normalization. We further conclude the per-epoch computational complexity results to clarify trade-offs between accuracy and efficiency.
- **Empirical comparisons.** We conduct extensive experiments on recommender systems to examine accuracy, learning-rate sensitivity, negative sampling, and wall-clock efficiency across representative backbones. The results complement our theoretical analysis and highlight practical trade-offs among surrogate objectives.

Together, these results connect statistical guarantees, optimization properties, and computational costs, providing a principled basis for selecting surrogate objectives. Tables 1 and 2 summarize the main findings across the studied methods.

2 Foundations of Learning Frameworks

2.1 Preliminaries

We consider a supervised learning framework encompassing multi-class classification, multi-label prediction, and ranking over a fixed set of C candidates. Let $\mathcal{X}$ denote the input space and $[C]=\{1,\dots,C\}$ the candidate set. A label vector $\boldsymbol{y}\in\{0,1\}^C$ indicates the relevance of each candidate to an input $\boldsymbol{x}\in\mathcal{X}$, with $y_i=1$ if candidate i is relevant and $y_i=0$ otherwise. We assume that each example has at least one relevant candidate and write $|\boldsymbol{y}|=\sum_{i=1}^{C}y_i, \mathcal{Y}=\{\boldsymbol{y}\in\{0,1\}^C:|\boldsymbol{y}|\geq 1\}$. This representation covers the following settings:

- **Multi-class classification:** $\boldsymbol{y}$ is one-hot, so $|\boldsymbol{y}|=1$ and there is a unique relevant class.
- **Multi-label prediction and ranking**[1]: $\boldsymbol{y}$ may contain multiple relevant candidates, whose indices form the support $\mathrm{supp}(\boldsymbol{y})=\{i\in[C]:y_i=1\}$.

[1]We focus on binary relevance labels, as commonly used in implicit-feedback recommendation and retrieval.

Training examples $(\boldsymbol{x}, \boldsymbol{y})$ are drawn i.i.d. from an unknown distribution $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$. For a fixed input $\boldsymbol{x}$, let $\nu_{\boldsymbol{x}} = \mathbb{P}(\boldsymbol{y} \mid \boldsymbol{x})$ denote the conditional distribution over label vectors, and define the marginal relevance probabilities $r_i(\boldsymbol{x}) = \mathbb{E}[y_i \mid \boldsymbol{x}], i \in [C]$. In the multi-class setting, these marginals form the class-posterior vector $\boldsymbol{\eta}(\boldsymbol{x}) = (\mathbb{P}(y_i = 1 \mid \boldsymbol{x}))_{i=1}^{C}$. In the multi-label setting, however, $\sum_i r_i(\boldsymbol{x}) = \mathbb{E}[|\boldsymbol{y}| \mid \boldsymbol{x}]$ need not equal one; the marginal vector is therefore distinct from a categorical probability distribution and from the full conditional law $\nu_{\boldsymbol{x}}$.

A parameterized model $f : \mathcal{X} \to \mathbb{R}^C$ produces scores (logits) $\boldsymbol{s} = f(\boldsymbol{x}; \theta)$. For losses equipped with a probability mapping, these scores are mapped to $\hat{\boldsymbol{p}}(\boldsymbol{s}) \in \Delta^C$, where

$$\Delta^C = \left\{ \boldsymbol{p} \in \mathbb{R}^C : p_i \geq 0 \text{ for all } i,\ \sum_{i=1}^{C} p_i = 1 \right\}.$$

In multi-label problems, such a simplex-valued prediction represents a normalized target specified by the surrogate. The evaluation metrics below act on the scores $\boldsymbol{s}$. Throughout, $\pi_{\boldsymbol{s}}$ denotes the permutation that sorts these scores in descending order, with ties broken by a fixed ordering of candidate indices, independently of the labels. For $1 \leq k \leq C$, define $\mathrm{Top}_k(\boldsymbol{s}) = \{\pi_{\boldsymbol{s}}(1), \ldots, \pi_{\boldsymbol{s}}(k)\}$.

2.2 Evaluation Metrics and Consistency

Classification and ranking metrics depend on discrete predictions, making gradient-based optimization difficult. We therefore utilize differentiable surrogates and assess their alignment with target metrics through conditional risks. For a metric M to be maximized, we use the target loss $\ell_M = 1 - M$.

Definition 1 (Conditional Bayes-consistency)**.** Let $\ell : \mathcal{Y} \times \mathbb{R}^C \to \mathbb{R}$ be a target loss and $\mathcal{L} : \mathcal{Y} \times \mathbb{R}^C \to \mathbb{R}$ a surrogate loss. For a conditional label distribution ν, define

$$\mathcal{R}_\ell(\nu, \boldsymbol{s}) = \mathbb{E}_{\boldsymbol{y} \sim \nu}[\ell(\boldsymbol{y}, \boldsymbol{s})], \quad \mathcal{R}_{\mathcal{L}}(\nu, \boldsymbol{s}) = \mathbb{E}_{\boldsymbol{y} \sim \nu}[\mathcal{L}(\boldsymbol{y}, \boldsymbol{s})]. \tag{1}$$

Their **optimal conditional risks** are

$$\mathcal{R}^*_\ell(\nu) = \inf_{\boldsymbol{s} \in \mathbb{R}^C} \mathcal{R}_\ell(\nu, \boldsymbol{s}), \quad \mathcal{R}^*_{\mathcal{L}}(\nu) = \inf_{\boldsymbol{s} \in \mathbb{R}^C} \mathcal{R}_{\mathcal{L}}(\nu, \boldsymbol{s}). \tag{2}$$

Assume these optimal risks are finite. The surrogate $\mathcal{L}$ is conditionally **Bayes-consistent** w.r.t. ℓ if, for every conditional label distribution ν in the task under consideration and every sequence $\{\boldsymbol{s}_n\}_{n=1}^{\infty}$,

$$\mathcal{R}_{\mathcal{L}}(\nu, \boldsymbol{s}_n) \xrightarrow[n \to \infty]{} \mathcal{R}^*_{\mathcal{L}}(\nu) \quad \Longrightarrow \quad \mathcal{R}_\ell(\nu, \boldsymbol{s}_n) \xrightarrow[n \to \infty]{} \mathcal{R}^*_\ell(\nu). \tag{3}$$

For multi-class classification, the basic target is the **Top-1 error**.

Definition 2 (Top-1 error)**.** Let y^* denote the unique index with $y_{y^*} = 1$. Under the fixed tie-breaking rule, the Top-1 error is $\ell_{\mathrm{Top}-1}(\boldsymbol{y}, \boldsymbol{s}) = \mathbf{1}\{\pi_{\boldsymbol{s}}(1) \neq y^*\}$.

Consistency w.r.t. Top-1 error captures the conditional requirement underlying classification calibration (Zhang, 2004; Bartlett et al., 2006). When more than one prediction is allowed, the corresponding target is transformed into the **Top-k error**.

Definition 3 (Top-k error)**.** For label $\boldsymbol{y}$ and $1 \leq k \leq C$, $\ell_{\mathrm{Top}-k}(\boldsymbol{y}, \boldsymbol{s}) = \mathbf{1}\{y^* \notin \mathrm{Top}_k(\boldsymbol{s})\}$.

Top-k consistency have been studied by Lapin et al. (2016); Yang & Koyejo (2020). Multi-class surrogates can also be used in reductions for multi-label prediction, but the target metric depends on how labels and examples are weighted (Menon et al., 2019). For multi-label prediction, we consider Precision@k (P@k) and Recall@k (R@k), which measure the fractions of relevant candidates among the top-k predictions and of relevant candidates retrieved. For ranking, DCG@k weights relevance by position, while NDCG@k (N@k) normalizes DCG@k by its maximum attainable value. Formal definitions and their corresponding Bayes rankings are provided in Appendix A.

We next introduce the softmax cross-entropy loss and examine its alignment with these targets.

2.3 Softmax Loss

Definition 4 (Softmax Mapping and Softmax Loss)**.** The **softmax mapping** $\hat{\boldsymbol{p}}_{\mathrm{SM}} : \mathbb{R}^C \to \Delta^C$ is defined element-wise as

$$\hat{p}_{\mathrm{SM},i}(\boldsymbol{s}) = \frac{\exp(s_i)}{\sum_{j=1}^{C} \exp(s_j)}. \tag{4}$$

For a non-empty binary label vector $\boldsymbol{y}$, we use the summed **softmax cross-entropy loss**

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{y}, \boldsymbol{s}) = -\sum_{i=1}^{C} y_i \log \hat{p}_{\mathrm{SM},i}(\boldsymbol{s}) = |\boldsymbol{y}| \log \sum_{j=1}^{C} \exp(s_j) - \langle \boldsymbol{s}, \boldsymbol{y} \rangle. \tag{5}$$

For one-hot labels, this loss is the categorical negative log-likelihood, and minimizing its empirical average corresponds to MLE. It is consistent with Top-k error for all k in multi-class setting (Yang & Koyejo, 2020). For multi-label scenario, Eq. 5 sums the contributions of all relevant candidates. Its conditional probability target is $p_i^*(\boldsymbol{x}) = r_i(\boldsymbol{x}) / \sum_{j=1}^{C} r_j(\boldsymbol{x})$, where $r_i(\boldsymbol{x}) = \mathbb{E}[y_i \mid \boldsymbol{x}]$, yielding consistency with Precision@k (Menon et al., 2019). It also yields Bayes-optimal DCG@k rankings, following the characterization of Cossock & Zhang (2008). Dividing the loss by $|\boldsymbol{y}|$ instead changes the target to $\mathbb{E}[\boldsymbol{y}/|\boldsymbol{y}| \mid \boldsymbol{x}]$, which corresponds to Recall@k (Menon et al., 2019).

2.4 Fenchel–Young Framework

The Fenchel-Young (F-Y) framework (Blondel et al., 2020) constructs convex surrogate losses and associated prediction mappings from a convex potential function Ω.

Definition 5 (Fenchel Conjugate)**.** Let $\Omega : \Delta^C \to \mathbb{R}$ be a lower semi-continuous convex potential function. Its conjugate is defined as $\Omega^*(\boldsymbol{s}) = \sup_{\boldsymbol{p} \in \Delta^C} \{\langle \boldsymbol{s}, \boldsymbol{p} \rangle - \Omega(\boldsymbol{p})\}, \boldsymbol{s} \in \mathbb{R}^C$.

Definition 6 (Fenchel–Young Loss)**.** For a target $\boldsymbol{t} \in \Delta^C$, the F–Y loss is

$$\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = \Omega^*(\boldsymbol{s}) + \Omega(\boldsymbol{t}) - \langle \boldsymbol{s}, \boldsymbol{t} \rangle. \tag{6}$$

For instances considered below, Ω is strongly convex on Δ^C. The prediction is defined by

$$\hat{\boldsymbol{p}}_\Omega(\boldsymbol{s}) = \operatorname*{argmax}_{\boldsymbol{p} \in \Delta^C} \{\langle \boldsymbol{s}, \boldsymbol{p} \rangle - \Omega(\boldsymbol{p})\} = \nabla \Omega^*(\boldsymbol{s}), \tag{7}$$

and the loss gradient is $\nabla_{\boldsymbol{s}} \mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = \hat{\boldsymbol{p}}_\Omega(\boldsymbol{s}) - \boldsymbol{t}$.

The F-Y loss is non-negative and convex in $\boldsymbol{s}$. Strong convexity of Ω ensures a Lipschitz-continuous loss gradient, whereas consistency with a target metric requires additional analysis of the potential and the prediction rule. To retain the multi-label weighting in Eq. 5, we sum the one-hot F-Y losses over relevant candidates. Writing $\boldsymbol{e}_i$ for the i-th standard basis vector, define

$$\mathcal{L}_\Omega^{\mathrm{sum}}(\boldsymbol{y}, \boldsymbol{s}) = \sum_{i=1}^{C} y_i \mathcal{L}_\Omega(\boldsymbol{e}_i, \boldsymbol{s}), \quad \nabla_{\boldsymbol{s}} \mathcal{L}_\Omega^{\mathrm{sum}}(\boldsymbol{y}, \boldsymbol{s}) = |\boldsymbol{y}| \hat{\boldsymbol{p}}_\Omega(\boldsymbol{s}) - \boldsymbol{y}. \tag{8}$$

For negative Shannon entropy, $\Omega(\boldsymbol{p}) = \sum_i p_i \log p_i$ with $0 \log 0 = 0$, the conjugate is $\Omega^*(\boldsymbol{s}) = \log \sum_i \exp(s_i)$ and the prediction map is softmax. The summed F-Y loss recovers Eq. 5 exactly.

Other examples. Choosing $\Omega(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p}\|_2^2$ yields **Sparsemax** (Martins & Astudillo, 2016). More generally, the negative Tsallis entropy (Tsallis, 1988) with $\Omega_\alpha(\boldsymbol{p}) = \frac{1}{\alpha(\alpha-1)} \sum_{i=1}^{C} (p_i^\alpha - p_i), 1 < \alpha \leq 2$, generates the α**-Entmax** family (Peters et al., 2019). It recovers softmax as $\alpha \to 1$ and sparsemax at $\alpha = 2$, and can produce sparse predictions for $\alpha > 1$.

The conditional risk of F-Y losses is determined by a mean target, with example weighting specifying which mean is recovered. The risk decomposition and probability-error bounds are given in Appendix B. The connection to Bregman divergences is presented in Appendix A.2.

2.5 Computational Bottleneck and Approximation Methods

Evaluating the softmax log-partition $\log Z(\boldsymbol{s}) = \log \sum_{j=1}^{C} \exp s_j$ requires a sum over all C candidates. Common approaches to reducing training cost include sampled softmax (**SSM**) (Jean et al., 2015), noise-contrastive estimation (**NCE**) (Gutmann & Hyvärinen, 2010), hierarchical softmax (**HSM**) (Morin & Bengio, 2005), and Taylor-based surrogates (Pu et al., 2024). Appendix A.3 specifies their formulations and assumptions for the comparisons that follow.

3 MAIN RESULTS

3.1 CONSISTENCY ANALYSIS OF FENCHEL-YOUNG LOSSES

For **Softmax** loss, minimizing the conditional risk recovers the class posterior for one-hot labels, with boundary targets approached in the limit. Since softmax strictly preserves score order, this recovery yields the following guarantees (Yang & Koyejo, 2020).

Proposition 7. *Softmax Cross-Entropy is **Top-k calibrated** for one-hot labels for every $1 \leq k \leq C$, including **classification calibration** at $k = 1$. For non-empty binary labels, its summed extension is **Bayes-consistent** with* Precision@k *and* DCG@k.

For **Sparsemax** and **α-Entmax** which arise from the squared ℓ_2 and negative Tsallis entropy respectively. These potentials permit boundary predictions, whose loss forms do not equal their associated Bregman divergence. However, the direct risk decomposition applies as well. The following proposition shows that, as the conditional surrogate risk approaches its infimum, the predicted probabilities converge to the target, eventually recovering every strict target ordering.

Proposition 8. *Sparsemax and α-Entmax losses with $1 < \alpha \leq 2$ are **Top-k calibrated** for one-hot labels for every $1 \leq k \leq C$. Their summed extensions are **Bayes-consistent** with* Precision@k *and* DCG@k *for non-empty binary labels.*

Proofs of both propositions are in Appendix B. The corresponding weighted losses for Recall@k and NDCG@k are established in Corollary 27.

3.2 ORDER PRESERVATION AND TRAINING SIGNALS

Although **Softmax** Loss and sparse F-Y losses share the consistency guarantees established above, their prediction maps retain different amounts of score-order information. Sparse maps can merge distinct below-threshold scores into zero probabilities, whereas softmax preserves their relative order.

Definition 9 (Order preservation)**.** For a prediction map $\boldsymbol{q} : \mathbb{R}^C \to \Delta^C$,

- **Strict order preservation (SOP)**: $s_i > s_j \iff q_i(\boldsymbol{s}) > q_j(\boldsymbol{s})$ for all $i \neq j$.
- **Weak order preservation (WOP)**: $s_i > s_j \Rightarrow q_i(\boldsymbol{s}) \geq q_j(\boldsymbol{s})$ for all $i \neq j$.
- **Tie-consistency**: $s_i = s_j \Rightarrow q_i(\boldsymbol{s}) = q_j(\boldsymbol{s})$ for all $i \neq j$.

Hence **WOP** includes **SOP**. A weak but not strict order preservation denotes a map that can merge a strict score inequality into a probability tie.

Proposition 10 (Order properties)**.** *Softmax is **SOP**. Sparsemax and α-Entmax are **WOP** and **tie-consistent**, but are not SOP when $C \geq 3$.*

The following theorem further demonstrate that **SOP** is not necessary for consistency, while the sufficient conditions extend the preceding consistency results beyond specific mapping functions.

