Size- and Dispersion-Corrected Two-Level Softmax Sampling

Walid Bendada [1] **Guillaume Salha-Galvan** [2]

Abstract

Sampling from a softmax distribution is a fundamental operation in machine learning, but its linear complexity in the number of items makes exact sampling impractical at scale. Two-level softmax (2LS) sampling is a popular alternative enabling sublinear-time sampling. Assuming items are partitioned into clusters, 2LS first samples a cluster and then an item within it. In this paper, we show that, despite its advantages, 2LS introduces systematic and undesirable sampling biases, which arise from misweighting clusters by ignoring both cluster size imbalance and intra-cluster similarity dispersion. We propose two sampling methods, Size-Corrected 2LS (S-2LS) and Size- and Dispersion-Corrected 2LS (SD-2LS), which correct these biases and provide provably better softmax approximations with negligible to non-existent computational overhead. In-depth experiments on five large-scale datasets validate the improved sampling properties of our methods. *This workshop paper is under review for presentation at an international conference.*

1. Introduction

Sampling items according to their similarity to a query vector in a shared embedding space is a fundamental operation in many machine learning applications, ranging from language modeling to information retrieval and recommendation (Bishop, 2006; Chen et al., 2019a; Covington et al., 2016; Vaswani et al., 2017). A standard and widely used approach is to sample from a *softmax distribution*, where each item is weighted by the exponential of its similarity to the query (Goodfellow et al., 2016). However, computing and sampling from the full softmax distribution scales linearly with the number of items and quickly becomes intractable in large-scale settings (Bendada et al., 2025; Chen et al., 2016; Jean et al., 2015). As an illustration, consider a recommender system where users and items (e.g., products, videos, or music tracks) are embedded in a shared vector space, and recommendations are generated by sampling items according to their similarity to a query user embedding (Covington et al., 2016; He et al., 2017). Modern recommender systems typically operate over millions of candidate items, making exact softmax sampling impractical (Bendada et al., 2025).

A popular alternative to address this issue is *two-level softmax* (2LS) sampling (Chen et al., 2022; 2025; Goodman, 2001; Johnson et al., 2019; Subbiah et al., 2025; Tranheden et al., 2026). Assuming items are partitioned into clusters, 2LS proceeds in two steps: (1) sampling a cluster according to a softmax over cluster representatives, e.g., centroids, and (2) sampling an item within the selected cluster via a softmax over its items. As detailed in Section 2, 2LS has been widely adopted across applications and offers several advantages. First and foremost, this decomposition enables sublinear-time sampling with respect to the number of items (Jegou et al., 2010; Morin & Bengio, 2005). Second, sampling probabilities still account for item–query similarity, as in standard softmax, unlike other scalable alternatives such as ε-greedy methods (Sutton & Barto, 2018). Third, 2LS allows sampling over the full set of items, in contrast to truncated softmax approaches that restrict sampling to a limited subset of items (Bendada et al., 2025).

In this paper, we show that, despite these advantages, 2LS exhibits systematic biases that, to our knowledge, have never been formally characterized. Backed by rigorous analysis, we propose corrected 2LS variants with improved properties. Specifically, our contributions are:

- We show that 2LS oversamples or undersamples items compared to softmax and, in particular, breaks its invariance: while softmax assigns identical sampling probabilities to items with equal similarity to the query, 2LS can assign substantially different probabilities to such items. This mismatch is undesirable in practice: in the recommendation example above, it implies that items equally relevant to a user may be recommended with different probabilities.
- We prove that 2LS systematically misweights clusters

[1]Spotify, London, United Kingdom (Part of this research was conducted while the author was at Deezer Research, Paris, France). [2]SJTU Paris Elite Institute of Technology, Shanghai, China. Correspondence to: Walid Bendada <walidb@spotify.com>, Guillaume Salha-Galvan <gsalhagalvan@sjtu.edu.cn>.

Accepted for presentation at the ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling (SPIGM), Seoul, South Korea.

along two complementary axes. First, it ignores cluster size imbalance, oversampling small clusters and undersampling large ones relative to their true softmax mass. Second, it ignores intra-cluster similarity dispersion, undersampling clusters whose items exhibit high variability in their similarity to the query.

- We introduce Size-Corrected 2LS (S-2LS) and Size- and Dispersion-Corrected 2LS (SD-2LS), two novel sampling methods that explicitly aim to address these issues. We demonstrate that these corrections effectively mitigate the systematic biases of standard 2LS while incurring negligible to non-existent additional computational overhead.

- We report experiments on five large-scale datasets, consistently validating the improved sampling properties of our approaches and their higher fidelity to the softmax distribution. Our code is available at: https://github.com/twolevelsoftmaxanon-creator/2ls.

In summary, our results show that S-2LS and SD-2LS consistently outperform 2LS in approximating softmax sampling, and we recommend their consistent use in future work. This paper is organized as follows. Section 2 introduces softmax and 2LS sampling more formally. Section 3 analyzes the biases of 2LS. Section 4 introduces and studies our proposed S-2LS and SD-2LS corrections. Section 5 presents our experiments, and Section 6 concludes.

2. Preliminaries

2.1. Softmax Sampling

Notation Throughout this paper, we consider a set $\mathcal{I} = \{1, 2, \dots, N\}$ of $N > 1$ items. Each item $i \in \mathcal{I}$ is represented by a low-dimensional vector $x_i \in \mathbb{R}^d$, i.e., an *embedding*, with $d \ll N$. We do not make assumptions regarding the specific methods used to learn these embeddings. We also consider a query vector $q \in \mathbb{R}^d$ in the same embedding space. Our goal is to sample items according to their similarity to the query, as measured by the dot product $q^\top x_i$ for all $i \in \mathcal{I}$.

Softmax Sampling A standard approach is to sample from the *softmax distribution* (Goodfellow et al., 2016), defined as:

$$\forall i \in \mathcal{I},\ p(i \mid q) = \frac{\exp\left(q^\top x_i/\tau\right)}{\sum_{j=1}^{N} \exp\left(q^\top x_j/\tau\right)} \in [0, 1], \quad (1)$$

where $\tau > 0$ is a temperature parameter controlling the sharpness of the distribution. The softmax distribution assigns higher probability to items with larger dot product similarity to the query. It has become ubiquitous in machine learning, appearing in applications such as classification, language modeling, reinforcement learning, information retrieval, and recommendation (Bendada et al., 2025; Bishop, 2006; Chen et al., 2019a; Covington et al., 2016; Sutton & Barto, 2018; Vaswani et al., 2017).

Sampling Complexity Sampling from the softmax distribution requires computing $q^\top x_j$ scores for all $j \in \mathcal{I}$, from which both the numerators and the normalization constant are derived. Consequently, exact sampling has $\mathcal{O}(dN)$ time complexity per query (Chen et al., 2016; Jean et al., 2015), which quickly becomes prohibitive in large-scale settings, such as the recommendation example with millions of items discussed in the introduction (Bendada et al., 2025). This challenge has motivated approximate methods, including 2LS, presented next.

2.2. Two-Level Softmax (2LS) Sampling

Notation In addition to the notation of Section 2.1, we assume that $\mathcal{I}$ is partitioned into $K < N$ disjoint clusters, and denote these clusters by $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$. We have $\bigcup_{k=1}^{K} \mathcal{C}_k = \mathcal{I}$ and $\mathcal{C}_k \cap \mathcal{C}_{k'} = \emptyset$ for $k \neq k'$. For each cluster $\mathcal{C}_k$, we define its centroid as $\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} x_i \in \mathbb{R}^d$.