Theorem 11 (WOP and consistency)**.** *Let Ω be continuous and μ-strongly convex on Δ^C in Euclidean norm with $\mu > 0$. If its F–Y prediction map is **WOP** and **tie-consistent**, then for every $1 \leq k \leq C$, its one-hot loss is **Top-k calibrated**. Its summed extension is **Bayes-consistent** with* Precision@k *and* DCG@k *for nonempty binary labels under the score-based ranking rule in Section 2.*

Proofs are provided in Appendices C.1 and C.2. Nevertheless, consistency does not characterize how different losses distribute training signals across candidates. At finite scores, the **Softmax** Jacobian has rank $C - 1$, with only the common-shift direction in its null space. In contrast, on a differentiable region with fixed positive support of size h, the **Sparsemax** and **α-Entmax** Jacobians have rank $h - 1$ (Martins & Astudillo, 2016; Peters et al., 2019). Whenever $h < C$, sparse losses introduce additional directions with zero local curvature. These directions reflect insensitivity of the prediction map to score changes. The derivations are in Appendix C.3.

A more direct distinction concerns which negative classes receive score-level updates.

Proposition 12 (Negative-class gradient coverage)**.** *For the summed F–Y loss with non-empty binary labels, let $\boldsymbol{q} = \hat{\boldsymbol{p}}_\Omega(\boldsymbol{s})$. Then $\frac{\partial \mathcal{L}_\Omega^{\mathrm{sum}}(\boldsymbol{y}, \boldsymbol{s})}{\partial s_j} = |\boldsymbol{y}| q_j - y_j$. For a negative class, this derivative is strictly*

positive under Softmax, whereas under Sparsemax or Entmax it is positive if and only if the class belongs to the prediction support. For a positive class outside the support, the derivative remains -1.

The proof is given in Appendix C.4. Proposition 12 identifies a concrete difference between dense and sparse training: softmax retains a direct suppression signal for every negative class, while sparse losses apply such signals selectively according to the current support. Sparsity therefore changes not only the output distribution but also the allocation of negative-class updates. This difference is consequential even though both families remain consistent.

Besides, **WOP** also permits distinct scores to be mapped to identical probabilities. If these probabilities are used for ranking, the resulting ties can obscure within-block score order. The following theorem quantifies the effect of uniformly random tie-breaking, providing an auxiliary interpretation of this difference between weak and strict order preservation.

Theorem 13 (Expected DCG under random ties)**.** *Fix a non-empty binary label vector* $\boldsymbol{y}$ *and a tied block occupying positions* $z+1, \ldots, z+h$*, containing* r *relevant candidates. For* $1 \le k \le C$*, define* $d_a = 1/\log_2(a+1)$ *for* $a \le k$ *and* $d_a = 0$ *otherwise. Under uniformly random permutations within the block, the expected* DCG@k *deficit relative to the optimal arrangement of that block is*

$$\Delta^{(k)}_{\text{block}} = \sum_{a=1}^{r} d_{z+a} - \frac{r}{h}\sum_{a=1}^{h} d_{z+a} \ge 0. \tag{9}$$

For full-list DCG*, the deficit is strictly positive whenever* $0 < r < h$*. Dividing by* IDCG@$k(\boldsymbol{y})$ *gives the corresponding expected* NDCG@k *deficit.*

The proof is provided in Appendix C.5. Theorem 13 makes precise one consequence of weak rather than strict order preservation: a mixed-relevance probability tie creates ambiguity whose random resolution can be suboptimal within the block.

3.3 A Sparse Alternative: Rankmax

Rankmax (Kong et al., 2020) provides a complementary, label-adaptive approach to sparsity. Unlike **Sparsemax** and **Entmax**, **Rankmax** anchors its threshold to the score of the observed positive class. We consider its standard-simplex, unit-margin form. For a positive class index c, define

$$q_i^{\text{rm}}(c, \boldsymbol{s}) = \frac{[s_i - s_c + 1]_+}{Z_c(\boldsymbol{s})}, \quad Z_c(\boldsymbol{s}) = \sum_{j=1}^{C} [s_j - s_c + 1]_+, \tag{10}$$

where $[a]_+ = \max\{a, 0\}$.

Proposition 14 (Order preservation and positive-class inclusion)**.** *For each fixed* c*, the map in Eq. 10 is* ***WOP*** *and* ***tie-consistent****, and* $q_c^{\text{rm}} > 0$*. Every class with* $s_j \ge s_c$ *is included in its support, whereas classes with* $s_j \le s_c - 1$ *are excluded. For* $C \ge 3$*, the map is not SOP.*

The proof is in Appendix D.1. However, since **Rankmax** depends on the observed class c and falls outside the fixed-potential F-Y framework of Theorem 11, these properties do not ensure Bayes-consistency. More strongly, its standard-simplex cross-entropy objective,

$$\ell_{\text{rm}}(c, \boldsymbol{s}) = -\log q_c^{\text{rm}}(c, \boldsymbol{s}) = \log Z_c(\boldsymbol{s}), \tag{11}$$

is not generally consistent with evaluation metrics in our discussion (A counterexample in Appendix D.2), which shows that label-wise **WOP** and **tie-consistency** do not suffice for consistency. This limitation concerns statistical consistency, rather than the structure of individual training updates. Rankmax's label-adaptive support still induces a distinctive update rule, as formalized below.

Proposition 15 (Margin-selective updates)**.** *For a positive class* c*, define the active competitor set* $\mathcal{A}_c = \{j \ne c : s_j > s_c - 1\}$ *and its size* $a_c = |\mathcal{A}_c|$*. At scores satisfying* $s_j \ne s_c - 1$ *for every* $j \ne c$*, the loss gradient is*

$$\nabla_{\boldsymbol{s}} \ell_{\text{rm}}(c, \boldsymbol{s}) = \frac{1}{Z_c(\boldsymbol{s})} \left(\sum_{j \in \mathcal{A}_c} \boldsymbol{e}_j - a_c \boldsymbol{e}_c \right). \tag{12}$$

The loss is zero whenever $s_c - s_j \ge 1$ *for every* $j \ne c$*, and its gradient is zero whenever all these inequalities are strict.*

The proof is in Appendix D.3. Proposition 15 highlights Rankmax's targeted allocation of training signals: its label-adaptive support retains every competitor scoring at least as highly as the positive class, excludes competitors already beyond the unit margin, and assigns equal gradient weights to active competitors. Rankmax therefore offers an explicit mechanism for selecting and balancing competing-class updates, providing a potential practical benefit despite its lack of general ranking-consistency guarantees.

3.4 Smoothness and Convergence Guarantees

The Jacobian analysis also characterizes the score-level smoothness of the standard F-Y losses. For the summed objective, its Hessian is $|\boldsymbol{y}|J_\Omega$ wherever it exists. Under the unit-scale parameterization, Euclidean Lipschitz bounds for the prediction maps are

$$B_{\mathrm{SM}} = \tfrac{1}{2}, \quad B_{\mathrm{Sparsemax}} = 1, \quad B_{\alpha\text{-Entmax}} = 1 \quad (1 < \alpha \le 2). \tag{13}$$

The first two are sharp, while the latter is a uniform upper bound. Proofs appear in Appendix E.1. Since convexity in scores does not imply convexity in arbitrary model parameters, we state a parameter-level guarantee for a frozen-feature affine head, rather than an end-to-end neural network.

Theorem 16 (Convergence of a regularized affine head)**.** *Let $\boldsymbol{s}_n(\theta) = A_n\theta + \boldsymbol{b}_n$, where A_n and $\boldsymbol{b}_n$ are fixed, and consider*

$$F_\Omega(\theta) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_\Omega^{\mathrm{sum}}(\boldsymbol{y}_n, A_n\theta + \boldsymbol{b}_n) + \frac{\gamma}{2}\|\theta\|_2^2, \quad \gamma > 0. \tag{14}$$

For Softmax, Sparsemax, or α-Entmax as above, define $\Lambda = \left\| \frac{1}{N}\sum_{n=1}^{N} |\boldsymbol{y}_n| A_n^\top A_n \right\|_2$, $\overline{L}_\Omega = \gamma + B_\Omega\Lambda$. Then F_Ω is γ-strongly convex and $\overline{L}_\Omega$-smooth. Full-batch gradient descent with $\eta = 2/(\overline{L}_\Omega + \gamma)$ satisfies

$$\|\theta_t - \theta^*\|_2 \le \overline{\rho}_\Omega^t \|\theta_0 - \theta^*\|_2, \quad \overline{\rho}_\Omega = \frac{B_\Omega\Lambda}{B_\Omega\Lambda + 2\gamma}. \tag{15}$$

The proof follows from the curvature bounds and standard smooth strongly convex optimization, and is given in Appendix E.2. For matched affine heads, label weighting, and regularization, these bounds give, when $\Lambda > 0$,

$$\overline{\rho}_{\mathrm{SM}} < \overline{\rho}_{\alpha\text{-Entmax}} = \overline{\rho}_{\mathrm{Sparsemax}}. \tag{16}$$

These worst-case bounds give softmax a wider certified descent interval than Entmax and Sparsemax. SSM, NCE, and HSM require separate coordinate and scaling analyses, while Rankmax's non-convex cross-entropy falls outside the theorem. Details are deferred to Appendix E.

3.5 Bias-Variance Decomposition for Softmax Approximations

Softmax approximations reduce computation through sampling or adjusting training objective. We compare them at the loss level using exact softmax cross-entropy as a common reference.

Let $p_i = \exp s_i / Z$, $Z = \sum_j \exp s_j$, and $\ell_* = -\log p_c$. For a given method A with loss $\widetilde{\ell}_A$, let ω denote approximation randomness. Define $\mathrm{Bias}_A = \mathbb{E}_\omega[\widetilde{\ell}_A] - \ell_*, \mathrm{Var}_A = \mathrm{Var}_\omega(\widetilde{\ell}_A)$. Bias measures systematic loss discrepancy, while variance measures fluctuations across approximation draws. Whenever the second moment exists,

$$\mathbb{E}_\omega[(\widetilde{\ell}_A - \ell_*)^2] = \mathrm{Bias}_A^2 + \mathrm{Var}_A\,. \tag{17}$$

Table 3 summarizes exact identities and delta-method expansions. For importance-corrected SSM, the leading bias and variance are $-D_c/(2K)$ and D_c/K, linking approximation error to sample count and proposal mismatch. Deterministic alternatives have zero sampling variance, with their discrepancy captured entirely by bias. These results complement the training-cost analysis by quantifying fidelity to the reference loss. Method definitions and derivations are provided in Appendices A.3 and F.

Table 3: Bias-variance Decomposition Results. K is the sample size of i.i.d. proposal draws. 'Status' with Large-K entries omit $O(K^{-2})$ remainders. Notation descriptions are deferred to Appendix F.3

Method	Bias_A	Var_A	**Status**
Softmax	0	0	Exact
SSM-Simple	$\log(M_K/Z) - K\sigma_c^2/(2M_K^2)$	$K\sigma_c^2/M_K^2$	Large K
SSM	$-D_c/(2K)$	D_c/K	Large K
NCE	$a_c + K\mathbb{E}_Q[h_i] + \log p_c$	$K\,\mathrm{Var}_Q(h_i)$	Exact
HSM	$\log(p_c/H_c)$	0	Exact
Quadratic RG$^\times$	$\Psi_{\mathrm{quad}} - \log Z = O(\|\boldsymbol{z}\|_2^3)$	0	Local at $\boldsymbol{z} = 0$

Table 4: Per-epoch computation Complexity.

Softmax / Rankmax	Sparsemax /1.5-Entmax(sorting)	α-Entmax(bisection)	SSM/NCE	HSM
$O(NCd)$	$O(NC(d+\log C))$	$O(NC(d+T))$	$O(N(K+1)d)$	$O(Nd\log C)$

3.6 Training Complexity Analysis

Table 4 reports per-epoch forward/backward costs for a linear output head, excluding the backbone, optimizer steps, and preprocessing. Here N is the number of single-positive training examples, d the feature dimension, C the number of classes, K the number of sampled negatives, and T the number of bisection iterations. Sampling assumes $O(1)$ cost per draw and evaluates only selected scores; HSM uses a balanced binary tree, and Rankmax uses the standard-simplex formulation. Sparse outputs alone do not avoid computing full logits. RG-ALS is excluded because its per-sweep cost depends on the linear systems and solver. Algorithmic discussions are given in Appendix F.4.

4 Experiments

We conduct extensive experiments on a typical recommendation scenario to examine how the choice of different surrogate losses within the Softmax-family affects predictive accuracy and training efficiency. Recommendation requires ranking relevant items from a large candidate set, making it a natural setting for studying both statistical and computational trade-offs. We evaluate three representative backbones, **MF** (Rendle et al., 2009), **SASRec** (Kang & McAuley, 2018) and graph-based **LightGCN** (He et al., 2020) on datasets including **MovieLens-1M**, **Amazon-Electronics**, and **Gowalla**. We compare **Softmax**, **Sparsemax**, 1.5**-Entmax**, **Rankmax**, **SSM**, and **NCE** on all backbones, while **HSM** and **RG** are evaluated solely on **MF** due to their implementations. Dataset preparation, model settings, and training details are provided in Appendix G.1. In all, the experiments mainly focus on the following three questions:

Q1: Accuracy and learning-rate sensitivity. How do surrogate losses compare across backbones and learning rates? We report P@20, R@20 and N@20 over a shared learning-rate grid (Appendix G.2). The results show a clear interaction between surrogate choice and backbone. **Sparsemax** and **Entmax** can match or outperform **Softmax** with **MF**, but achieve lower best-reported metrics on **SASRec**. **Rankmax** is particularly competitive with **MF** and **SASRec**: it attains the highest reported N@20 among the compared methods on **ML-1M** and **Gowalla** with **MF**, and on all datasets with **SASRec**. However, this advantage does not carry over to **LightGCN**, where **Softmax** and **SSM** perform better. Besides, learning-rate preferences also vary across settings, in which our smoothness analysis provides a useful perspective on this sensitivity.

Q2: Sample count and proposal. How do the number and selection of negatives affect **SSM** and **NCE**? We vary $K \in \{5, 10, 50, 100\}$ and compare uniform sampling with dynamic negative sampling (DNS)(Appendix G.3). **SSM**'s N@20 increases with K in all six **MF/SASRec** dataset combinations, while **NCE**'s response on **MF** change relatively little. In comparison, **DNS** improves both losses over uniform sampling in the displayed comparisons. These results align with Table 3, which identifies both sample count and proposal as important factors.

Q3: Computational efficiency. How does progress per epoch translate into wall-clock efficiency? We compare N@20 against both epochs and elapsed time for **MF** and **SASRec**(Appendix G.4). The two views assess complementary aspects of training: predictive progress per data pass and the computational cost of achieving it. Our complexity analysis helps interpret this comparison: sparse heads require threshold computation after evaluating full scores, whereas sampled methods can reduce the number of evaluated candidates. Practical loss selection therefore considers achieved accuracy together with elapsed training time, rather than epoch count alone.

Supplementary matrix visualizations accompany the derivative analysis (Appendix G.5). Overall, the experiments identify Softmax as a competitive reference, show that sparse alternatives have backbone-dependent strengths, and demonstrate the practical importance of negative sampling.

5 Related Work

Building on Section 2, we highlight two directions most relevant to our study.

Statistical foundations. Surrogate consistency has been extensively studied for classification (Zhang, 2004; Bartlett et al., 2006) including Top-k prediction (Lapin et al., 2016; Yang & Koyejo, 2020). For ranking, existing results establish metric-specific guarantees and limitations, emphasizing the role of the target metric and label normalization (Cossock & Zhang, 2008; Ravikumar et al., 2011; Calauzènes et al., 2012). The F-Y framework (Blondel et al., 2020) unifies regularized prediction maps and convex losses, encompassing Softmax, Sparsemax (Martins & Astudillo, 2016), and α-Entmax (Peters et al., 2019). Rankmax (Kong et al., 2020) provides a related adaptive alternative, whose label-dependent formulation considered here requires separate analysis. Sparse prediction methods have also been explored in structured prediction and neural sequence modeling (Niculae et al., 2018; Correia et al., 2019). Our study connects these statistical foundations to differences in training signals and optimization, while keeping guarantees tied to the relevant objective.

Computational scalability. Large output spaces motivate methods that reduce or avoid full softmax normalization. Sampled softmax (Jean et al., 2015) estimates normalization using sampled candidates, whereas NCE (Gutmann & Hyvärinen, 2010) replaces likelihood training with data–noise discrimination. Their computational savings come with objective- and sampling-dependent bias and variance. Adaptive kernel-based proposals further improve the sampling distribution (Blanc & Rendle, 2018). Deterministic alternatives include hierarchical factorization (Morin & Bengio, 2005) and Taylor-based replacements of the exponential (Banerjee et al., 2021). These approaches change different aspects of prediction and training; we therefore compare their loss discrepancies and computational costs without assuming that all share a fixed F-Y potential.