Two-Level Softmax (2LS) Sampling The objective of 2LS (Chen et al., 2025; Goodman, 2001; Johnson et al., 2019; Subbiah et al., 2025) is to first sample a cluster index $k \in \{1, \dots, K\}$, and then an item $i \in \mathcal{C}_k$ *within the selected cluster*, thereby factorizing the sampling distribution as $p_{\text{2LS}}(i \mid q) = p_{\text{2LS}}(k \mid q)\, p_{\text{2LS}}(i \mid k, q)$. Clusters are sampled using a softmax over centroids, while items are sampled using another softmax restricted to the selected cluster:

$$p_{\text{2LS}}(k \mid q) = \frac{\exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^{K} \exp(q^\top \mu_{k'}/\tau)}, \text{ for } k \in \{1, \dots, K\}, \quad (2)$$

$$p_{\text{2LS}}(i \mid k, q) = \frac{\exp(q^\top x_i/\tau)}{\sum_{j \in \mathcal{C}_k} \exp(q^\top x_j/\tau)}, \text{ for } i \in \mathcal{C}_k. \quad (3)$$

Sampling Complexity Sampling a cluster index requires computing a softmax over K centroids, with complexity $\mathcal{O}(dK)$. Sampling an item within the selected cluster $\mathcal{C}_k$ then requires computing a softmax over $|\mathcal{C}_k|$ items, with complexity $\mathcal{O}(d|\mathcal{C}_k|)$. Therefore, the cost per query of 2LS is $\mathcal{O}(dK + d|\mathcal{C}_k|)$, which is sublinear in N when $K \ll N$ and clusters are sufficiently balanced. In particular, if clusters have comparable sizes, $|\mathcal{C}_k| \approx N/K$, yielding an average complexity of $\mathcal{O}(dK + dN/K)$. This is minimized when $K \approx \sqrt{N}$, leading to a complexity of $\mathcal{O}(d\sqrt{N})$.

The item set partition, required prior to sampling, can be obtained using clustering procedures such as k-means on embeddings (Bishop, 2006). While this partitioning step

can be costly, it is typically performed offline and only once. In many applications, the same partition is reused across many sampling operations: in recommendation, items are repeatedly sampled for many queries (Subbiah et al., 2025); in NLP, tokens are repeatedly sampled for sequence generation (Shao et al., 2025). In such regimes, the dominant cost is thus the *inference complexity for a fixed partition*, making the sublinearity derived above a key advantage.

2.3. Related Work

Applications of 2LS and Related Methods 2LS and closely related variants have been widely adopted in large-scale systems where sampling over large item sets renders exact softmax computation intractable. Applications include (large) language model inference for text generation with large vocabularies (Chen et al., 2025; Liu et al., 2025; Shao et al., 2025; Tranheden et al., 2026; Zhang et al., 2025), machine translation (Chen et al., 2019b; Zhang et al., 2025), text classification (Chen et al., 2025), image classification (Liao et al., 2019), and item recommendation (Chen et al., 2022; 2025; Subbiah et al., 2025). Despite differences in terminology and formulation, these methods share the common 2LS two-stage structure, selecting a cluster from a query q via scores $q^\top \mu_k$ or learned variants, followed by a conditional softmax over items within this cluster.

2LS is also closely related to *inverted file (IVF) indexing* (Douze et al., 2025; Guo et al., 2020; Jegou et al., 2010; Johnson et al., 2019), an efficient approximate nearest neighbor (ANN) search technique. IVF partitions the item space into clusters, typically via k-means, and restricts search to a subset of clusters to retrieve similar neighbors to a query. This strategy mirrors the two-stage procedure of 2LS, where cluster selection guides subsequent item-level processing. IVF-based methods are implemented in widely used ANN libraries such as Faiss (Douze et al., 2025). 2LS also shares similarities with *codebook-based methods* (Jegou et al., 2010; Shao et al., 2025), which partition embeddings and assign items to discrete codes, while 2LS defines a probabilistic sampling procedure.

Moreover, 2LS is related to *hierarchical softmax* (Goodman, 2001; Mnih & Hinton, 2008; Mohammed & Umaashankar, 2018; Morin & Bengio, 2005), which factorizes the softmax along a tree, scoring items using products of conditional probabilities along root-to-leaf paths, yielding $\mathcal{O}(d \log N)$ complexity for balanced trees. In this sense, 2LS can be viewed as a two-level instance of this idea. However, while hierarchical softmax relies on predefined trees and provides an exact factorization, 2LS uses a cluster-based approximation and a query-dependent sampling procedure.

Other Softmax Approximations For completeness, we note that a broad body of work, less directly related to 2LS, has also explored alternative methods to accelerate, approximate, or replace softmax computation for various applications (Chen et al., 2016). In particular, during training, approaches such as *sampled softmax* (Jean et al., 2015) approximate the partition function using a subset of classes, while *adaptive softmax* (Joulin et al., 2017) partitions classes by frequency to reduce computation in skewed distributions.

In large-scale inference settings, such as large language models or recommender systems with large item sets, simple *truncation-based methods* remain among the most widely used in practice. They restrict the softmax computation to the top-k items in terms of similarity (Chen et al., 2021), with $k \ll N$, or to the smallest set of items whose cumulative probability exceeds a threshold $p < 1$, yielding nucleus (top-p) sampling (Brown et al., 2020; Holtzman et al., 2020). These methods can be combined with ANN search to reduce computational cost (Chen et al., 2019a). However, they restrict sampling to a truncated subset of items. This limitation is detrimental in applications such as recommendation, where constraining the candidate item set based on computational considerations may exclude relevant items beyond the selected subset (Bendada et al., 2025).

Finally, one may also replace softmax with methods such as von Mises–Fisher sampling (which still samples proportionally to $q^\top x_j$, in sublinear time, but requires hyperspherical assumptions (Bendada et al., 2025)) and ε-greedy sampling (which runs in constant time but, in contrast, ignores item-query similarity (Sutton & Barto, 2018)). In this context, 2LS appears as an alternative to simultaneously (1) account for $q^\top x_j$ similarities as in softmax, (2) avoid restricting sampling to a subset of items, and (3) enable sublinear-time complexity.

3. Bias Characterization of Two-Level Softmax (2LS) Sampling

In this section, we show that, despite its advantages, 2LS exhibits systematic sampling biases, and we analyze their underlying causes. We report all mathematical proofs in Appendix A of this paper.

3.1. Sampling Ratio

To analyze the discrepancy between exact softmax and 2LS in this paper, we introduce the ratio

$$R_i^{(N)}(q) = \frac{p_{\text{2LS}}(i \mid q)}{p(i \mid q)}. \tag{4}$$

For any item $i \in \mathcal{I}$, query $q \in \mathbb{R}^d$, and number of items N, this ratio captures the relative sampling distortion induced by the 2LS procedure: values greater than 1 indicate that i is oversampled by 2LS compared to the exact softmax, while values smaller than 1 indicate undersampling.

A notable property of this ratio is that it depends only on the 2LS cluster containing i, and not directly on the similarity $q^\top x_i$, as formalized in Proposition 1. Consequently, although items within the same cluster may have different sampling probabilities, they are over- or undersampled by the same factor.

Proposition 1. *Let $q \in \mathbb{R}^d$, and let $i, j \in \mathcal{I}$ with $i \neq j$. If $i, j \in \mathcal{C}_k$ for any $\mathcal{C}_k \in \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$, i.e., if the items i and j belong to the same 2LS cluster, then $R_i^{(N)}(q) = R_j^{(N)}(q)$.*

3.2. Systematic Bias in 2LS Sampling

Setting In the following, we study $R_i^{(N)}(q)$ under a general probabilistic setting where, for each N, item embeddings $x_1, \dots, x_N$ are i.i.d. samples from a mixture distribution $\sum_{k=1}^K \pi_k \mathcal{D}_k$, where π_k denotes the probability of sampling from the component distribution $\mathcal{D}_k$, and such that, for each $k \in \{1, \dots, K\}$, the item embeddings $\{x_i : i \in \mathcal{C}_k\}$ are samples from $\mathcal{D}_k$.

Asymptotic Behavior $R_i^{(N)}(q)$ is a random variable. We study its behavior as N increases. Next, expectations are taken with respect to the mixture $x \sim \sum_{k=1}^K \pi_k \mathcal{D}_k$, unless specified otherwise.

Proposition 2. *Let $i \in \mathcal{I}$, $q \in \mathbb{R}^d$, and k be the cluster index such that $x_i \in \mathcal{C}_k$. Assume $\pi_k > 0$. Then,*

$$R_i^{(N)}(q) \xrightarrow[N\to\infty]{a.s.} R_k^{(\infty)}(q) = c(q) \frac{\exp(q^\top \mu_k/\tau)}{\pi_k \, \mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}, \tag{5}$$

with

$$c(q) = \frac{\mathbb{E}[\exp(q^\top x/\tau)]}{\sum_{k'=1}^{K} \exp(q^\top \mu_{k'}/\tau)}. \tag{6}$$

Proposition 3 (Corollary with Gaussian approximation)**.** *In addition, assume that $q^\top x_i/\tau$ follows a Gaussian distribution with mean $q^\top \mu_k/\tau$ and variance $\sigma_k^2(q) = \mathrm{Var}_{x\sim\mathcal{D}_k}(q^\top x/\tau)$. Then,*

$$R_k^{(\infty)}(q) \propto \frac{1}{\pi_k \, \exp\left(\frac{1}{2}\sigma_k^2(q)\right)}. \tag{7}$$

Discussion Our results show that $R_k^{(\infty)}(q) \neq 1$ in general, i.e., that 2LS systematically oversamples or undersamples items compared to exact softmax sampling. In particular, two items $i, j \in \mathcal{I}$ with equal similarity to the query q, i.e., $q^\top x_i = q^\top x_j$, would receive identical sampling probabilities under softmax, but this invariance can be broken under 2LS if i and j belong to different clusters, as they may be oversampled or undersampled by different relative factors.

Importantly, our results provide a theoretical understanding of this phenomenon, which is particularly clear in the simplified ratio form given by our corollary under a Gaussian approximation. They show that 2LS misweights clusters along two complementary axes. First, it ignores *cluster size imbalance*, thereby oversampling smaller clusters and undersampling larger ones. Second, it ignores the *intra-cluster item-query similarity dispersion*, undersampling clusters whose items exhibit higher variability in their similarity to the query. Our experiments in Section 5 will empirically confirm and illustrate the practical significance of these biases on various large-scale datasets.

To our knowledge, these biases have never been formally characterized in previous work. The closest connection is a recent observation by Tranheden et al. (2026), who empirically noted a loss of efficiency under cluster size imbalance, consistent with our findings. Our results suggest corrections accounting for cluster size imbalance and intra-cluster dispersion, which we develop in the next section.

4. 2LS Sampling Done Right: Correcting Bias from Size Imbalance and Dispersion

We now propose two corrected sampling methods designed to address the sampling biases of 2LS. We report all mathematical proofs from this section in Appendix B.

4.1. Size-Corrected 2LS Sampling (S-2LS)

S-2LS Sampling We first propose S-2LS, a corrected 2LS procedure that incorporates cluster size into the cluster-level score, while the within-cluster distribution remains unchanged:

$$p_{\text{S-2LS}}(k \mid q) = \frac{|\mathcal{C}_k| \exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^{K} |\mathcal{C}_{k'}| \exp(q^\top \mu_{k'}/\tau)}, \quad \text{for } k \in \{1, \dots, K\}, \tag{8}$$

$$p_{\text{S-2LS}}(i \mid k, q) = \frac{\exp(q^\top x_i/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}, \text{ for } i \in \mathcal{C}_k. \tag{9}$$

Equation (8) requires computing a softmax over K d-dimensional dot products, with complexity $\mathcal{O}(dK)$. The within-cluster step is unchanged, so the overall sampling complexity of S-2LS is $\mathcal{O}(dK + d|\mathcal{C}_k|)$, i.e., identical to that of 2LS. In particular, following the reasoning of Section 2.2, choosing $K \approx \sqrt{N}$ and balancing clusters yields an $\mathcal{O}(d\sqrt{N})$ complexity.

Asymptotic Behavior Analogous to the ratio of Equation (4), we introduce $R_{i,\text{S-2LS}}^{(N)}(q) = \frac{p_{\text{S-2LS}}(i|q)}{p(i|q)}$, a variant of the same ratio for S-2LS, and study its asymptotic behavior in the setting of Section 3.

Proposition 4. *Let $i \in \mathcal{I}$, $q \in \mathbb{R}^d$, and k be such that $x_i \in \mathcal{C}_k$. Assume $\pi_k > 0$. Then,*

$$R_{i,\text{S-2LS}}^{(N)}(q) \xrightarrow[N\to\infty]{a.s.} c_{\text{S-2LS}}(q) \frac{\exp(q^\top \mu_k/\tau)}{\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}, \tag{10}$$

where $c_{S\text{-}2LS}(q)$ does not depend on i or k.

In summary, the proposed S-2LS correction removes the dependence on cluster size. Our experiments will confirm this result empirically across several datasets, showing that S-2LS does not undersample (respectively, oversample) items from larger (resp., smaller) clusters.

4.2. Size- and Dispersion-Corrected 2LS Sampling (SD-2LS)

SD-2LS Sampling S-2LS alone does not address the second source of bias identified in Section 3, namely the variability of intra-cluster item-query similarities. We therefore propose a second method, SD-2LS, which, in addition to accounting for cluster size imbalance as in S-2LS, also accounts for dispersion. Denoting $\sigma_k^2(q) = \mathrm{Var}_{x\sim\mathcal{D}_k}(q^\top x/\tau)$, we propose to sample items as follows:

$$p_{\text{SD-2LS}}(k \mid q) = \frac{|\mathcal{C}_k| \exp\big(q^\top \mu_k/\tau + \frac{1}{2}\sigma_k^2(q)\big)}{\sum_{k'=1}^{K} |\mathcal{C}_{k'}| \exp\big(q^\top \mu_{k'}/\tau + \frac{1}{2}\sigma_{k'}^2(q)\big)} \quad \text{for } k \in \{1,\dots,K\}, \tag{11}$$

$$p_{\text{SD-2LS}}(i \mid k, q) = \frac{\exp(q^\top x_i/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}, \text{ for } i \in \mathcal{C}_k. \tag{12}$$

In addition to dot products $q^\top \mu_k$, this variant requires evaluating the quadratic forms $q^\top \Sigma_k q$ for all K clusters, where the $d \times d$ matrices Σ_k denote the intra-cluster covariance matrices, yielding an $\mathcal{O}(d^2K)$ cost. The within-cluster step is unchanged, so the overall sampling cost is $\mathcal{O}(d^2K + d|\mathcal{C}_k|)$. In practice, since d is typically much smaller than N, which can reach millions, the additional cost compared to 2LS remains negligible. In particular, following the reasoning of Section 2.2, choosing $K \approx \sqrt{N}$ and balancing clusters yields an $\mathcal{O}(d^2\sqrt{N})$ complexity, preserving sublinear scaling in N.

Asymptotic Behavior Analogous to Equation (4), we introduce $R^{(N)}_{i,\text{SD-2LS}}(q) = \frac{p_{\text{SD-2LS}}(i|q)}{p(i|q)}$, a variant of the same ratio for SD-2LS, and study its asymptotic behavior in the setting of Section 3.