6 Conclusion

This work provides a comparative analysis of Softmax-family objectives across statistical consistency, optimization, and computational cost. Within the Fenchel-Young framework, we establish consistency guarantees and characterize how surrogate design shapes training behavior. Our curvature, bias-variance, and complexity analyses bring these perspectives together, clarifying the strengths and trade-offs of Softmax and its alternatives. Supported by recommendation experiments, these findings provide a principled foundation for selecting surrogate losses that balance predictive accuracy and computational efficiency.

For practical guidance, our theoretical analysis and experiments support that:

- **When full-score training is affordable,** use Softmax as the default starting point. Choose sparse alternatives for how they allocate training signals, rather than sparsity alone, and tune them jointly with the backbone and learning rate.
- **When the output space is large,** use sampling or structured alternatives to reduce training cost, and select them by jointly assessing predictive accuracy and wall-clock efficiency. Sample count and proposal matching are key factors for balancing approximation quality and cost.

AI USE STATEMENT

The research motivation, original theoretical derivations and proofs, initial manuscript, and experimental code were developed entirely by the authors without AI assistance. The authors then use GPT-6 to assist in checking grammar issues, theoretical rigor and for language polishing. The authors carefully reviewed all AI-assisted revisions and take full responsibility for the entire paper, including its theoretical claims, proofs, code, and experimental results.

REPRODUCIBILITY STATEMENT

The appendices provide the metric definitions, assumptions, proofs, and derivations supporting our consistency, smoothness, bias–variance, and computational complexity analyses. Experimental settings and results are reported in Appendix G, including dataset preparation, backbone configurations, training settings, evaluation metrics, and comparisons across learning rates and negative-sampling strategies. Code and configurations for reproducing the experiments is available in the anonymized repository: `https://anonymous.4open.science/r/softmax-family`.

REFERENCES

Kunal Banerjee, Vishak Prasad C., Rishi Raj Gupta, Kartik Vyas, Anushree H., and Biswajit Mishra. Exploring alternatives to softmax function. pp. 81–86, 2021. doi: 10.5220/0010502000810086. URL https://doi.org/10.5220/0010502000810086.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. doi: 10.1198/016214505000000907. URL https://doi.org/10.1198/016214505000000907.

Guy Blanc and Steffen Rendle. Adaptive sampled softmax with kernel based sampling. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 589–598. PMLR, 2018. URL http://proceedings.mlr.press/v80/blanc18a.html.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with fenchel-young losses. *J. Mach. Learn. Res.*, 21:35:1–35:69, 2020. URL https://jmlr.org/papers/v21/19-021.html.

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Clément Calauzènes, Nicolas Usunier, and Patrick Gallinari. "on the (non-)existence of convex, calibrated surrogate losses for ranking". pp. 197–205, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/50f3f8c42b998a48057e9d33f4144b8b-Abstract.html.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. pp. 2174–2184, 2019. doi: 10.18653/V1/D19-1223. URL https://doi.org/10.18653/v1/D19-1223.

David Cossock and Tong Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Trans. Inf. Theory*, 54(11):5140–5154, 2008. doi: 10.1109/TIT.2008.929939. URL https://doi.org/10.1109/TIT.2008.929939.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and D. Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pp. 297–304. JMLR.org, 2010. URL http://proceedings.mlr.press/v9/gutmann10a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pp. 639–648. ACM, 2020. doi: 10.1145/3397271.3401063. URL https://doi.org/10.1145/3397271.3401063.

Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 1–10. The Association for Computer Linguistics, 2015. doi: 10.3115/V1/P15-1001. URL https://doi.org/10.3115/v1/p15-1001.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pp. 197–206. IEEE Computer Society, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. URL http://arxiv.org/abs/1412.6980.

Weiwei Kong, Walid Krichene, Nicolas Mayoraz, Steffen Rendle, and Li Zhang. Rankmax: An adaptive projection alternative to the softmax function. 2020. URL https://proceedings.neurips.cc/paper/2020/hash/070dbb6024b5ef93784428afc71f2146-Abstract.html.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 1468–1477. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.163. URL https://doi.org/10.1109/CVPR.2016.163.

André F. T. Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1614–1623. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/martins16.html.

Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Multilabel reductions: what is my loss optimising? In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, volume 32, pp. 10599–10610, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/da647c549dde572c2c5edc4f5bef039c-Abstract.html.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In Robert G. Cowell and Zoubin Ghahramani (eds.), *Proceedings of the Tenth International*

Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005, pp. 246–252. Society for Artificial Intelligence and Statistics, 2005. URL http://www.gatsby.ucl.ac.uk/aistats/fullpapers/208.pdf.

Yurii Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5, 1998.

Vlad Niculae, André F. T. Martins, Mathieu Blondel, and Claire Cardie. Sparsemap: Differentiable sparse structured inference. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3796–3805. PMLR, 2018. URL http://proceedings.mlr.press/v80/niculae18a.html.

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 1504–1519. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1146. URL https://doi.org/10.18653/v1/p19-1146.

Yuanhao Pu, Xiaolong Chen, Xu Huang, Jin Chen, Defu Lian, and Enhong Chen. Learning-efficient yet generalizable collaborative filtering for item recommendation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=D5IRvFF1lN.

Pradeep Ravikumar, Ambuj Tewari, and Eunho Yang. On NDCG consistency of listwise ranking methods. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pp. 618–626. JMLR.org, 2011. URL http://proceedings.mlr.press/v15/ravikumar11a/ravikumar11a.pdf.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. pp. 452–461, 2009. URL https://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3104–3112, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of statistical physics*, 52(1):479–487, 1988.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Forest Yang and Sanmi Koyejo. On the consistency of top-k surrogate losses. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10727–10735. PMLR, 2020. URL http://proceedings.mlr.press/v119/yang20f.html.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004.

APPENDIX

A Foundations of Learning Frameworks **14**
A.1 Evaluation Metric and Bayes-Optimal Ranking . . . 14
A.1.1 Metric Definitions . . . 14
A.1.2 Metric-Specific Bayes Rankings . . . 14
A.2 Fenchel-Young Losses and Bregman Divergence . . . 15
A.3 Softmax Approximation Methods . . . 15

B Proofs of Consistency for Softmax and Sparse F–Y Losses **16**
B.1 Conditional targets and probability-error control . . . 16
B.2 Optimal scores and score ordering . . . 17
B.3 Top-k calibration . . . 18
B.4 Precision and DCG consistency . . . 18

C Order Preservation, Local Geometry, and Sparse Training Signals **19**
C.1 Proof of Proposition 10 . . . 19
C.2 Proof of Theorem 11 . . . 19
C.3 Jacobians and local curvature . . . 19
C.4 Proof of Proposition 12 . . . 20
C.5 Proof of Theorem 13 . . . 21

D Discussions on Rankmax **21**
D.1 Proof of Proposition 14 . . . 21
D.2 A counterexample to Rankmax consistency . . . 21
D.3 Proof of Proposition 15 . . . 22
D.4 Curvature of the Rankmax loss . . . 22

E Smoothness Analysis **22**
E.1 Score-level smoothness . . . 22
E.2 Proof of Theorem 16 . . . 23
E.3 Rankmax Mapping Smoothness . . . 23
E.4 Smoothness of softmax approximation objectives . . . 23

F Conditional Discrepancies and the Delta Method **24**
F.1 Conditional bias-variance decomposition . . . 24
F.2 Delta-method expansion . . . 24
F.3 Bias-variance Decomposition of softmax Approximations . . . 24
F.4 Training Complexity . . . 26

G Experimental Results **26**
G.1 Settings . . . 26
G.1.1 Dataset and Evaluation . . . 26
G.1.2 Evaluation Metrics . . . 27
G.1.3 Baselines . . . 27
G.1.4 Implementation Details . . . 28
G.2 Q1: Accuracy under aligned protocols . . . 28
G.3 Q2: Bias-variance decomposition . . . 30
G.4 Q3: Training dynamics and efficiency . . . 32
G.5 Gradient-magnitude diagnostics . . . 34

A Foundations of Learning Frameworks

A.1 Evaluation Metric and Bayes-Optimal Ranking

We provide the formal definitions of the multi-label and ranking metrics used in the main text and characterize their Bayes-optimal rankings.

A.1.1 Metric Definitions

Precision@k and Recall@k. These metrics evaluate the set of retrieved candidates:

$$\mathrm{P}\,@k(\boldsymbol{y}, \boldsymbol{s}) = \frac{1}{k}\sum_{i=1}^{k} y_{\pi_{\boldsymbol{s}}(i)}, \quad \mathrm{R}\,@k(\boldsymbol{y}, \boldsymbol{s}) = \frac{1}{|\boldsymbol{y}|}\sum_{i=1}^{k} y_{\pi_{\boldsymbol{s}}(i)}. \tag{18}$$

Precision@k measures the fraction of retrieved candidates that are relevant, whereas Recall@k measures the fraction of relevant candidates that are retrieved.

DCG@k and NDCG@k. Let $w_i = 1/\log_2(i+1)$ denote the position discount. Discounted cumulative gain is

$$\mathrm{DCG}\,@k(\boldsymbol{y}, \boldsymbol{s}) = \sum_{i=1}^{k} w_i y_{\pi_{\boldsymbol{s}}(i)}. \tag{19}$$

For binary relevance labels, its maximum attainable value is

$$\mathrm{IDCG}\,@k(\boldsymbol{y}) = \max_{\boldsymbol{s}} \mathrm{DCG}\,@k(\boldsymbol{y}, \boldsymbol{s}) = \sum_{i=1}^{\min(k,|\boldsymbol{y}|)} w_i. \tag{20}$$

Normalized DCG is therefore

$$\mathrm{N}\,@k(\boldsymbol{y}, \boldsymbol{s}) = \frac{\mathrm{DCG}\,@k(\boldsymbol{y}, \boldsymbol{s})}{\mathrm{IDCG}\,@k(\boldsymbol{y})}. \tag{21}$$

The full-list versions are

$$\mathrm{DCG}(\boldsymbol{y}, \boldsymbol{s}) = \mathrm{DCG}\,@C(\boldsymbol{y}, \boldsymbol{s}), \quad \mathrm{NDCG}(\boldsymbol{y}, \boldsymbol{s}) = \mathrm{N}\,@C(\boldsymbol{y}, \boldsymbol{s}). \tag{22}$$

For each metric M above, consistency is understood w.r.t. the target loss $\ell_M = 1 - M$ in Definition 1. Although this loss may be negative for unnormalized DCG, minimizing its conditional risk is equivalent to maximizing expected DCG.

A.1.2 Metric-Specific Bayes Rankings

For a fixed input $\boldsymbol{x}$, define

$$r_i(\boldsymbol{x}) = \mathbb{E}[y_i \mid \boldsymbol{x}], \quad a_i(\boldsymbol{x}) = \mathbb{E}\left[\frac{y_i}{|\boldsymbol{y}|}\,\middle|\,\boldsymbol{x}\right], \quad b_i^{(k)}(\boldsymbol{x}) = \mathbb{E}\left[\frac{y_i}{\mathrm{IDCG}\,@k(\boldsymbol{y})}\,\middle|\,\boldsymbol{x}\right]. \tag{23}$$

Here, r_i is the marginal relevance probability, while a_i and $b_i^{(k)}$ incorporate the label-dependent normalizations used by Recall@k and NDCG@k. By linearity of conditional expectation,

$$\begin{aligned} \mathbb{E}[\mathrm{P}\,@k(\boldsymbol{y}, \boldsymbol{s}) \mid \boldsymbol{x}] &= \frac{1}{k}\sum_{i=1}^{k} r_{\pi_{\boldsymbol{s}}(i)}(\boldsymbol{x}), \quad \mathbb{E}[\mathrm{R}\,@k(\boldsymbol{y}, \boldsymbol{s}) \mid \boldsymbol{x}] = \sum_{i=1}^{k} a_{\pi_{\boldsymbol{s}}(i)}(\boldsymbol{x}), \\ \mathbb{E}[\mathrm{DCG}\,@k(\boldsymbol{y}, \boldsymbol{s}) \mid \boldsymbol{x}] &= \sum_{i=1}^{k} w_i r_{\pi_{\boldsymbol{s}}(i)}(\boldsymbol{x}), \quad \mathbb{E}[\mathrm{N}\,@k(\boldsymbol{y}, \boldsymbol{s}) \mid \boldsymbol{x}] = \sum_{i=1}^{k} w_i b^{(k)}_{\pi_{\boldsymbol{s}}(i)}(\boldsymbol{x}). \end{aligned} \tag{24}$$

For Precision@k and Recall@k, the order within the selected set is irrelevant while for DCG@k and NDCG@k, the ordering follows from the non-increasing position weights. If $|\boldsymbol{y}|$ is a constant almost surely conditional on $\boldsymbol{x}$, then

$$a_i(\boldsymbol{x}) = \frac{r_i(\boldsymbol{x})}{|\boldsymbol{y}|}, \quad b_i^{(k)}(\boldsymbol{x}) = \frac{r_i(\boldsymbol{x})}{\sum_{j=1}^{\min(k,|\boldsymbol{y}|)} w_j}.$$

Hence the three vectors induce the same ordering.

A.2 Fenchel-Young Losses and Bregman Divergence

For a strictly convex potential Ω on Δ^C differentiable at $\boldsymbol{q}$, its Bregman divergence (Bregman, 1967) is

$$D_\Omega(\boldsymbol{t}, \boldsymbol{q}) = \Omega(\boldsymbol{t}) - \Omega(\boldsymbol{q}) - \langle \nabla\Omega(\boldsymbol{q}), \boldsymbol{t} - \boldsymbol{q} \rangle. \tag{25}$$

If $\boldsymbol{t} \in \mathcal{A}$ has mean $\boldsymbol{\mu}$ and all terms are integrable, then

$$\mathbb{E}[D_\Omega(\boldsymbol{t}, \boldsymbol{q}) \mid \boldsymbol{x}] = \mathbb{E}[\Omega(\boldsymbol{t}) \mid \boldsymbol{x}] - \Omega(\boldsymbol{\mu}) + D_\Omega(\boldsymbol{\mu}, \boldsymbol{q}). \tag{26}$$

For $\boldsymbol{q} = \hat{\boldsymbol{p}}_\Omega(\boldsymbol{s})$ at which Ω is differentiable, the F-Y loss satisfies

$$\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = D_\Omega(\boldsymbol{t}, \boldsymbol{q}) + \langle \boldsymbol{s} - \nabla\Omega(\boldsymbol{q}), \boldsymbol{q} - \boldsymbol{t} \rangle, \tag{27}$$

whose additional term is non-negative by constrained optimality. For negative Shannon entropy and finite softmax scores, this recovers $\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = \mathrm{KL}(\boldsymbol{t} \| \hat{\boldsymbol{p}}_{\mathrm{SM}}(\boldsymbol{s}))$. However, for sparse mapping predictions, the additional term may be positive, so the loss does not equal the Bregman divergence.

A.3 Softmax Approximation Methods

Let $Z(\boldsymbol{s}) = \sum_{j=1}^{C} \exp s_j$, the summed softmax loss is formed as $|\boldsymbol{y}| \log Z(\boldsymbol{s}) - \langle \boldsymbol{y}, \boldsymbol{s} \rangle$. The following contents introduce typical approximation methods of softmax loss, in which c denotes one relevant candidate and $K \geq 1$ the number of noise samples.

SSM (Jean et al., 2015) approximates the full softmax by computing scores only on a small sampled subset of candidates. A simple realization is formed as

$$\ell_{\mathrm{simple}}(c, \boldsymbol{s}) = -s_c + \log \left(\exp s_c + \sum_{j=1}^{K} \exp s_{n_j} \right). \tag{28}$$

Such realization is often combined with a importance correction, in which n_j denotes independent draws with replacement from Q_c, with $Q_c(i) > 0$ for $i \neq c$. Define

$$\widehat{Z}_c = \exp s_c + \frac{1}{K} \sum_{j=1}^{K} \frac{\exp s_{n_j}}{Q_c(n_j)}, \quad \ell_{\mathrm{SSM}}(c, \boldsymbol{s}) = -s_c + \log \widehat{Z}_c. \tag{29}$$

Such design corrects sampled negative logits by $-\log(KQ_c(n_j))$, while retaining the positive logit. For fixed scores and proposal, $\mathbb{E}[\widehat{Z}_c] = Z$. The estimator and its score gradient converge to their full-softmax counterparts as $K \to \infty$.

NCE (Gutmann & Hyvärinen, 2010) learns a density by discriminating data from noise. Let $Q(i) > 0$ for every class and u_i denote the model's log-density parameter. For $n_j \overset{\text{iid}}{\sim} Q$,

$$\ell_{\mathrm{NCE}}(c, \boldsymbol{u}) = -\log \sigma(u_c - \log(KQ(c))) - \sum_{j=1}^{K} \log \sigma\big(-u_{n_j} + \log(KQ(n_j))\big), \tag{30}$$

where $\sigma(a) = (1 + e^{-a})^{-1}$. NCE can be statistically consistent as the number of data observations grows even for fixed K.