Proposition 5. *Let $i \in \mathcal{I}$, $q \in \mathbb{R}^d$, and k be such that $x_i \in \mathcal{C}_k$. Assume $\pi_k > 0$. Then,*

$$R^{(N)}_{i,SD\text{-}2LS}(q) \xrightarrow[N\to\infty]{a.s.} c_{SD\text{-}2LS}(q)\, \frac{\exp\big(q^\top \mu_k/\tau + \frac{1}{2}\sigma_k^2(q)\big)}{\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}, \tag{13}$$

where $c_{SD\text{-}2LS}(q)$ does not depend on i or k.

Proposition 5 shows that, regardless of the underlying distribution, the denominator involves the *moment generating function* (MGF) of the scalar projection $q^\top x/\tau$. This perspective provides a natural interpretation of the remaining distortion. In general, the MGF can be expanded in terms of the moments of the distribution, and is well approximated by its leading terms when higher-order moments (e.g., skewness and kurtosis) are small (Casella & Berger, 2024).

This observation suggests a natural correction strategy: ideally, one would reweight each cluster by $\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]$. However, computing this quantity exactly would require evaluating all similarities within each cluster, leading to a cost of $\mathcal{O}(N)$, which defeats the purpose of sublinear sampling. Instead, in this paper, our strategy is to approximate the MGF using a *truncated expansion*. In particular, retaining terms up to second order as done in SD-2LS yields a correction that depends on the variance $\sigma_k^2(q)$, which can be computed efficiently.

Notably, as formalized in Proposition 6, this approximation is exact under Gaussian assumptions, as all higher-order moments beyond the variance vanish. Therefore, SD-2LS asymptotically recovers exact softmax sampling in this case. Empirically, our results on real-world datasets in the next section consistently show that this second-order correction remains effective even when the Gaussian assumption does not strictly hold, confirming the practical utility of SD-2LS on real-world data.

Proposition 6 (Corollary with Gaussian approximation)**.** *In addition, assume that $q^\top x_i/\tau$ follows a Gaussian distribution with mean $q^\top \mu_k/\tau$ and variance $\sigma_k^2(q) = \mathrm{Var}_{x\sim\mathcal{D}_k}(q^\top x/\tau)$. Then,*

$$\forall i \in \mathcal{I}, R^{(N)}_{i,SD\text{-}2LS}(q) \xrightarrow[N\to\infty]{a.s.} 1. \tag{14}$$

4.3. Discussion

Usage of S-2LS vs SD-2LS SD-2LS provides stronger guarantees than S-2LS; it is therefore natural to ask why S-2LS was introduced. The two variants are in fact suited to different regimes. S-2LS compensates for cluster size imbalance at the same sampling complexity as 2LS. It is the preferred choice when clusters are highly imbalanced and low-latency sampling is critical. SD-2LS additionally accounts for intra-cluster dispersion. This yields a more faithful softmax approximation, at the cost of slightly higher computation due to covariance evaluations, introducing a d^2 term in the complexity instead of d. While negligible in many cases, this overhead may become more significant in high-dimensional settings with large d, making SD-2LS preferable only when intra-cluster dispersion is substantial and sufficient resources are available. In summary, S-2LS provides a lightweight correction, while SD-2LS trades increased computation for improved sampling accuracy.

Limitations and Future Work Our analysis of SD-2LS relies on truncated expansions of the MGF of the similarity. While this paper shows this approximation is exact

under Gaussian assumptions and empirically effective more broadly, it may be less accurate for embedding distributions exhibiting significant higher-order moments (e.g., high skewness or kurtosis). In such cases, higher-order corrections to 2LS could further improve sampling accuracy. Extending our analysis to account for these effects, while maintaining computational efficiency, is an interesting direction for future work. Moreover, while we focus on 2LS, the bias study extends to hierarchical softmax constructions based on trees (Goodman, 2001; Morin & Bengio, 2005). In such models, the mismatch identified in Section 3 recurs at each level of the root-to-leaf hierarchy, accumulating multiplicatively along the path. This suggests that our approach could be extended to hierarchical softmax by incorporating corrections at each node, which we leave for future work.

5. Experimental Analysis

In this section, we empirically validate our theoretical results and the practical effectiveness of the proposed methods. Our experiments are designed to address the following questions:

- **Q1**: Are the sampling biases of standard 2LS, theoretically characterized in Section 3, observable and practically significant?
- **Q2**: Do S-2LS and SD-2LS effectively correct these biases, yielding samples that more closely match the exact softmax distribution than 2LS?
- **Q3**: How do S-2LS and SD-2LS compare in terms of the fidelity-latency trade-off relative to other approximation methods, including 2LS, hierarchical softmax, and top-k softmax?

5.1. Experimental Setup

Datasets We consider five large-scale datasets, including three real-world embedding corpora and two controlled synthetic corpora. We use two item embedding datasets, *VK-LSVD* (Poslavsky et al., 2026) ($N = 19.6$M videos, $d = 64$) and *YAMBDA* (Ploshkin et al., 2025) ($N = 7.7$M music tracks, $d = 128$), extracted from industrial recommender systems. We additionally use *GloVe-100* (Pennington et al., 2014) ($N = 1.2$M word embeddings, $d = 100$) to extend the evaluation to NLP data. Finally, we construct two synthetic Gaussian mixture corpora, *Synth-balanced* and *Synth-unbalanced* (both $N = 10^6$, $d = 100$), to isolate the role of cluster-size imbalance. They are obtained by generating 1024 mean vectors uniformly on the d-dimensional unit hypersphere, and forming embedding clusters by adding isotropic Gaussian noise ($\sigma = 0.1$) around these means. Synth-balanced has an equal number of points per component. In contrast, Synth-unbalanced follows a heavy-tailed distribution resembling real-world corpora, with component sizes drawn from a multinomial distribution with log-normal weights $w_c \propto \exp(0.8\, Z_c)$, with $Z_c \sim \mathcal{N}(0, 1)$. GloVe-100 and YAMBDA embeddings are ℓ_2-normalized, while VK-LSVD and the synthetic corpora are left unnormalized, ensuring coverage of a broad range of regimes.

Methods We compare six sampling methods. *Exact softmax* performs the full $\mathcal{O}(N)$ pass and serves as the reference. We compare it against *2LS*, as well as our proposed *S-2LS* and *SD-2LS*. For all three methods, we run k-means with $K = 1024$ clusters on each dataset, which approximates $\sqrt{N}$ for the natural corpora. The same partition and centroids are shared across the three variants, so the only varying factor is the cluster sampling rule (Appendix C reports histograms of cluster sizes). SD-2LS uses full-rank covariance matrices $\Sigma_k \in \mathbb{R}^{d\times d}$, so that latency reflects a faithful, non-approximated implementation. We further consider *top-k softmax* (Bendada et al., 2024), which restricts the softmax to the $k = 1000$ most similar items, implemented via Faiss IVFFlat (Douze et al., 2025), and *hierarchical softmax* (Morin & Bengio, 2005) as a representative tree-based method. We adopt a recursive binary k-means tree to control the memory footprint; details are provided in Appendix C. Across all methods, we vary the temperature parameter $\tau \in \{0.05, 0.1, 0.2\}$ to consider different sharpness levels of the target softmax distribution.

Evaluation Protocol For each dataset, we sample $n_q = 1000$ embeddings and use them as queries. For each query, we compare the item sampling probabilities of exact softmax to those of the five approximate methods for each temperature. We report three evaluation metrics. First, the *sampling ratio* from Section 3 captures biases compared to softmax. Second, the *sampling fidelity* measures how closely each approximate method matches softmax, and is quantified by the Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) $\mathrm{KL}(p_{\text{approx}} \,\|\, p_{\text{exact}})$, averaged over queries. KL is computed in the approximate-to-exact direction, which remains finite when the approximate distribution assigns zero probability to some items, as in top-k softmax. Finally, we compute the *per-query latency*, measured in milliseconds and averaged over queries and the three temperatures, as latency is effectively τ-invariant for all methods. All experiments are run single-threaded on CPU with sequential queries, reflecting the inference regime in which 2LS-style samplers are typically deployed. Our Python code and data are released on GitHub to reproduce all results: `https://github.com/twolevelsoftmaxanon-creator/2ls`.