HSM (Morin & Bengio, 2005) replaces the flat output parameterization by a tree over classes. For an internal node u with children $\mathcal{C}(u)$ and local logits $a_u(v)$, define

$$p(v \mid u) = \frac{\exp a_u(v)}{\sum_{z \in \mathcal{C}(u)} \exp a_u(z)}, \quad p_{\mathrm{HSM}}(i) = \prod_{u \in \mathrm{path}(i)} p(c_i(u) \mid u), \tag{31}$$

where $\mathrm{path}(i)$ contains the internal nodes on the path to leaf i, and $c_i(u)$ is the next child on that path. The summed training loss is $-\sum_i y_i \log p_{\mathrm{HSM}}(i)$. Its valid implicit global scores are

$$g_i = \log p_{\mathrm{HSM}}(i) = \sum_{u \in \mathrm{path}(i)} \left[a_u(c_i(u)) - \log \sum_{v \in \mathcal{C}(u)} \exp a_u(v) \right]. \tag{32}$$

Since the leaf probabilities sum to one, $\mathrm{softmax}(\boldsymbol{g})_i = p_{\mathrm{HSM}}(i)$.

Taylor-based surrogates

(i) Approximating the exponential. Taylor-softmax (Banerjee et al., 2021) replaces $\exp s_i$ by a polynomial before normalization, i.e.,

$$p_{\mathrm{Taylor},i}(\boldsymbol{s}) = \frac{1 + s_i + s_i^2/2}{\sum_j (1 + s_j + s_j^2/2)}. \tag{33}$$

The numerator is positive but not globally increasing in s_i, thus need not preserve score order.

(ii) Approximating the log-partition. Let $\bar{s} = C^{-1} \sum_i s_i$ and $\mathbf{1}$ be the all-ones vector. The second-order expansion of the softmax log-partition at zero is

$$\Psi_{\mathrm{quad}}(\boldsymbol{s}) = \log C + \bar{s} + \frac{1}{2C} \|\boldsymbol{s} - \bar{s}\mathbf{1}\|_2^2, \quad \mathcal{L}_{\mathrm{quad}}(\boldsymbol{y}, \boldsymbol{s}) = |\boldsymbol{y}| \Psi_{\mathrm{quad}}(\boldsymbol{s}) - \langle \boldsymbol{y}, \boldsymbol{s} \rangle. \tag{34}$$

Here Ψ_{quad} approximates $\log Z$ rather than Z. For fixed C,

$$\log Z(\boldsymbol{s}) - \Psi_{\mathrm{quad}}(\boldsymbol{s}) = O\big(\|\boldsymbol{s} - \bar{s}\mathbf{1}\|_2^3\big) \quad \text{as } \boldsymbol{s} - \bar{s}\mathbf{1} \to \mathbf{0}. \tag{35}$$

The full centered quadratic corresponds to the $\mathrm{RG}^{\times}$ construction of Pu et al. (2024), up to score-independent constants and aggregation conventions. Their RG^2 variant drops the interaction term $-(\sum_i s_i)^2/(2C^2)$ in the per-positive log-partition approximation.

B Proofs of Consistency for Softmax and Sparse F–Y Losses

We consider Softmax, Sparsemax, and α-Entmax with the potentials in Section 2.4. Write

$$\boldsymbol{q}(\boldsymbol{s}) = \arg\max_{\boldsymbol{p} \in \Delta^C} \{\langle \boldsymbol{s}, \boldsymbol{p} \rangle - \Omega(\boldsymbol{p})\}, \quad \mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = \Omega^*(\boldsymbol{s}) + \Omega(\boldsymbol{t}) - \langle \boldsymbol{s}, \boldsymbol{t} \rangle.$$

B.1 Conditional targets and probability-error control

For one-hot labels, let $\boldsymbol{t} = \boldsymbol{\eta} = \mathbb{E}[\boldsymbol{y} \mid \boldsymbol{x}]$ and $W = 1$, while for non-empty binary labels and the summed loss $\sum_i y_i \mathcal{L}_\Omega(\boldsymbol{e}_i, \boldsymbol{s})$, let $r_i = \mathbb{E}[y_i \mid \boldsymbol{x}], W = \sum_i r_i = \mathbb{E}[|\boldsymbol{y}| \mid \boldsymbol{x}] > 0, \boldsymbol{t} = \boldsymbol{r}/W$. In both cases, denote the conditional surrogate risk by $\mathcal{R}_\Omega$ and its excess by

$$\mathcal{E}_\Omega(\boldsymbol{s}) = \mathcal{R}_\Omega(\boldsymbol{s}) - \inf_{\boldsymbol{v} \in \mathbb{R}^C} \mathcal{R}_\Omega(\boldsymbol{v}).$$

Lemma 17 (Strong convexity of the potentials)**.** *Negative Shannon entropy, the Sparsemax potential, and negative Tsallis entropy are* 1*-strongly convex on* Δ^C *in Euclidean norm.*

Proof. On the relative interior of the simplex, their Hessians are

$$\nabla^2 \Omega_{\mathrm{SM}}(\boldsymbol{p}) = \mathrm{diag}(1/p_i), \quad \nabla^2 \Omega_{\mathrm{SP}}(\boldsymbol{p}) = I, \quad \nabla^2 \Omega_\alpha(\boldsymbol{p}) = \mathrm{diag}(p_i^{\alpha - 2}).$$

Since $0 < p_i \leq 1$ and $\alpha \leq 2$, each quadratic form is bounded below by the squared Euclidean norm. Integrating along line segments proves strong convexity on the relative interior. □

Lemma 18 (Conditional excess-risk identity)**.** *For either label setting,* $\mathcal{E}_\Omega(\boldsymbol{s}) = W \mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s})$.

Proof. Expanding the conditional expectation gives

$$\mathcal{R}_\Omega(\boldsymbol{s}) = W \Omega^*(\boldsymbol{s}) - W \langle \boldsymbol{s}, \boldsymbol{t} \rangle + A,$$

where $A = \sum_i r_i \Omega(\boldsymbol{e}_i)$ for summed labels and $r_i = \eta_i$ for one-hot labels. Thus

$$\mathcal{R}_\Omega(\boldsymbol{s}) = W \mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) + A - W \Omega(\boldsymbol{t}).$$

Extend Ω by $+\infty$ outside the simplex. It is closed and convex, so Fenchel bi-conjugacy gives

$$\inf_{\boldsymbol{s}} \mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = \Omega(\boldsymbol{t}) - \sup_{\boldsymbol{s}} \{\langle \boldsymbol{s}, \boldsymbol{t} \rangle - \Omega^*(\boldsymbol{s})\} = 0.$$

Subtracting the infimum proves the lemma. □

Lemma 19 (Probability-error control)**.** *For every score vector, $\mathcal{E}_\Omega(\boldsymbol{s}) = W\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) \geq \frac{W}{2}\|\boldsymbol{q}(\boldsymbol{s}) - \boldsymbol{t}\|_2^2$. Consequently, $\mathcal{E}_\Omega(\boldsymbol{s}^{(n)}) \to 0$ implies $\boldsymbol{q}(\boldsymbol{s}^{(n)}) \to \boldsymbol{t}$.*

Proof. Set $f(\boldsymbol{p}) = \Omega(\boldsymbol{p}) - \langle \boldsymbol{s}, \boldsymbol{p}\rangle$ and $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{s})$. By Lemma 17, f is 1-strongly convex, and $\boldsymbol{q}$ minimizes it over the simplex. For $0 < \lambda < 1$, optimality and strong convexity give

$$f(\boldsymbol{q}) \leq f((1-\lambda)\boldsymbol{q} + \lambda\boldsymbol{t}) \leq (1-\lambda)f(\boldsymbol{q}) + \lambda f(\boldsymbol{t}) - \frac{\lambda(1-\lambda)}{2}\|\boldsymbol{t} - \boldsymbol{q}\|_2^2.$$

Rearranging and letting $\lambda \downarrow 0$ yields $f(\boldsymbol{t}) - f(\boldsymbol{q}) \geq \frac{1}{2}\|\boldsymbol{t} - \boldsymbol{q}\|_2^2$. Since $\Omega^*(\boldsymbol{s}) = -f(\boldsymbol{q})$, the left-hand side is $\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s})$. Applying Lemma 18 completes the proof. □

B.2 Optimal scores and score ordering

Lemma 20 (Softmax)**.** *For negative Shannon entropy, $\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) = \mathrm{KL}(\boldsymbol{t}\|\boldsymbol{q}(\boldsymbol{s}))$ with $q_i(\boldsymbol{s}) = \frac{e^{s_i}}{\sum_j e^{s_j}}$. If $\boldsymbol{t}$ is strictly positive, all optimal scores have the form $s_i^* = \log t_i + a$ for a common $a \in \mathbb{R}$. For every finite score vector, $q_i(\boldsymbol{s}) > q_j(\boldsymbol{s}) \iff s_i > s_j$.*

Proof. Substitute $\Omega(\boldsymbol{t}) = \sum_i t_i \log t_i$ and $\Omega^*(\boldsymbol{s}) = \log \sum_i e^{s_i}$ into the F-Y loss. The KL divergence vanishes exactly when $\boldsymbol{q} = \boldsymbol{t}$, giving the optimal-score formula for positive targets. The ordering equivalence follows from $q_i/q_j = \exp(s_i - s_j)$. □

Lemma 21 (Sparse optimal scores and support separation)**.** *For Sparsemax and α-Entmax, a score vector is conditionally optimal if and only if $\boldsymbol{q}(\boldsymbol{s}^*) = \boldsymbol{t}$. Let $\mathcal{P} = \{i : t_i > 0\}$. Sparsemax optimal scores satisfy*

$$s_i^* = t_i + \tau^* \quad (i \in \mathcal{P}), \quad s_j^* \leq \tau^* \quad (j \notin \mathcal{P}).$$

For α-Entmax they satisfy

$$s_i^* = \tau^* + \frac{t_i^{\alpha-1}}{\alpha - 1} \quad (i \in \mathcal{P}), \quad s_j^* \leq \tau^* \quad (j \notin \mathcal{P}).$$

In each case, finite optimal scores exist for every $\boldsymbol{t} \in \Delta^C$.

Proof. Strong convexity makes the conjugate differentiable with gradient $\boldsymbol{q}$, so

$$\nabla_{\boldsymbol{s}} \mathcal{R}_\Omega(\boldsymbol{s}) = W(\boldsymbol{q}(\boldsymbol{s}) - \boldsymbol{t}).$$

The risk is convex and differentiable on $\mathbb{R}^C$, hence optimality is equivalent to $\boldsymbol{q}(\boldsymbol{s}^*) = \boldsymbol{t}$. The threshold representations are

$$q_i(\boldsymbol{s}) = [s_i - \tau]_+ \text{ (Sparsemax)}, \quad q_i(\boldsymbol{s}) = [(\alpha-1)(s_i - \tau)]_+^{1/(\alpha-1)} \text{ (α-Entmax)}.$$

Substituting $\boldsymbol{q} = \boldsymbol{t}$ gives the displayed score formulas. Taking $\tau^* = 0$ and setting zero-target scores to zero constructs finite optima. □

Lemma 22 (Ordering for arbitrary scores)**.** *For all three maps and every score vector,*

$$s_i \leq s_j \implies q_i(\boldsymbol{s}) \leq q_j(\boldsymbol{s}), \quad q_i(\boldsymbol{s}) > q_j(\boldsymbol{s}) \implies s_i > s_j. \tag{36}$$

Consequently, any non-increasing score permutation also sorts $\boldsymbol{q}(\boldsymbol{s})$ non-increasingly.

Proof. For Softmax, use Lemma 20 and equality of probabilities at equal scores. For the sparse maps, apply their common non-decreasing scalar threshold function to s_i and s_j. □

B.3 TOP-k CALIBRATION

Lemma 23 (Excess risk of an inverted pair)**.** *If $t_i > t_j$ but $s_i \le s_j$, then $\mathcal{E}_\Omega(\boldsymbol{s}) \ge \frac{W}{4}(t_i - t_j)^2$.*

Proof. Lemma 22 gives $q_i \le q_j$. Therefore

$$t_i - t_j = (t_i - q_i) + (q_i - q_j) + (q_j - t_j) \le |t_i - q_i| + |q_j - t_j|.$$

Squaring and applying inequality $(a+b)^2 \le 2(a^2+b^2)$ yields

$$(q_i - t_i)^2 + (q_j - t_j)^2 \ge \frac{(t_i - t_j)^2}{2}.$$

Combine this with Lemma 19 finishes this proof. □

Theorem 24 (Top-k calibration)**.** *Softmax, Sparsemax, and α-Entmax are Top-k calibrated for one-hot labels and every $1 \le k < C$.*

Proof. Let $\eta_{[k]}$ and $s_{[k]}$ denote the k-th largest coordinates. A score vector $\boldsymbol{s}$ is **Top-k preserving** w.r.t. $\boldsymbol{\eta}$ if for every i,

$$\eta_i > \eta_{[k+1]} \implies s_i > s_{[k+1]}, \quad \eta_i < \eta_{[k]} \implies s_i < s_{[k]}.$$

Top-k calibration requires a positive conditional excess-risk gap over scores violating this property (Yang & Koyejo, 2020). For non-uniform $\boldsymbol{\eta}$, let $\delta(\boldsymbol{\eta}) = \min_{i,j:\eta_i > \eta_j}(\eta_i - \eta_j) > 0$. Failure of either implication yields a pair $\eta_i > \eta_j$ with $s_i \le s_j$. Since $W = 1$ and $\boldsymbol{t} = \boldsymbol{\eta}$, Lemma 23 gives

$$\mathcal{E}_\Omega(\boldsymbol{s}) \ge \frac{(\eta_i - \eta_j)^2}{4} \ge \frac{\delta(\boldsymbol{\eta})^2}{4} > 0. \tag{37}$$

This establishes Top-k calibration. For uniform $\boldsymbol{\eta}$, both preserving conditions are vacuous. □

B.4 PRECISION AND DCG CONSISTENCY

Let $d_1 \ge \cdots \ge d_C \ge 0$ be fixed position weights and define $M_d(\boldsymbol{y}, \boldsymbol{s}) = \sum_{a=1}^{C} d_a y_{\pi_{\boldsymbol{s}}(a)}$. For Precision@$k$, set $d_a = 1/k$ for $a \le k$ and zero otherwise. For DCG@k, set $d_a = 1/\log_2(a+1)$ for $a \le k$ and zero otherwise.

Lemma 25 (Conditional ranking-regret bound)**.** *For the summed multi-label loss, the conditional regret of M_d satisfies*

$$\mathrm{Reg}_d(\boldsymbol{s}) \le 2W\|\boldsymbol{d}\|_2\|\boldsymbol{t} - \boldsymbol{q}(\boldsymbol{s})\|_2 \le 2\|\boldsymbol{d}\|_2\sqrt{2W\mathcal{E}_\Omega(\boldsymbol{s})}. \tag{38}$$

Proof. Let π^* be a permutation satisfying $t_{\pi^*(1)} \ge \cdots \ge t_{\pi^*(C)}$, and let $\pi = \pi_{\boldsymbol{s}}$ denote the permutation induced by sorting the scores in non-increasing order. Since $\boldsymbol{r} = W\boldsymbol{t}$, the rearrangement inequality gives

$$\mathrm{Reg}_d(\boldsymbol{s}) = W\sum_a d_a(t_{\pi^*(a)} - t_{\pi(a)}).$$

By Lemma 22, π also sorts $\boldsymbol{q}$, therefore gives $\sum_a d_a(q_{\pi^*(a)} - q_{\pi(a)}) \le 0$. Adding and subtracting the corresponding probability terms yields

$$\frac{\mathrm{Reg}_d(\boldsymbol{s})}{W} \le \sum_a d_a(t-q)_{\pi^*(a)} - \sum_a d_a(t-q)_{\pi(a)} \le 2\|\boldsymbol{d}\|_2\|\boldsymbol{t} - \boldsymbol{q}\|_2.$$

The last step uses Cauchy–Schwarz and permutation invariance of the norm. Lemma 19 directly gives the second bound. □

Theorem 26 (Precision and DCG consistency)**.** *The summed multi-label losses associated with Softmax, Sparsemax, and α-Entmax are consistent with* Precision@k *and* DCG@k *for every $1 \le k \le C$.*

Proof. Insert their corresponding position weights into Lemma 25. As $\mathcal{E}_\Omega(\boldsymbol{s}^{(n)}) \to 0$, the conditional metric regret tends to zero. □

This proves the ranking claims in Propositions 7 and 8.

Corollary 27 (Normalized ranking targets)**.** *For non-empty binary labels, dividing the summed F-Y loss by $|\boldsymbol{y}|$ yields* Recall@k *consistency. while* IDCG $@k(\boldsymbol{y})$ *yields* NDCG@k *consistency.*

Proof. For a positive example weight $w(\boldsymbol{y})$, define

$$r_i^w = \mathbb{E}[w(\boldsymbol{y})y_i \mid \boldsymbol{x}], \quad W_w = \sum_i r_i^w, \quad \boldsymbol{t}_w = \boldsymbol{r}^w / W_w.$$

The weighted conditional risk has the same decomposition as in Lemma 18, with $W, \boldsymbol{t}$ replaced by $W_w, \boldsymbol{t}_w$. Its target metric is $\mathbb{E}[w(\boldsymbol{y})M_d(\boldsymbol{y}, \boldsymbol{s}) \mid \boldsymbol{x}]$, so Lemma 25 applies with the same replacements.

For Recall@k, use $w(\boldsymbol{y}) = 1/|\boldsymbol{y}|$ and $d_a = 1$ for $a \le k$, zero otherwise. Then $W_w = 1$ and $\boldsymbol{t}_w = \mathbb{E}[\boldsymbol{y}/|\boldsymbol{y}| \mid \boldsymbol{x}]$. For NDCG@$k$, use $w(\boldsymbol{y}) = 1/\operatorname{IDCG} @k(\boldsymbol{y})$ and the DCG@k position weights. The target vector is proportional to $\mathbb{E}[\boldsymbol{y}/\operatorname{IDCG} @k(\boldsymbol{y}) \mid \boldsymbol{x}]$. In both cases vanishing weighted surrogate excess implies vanishing conditional metric regret. □

C Order Preservation, Local Geometry, and Sparse Training Signals

C.1 Proof of Proposition 10

Proof. For softmax, $q_i/q_j = \exp(s_i - s_j)$, which directly proves SOP. The sparse maps, however, have the common-threshold forms $[s_i - \tau]_+$ and $[(\alpha - 1)(s_i - \tau)]_+^{1/(\alpha-1)}$. The transformations are non-decreasing and strictly increasing above the threshold, proving WOP and tie-consistency.

For $C \ge 3$, take $s_1 = 1/(\alpha - 1)$, $s_2 = -1$, and $s_3 = -2$, with any remaining scores at most -2. The threshold $\tau = 0$ yields $\boldsymbol{q} = \boldsymbol{e}_1$, so $s_2 > s_3$ but $q_2 = q_3 = 0$. □

C.2 Proof of Theorem 11

Proof. Using the notation of Appendix B, the conditional-risk decomposition and μ-strong convexity give

$$\mathcal{E}_\Omega(\boldsymbol{s}) = W\mathcal{L}_\Omega(\boldsymbol{t}, \boldsymbol{s}) \ge \frac{W\mu}{2}\|\boldsymbol{q}(\boldsymbol{s}) - \boldsymbol{t}\|_2^2. \tag{39}$$

WOP and tie-consistency imply $s_i \le s_j \Rightarrow q_i \le q_j$, which is the ordering property used in preceding proofs. Those arguments therefore apply with the strong-convexity constant replaced by μ: the Top-k calibration gap becomes $\mu\delta(\boldsymbol{\eta})^2/4$, and the ranking-regret bound is

$$\operatorname{Reg}_d(\boldsymbol{s}) \le 2\|\boldsymbol{d}\|_2\sqrt{\frac{2W\mathcal{E}_\Omega(\boldsymbol{s})}{\mu}}.$$

These establish Top-k calibration and Precision@k/DCG@k consistency, respectively. □

C.3 Jacobians and local curvature

Let $\mathcal{P} = \{i : q_i > 0\}$ and $h = |\mathcal{P}|$. Since $\nabla\Omega^*(\boldsymbol{s}) = \boldsymbol{q}$, the summed F-Y loss is written as

$$\mathcal{L}_\Omega^{\text{sum}}(\boldsymbol{y}, \boldsymbol{s}) = |\boldsymbol{y}|\Omega^*(\boldsymbol{s}) + \sum_i y_i\Omega(\boldsymbol{e}_i) - \langle \boldsymbol{y}, \boldsymbol{s}\rangle.$$

Differentiating w.r.t. the scores gives

$$\nabla_{\boldsymbol{s}}\mathcal{L}_\Omega^{\text{sum}} = |\boldsymbol{y}|\boldsymbol{q} - \boldsymbol{y}, \quad \nabla_{\boldsymbol{s}}^2\mathcal{L}_\Omega^{\text{sum}} = |\boldsymbol{y}|J_\Omega, \text{ where } J_\Omega = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{s}}. \tag{40}$$

For non-empty labels, the loss Hessian has the same rank and kernel as J_Ω.

Softmax. For $q_i = e^{s_i}/\sum_j e^{s_j}$, we have $\frac{\partial q_i}{\partial s_j} = q_i(\delta_{ij} - q_j)$, $J_{\mathrm{SM}} = \mathrm{diag}(\boldsymbol{q}) - \boldsymbol{q}\boldsymbol{q}^\top$. For any $\boldsymbol{v} \in \mathbb{R}^C$, let $\bar{v}_q = \sum_i q_i v_i$. Since $\sum_i q_i = 1$,

$$\boldsymbol{v}^\top J_{\mathrm{SM}} \boldsymbol{v} = \sum_i q_i v_i^2 - \left(\sum_i q_i v_i\right)^2 = \sum_i q_i (v_i - \bar{v}_q)^2.$$

At finite scores, every q_i is positive. The sum vanishes if and only if $v_i = \bar{v}_q$ for every i. Hence $\ker J_{\mathrm{SM}} = \mathrm{span}\{\mathbf{1}\}, \mathrm{rank}(J_{\mathrm{SM}}) = C - 1$. In particular, every nonzero $\boldsymbol{v} \in \mathbf{1}^\perp$ has $\boldsymbol{v}^\top J_{\mathrm{SM}} \boldsymbol{v} > 0$.

Sparsemax and α-Entmax. For $1 < \alpha \leq 2$, the threshold map is $[(\alpha-1)(s_i - \tau)]_+^{1/(\alpha-1)}$. On its active support, $q_i^{\alpha-1} = (\alpha-1)(s_i - \tau)$. Differentiating both sides yields

$$(\alpha-1) q_i^{\alpha-2}\, dq_i = (\alpha-1)(ds_i - d\tau),$$

and hence

$$dq_i = q_i^{2-\alpha}(ds_i - d\tau).$$

Inactive probabilities remain zero on the fixed-support region, so $dq_i = 0$ there. Define

$$a_i = \begin{cases} q_i^{2-\alpha}, & i \in \mathcal{P}, \\ 0, & i \notin \mathcal{P}, \end{cases} \qquad A = \sum_i a_i > 0.$$

Differentiating $\sum_i q_i = 1$ now gives

$$0 = \sum_i dq_i = \sum_i a_i ds_i - A\, d\tau, \quad d\tau = \frac{\sum_i a_i ds_i}{A}.$$

Substitution gives

$$dq_i = \sum_j \left(a_i \delta_{ij} - \frac{a_i a_j}{A}\right) ds_j,$$

so

$$J_\alpha = \mathrm{diag}(\boldsymbol{a}) - \frac{\boldsymbol{a}\boldsymbol{a}^\top}{A}. \tag{41}$$

This recovers the Entmax Jacobian of Peters et al. (2019).

For $\alpha = 2$, $a_i = 1$ on $\mathcal{P}$ and $A = h$. The active block therefore becomes

$$J_{\mathrm{SP}}\big|_{\mathcal{P}\times\mathcal{P}} = I_h - \frac{1}{h}\mathbf{1}\mathbf{1}^\top,$$

with all remaining entries zero, recovering Sparsemax (Martins & Astudillo, 2016).

For any $\boldsymbol{v} \in \mathbb{R}^C$, define $\bar{v}_a = A^{-1}\sum_i a_i v_i$. Then

$$\boldsymbol{v}^\top J_\alpha \boldsymbol{v} = \sum_i a_i v_i^2 - \frac{(\sum_i a_i v_i)^2}{A} = \sum_{i\in\mathcal{P}} a_i (v_i - \bar{v}_a)^2.$$

Since $a_i > 0$ on $\mathcal{P}$, this quadratic form vanishes exactly when the active coordinates of $\boldsymbol{v}$ are equal. Therefore

$$\ker J_\alpha = \{\boldsymbol{v} :\ v_i = c \text{ for all } i \in \mathcal{P},\ \text{for some } c \in \mathbb{R}\}.$$

The kernel has one degree of freedom on the active support and $C - h$ on the inactive coordinates:

$$\dim\ker J_\alpha = C - h + 1, \quad \mathrm{rank}(J_\alpha) = h - 1.$$

Compared with Softmax, a sparse map with $h < C$ has $C - h$ zero-curvature directions.

C.4 Proof of Proposition 12

Proof. Since $\nabla\Omega^*(\boldsymbol{s}) = \boldsymbol{q}(\boldsymbol{s})$, the differentiation of the summed F-Y loss gives

$$\nabla_{\boldsymbol{s}} \mathcal{L}_\Omega^{\mathrm{sum}} = \sum_i y_i(\boldsymbol{q} - \boldsymbol{e}_i) = |\boldsymbol{y}|\boldsymbol{q} - \boldsymbol{y}.$$

Thus, for a negative class j,

$$\frac{\partial \mathcal{L}_\Omega^{\mathrm{sum}}}{\partial s_j} = |\boldsymbol{y}| q_j.$$

This derivative is strictly positive for Softmax at finite scores, whereas for sparse maps it vanishes exactly when $q_j = 0$. For an excluded positive class i, the derivative is instead $|\boldsymbol{y}|q_i - y_i = -1$. □

C.5 Proof of Theorem 13

Proof. Consider a tied block occupying positions $b+1, \dots, b+h$, with r relevant candidates. Let Y_a indicate whether the candidate assigned to position $b+a$ is relevant. Under a uniformly random permutation of the block, each candidate appears at each position with probability $1/h$, so $\mathbb{E}[Y_a] = \frac{r}{h}$. The expected contribution of the block is

$$\mathbb{E}\left[\sum_{a=1}^{h} d_{b+a} Y_a\right] = \frac{r}{h} \sum_{a=1}^{h} d_{b+a}.$$

Since the position weights are non-increasing, a label-aware optimal permutation places the r relevant candidates first, contributing $\sum_{a=1}^{r} d_{b+a}$. The expected deficit is therefore $\sum_{a=1}^{r} d_{b+a} - \frac{r}{h}\sum_{a=1}^{h} d_{b+a}$, which proves Eq. 9. For $0 < r < h$, this difference equals

$$\frac{1}{h} \sum_{a=1}^{r} \sum_{c=r+1}^{h} \left(d_{b+a} - d_{b+c}\right) \geq 0.$$

For DCG@k, set

$$d_a = \begin{cases} 1/\log_2(a+1), & a \leq k, \\ 0, & a > k. \end{cases}$$

For NDCG@k, divide the deficit by $\mathrm{IDCG}\,@k(\boldsymbol{y}) > 0$ for the fixed nonempty label vector. □

D Discussions on Rankmax

D.1 Proof of Proposition 14

Proof. Since $[s_c - s_c + 1]_+ = 1$, we have $Z_c \geq 1$ and $q_c^{\mathrm{rm}} > 0$. All coordinates share the threshold $s_c - 1$ and the positive denominator Z_c, so monotonicity of $[\cdot]_+$ gives WOP and tie-consistency. Moreover, $s_j \geq s_c$ implies $q_j^{\mathrm{rm}} > 0$, while $s_j \leq s_c - 1$ implies $q_j^{\mathrm{rm}} = 0$.

For $s_i > s_j$, their probabilities are equal exactly when $s_i \leq s_c - 1$. Thus SOP fails for $C \geq 3$: take $c = 1$, $\boldsymbol{s} = (0, -2, -3)$, and any additional scores at most -3 as a counterexample. □

D.2 A counterexample to Rankmax consistency

Consider the posterior $\boldsymbol{\eta} = (0.50, 0.24, 0.26)$. Its conditional risk is minimized at $\boldsymbol{s}^* = (1, 0, 0)$, which ties two classes with different posterior probabilities. This yields the following counterexample.

Proposition 28. *The loss ℓ_{rm} is neither Top-2 consistent nor DCG consistent.*

Proof. We first show that $\inf_{\boldsymbol{s}} R(\boldsymbol{s}; \boldsymbol{\eta}) = \log 2$, attained at $\boldsymbol{s}^* = (1, 0, 0)$.

For a fixed set of scores, the class loss $\log \sum_j [1 + s_j - s_c]_+$ decreases with s_c. Exchanging an inverted pair therefore assigns larger scores to larger posterior probabilities without increasing risk. Relabel the classes in posterior order $(0.50, 0.26, 0.24)$ and, by shift invariance, write their scores as $(u+v, v, 0)$ with $u, v \geq 0$. Any adjacent gap exceeding one can be reduced to one by raising the lower group of scores. This leaves upper-label losses unchanged and decreases lower-label losses. It therefore suffices to consider $u, v \in [0, 1]$. The normalizers are

$$Z_1 = 2 - u + [1 - u - v]_+, \quad Z_2 = 3 + u - v, \quad Z_3 = 3 + u + 2v.$$

On each triangle separated by $u + v = 1$, the risk $\frac{1}{2}\log Z_1 + \frac{13}{50}\log Z_2 + \frac{6}{25}\log Z_3$ is concave, so its minimum occurs at a vertex. The four candidate values are

(u, v)	$(0, 0)$	$(1, 0)$	$(0, 1)$	$(1, 1)$
R	$\log 3$	$\log 2$	$\frac{19}{25}\log 2 + \frac{6}{25}\log 5$	$\frac{1}{2}\log 3 + \frac{6}{25}\log 2$

The minimum is $\log 2$: the third value exceeds it by $\frac{6}{25}\log(5/2)$, and the fourth by $\frac{1}{50}\log(3^{25}/2^{38}) > 0$. Restoring the original indices gives $\boldsymbol{s}^* = (1, 0, 0)$. Now take $\boldsymbol{s}^{(n)} = (1, 1/n, 0)$ for $n \geq 2$. By continuity,

$$R(\boldsymbol{s}^{(n)}; \boldsymbol{\eta}) \longrightarrow R(\boldsymbol{s}^*; \boldsymbol{\eta}) = \inf_{\boldsymbol{s}} R(\boldsymbol{s}; \boldsymbol{\eta}).$$

Every $\boldsymbol{s}^{(n)}$ ranks the classes as $1, 2, 3$, whereas the Bayes order is $1, 3, 2$. The selected Top-2 set is therefore $\{1, 2\}$ instead of $\{1, 3\}$, giving constant regret

$$\mathrm{Reg}_{\text{Top-2}}(\boldsymbol{s}^{(n)}) = \eta_3 - \eta_2 = \frac{1}{50}.$$

Likewise, with $w_a = 1/\log_2(a+1)$,

$$\mathrm{Reg}_{\mathrm{DCG}}(\boldsymbol{s}^{(n)}) = (\eta_3 - \eta_2)(w_2 - w_3) = \frac{1}{50}\left(\frac{1}{\log_2 3} - \frac{1}{2}\right) > 0.$$

Thus the surrogate excess risk tends to zero while both target regrets remain positive, proving inconsistency. □

D.3 Proof of Proposition 15

Proof. Let $Z_c = 1 + \sum_{j \neq c}[s_j - s_c + 1]_+, \mathcal{A}_c = \{j \neq c : s_j > s_c - 1\}, a_c = |\mathcal{A}_c|$. The active set is locally fixed, so

$$\nabla_{\boldsymbol{s}} Z_c = \sum_{j \in \mathcal{A}_c} \boldsymbol{e}_j - a_c \boldsymbol{e}_c.$$

Differentiating $\ell_{\mathrm{rm}} = \log Z_c$ gives Eq. 12. If $s_c - s_j \geq 1$ for every $j \neq c$, then $Z_c = 1$ and $\ell_{\mathrm{rm}} = 0$. Under strict inequalities, $\mathcal{A}_c$ is locally empty and the gradient is zero. □

D.4 Curvature of the Rankmax loss

On a fixed active-set region, let $\boldsymbol{v}_c = \sum_{j \in \mathcal{A}_c}(\boldsymbol{e}_j - \boldsymbol{e}_c)$. Since Z_c is affine and $\nabla \ell_{\mathrm{rm}} = \boldsymbol{v}_c / Z_c$,

$$\nabla^2_{\boldsymbol{s}} \ell_{\mathrm{rm}} = -\frac{\boldsymbol{v}_c \boldsymbol{v}_c^\top}{Z_c^2}. \tag{42}$$

For $a_c > 0$, its nonzero eigenvalue is $-\|\boldsymbol{v}_c\|_2^2 / Z_c^2 < 0$. Hence the Rankmax cross-entropy is non-convex in the scores and falls outside the F-Y loss framework.