5.2. Results and Discussion

Q1: 2LS biases are substantial in practice Our theoretical analysis in Section 3 demonstrates that 2LS mis-

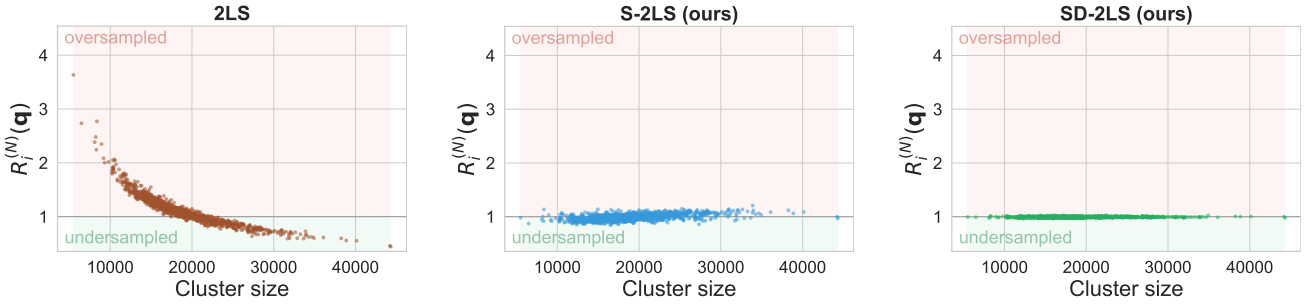

Figure 1. Sampling ratio on VK-LSVD ($\tau = 0.1$) as a function of cluster size, averaged over queries. Standard 2LS undersamples large clusters and oversamples small ones relative to softmax ($R_i^{(N)}(q) \neq 1$, p-value $p < 0.01$), while S-2LS and SD-2LS correct these biases.

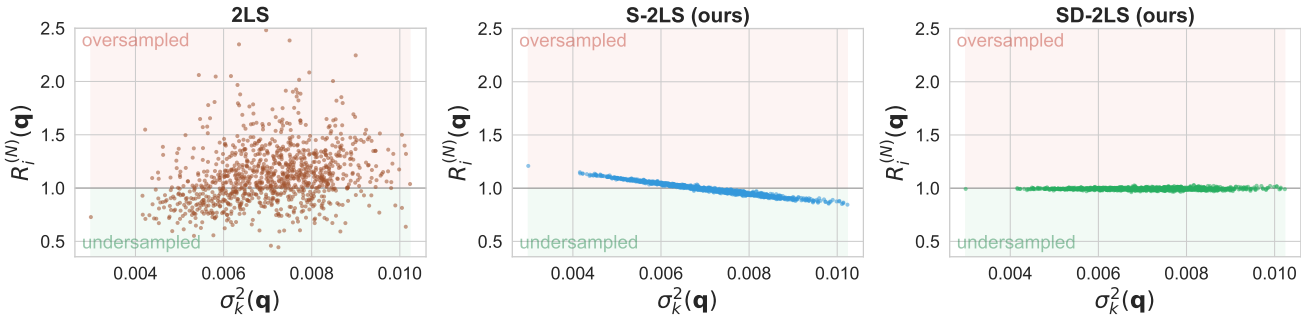

Figure 2. Sampling ratio on VK-LSVD ($\tau = 0.1$) as a function of intra-cluster similarity variance, averaged over queries. The effect is entangled with cluster-size bias for 2LS; S-2LS corrects the cluster-size bias but not the variance bias, isolating the latter and revealing that high- (resp., low-) variance clusters are undersampled (resp., oversampled) relative to softmax ($R_i^{(N)}(q) \neq 1$, $p < 0.01$).

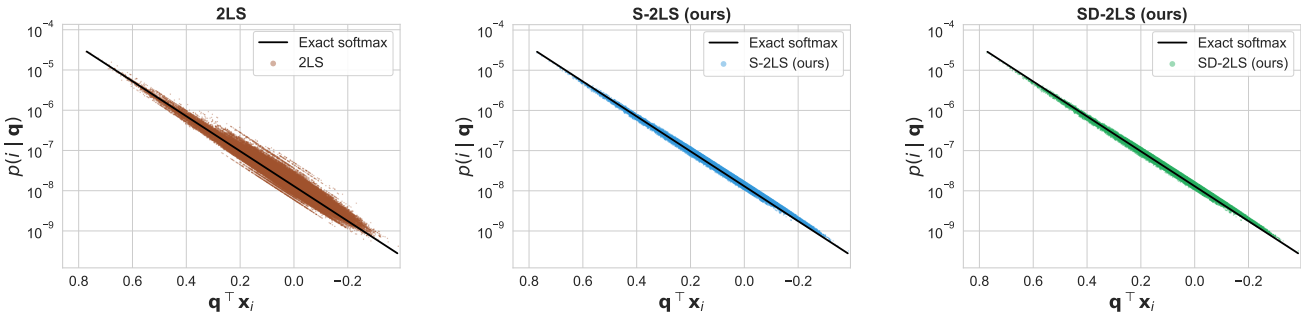

Figure 3. Sampling probability on VK-LSVD ($\tau = 0.1$) per item as a function of query-item similarity for a fixed query. 2LS assigns widely different probabilities to items with identical similarity to the query, while S-2LS and SD-2LS markedly reduce this dispersion.

weights clusters by ignoring both cluster-size imbalance and intra-cluster dispersion. Figure 1 confirms empirically on VK-LSVD that 2LS significantly undersamples large clusters and oversamples small ones relative to exact softmax. In addition, Figure 2 illustrates the impact of intra-cluster similarity variance. While this effect is entangled with the (stronger) cluster-size bias for 2LS, it becomes clearer in the S-2LS figure, which corrects the cluster-size bias but not the intra-cluster variance bias, thereby isolating the latter. Results confirms that 2LS (and S-2LS) undersample clusters with high variability in item-query similarity and oversample those with low variability. Similar results are consistently observed on the four other datasets and reported in Appendix C. These observations empirically validate the theoretical predictions of Section 3 and show that the resulting biases are clearly observable and practically significant across all datasets.

Q2: S-2LS and SD-2LS effectively correct 2LS biases Figure 1 shows on VK-LSVD that S-2LS and SD-2LS effectively correct the cluster-size bias, with sampling ratios tightly concentrated around $R_i^{(N)}(q) = 1$. Figure 2 further shows that, while S-2LS still exhibits a residual bias with respect to within-cluster dispersion, SD-2LS additionally corrects this second source of distortion. Similar patterns are observed across the four remaining datasets and reported in Appendix C. These results confirm that the proposed corrections substantially reduce the biases of 2LS and yield

Table 1. Sampling fidelity $\mathrm{KL}(p_{\text{approx}} \,\|\, p_{\text{exact}})$ relative to exact softmax, averaged over $n_q = 1000$ queries with 95% confidence intervals. Best results are in bold, and second-best are underlined.