E Smoothness Analysis

E.1 Score-level smoothness

Proposition 29 (Global smoothness bounds)**.** *For $C \geq 2$, the prediction maps of Softmax, Sparsemax, and α-Entmax ($1 < \alpha \leq 2$) are globally Lipschitz with bounds*

$$B_{\mathrm{SM}} = \frac{1}{2}, \quad B_{\mathrm{SP}} = 1, \quad B_\alpha = 1.$$

The first two bounds are sharp. Their summed F–Y losses have $|\boldsymbol{y}|B_\Omega$-Lipschitz score gradients.

Proof. For Softmax, the variance identity in Appendix C.3 gives, for $\|\boldsymbol{v}\|_2 = 1$,

$$\boldsymbol{v}^\top J_{\mathrm{SM}} \boldsymbol{v} \leq \frac{1}{4}(\max_i v_i - \min_i v_i)^2 \leq \frac{1}{2}.$$

The bound is attained at $\boldsymbol{q} = (1/2, 1/2)$ when $C = 2$, and approached by concentrating probability equally on two classes when $C > 2$. Integrating the Jacobian along line segments gives global $1/2$-Lipschitz continuity.

Sparsemax is globally 1-Lipschitz. Its active Jacobian block is an orthogonal projector with eigenvalue one whenever at least two coordinates are active. For α-Entmax, the potential is 1-strongly convex by Lemma 17. Its conjugate gradient is therefore globally 1-Lipschitz.

Finally, Eq. 40 gives $\nabla_{\boldsymbol{s}} \mathcal{L}_\Omega^{\mathrm{sum}} = |\boldsymbol{y}|\boldsymbol{q}(\boldsymbol{s}) - \boldsymbol{y}$, so the loss-gradient Lipschitz bound is $|\boldsymbol{y}|B_\Omega$. □

E.2 Proof of Theorem 16

Proof. Since the scores are affine in θ and the F-Y loss is convex in the scores, the empirical loss is convex in θ. Adding $\gamma\|\theta\|_2^2/2$ therefore makes F_Ω γ-strongly convex. The regularized objective is also coercive, so it has a unique minimizer θ^*. To bound its smoothness, let

$$M = \frac{1}{N}\sum_n |\boldsymbol{y}_n| A_n^\top A_n, \quad \Lambda = \|M\|_2.$$

For a parameter displacement $\boldsymbol{d} = \theta' - \theta$, the corresponding score displacement of example n is $A_n\boldsymbol{d}$. By Proposition 29, its summed loss has a $|\boldsymbol{y}_n|B_\Omega$-Lipschitz gradient. Applying the descent bound to each example gives

$$\begin{aligned} F_\Omega(\theta') &\le F_\Omega(\theta) + \langle \nabla F_\Omega(\theta), \boldsymbol{d}\rangle + \frac{1}{2}\boldsymbol{d}^\top(\gamma I + B_\Omega M)\boldsymbol{d} \\ &\le F_\Omega(\theta) + \langle \nabla F_\Omega(\theta), \boldsymbol{d}\rangle + \frac{\gamma + B_\Omega\Lambda}{2}\|\boldsymbol{d}\|_2^2. \end{aligned}$$

Thus F_Ω is globally smooth with constant $\overline{L}_\Omega = \gamma + B_\Omega\Lambda$. We now apply the standard GD result for a γ-strongly convex, $\overline{L}_\Omega$-smooth objective (Nesterov, 1998). With step size $\eta = 2/(\overline{L}_\Omega + \gamma)$, each iteration contracts the distance to θ^*:

$$\|\theta_{t+1} - \theta^*\|_2 \le \frac{\overline{L}_\Omega - \gamma}{\overline{L}_\Omega + \gamma}\|\theta_t - \theta^*\|_2.$$

Substituting $\overline{L}_\Omega = \gamma + B_\Omega\Lambda$ and iterating yields Eq. 15. □

Step sizes and comparison. The same smoothness bound gives

$$F_\Omega(\theta - \eta\nabla F_\Omega(\theta)) \le F_\Omega(\theta) - \eta\left(1 - \frac{\eta\overline{L}_\Omega}{2}\right)\|\nabla F_\Omega(\theta)\|_2^2.$$

Hence every $0 < \eta < 2/\overline{L}_\Omega$ guarantees descent whenever the gradient is non-zero. The strong-convexity and smoothness bounds also give

$$\kappa(F_\Omega) \le \frac{\overline{L}_\Omega}{\gamma} = 1 + \frac{B_\Omega\Lambda}{\gamma}. \tag{43}$$

When features, loss scaling, and regularization are held fixed, a smaller B_Ω gives a wider guaranteed descent interval and a tighter condition-number bound. This is the comparison made in Eq. 16.

E.3 Rankmax Mapping Smoothness

As shown in Appendix D.4, the Rankmax Loss has negative curvature whenever a competitor is active. We now examine the Jacobian of its prediction map. Fix a positive class c and a support $\mathcal{P}$, the prediction Jacobian has active block

$$J_{\mathrm{rm}}\big|_{\mathcal{P}\times\mathcal{P}} = \frac{1}{Z_c}(I - \boldsymbol{q}\mathbf{1}^\top)(I - \mathbf{1}\boldsymbol{e}_c^\top), \tag{44}$$

For $h \ge 2$, submultiplicativity gives

$$\|J_{\mathrm{rm}}\|_2 \le \frac{\|I - \boldsymbol{q}\mathbf{1}^\top\|_2\|I - \mathbf{1}\boldsymbol{e}_c^\top\|_2}{Z_c} \le \frac{h + \sqrt{h}}{Z_c}, \tag{45}$$

which measures the sensitivity of the prediction map.

E.4 Smoothness of softmax approximation objectives

Sampled softmax. For a fixed sample, $\ell = -s_c + \log\sum_i a_i e^{s_i}$. Writing $p_i = a_i e^{s_i}/\sum_j a_j e^{s_j}$ gives

$$\nabla_{\boldsymbol{s}}^2\ell = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top, \quad \|\nabla_{\boldsymbol{s}}^2\ell\|_2 \le \tfrac{1}{2}.$$

NCE. Each logistic term has curvature $\sigma(z)(1-\sigma(z)) \le 1/4$. The Hessian is therefore bounded by $I/4$ in separate occurrence-logit coordinates. When repeated occurrences share a class score, the bound for class i becomes $b_i/4$, where b_i is its occurrence count.

Hierarchical softmax. A leaf loss sums cross-entropies along its path. In separate child-logit coordinates, the Hessian is block diagonal, with each node block bounded by $I/2$. A binary node using a single log-odds coordinate instead has curvature at most $1/4$.

Quadratic RG Loss. For the centered quadratic proxy,

$$\nabla^2_{\boldsymbol{s}} \ell_{\mathrm{quad}} = \frac{1}{C}\left(I - \frac{\mathbf{1}\mathbf{1}^\top}{C}\right), \quad \|\nabla^2_{\boldsymbol{s}} \ell_{\mathrm{quad}}\|_2 = \frac{1}{C} \quad (C \ge 2).$$

F Conditional Discrepancies and the Delta Method

F.1 Conditional bias-variance decomposition

Let $p_i = \frac{e^{s_i}}{\sum_j e^{s_j}}, \ell_* = -\log p_c$ and define

$$\mathrm{Bias}_A = \mathbb{E}_\omega[\widetilde{\ell}_A] - \ell_*, \quad \mathrm{Var}_A = \mathrm{Var}_\omega(\widetilde{\ell}_A).$$

Proposition 30 (Conditional bias–variance decomposition)**.** *For any loss $\widetilde{\ell}_A$,*

$$\mathbb{E}_\omega[(\widetilde{\ell}_A - \ell_*)^2] = \mathrm{Bias}_A^2 + \mathrm{Var}_A\,.$$

For a fixed dataset with independent draws across examples, the average-loss discrepancy has bias $N^{-1}\sum_n \mathrm{Bias}_{A,n}$ and variance $N^{-2}\sum_n \mathrm{Var}_{A,n}$.

Proof. The proof is trivial, which follows by expanding the squared error around $\mathbb{E}_\omega[\widetilde{\ell}_A]$. The aggregation formulas follow from linearity of expectation and independence. □

F.2 Delta-method expansion

Let $U_1, \dots, U_K$ be i.i.d. with $0 < a \le U_j \le b < \infty$ and mean μ, variance σ^2. Define $X_K = K^{-1}\sum_{j=1}^K U_j$. For function g with bounded derivatives on $[a, b]$, Taylor expansion around μ gives

$$\mathbb{E}[g(X_K)] = g(\mu) + \frac{g''(\mu)\sigma^2}{2K} + O(K^{-2}), \quad \mathrm{Var}(g(X_K)) = \frac{g'(\mu)^2\sigma^2}{K} + O(K^{-2}). \tag{46}$$

For a log-partition estimator, the resulting bias relative to $\log Z$ is

$$\mathrm{Bias} = g(\mu) - \log Z + \frac{g''(\mu)\sigma^2}{2K} + O(K^{-2}).$$

The first term is the plug-in discrepancy, while the second is the curvature correction.

F.3 Bias-variance Decomposition of softmax Approximations

Recall that $Z = \sum_i \exp s_i, p_i = \exp s_i / Z$ and $\ell_* = -s_c + \log Z = -\log p_c$. For each method, we compute $\mathrm{Bias} = \mathbb{E}[\widetilde{\ell}] - \ell_*$ and $\mathrm{Var} = \mathrm{Var}(\widetilde{\ell})$.

Uncorrected SSM Let $U_j = \exp s_{n_j}$ with $n_j \sim Q_c$, define $X_K = \frac{1}{K}\sum_{j=1}^K U_j, \mu_c = \mathbb{E}[U_j], \sigma_c^2 = \mathrm{Var}(U_j)$ and $M_K = \exp s_c + K\mu_c$. The loss is $\widetilde{\ell}_{\mathrm{simple}} = -s_c + g_K(X_K)$, where

$$g_K(x) = \log(\exp s_c + Kx), \quad g_K'(\mu_c) = \frac{K}{M_K}, \quad g_K''(\mu_c) = -\frac{K^2}{M_K^2}.$$

Since $\mathrm{Var}(X_K) = \sigma_c^2/K$, Eq. 46 gives

$$\mathbb{E}[g_K(X_K)] = \log M_K - \frac{K\sigma_c^2}{2M_K^2} + O(K^{-2}).$$

The required derivatives are uniformly bounded on the positive sampling range, as $g_K(x) = \log K + \log(x + \exp s_c/K)$. Subtracting $\log Z$ and applying the variance expansion yields

$$\boxed{\mathrm{Bias}_{\mathrm{simple}} = \log \frac{M_K}{Z} - \frac{K\sigma_c^2}{2M_K^2} + O(K^{-2}), \quad \mathrm{Var}_{\mathrm{simple}} = \frac{K\sigma_c^2}{M_K^2} + O(K^{-2}).} \tag{47}$$

Importance-corrected SSM. Define $U_j = \frac{\exp s_{n_j}}{Q_c(n_j)}, \widehat{Z}_c = \exp s_c + \frac{1}{K}\sum_{j=1}^K U_j$ and $\widetilde{\ell}_{\mathrm{SSM}} = -s_c + \log \widehat{Z}_c$. The first two moments are

$$\mathbb{E}[U_j] = Z - \exp s_c, \quad \sigma_U^2 = \sum_{i \neq c} \frac{\exp 2s_i}{Q_c(i)} - (Z - \exp s_c)^2.$$

Thus $\mathbb{E}[\widehat{Z}_c] = Z$ and $\mathrm{Var}(\widehat{Z}_c) = \sigma_U^2/K$. Expanding the logarithm around Z gives

$$\mathbb{E}[\log \widehat{Z}_c] = \log Z - \frac{\sigma_U^2}{2KZ^2} + O(K^{-2}), \quad \mathrm{Var}(\log \widehat{Z}_c) = \frac{\sigma_U^2}{KZ^2} + O(K^{-2}).$$

To express the coefficient in probability coordinates, write $p_{-c,i} = p_i/(1-p_c)$ for $i \neq c$. Then

$$\frac{\sigma_U^2}{Z^2} = \sum_{i \neq c} \frac{p_i^2}{Q_c(i)} - (1-p_c)^2 = (1-p_c)^2 \left(\sum_{i \neq c} \frac{p_{-c,i}^2}{Q_c(i)} - 1 \right) = (1-p_c)^2 \chi^2(\boldsymbol{p}_{-c} \| Q_c) =: D_c,$$

where $\chi^2(P\|Q) = \sum_i P_i^2/Q_i - 1$. Therefore

$$\boxed{\mathrm{Bias}_{\mathrm{SSM}} = -\frac{D_c}{2K} + O(K^{-2}), \quad \mathrm{Var}_{\mathrm{SSM}} = \frac{D_c}{K} + O(K^{-2}).} \tag{48}$$

NCE. Let $v_i = \exp u_i > 0$ be the modeled density and set

$$a_c = \log\left(1 + \frac{KQ(c)}{v_c}\right), \quad h_i = \log\left(1 + \frac{v_i}{KQ(i)}\right).$$

The loss is $\widetilde{\ell}_{\mathrm{NCE}} = a_c + \sum_{j=1}^K h_{n_j}$ with $n_j \sim Q$. Since a_c is fixed, we have

$$\mathbb{E}[\widetilde{\ell}_{\mathrm{NCE}}] = a_c + K\sum_i Q(i)h_i, \quad \mathrm{Var}(\widetilde{\ell}_{\mathrm{NCE}}) = K\left[\sum_i Q(i)h_i^2 - \left(\sum_i Q(i)h_i\right)^2\right].$$

Subtracting $\ell_* = -\log p_c$ gives the exact formulas

$$\boxed{\mathrm{Bias}_{\mathrm{NCE}} = a_c + K\mathbb{E}_Q[h_i] + \log p_c, \quad \mathrm{Var}_{\mathrm{NCE}} = K\,\mathrm{Var}_Q(h_i).} \tag{49}$$

For large K, $h_i = v_i/(KQ(i)) + O(K^{-2})$, so

$$\mathrm{Var}_{\mathrm{NCE}} = \frac{1}{K}\,\mathrm{Var}_Q\left(\frac{v_i}{Q(i)}\right) + O(K^{-2}).$$

HSM. Let $H_c > 0$ be the probability assigned to class c by the tree. The loss $\widetilde{\ell}_{\mathrm{HSM}} = -\log H_c$ is deterministic at fixed model outputs, and

$$\widetilde{\ell}_{\mathrm{HSM}} - \ell_* = -\log H_c + \log p_c.$$

Hence

$$\boxed{\mathrm{Bias}_{\mathrm{HSM}} = \log \frac{p_c}{H_c}, \quad \mathrm{Var}_{\mathrm{HSM}} = 0.} \tag{50}$$

Quadratic RG$^\times$ Loss. Let $\bar{s} = C^{-1}\sum_i s_i$, $\boldsymbol{z} = \boldsymbol{s} - \bar{s}\mathbf{1}$, and

$$\Psi_{\mathrm{quad}}(\boldsymbol{s}) = \log C + \bar{s} + \frac{1}{2C}\|\boldsymbol{z}\|_2^2, \quad \widetilde{\ell}_{\mathrm{quad}} = -s_c + \Psi_{\mathrm{quad}}(\boldsymbol{s}).$$

Since $\sum_i z_i = 0$, Taylor expansion at $\boldsymbol{z} = 0$ for fixed C gives

$$\frac{1}{C}\sum_i e^{z_i} = 1 + \frac{1}{2C}\sum_i z_i^2 + \frac{1}{6C}\sum_i z_i^3 + O(\|\boldsymbol{z}\|_2^4).$$

Using $\log(1+u) = u + O(u^2)$ and $u = O(\|\boldsymbol{z}\|_2^2)$, we obtain

$$\log Z = \bar{s} + \log C + \frac{1}{2C}\sum_i z_i^2 + \frac{1}{6C}\sum_i z_i^3 + O(\|\boldsymbol{z}\|_2^4).$$

Therefore

$$\boxed{\mathrm{Bias}_{\mathrm{quad}} = \Psi_{\mathrm{quad}} - \log Z = -\frac{1}{6C}\sum_i z_i^3 + O(\|\boldsymbol{z}\|_2^4), \quad \mathrm{Var}_{\mathrm{quad}} = 0.} \tag{51}$$

Uncentered RG^2 Loss. Removing the interaction term from RG$^\times$ gives

$$\Psi_{\mathrm{RG}^2} = \log C + \bar{s} + \frac{1}{2C}\|\boldsymbol{s}\|_2^2.$$

The identity $\|\boldsymbol{s}\|_2^2 = \|\boldsymbol{z}\|_2^2 + C\bar{s}^2$ yields $\Psi_{\mathrm{RG}^2} = \Psi_{\mathrm{quad}} + \bar{s}^2/2$. Consequently,

$$\boxed{\mathrm{Bias}_{\mathrm{RG}^2} = \frac{\bar{s}^2}{2} + \mathrm{Bias}_{\mathrm{quad}} = \frac{\bar{s}^2}{2} - \frac{1}{6C}\sum_i z_i^3 + O(\|\boldsymbol{z}\|_2^4), \quad \mathrm{Var}_{\mathrm{RG}^2} = 0.} \tag{52}$$

F.4 Training Complexity

Table 4 counts forward/backward operations for a linear head with feature dimension d. The bounds assume one positive label per example, while multiplying the costs by N gives the per-epoch costs.

Full-score methods. Computing all logits costs $O(Cd)$. Softmax and standard-simplex Rankmax add $O(C)$ work for loss and gradient evaluation, giving $O(Cd)$ overall. Sorting-based Sparsemax and 1.5-Entmax require $O(C\log C)$ threshold computation, giving $O(C(d+\log C))$. General α-Entmax with T bisection steps costs $O(C(d+T))$(Martins & Astudillo, 2016; Peters et al., 2019). Direct evaluation of the quadratic RG proxy also costs $O(Cd)$, but RG-ALS instead has solver-dependent per-sweep complexity.

Sampling and hierarchical methods. SSM and NCE evaluate one positive and K sampled scores, costing $O((K+1)d)$ when draws take constant time and only selected scores are computed. Additional candidate scoring, such as DNS, must be counted separately. HSM costs $O(dL)$ for a target path of length L, in which a balanced binary tree gives $L = O(\log C)$.

G Experimental Results

G.1 Settings

G.1.1 Dataset and Evaluation

Dataset. We evaluate our method on three public datasets: **MovieLens-1M(ML-1M)**, **Gowalla**, and **Amazon-Electronics(Electronics)**, collected from different real-world online platforms. **ML-1M** contains explicit user ratings on movies with a 1–5 scale. **Gowalla** is a location-based social networking service where users share their locations via check-ins. **Electronics** collects customers' reviews and ratings (1–5) on electronics products on the Amazon platform. For **ML-1M** and **Amazon-Electronics**, we treat items rated below 3 as negatives and the remaining ones as positives. We employ

Table 5: Statistics of datasets.

Dataset	#User	#Item	#Interact	Sparsity
ML-1M	6,038	3,307	835,789	95.81%
Gowalla	29,858	40,981	1,027,370	99.92%
Electronics	192,403	63,001	1,689,188	99.99%

the widely used k-core filtering strategy to remove users and items with fewer than 10 interactions. The detailed statistics after filtering are shown in Table 5.

Backbones.

MF (Rendle et al., 2009): Matrix Factorization is one of the most fundamental and widely adopted two-tower architectures. It learns linear embeddings for users and items to obtain their latent representations. Owing to its simplicity, rapid convergence, and strong scalability, **MF** is often employed as a baseline for benchmarking and for systematically comparing the performance of different loss functions.

SASRec (Kang & McAuley, 2018) represents a state-of-the-art sequential recommendation approach that applies self-attention mechanisms to model user–item interaction sequences. By capturing both short- and long-term dependencies, this model is capable of effectively representing dynamic user preferences. Its flexibility in handling variable-length sequences and its strong predictive accuracy for next-item recommendation make **SASRec** particularly suitable for studying the influence of sequential patterns in user behavior.

LightGCN (He et al., 2020) is a leading graph-based collaborative filtering model within the two-tower paradigm. It employs simplified graph convolutional networks to learn user and item representations by propagating collaborative signals over the user–item interaction graph. With its efficient message-passing mechanism, concise design, and strong empirical performance, **LightGCN** serves as a representative benchmark for evaluating the effectiveness of graph-based recommendation techniques.

Data Split. For each dataset, we divide the interactions of every user into training, validation, and test subsets with a ratio of $\{0.8, 0.1, 0.1\}$. The validation subset is employed to monitor and tune model performance during training, while the test subset provides the basis for the final comparative evaluation.

G.1.2 Evaluation Metrics

Following prior discussion, we adopt both classification-oriented and ranking-oriented metrics to comprehensively evaluate model performance. Specifically, we report **P@k**, **R@k**, and **N@k** with $k = 20$ across all datasets. These metrics jointly capture both classification and ranking accuracy, which are critical for large-scale recommendation scenarios.

G.1.3 Baselines

We compare **Softmax-family** surrogate losses discussed in our prior sections:

- **Exact losses: Softmax**, **Sparsemax** (Martins & Astudillo, 2016), α**-Entmax** (Peters et al., 2019)($\alpha = 1.5$ as representative), and **Rankmax** (Kong et al., 2020) (with simplest $\Delta_{n,1}$ form).

- **Approximate losses: SSM** (Jean et al., 2015), **NCE** (Gutmann & Hyvärinen, 2010), **HSM** (Morin & Bengio, 2005), and **RG** (Pu et al., 2024).

It is worth highlighting that **HSM** and **RG** differ from other Softmax-family losses in terms of their structural and optimization requirements. Specifically, **HSM** leverages a hierarchical decomposition of the output space through a Huffman tree, which tightly couples the loss with the model architecture. This design makes it difficult to seamlessly integrate **HSM** into diverse backbones such as SASRec or LightGCN. To ensure a fair comparison, we therefore report **HSM** results only under the MF backbone. Similarly, **RG** employs an optimization strategy based on Alternating Least Squares (ALS),

rather than the commonly used gradient descents. This reliance on ALS prevents its straightforward adaptation to sequential or graph-based recommenders. Consequently, we restrict the evaluation of **RG** to the MF backbone and present its results exclusively in the corresponding tables for direct comparison.

G.1.4 Implementation Details

To ensure fair and consistent comparisons, all backbone models are implemented with an identical architectural configuration across different loss functions. The embedding dimension is fixed at 64 for all models. For **SASRec**, we adopt two self-attention blocks with a dropout rate of 0.5, while for **LightGCN**, a two-layer graph convolutional network is employed. All methods are trained on a single NVIDIA RTX-3090 GPU with 24GB of memory.

All models and baselines(except for RG) are trained under aligned protocols to ensure fairness:

- Optimizer: Adam (Kingma & Ba, 2015) with batch size of 2048.

- Learning rate: tuned over $\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$ using validation N@20, results are reported in G.2.

- For sampling-based approximations, we choose $k = 100$ for comparision with exact methods and sweep the number of negative samples $k \in \{5, 10, 50, 100\}$ in further discussions.

G.2 Q1: Accuracy under aligned protocols

Tables 6 and 7 report the results of different learning rate selections under MF and SASRec, respectively. Notice that RG utilizes ALS optimization, which need not fine-tuning on learning rate.

The results reveal several consistent patterns regarding the behavior of different softmax-family losses under aligned training protocols. The key findings can be summarized as follows:

- *Finding 1.* Softmax performs well at relatively large learning rates. Across the MF and SASRec comparisons, Softmax achieves its best N@20 at 10^{-3} or 5×10^{-4}, while Sparsemax and Entmax-1.5 show more varied learning-rate preferences. This pattern complements our smoothness analysis: Softmax's smaller worst-case curvature bound gives a wider certified descent interval in the fixed-feature setting. The experiments highlight learning-rate selection as an important part of comparing dense and sparse objectives.
- *Finding 2.* Rankmax highlights the value of label-adaptive negative selection. It achieves strong results with MF and SASRec, particularly on ML-1M and Gowalla. Our gradient analysis identifies its distinctive mechanism: every margin-violating competitor receives the same downward score update, while sufficiently separated negatives receive none. Rankmax's competitive performance makes this margin-selective allocation of training signals a promising design choice for recommendation.
- *Finding 3.* SSM achieves stronger NDCG performance than NCE under the tested uniform-sampling configurations. At $K = 100$, SSM attains higher best-reported N@20 in all six MF/SASRec dataset combinations, with particularly clear advantages on ML-1M and Gowalla. Our analysis distinguishes the two training targets: SSM approximates the softmax normalization, whereas NCE solves a data–noise discrimination problem. The results favor the normalization-based approach for NDCG in these comparisons. The following sampling ablations further examine how sample count and negative selection affect both methods.
- *Finding 4.* Sparse objectives have backbone-dependent strengths. Sparsemax and Entmax-1.5 can match or outperform Softmax with MF, but achieve lower best-reported N@20 on all three SASRec datasets. With LightGCN on ML-1M (Table 8), Softmax and SSM also outperform the sparse alternatives, including Rankmax. These differences reinforce a central distinction in our analysis: consistency describes the conditional optimum, while support selection determines which negative coordinates receive training signals. The results make backbone compatibility a key consideration when choosing a sparse objective.
- *Finding 5.* RG is a competitive non-sampling alternative for MF. It outperforms HSM on every reported metric across the three datasets and achieves the best overall results on Electronics. Our analysis places these methods in distinct categories: HSM factorizes output probabilities through a tree, while RG replaces the objective with a quadratic surrogate

Table 6: MF performance under different learning rates. **Bold** denotes the best performance of a method under different lr. **Blue** indicates the overall best performance under each data–metric pair.

Loss	Metric	ML-1M			Electronics			Gowalla		
		lr=1e-3	lr=5e-4	lr=1e-4	lr=1e-3	lr=5e-4	lr=1e-4	lr=1e-3	lr=5e-4	lr=1e-4
Softmax	N@20	**0.2661**	0.2658	0.2482	**0.0235**	0.0231	0.0146	0.0947	**0.0953**	0.0571
	P@20	**0.1465**	0.1462	0.1391	**0.0027**	0.0027	0.0018	0.0188	**0.0190**	0.0122
	R@20	**0.2937**	0.2930	0.2596	**0.0496**	0.0494	0.0333	**0.1716**	0.1704	0.0970
Sparsemax	N@20	0.2353	**0.2536**	0.2463	0.0214	**0.0240**	0.0166	0.0782	0.0948	**0.0992**
	P@20	0.1228	**0.1345**	0.1344	0.0023	**0.0027**	0.0019	0.0150	0.0180	**0.0203**
	R@20	0.2620	**0.2769**	0.2651	0.0421	**0.0491**	0.0347	0.1375	0.1636	**0.1739**
Entmax-1.5	N@20	**0.2643**	0.2631	0.2557	0.0244	**0.0255**	0.0187	0.0704	0.0733	**0.0759**
	P@20	0.1410	**0.1426**	0.1383	0.0028	**0.0029**	0.0022	0.0148	0.0156	**0.0161**
	R@20	**0.2897**	0.2852	0.2778	0.0510	**0.0529**	0.0410	0.1232	0.1297	**0.1353**
Rankmax	N@20	**0.2851**	0.2835	0.2621	0.0218	**0.0225**	0.0145	0.1030	**0.1031**	0.0654
	P@20	**0.1543**	0.1535	0.1452	0.0025	**0.0026**	0.0018	0.0206	**0.0206**	0.0141
	R@20	**0.3005**	0.2985	0.2710	0.0461	**0.0472**	0.0331	0.1790	**0.1794**	0.1132
SSM	N@20	0.2644	**0.2645**	0.2465	0.0236	**0.0246**	0.0147	**0.0848**	0.0816	0.0539
	P@20	**0.1462**	0.1461	0.1388	0.0028	**0.0028**	0.0018	**0.0173**	0.0169	0.0116
	R@20	0.2908	**0.2910**	0.2569	0.0509	**0.0525**	0.0336	**0.1522**	0.1458	0.0955
NCE	N@20	**0.2423**	0.2420	0.2227	**0.0238**	0.0232	0.0150	**0.0518**	0.0514	0.0451
	P@20	**0.1374**	0.1364	0.1280	**0.0029**	0.0028	0.0018	**0.0110**	0.0109	0.0094
	R@20	0.2661	**0.2687**	0.2283	**0.0528**	0.0520	0.0339	**0.0903**	0.0899	0.0794
HSM	N@20	0.1120	0.1193	**0.1202**	**0.0134**	0.0129	0.0123	**0.0512**	0.0496	0.0464
	P@20	0.0739	0.0736	**0.0742**	**0.0016**	0.0015	0.0014	**0.0114**	0.0111	0.0104
	R@20	**0.1497**	0.1236	0.1249	**0.0297**	0.0283	0.0267	**0.0938**	0.0915	0.0845
RG	N@20		**0.2785**			**0.0274**			**0.0894**	
	P@20		**0.1521**			**0.0030**			**0.0200**	
	R@20		**0.2945**			**0.0563**			**0.1476**	

Table 7: SASRec results under different learning rates. **Bold** denotes the best performance of a method under different lr. **Blue** indicates the overall best performance under each data–metric pair.

Loss	Metric	ML-1M			Electronics			Gowalla		
		lr=1e-3	lr=5e-4	lr=1e-4	lr=1e-3	lr=5e-4	lr=1e-4	lr=1e-3	lr=5e-4	lr=1e-4
Softmax	N@20	**0.1647**	0.1643	0.1630	**0.0355**	0.0351	0.0348	0.0459	**0.0470**	0.0462
	P@20	**0.0180**	0.0178	0.0177	**0.0036**	0.0036	0.0036	0.0048	**0.0049**	0.0049
	R@20	**0.3599**	0.3566	0.3546	**0.0728**	0.0727	0.0723	0.0968	**0.0988**	0.0977
Sparsemax	N@20	0.1480	**0.1490**	0.0229	0.0170	**0.0246**	0.0117	0.0338	0.0347	**0.0351**
	P@20	0.0158	**0.0160**	0.0027	0.0020	**0.0027**	0.0014	0.0037	0.0038	**0.0038**
	R@20	0.3165	**0.3198**	0.0542	0.0399	**0.0540**	0.0276	0.0746	0.0754	**0.0760**
Entmax-1.5	N@20	0.0952	0.0932	**0.0972**	0.0171	**0.0172**	0.0167	0.0154	**0.0154**	0.0153
	P@20	0.0111	0.0108	**0.0113**	0.0020	**0.0020**	0.0019	0.0017	0.0017	**0.0017**
	R@20	0.2229	0.2158	**0.2264**	0.0398	**0.0407**	0.0389	0.0337	0.0336	**0.0338**
Rankmax	N@20	0.1753	**0.1789**	0.1729	**0.0357**	0.0353	0.0344	0.0469	0.0467	**0.0474**
	P@20	0.0185	**0.0188**	0.0183	**0.0036**	0.0036	0.0035	0.0049	0.0049	**0.0050**
	R@20	0.3695	**0.3756**	0.3667	**0.0721**	0.0716	0.0694	0.0984	0.0983	**0.0993**
SSM	N@20	**0.1524**	0.1501	0.1486	0.0251	**0.0260**	0.0134	0.0394	**0.0398**	0.0394
	P@20	**0.0171**	0.0169	0.0166	0.0028	**0.0029**	0.0017	0.0044	**0.0044**	0.0044
	R@20	**0.3428**	0.3385	0.3319	0.0567	**0.0576**	0.0331	0.0872	**0.0873**	0.0873
NCE	N@20	0.1274	**0.1325**	0.1281	0.0173	0.0171	**0.0175**	**0.0309**	0.0304	0.0262
	P@20	0.0153	**0.0158**	0.0151	0.0020	0.0019	**0.0020**	**0.0035**	0.0034	0.0029
	R@20	0.3062	**0.3158**	0.3029	0.0394	0.0384	**0.0399**	**0.0702**	0.0684	0.0581

suited to ALS. Their contrasting performance highlights the practical importance of combining the surrogate with an appropriate parameterization and optimizer, and supports RG's effectiveness in the tested factorization setting.

Table 8: LightGCN results on ML-1M under different learning rates. **Bold** denotes the best performance of each method across learning rates.

Loss	N@20			P@20			R@20		
	lr=1e-3	lr=5e-4	lr=1e-4	lr=1e-3	lr=5e-4	lr=1e-4	lr=1e-3	lr=5e-4	lr=1e-4
Softmax	**0.2279**	0.2150	0.1268	**0.1316**	0.1239	0.0740	**0.2488**	0.2194	0.1281
Sparsemax	**0.1695**	0.1496	0.1226	**0.0983**	0.0881	0.0748	**0.1755**	0.1544	0.1278
Entmax-1.5	0.1208	0.1207	**0.1212**	0.0731	0.0731	**0.0734**	0.1272	0.1272	**0.1280**
Rankmax	0.1208	0.1208	**0.1212**	0.0731	0.0732	**0.0734**	0.1273	0.1273	**0.1277**
SSM	**0.2358**	0.2155	0.1281	**0.1339**	0.1242	0.0738	**0.2523**	0.2234	0.1281

Overall, these experiments demonstrate that Softmax performs well at relatively large learning rates, Rankmax's margin-selective updates accompany strong MF and SASRec results, and SSM achieves higher NDCG than NCE in the uniform-sampling comparisons. Sparsemax and Entmax show clear backbone-dependent strengths, while RG outperforms HSM in the tested MF setting. These findings complement our analysis of smoothness, support-dependent gradients, and alternative training objectives, highlighting the importance of matching the loss and training configuration to the backbone.