τ	Dataset	Top-k	2LS	Hierarchical	S-2LS (ours)	SD-2LS (ours)
0.05	GloVe-100	1.9137 ± 0.0516	0.0961 ± 0.0110	0.5231 ± 0.0272	$\underline{0.0711 \pm 0.0115}$	$\mathbf{0.0288 \pm 0.0076}$
	VK-LSVD	2.4902 ± 0.0442	$\underline{0.0541 \pm 0.0023}$	0.7610 ± 0.0253	0.0551 ± 0.0023	$\mathbf{0.0129 \pm 0.0008}$
	YAMBDA	1.7466 ± 0.0285	$\underline{0.0485 \pm 0.0018}$	1.4484 ± 0.0563	0.0551 ± 0.0016	$\mathbf{0.0177 \pm 0.0056}$
	Synth-balanced	$\mathbf{0.0003 \pm 0.0001}$	0.5968 ± 0.1051	9.4151 ± 0.0980	$\underline{0.3929 \pm 0.0801}$	0.6586 ± 0.1403
	Synth-unbalanced	$\mathbf{0.0090 \pm 0.0020}$	0.8959 ± 0.1504	10.3033 ± 0.1216	0.7096 ± 0.1279	$\underline{0.1740 \pm 0.0584}$
0.1	GloVe-100	3.9907 ± 0.0395	0.0615 ± 0.0003	0.2897 ± 0.0018	$\underline{0.0022 \pm 0.0002}$	$\mathbf{0.0001 \pm 0.0000}$
	VK-LSVD	5.2351 ± 0.0334	0.0297 ± 0.0003	0.3425 ± 0.0027	$\underline{0.0051 \pm 0.0001}$	$\mathbf{0.0002 \pm 0.0000}$
	YAMBDA	3.5266 ± 0.0292	0.0300 ± 0.0007	0.6049 ± 0.0154	$\underline{0.0098 \pm 0.0003}$	$\mathbf{0.0005 \pm 0.0000}$
	Synth-balanced	$\underline{0.1580 \pm 0.0087}$	0.3518 ± 0.0436	2.6158 ± 0.0438	0.2334 ± 0.0327	$\mathbf{0.0293 \pm 0.0042}$
	Synth-unbalanced	0.2318 ± 0.0116	0.2858 ± 0.0391	3.1780 ± 0.0546	$\underline{0.1728 \pm 0.0295}$	$\mathbf{0.0314 \pm 0.0070}$
0.2	GloVe-100	5.4103 ± 0.0217	0.0677 ± 0.0001	0.3007 ± 0.0005	$\underline{0.0001 \pm 0.0000}$	$\mathbf{1.2e\text{-}06 \pm 9e\text{-}08}$
	VK-LSVD	7.3333 ± 0.0174	0.0337 ± 0.0001	0.3153 ± 0.0007	$\underline{0.0003 \pm 0.0000}$	$\mathbf{2.7e\text{-}06 \pm 1e\text{-}07}$
	YAMBDA	5.6988 ± 0.0202	0.0388 ± 0.0003	0.3263 ± 0.0020	$\underline{0.0008 \pm 0.0000}$	$\mathbf{5.3e\text{-}06 \pm 2e\text{-}07}$
	Synth-balanced	2.2841 ± 0.0261	0.1296 ± 0.0026	0.6342 ± 0.0029	$\underline{0.0117 \pm 0.0015}$	$\mathbf{0.0083 \pm 0.0021}$
	Synth-unbalanced	2.2393 ± 0.0345	0.1935 ± 0.0057	0.9311 ± 0.0070	$\underline{0.0059 \pm 0.0009}$	$\mathbf{0.0023 \pm 0.0011}$

distributions that more closely match exact softmax. This is further supported by Table 1, where S-2LS and SD-2LS consistently achieve higher sampling fidelity than 2LS, as discussed in the next paragraph.

Q3: S-2LS and SD-2LS achieve a superior fidelity–latency trade-off Table 1 reports sampling fidelity across all datasets and temperatures. SD-2LS consistently achieves the lowest KL on all natural corpora (VK-LSVD, YAMBDA, GloVe-100) across all temperatures. The gains over 2LS are substantial, ranging from 3–5$\times$ at $\tau = 0.05$ to one to four orders of magnitude at $\tau \in \{0.1, 0.2\}$. S-2LS is a consistent second best, already reducing KL by one to two orders of magnitude in the low-temperature regime most relevant for recommendation. In contrast, top-k truncation exhibits large errors even at $k = 1000$, and hierarchical softmax remains significantly less accurate, particularly on YAMBDA, likely due to accumulated routing errors along the tree.

The gap between methods widens with temperature. As τ increases, the softmax distribution flattens, and accurate estimation of cluster mass becomes the dominant challenge. In this regime, the dispersion correction in SD-2LS becomes critical, explaining the large performance gains observed at $\tau \in \{0.1, 0.2\}$. Synthetic datasets further confirm this mechanism: on Synth-balanced, where cluster sizes are nearly uniform, the advantage of the corrections is limited and top-k can perform well at low temperature; on Synth-unbalanced, which mirrors the heavy-tailed structure of natural corpora, SD-2LS consistently dominates and the gap to 2LS reaches an order of magnitude.

These improvements come at minimal additional cost, as shown in Table 2 from Appendix C. S-2LS matches the latency of 2LS within measurement noise, and therefore Pareto-dominates 2LS (same cost, lower KL). SD-2LS incurs a moderate overhead due to the computation of per-cluster quadratic forms, but this cost is efficiently amortized in practice, resulting in only a 1.2–1.8$\times$ slowdown on higher-dimensional datasets and up to $\sim 3\times$ on lower-dimensional ones. Even in this regime, SD-2LS remains within a small constant factor of the fastest approximate methods while delivering KL reductions of several orders of magnitude. Overall, S-2LS and SD-2LS provide a strictly better fidelity-latency trade-off, making them practical drop-in replacements for 2LS in large-scale settings.

6. Conclusion

In this paper, we showed that 2LS sampling introduces systematic biases due to misweighting clusters. We characterized these biases theoretically and demonstrated that they are substantial in practice across multiple large-scale datasets. We proposed two corrections, S-2LS and SD-2LS, which preserve the computational efficiency of 2LS while correcting its biases and improving softmax sampling fidelity. Extensive theoretical and experimental results show that S-2LS consistently improves upon 2LS at no cost, while SD-2LS further reduces approximation error with only moderate overhead. These results support clear practical recommendations: S-2LS should replace standard 2LS unconditionally, as it incurs no additional cost and consistently improves sampling fidelity. SD-2LS should be preferred when intra-cluster variance is non-negligible and the additional computational cost is acceptable, as in many large-scale recommendation settings where $d \ll N$. Finally, our analysis suggests directions for future research. In particular, as SD-2LS relies on a second-order approximation of the similarity distribution, extending our method to account for higher-order moments could further improve sampling accuracy for embedding distributions with high skewness or kurtosis.

References

Bendada, W., Salha-Galvan, G., Hennequin, R., Bontempelli, T., Bouabça, T., and Cazenave, T. vmf-exp: von mises-fisher exploration of large action sets with hyperspherical embeddings. In *ICML 2024 Workshop on Aligning Reinforcement Learning Experimentalists and Theorists*, 2024.

Bendada, W., Salha-Galvan, G., Hennequin, R., Bontempelli, T., Bouabça, T., and Cazenave, T. Exploring Large Action Sets with Hyperspherical Embeddings using von Mises-Fisher Sampling. *Proceedings of the 42nd International Conference on Machine Learning*, pp. 3677–3711, 2025.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020.

Casella, G. and Berger, R. *Statistical Inference*. Chapman and Hall/CRC, 2024.

Chen, J., Lian, D., Jin, B., Huang, X., Zheng, K., and Chen, E. Fast Variational AutoEncoder with Inverted Multi-Index for Collaborative Filtering. In *Proceedings of the ACM Web Conference*, pp. 1944–1954, 2022.

Chen, J., Zhang, J., Yang, Y., Lian, D., and Chen, E. Adaptive Sampled Softmax with Inverted Multi-Index: Methods, Theory and Applications. *arXiv preprint arXiv:2501.08563*, 2025.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. Top-k Off-Policy Correction for a REINFORCE Recommender System. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pp. 456–464, 2019a.