G.3 Q2: Bias-variance decomposition

Our second set of experiments directly examines the bias–variance trade-offs predicted by our theoretical analysis. The results are summarized in Tables 9, 10, 11, and 12. We highlight two key findings:

Table 9: MF results under different negative sample sizes k. **Bold** denotes the best performance of each method across k within each dataset.

Loss	N@20				P@20				R@20			
	k=5	k=10	k=50	k=100	k=5	k=10	k=50	k=100	k=5	k=10	k=50	k=100
ML-1M												
SSM	0.2504	0.2554	0.2630	**0.2644**	0.1406	0.1430	0.1452	**0.1462**	0.2738	0.2821	0.2905	**0.2908**
NCE	0.2403	0.2410	0.2409	**0.2423**	0.1360	0.1360	0.1362	**0.1374**	0.2643	0.2650	**0.2681**	0.2661
Electronics												
SSM	0.0235	0.0237	0.0245	**0.0246**	0.0028	0.0028	0.0028	**0.0028**	0.0514	0.0516	0.0520	**0.0525**
NCE	0.0232	0.0235	0.0235	**0.0238**	0.0028	0.0028	**0.0029**	0.0029	0.0523	0.0525	**0.0530**	0.0528
Gowalla												
SSM	0.0611	0.0675	0.0806	**0.0848**	0.0129	0.0143	0.0167	**0.0173**	0.1074	0.1197	0.1450	**0.1522**
NCE	0.0514	0.0515	0.0516	**0.0518**	0.0109	0.0110	0.0109	**0.0110**	0.0893	0.0900	0.0899	**0.0903**

Table 10: SASRec results under different negative sample sizes k. **Bold** denotes the best performance of each method across k within each dataset.

Loss	N@20				P@20				R@20			
	k=5	k=10	k=50	k=100	k=5	k=10	k=50	k=100	k=5	k=10	k=50	k=100
ML-1M												
SSM	0.0179	0.1248	0.1400	**0.1524**	0.0024	0.0149	0.0160	**0.0171**	0.0480	0.2978	0.3203	**0.3428**
NCE	0.0174	0.1024	0.1291	**0.1325**	0.0024	0.0124	0.0153	**0.0158**	0.0474	0.2473	0.3064	**0.3158**
Electronics												
SSM	0.0148	0.0170	0.0231	**0.0260**	0.0016	0.0017	0.0024	**0.0029**	0.0320	0.0344	0.0485	**0.0576**
NCE	0.0157	0.0158	**0.0214**	0.0175	0.0017	0.0017	**0.0023**	0.0020	0.0340	0.0342	**0.0464**	0.0399
Gowalla												
SSM	0.0301	0.0308	0.0368	**0.0398**	0.0033	0.0034	0.0041	**0.0044**	0.0668	0.0689	0.0824	**0.0873**
NCE	0.0270	0.0261	0.0292	**0.0309**	0.0029	0.0028	0.0030	**0.0035**	0.0584	0.0561	0.0595	**0.0702**

- *Finding 6.* Increasing the negative sample count consistently benefits SSM, while NCE shows a more varied response. As shown in Tables 9 and 10, SSM's N@20 increases with K across all dataset–backbone combinations, and P@20 and R@20 are nondecreasing at the reported precision. NCE changes relatively little with MF, but benefits substantially from more negatives with SASRec on ML-1M; on Electronics, its SASRec performance peaks at $K = 50$. Our analysis provides a quantitative explanation for SSM's approximation behavior: at fixed scores and proposal, the leading bias magnitude and variance decrease at rate $1/K$. The experiments establish sample count as an important tuning choice for predictive performance.
- *Finding 7.* Hard-negative selection improves both SSM and NCE. Tables 11 and 12 show that DNS outperforms uniform sampling on every reported metric across the tested datasets and backbones. By retaining higher-scoring candidates, DNS directs training toward more competitive negatives. This complements our analysis of proposal-dependent approximation error and the margin-selective mechanism of Rankmax: the selection of negative candidates is a central part of surrogate design. The gains support negative selection as an effective way to improve sampling-based training.

These ablations identify sample count and negative selection as two practical controls for improving recommendation quality. The bias-variance analysis in Appendix F explains how sampling configurations affect loss approximation, while the experiments demonstrate their importance for predictive performance.

Table 11: MF results under different proposal distributions Q (DNS draws Top-100 from $k = 500$ candidates). **Bold** denotes the best performance of each method across proposal distributions within each dataset.

Loss	N@20		P@20		R@20	
	Uniform	DNS	Uniform	DNS	Uniform	DNS
ML-1M						
SSM	0.2644	**0.2739**	0.1462	**0.1478**	0.2908	**0.2991**
NCE	0.2423	**0.2769**	0.1374	**0.1498**	0.2661	**0.3002**
Amazon-Electronics						
SSM	0.0246	**0.0255**	0.0028	**0.0029**	0.0525	**0.0536**
NCE	0.0238	**0.0273**	0.0029	**0.0032**	0.0528	**0.0599**
Gowalla						
SSM	0.0848	**0.0908**	0.0173	**0.0180**	0.1522	**0.1651**
NCE	0.0518	**0.0599**	0.0110	**0.0129**	0.0903	**0.1048**

G.4 Q3: Training dynamics and efficiency

We further examine the optimization dynamics of different normalization losses by plotting NDCG@20 against both training epochs and cumulative wall-clock time. The results are shown in Figures 1 and 2. The findings are:

- *Finding 8.* Early progress with sparse objectives depends on the backbone. The NDCG@20-versus-epoch curves in Figures 1 and 2 show that Sparsemax and Entmax reach their performance plateaus in relatively few epochs with MF. This early-epoch advantage is less pronounced with SASRec, where their predictive performance also falls below Softmax. Together with the accuracy comparisons, these trajectories highlight the interaction between support-dependent training signals and the backbone.
- *Finding 9.* Progress per epoch does not translate directly into wall-clock efficiency. The NDCG@20-versus-time curves in Figures 1 and 2 show that the early-epoch progress of sparse methods does not consistently yield faster training to competitive accuracy. Our complexity analysis identifies the additional work involved: sparse heads evaluate full scores

Table 12: SASRec results under different proposal distributions Q (DNS draws Top-100 from $k = 500$ candidates). **Bold** denotes the best performance of each method across proposal distributions within each dataset.

Loss	N@20		P@20		R@20	
	Uniform	DNS	Uniform	DNS	Uniform	DNS
ML-1M						
SSM	0.1524	**0.1570**	0.0171	**0.0176**	0.3428	**0.3497**
NCE	0.1325	**0.1457**	0.0158	**0.0173**	0.3158	**0.3474**
Amazon-Electronics						
SSM	0.0260	**0.0265**	0.0029	**0.0030**	0.0576	**0.0588**
NCE	0.0175	**0.0201**	0.0020	**0.0024**	0.0399	**0.0459**
Gowalla						
SSM	0.0398	**0.0418**	0.0044	**0.0045**	0.0873	**0.0895**
NCE	0.0309	**0.0387**	0.0035	**0.0042**	0.0702	**0.0836**

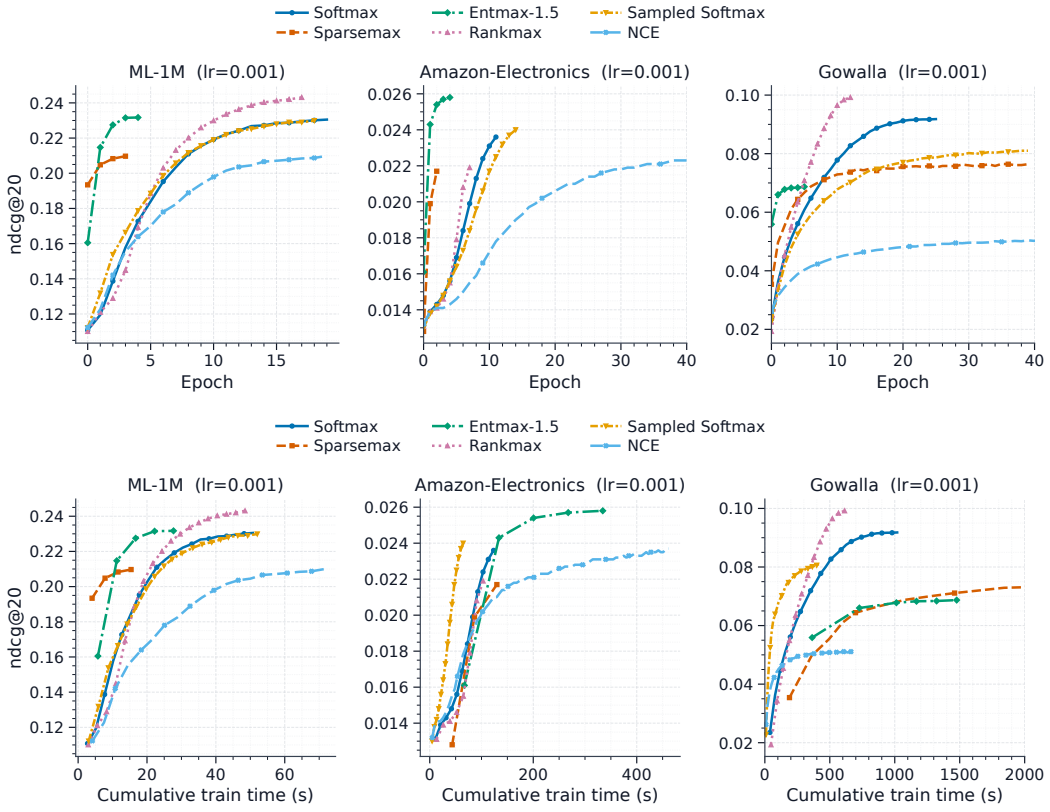

Figure 1: MF backbone: NDCG@20 curves with respect to (top) training epochs and (bottom) cumulative wall-clock time.

and then compute their support thresholds. Sampling-based methods instead evaluate a smaller candidate set and can reach competitive results in less elapsed time, particularly with MF. These comparisons highlight the practical value of reducing candidate-scoring cost.

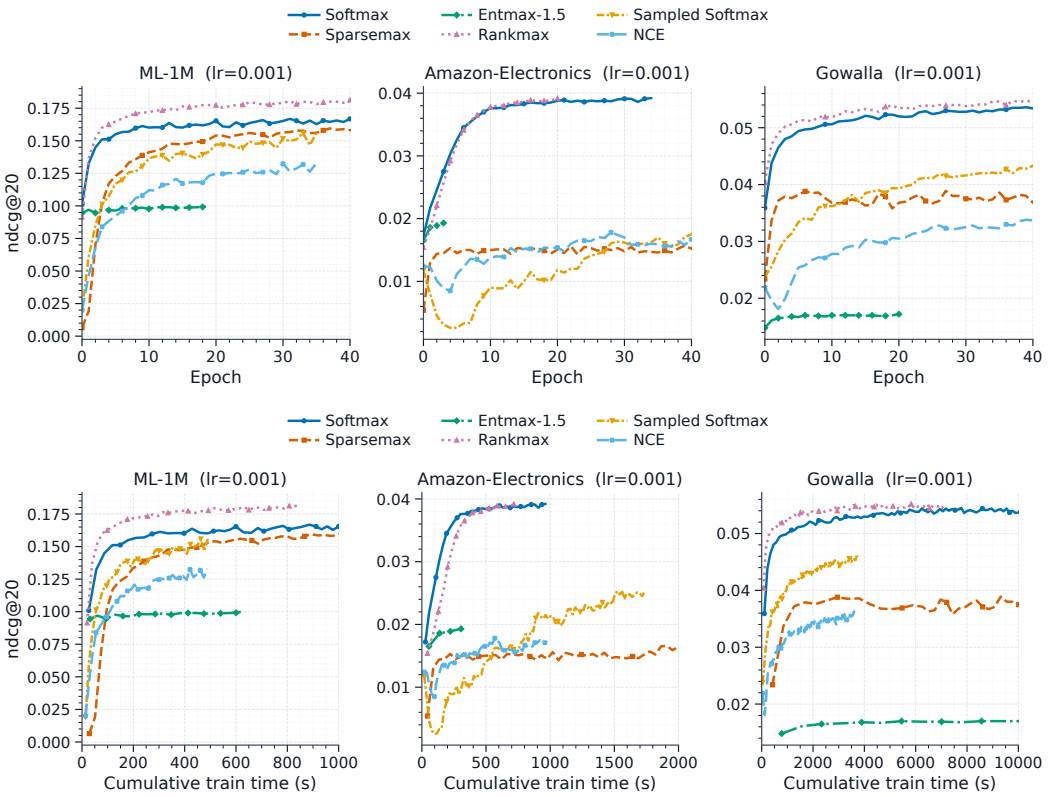

Figure 2: SASRec backbone: NDCG@20 curves with respect to (top) training epochs and (bottom) cumulative wall-clock time.

These findings connect backbone-dependent training behavior with the computational costs of different objectives. They emphasize predictive quality at a given time budget as a practical criterion for choosing between full-score and sampling-based methods.

G.5 Gradient-magnitude diagnostics

Figure 3 visualizes the magnitudes of negative-class score gradients across a batch. Softmax provides dense negative-class gradients, whereas Sparsemax, Entmax, and Rankmax exhibit selective gradient coverage, with black regions indicating zero entries. These patterns illustrate the mechanisms established in Propositions 12 and 15: Sparsemax and Entmax suppress negatives within their prediction support, while Rankmax updates competitors within its label-dependent margin. The heatmaps thus show how different objectives allocate training signals across negative candidates.

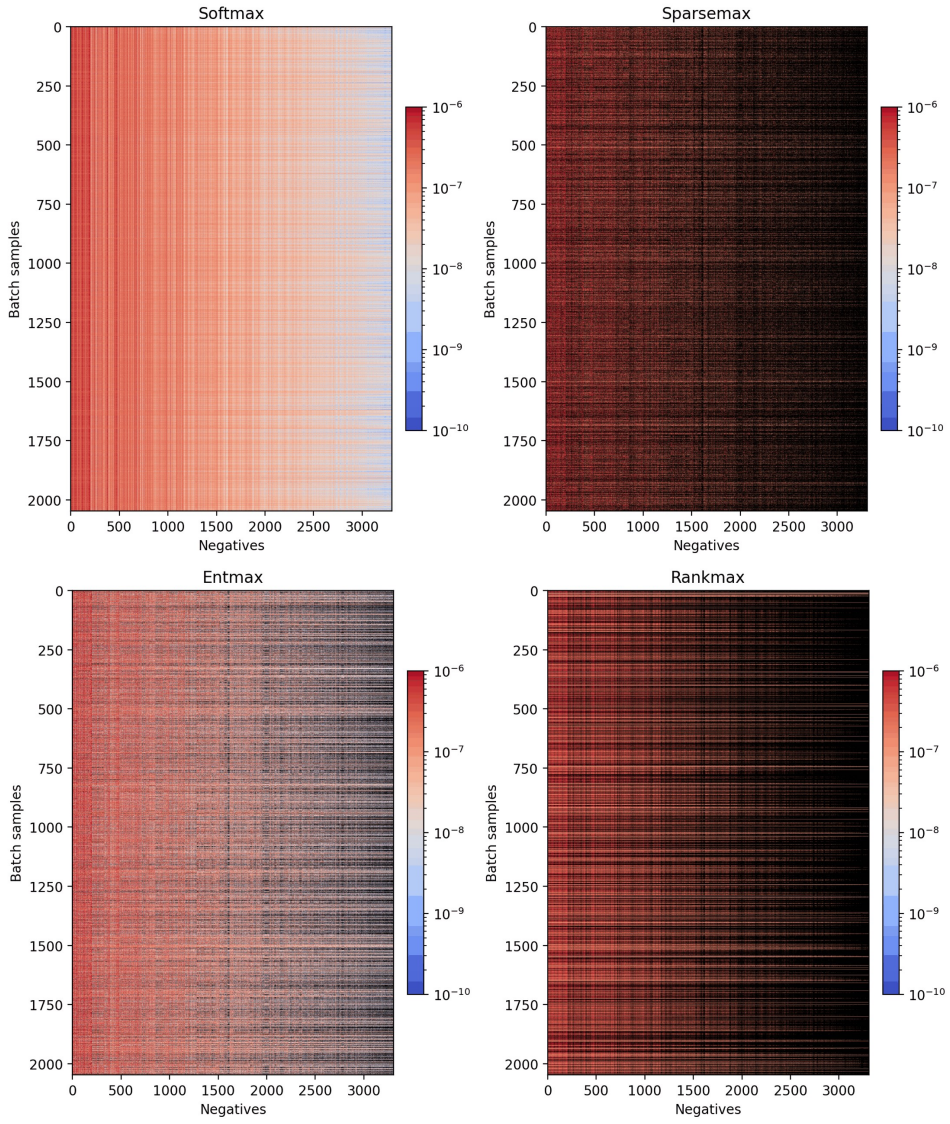

Figure 3: Heatmaps of negative-class score-gradient magnitudes across a batch for full-score objectives. Black regions indicate zero gradients.