Chen, M., Chang, B., Xu, C., and Chi, E. H. User Response Models to Improve a REINFORCE Recommender System. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 121–129, 2021.

Chen, P. H., Si, S., Kumar, S., Li, Y., and Hsieh, C.-J. Learning to Screen for Fast Softmax Inference on Large Vocabulary Neural Networks. In *Proceedings of the 7th International Conference on Learning Representations*, 2019b.

Chen, W., Grangier, D., and Auli, M. Strategies for Training Large Vocabulary Neural Language Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1975–1985, 2016.

Covington, P., Adams, J., and Sargin, E. Deep Neural Networks for Youtube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191–198, 2016.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. The Faiss Library. *IEEE Transactions on Big Data*, 2025.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Goodman, J. Classes for Fast Maximum Entropy Training. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 561–564, 2001.

Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., and Kumar, S. Accelerating Large-Scale Inference with Anisotropic Vector Auantization. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3887–3896, 2020.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The Curious Case of Neural Text Degeneration. *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. 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 (Volume 1: Long Papers)*, pp. 1–10, 2015.

Jegou, H., Douze, M., and Schmid, C. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

Johnson, J., Douze, M., and Jégou, H. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7 (3):535–547, 2019.

Joulin, A., Cissé, M., Grangier, D., Jégou, H., et al. Efficient Softmax Approximation for GPUs. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1302–1310, 2017.

Kullback, S. and Leibler, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1): 79–86, 1951.

Liao, S., Chen, T., Lin, T., Zhou, D., and Wang, C. Doubly Sparse: Sparse Mixture of Sparse Experts for Efficient Softmax Inference. *arXiv preprint arXiv:1901.10668*, 2019.

Liu, D., Yu, Y., and Lengerich, B. CSV-Decode: Certifiable Sub-Vocabulary Decoding for Efficient Large Language Model Inference. *arXiv preprint arXiv:2511.21702*, 2025.

Mnih, A. and Hinton, G. E. A Scalable Hierarchical Distributed Language Model. *Advances in Neural Information Processing Systems*, 21, 2008.

Mohammed, A. A. and Umaashankar, V. Effectiveness of Hierarchical Softmax in Large Scale Classification Tasks. In *Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics*, pp. 1090–1094, 2018.

Morin, F. and Bengio, Y. Hierarchical Probabilistic Neural Network Language Model. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pp. 246–252. PMLR, 2005.

Pennington, J., Socher, R., and Manning, C. D. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

Ploshkin, A., Tytskiy, V., Pismenny, A., Baikalov, V., Taychinov, E., Permiakov, A., Burlakov, D., and Krofto, E. Yambda-5B—A Large-Scale Multi-Modal Dataset for Ranking and Retrieval. In *Proceedings of the 19th ACM Conference on Recommender Systems*, pp. 894–901, 2025.

Poslavsky, A., D'yakonov, A., Dorn, Y., and Zimovnov, A. VK-LSVD: A Large-Scale Industrial Dataset for Short-Video Recommendation. In *Proceedings of the ACM Web Conference 2026*, pp. 8657–8660, 2026.

Shao, J., Huang, H., Wu, J., Cheng, Y., Wu, Z., Shan, Y., and Zheng, M. VQ-Logits: Compressing the Output Bottleneck of Large Language Models via Vector Quantized Logits. *arXiv preprint arXiv:2505.10202*, 2025.

Subbiah, A., Aggarwal, V., Pine, J., Rendle, S., Sayana, K., and Su, K. Efficient Item ID Generation for Large-Scale LLM-Based Recommendation. *arXiv preprint arXiv:2509.03746*, 2025.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Tranheden, W., Ahmed, S., Dubhashi, D., Matthiesen, J., and von Essen, H. FlashHead: Efficient Drop-In Replacement for the Classification Head in Language Model Inference. *arXiv preprint arXiv:2603.14591*, 2026.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.

Zhang, J., Ullah, N., Schultheis, E., and Babbar, R. DynaSpec: Context-aware Dynamic Speculative Sampling for Large-Vocabulary Language Models. *arXiv preprint arXiv:2510.13847*, 2025.

Appendix

This appendix supplements the article "*Size- and Dispersion-Corrected Two-Level Softmax Sampling*" and is organized as follows:

- Appendix A presents detailed proofs for all theoretical results presented in Section 3.
- Appendix B reports detailed proofs for all theoretical results presented in Section 4.
- Appendix C provides additional results and technical details complementing the experimental analysis in Section 5.

A. Proofs for Section 3: Bias Characterization of 2LS Sampling

A.1. Proof of Proposition 1

Equation (4) in Section 3 introduced the sampling ratio $R_i^{(N)}(q) = p_{\text{2LS}}(i \mid q)/p(i \mid q)$. Substituting the expressions of the sampling probabilities $p_{\text{2LS}}(i \mid q)$ and $p(i \mid q)$ from Section 2, we obtain

$$R_i^{(N)}(q) = \frac{\frac{\exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\exp(q^\top x_i/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}}{\frac{\exp(q^\top x_i/\tau)}{\sum_{j=1}^N \exp(q^\top x_j/\tau)}} \tag{15}$$

$$= \frac{\exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\sum_{j=1}^N \exp(q^\top x_j/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}. \tag{16}$$

As the $\exp(q^\top x_k/\tau)$ terms cancel out, the resulting expression depends only on the cluster $\mathcal{C}_k$ containing i, and not on the dot product similarity $q^\top x_i$ directly. Consequently, $R_i^{(N)}(q)$ is constant across all items within the same cluster $\mathcal{C}_k$, which proves the proposition. □

A.2. Proof of Proposition 2

From the previous proof, we have:

$$R_i^{(N)}(q) = \frac{\exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\sum_{j=1}^N \exp(q^\top x_j/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}. \tag{17}$$

We rewrite the second factor in a form amenable to asymptotic analysis. To begin with,

$$\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau) = |\mathcal{C}_k| \cdot \frac{1}{|\mathcal{C}_k|} \sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau). \tag{18}$$

Also,

$$\sum_{j=1}^N \exp(q^\top x_j/\tau) = N \cdot \frac{1}{N} \sum_{j=1}^N \exp(q^\top x_j/\tau). \tag{19}$$

Substituting these identities into Equation (17), we obtain

$$R_i^{(N)}(q) = \frac{\exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\frac{1}{N}\sum_{j=1}^N \exp(q^\top x_j/\tau)}{\frac{|\mathcal{C}_k|}{N} \cdot \frac{1}{|\mathcal{C}_k|}\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}. \tag{20}$$

We analyze each factor separately. First, since $x_1, \ldots, x_N$ are i.i.d. samples from the mixture distribution, the strong law of large numbers yields

$$\frac{1}{N}\sum_{j=1}^N \exp(q^\top x_j/\tau) \xrightarrow[N\to\infty]{a.s.} \mathbb{E}[\exp(q^\top x/\tau)]. \tag{21}$$

Second, for each $k \in \{1, \dots, K\}$, the empirical cluster proportion converges almost surely to the corresponding mixture weight, i.e.,

$$\frac{|\mathcal{C}_k|}{N} \xrightarrow[N\to\infty]{a.s.} \pi_k. \tag{22}$$

Third,the embeddings from component $\mathcal{C}_k$ are i.i.d. samples from $\mathcal{D}_k$. Hence, by the strong law of large numbers,

$$\frac{1}{|\mathcal{C}_k|} \sum_{j \in \mathcal{C}_k} \exp(q^\top x_j/\tau) \xrightarrow[N\to\infty]{a.s.} \mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]. \tag{23}$$

Combining Equation (20), Equation (21), Equation (22), and Equation (23), we obtain

$$R_i^{(N)}(q) \xrightarrow[N\to\infty]{a.s.} \frac{\exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\mathbb{E}[\exp(q^\top x/\tau)]}{\pi_k \, \mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}. \tag{24}$$

Defining

$$c(q) = \frac{\mathbb{E}[\exp(q^\top x/\tau)]}{\sum_{k'=1}^K \exp(q^\top \mu_{k'}/\tau)}, \tag{25}$$

this limit can be written as

$$R_i^{(N)}(q) \xrightarrow[N\to\infty]{a.s.} c(q) \, \frac{\exp(q^\top \mu_k/\tau)}{\pi_k \, \mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]} = R_k^{(\infty)}(q). \tag{26}$$

This concludes the proof. □

A.3. Proof of Proposition 3

By Proposition 2, for any $k \in \{1, \dots, K\}$,

$$R_k^{(\infty)}(q) = c(q) \, \frac{\exp(q^\top \mu_k/\tau)}{\pi_k \, \mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}. \tag{27}$$

Under a Gaussian approximation, the scalar projection $q^\top x/\tau$ with $x \sim \mathcal{D}_k$ is modeled as a Gaussian random variable with mean $q^\top \mu_k/\tau$ and variance $\sigma_k^2(q)$. Using the moment generating function of a Gaussian distribution (Casella & Berger, 2024), we have

$$\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)] = \exp\left(q^\top \mu_k/\tau + \tfrac{1}{2}\sigma_k^2(q)\right). \tag{28}$$

Substituting Equation (28) into Equation (27), we obtain

$$R_k^{(\infty)}(q) = c(q) \, \frac{\exp(q^\top \mu_k/\tau)}{\pi_k \, \exp\left(q^\top \mu_k/\tau + \frac{1}{2}\sigma_k^2(q)\right)}. \tag{29}$$

Canceling the factor $\exp(q^\top \mu_k/\tau)$ gives

$$R_k^{(\infty)}(q) = c(q) \, \frac{1}{\pi_k \, \exp\left(\frac{1}{2}\sigma_k^2(q)\right)}. \tag{30}$$

Since $c(q)$ does not depend on k, this implies

$$R_k^{(\infty)}(q) \propto \frac{1}{\pi_k \, \exp\left(\frac{1}{2}\sigma_k^2(q)\right)}. \tag{31}$$

□

B. Proofs for Section 4: Size- and Dispersion-Corrected 2LS Sampling

B.1. Proof of Proposition 4

The proof follows the steps of that of Proposition 2, with the difference that the cluster-level distribution is now weighted by $|\mathcal{C}_k|$. Accordingly, for any $i \in \mathcal{C}_k$,

$$R^{(N)}_{i,\text{S-2LS}}(q) = \frac{|\mathcal{C}_k| \exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K |\mathcal{C}_{k'}| \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\sum_{j=1}^N \exp(q^\top x_j/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}. \tag{32}$$

Dividing the numerator and denominator of the first factor by N, and rewriting the denominator of the second factor as in Proposition 2, we obtain

$$R^{(N)}_{i,\text{S-2LS}}(q) = \frac{\frac{|\mathcal{C}_k|}{N} \exp(q^\top \mu_k/\tau)}{\sum_{k'=1}^K \frac{|\mathcal{C}_{k'}|}{N} \exp(q^\top \mu_{k'}/\tau)} \cdot \frac{\frac{1}{N}\sum_{j=1}^N \exp(q^\top x_j/\tau)}{\frac{|\mathcal{C}_k|}{N} \cdot \frac{1}{|\mathcal{C}_k|}\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}. \tag{33}$$

Applying the same almost-sure limits as in Proposition 2, namely,

$$\frac{|\mathcal{C}_k|}{N} \xrightarrow[N\to\infty]{a.s.} \pi_k, \tag{34}$$

$$\frac{1}{N}\sum_{j=1}^N \exp(q^\top x_j/\tau) \xrightarrow[N\to\infty]{a.s.} \mathbb{E}_{x\sim\sum_{k'=1}^K \pi_{k'}\mathcal{D}_{k'}}[\exp(q^\top x/\tau)], \tag{35}$$

and

$$\frac{1}{|\mathcal{C}_k|}\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau) \xrightarrow[N\to\infty]{a.s.} \mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)], \tag{36}$$

yields

$$R^{(N)}_{i,\text{S-2LS}}(q) \xrightarrow[N\to\infty]{a.s.} c_{\text{S-2LS}}(q) \frac{\exp(q^\top \mu_k/\tau)}{\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x)/\tau]}, \tag{37}$$

with

$$c_{\text{S-2LS}}(q) = \frac{\mathbb{E}_{x\sim\sum_{k'=1}^K \pi_{k'}\mathcal{D}_{k'}}[\exp(q^\top x/\tau)]}{\sum_{k'=1}^K \pi_{k'} \exp(q^\top \mu_{k'}/\tau)}. \tag{38}$$

□

B.2. Proof of Proposition 5

Compared to previous proofs, the cluster-level score now includes both the cluster cardinality and the dispersion correction. For any $i \in \mathcal{C}_k$, we have

$$R^{(N)}_{i,\text{SD-2LS}}(q) = \frac{|\mathcal{C}_k| \exp\big(q^\top \mu_k/\tau + \frac{1}{2}\sigma_k^2(q)\big)}{\sum_{k'=1}^K |\mathcal{C}_{k'}| \exp\big(q^\top \mu_{k'}/\tau + \frac{1}{2}\sigma_{k'}^2(q)\big)} \cdot \frac{\sum_{j=1}^N \exp(q^\top x_j/\tau)}{\sum_{j\in\mathcal{C}_k} \exp(q^\top x_j/\tau)}. \tag{39}$$

Applying the same asymptotic argument as in Proposition 2 and 4, we obtain

$$R^{(N)}_{i,\text{SD-2LS}}(q) \xrightarrow[N\to\infty]{a.s.} c_{\text{SD-2LS}}(q) \frac{\exp\big(q^\top \mu_k/\tau + \frac{1}{2}\sigma_k^2(q)\big)}{\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}, \tag{40}$$

where

$$c_{\text{SD-2LS}}(q) = \frac{\mathbb{E}_{x\sim\sum_{k'=1}^K \pi_{k'}\mathcal{D}_{k'}}[\exp(q^\top x/\tau)]}{\sum_{k'=1}^K \pi_{k'} \exp\big(q^\top \mu_{k'}/\tau + \frac{1}{2}\sigma_{k'}^2(q)\big)}, \tag{41}$$

which is the result stated in Proposition 5. □

B.3. Proof of Proposition 6

In the proof of Proposition 5, we have obtained

$$R^{(N)}_{i,\text{SD-2LS}}(q) \xrightarrow[N\to\infty]{a.s.} c_{\text{SD-2LS}}(q)\,\frac{\exp\big(q^\top \mu_k/\tau + \frac{1}{2}\sigma_k^2(q)\big)}{\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)]}, \tag{42}$$

where

$$c_{\text{SD-2LS}}(q) = \frac{\mathbb{E}_{x\sim\sum_{k'=1}^K \pi_{k'}\mathcal{D}_{k'}}[\exp(q^\top x/\tau)]}{\sum_{k'=1}^K \pi_{k'}\exp\big(q^\top \mu_{k'}/\tau + \frac{1}{2}\sigma_{k'}^2(q)\big)}. \tag{43}$$

Under the Gaussian approximation of $q^\top x$ under each component $\mathcal{D}_k$,

$$\mathbb{E}_{x\sim\mathcal{D}_k}[\exp(q^\top x/\tau)] = \exp\big(q^\top \mu_k/\tau + \tfrac{1}{2}\sigma_k^2(q)\big)\,, \tag{44}$$

so the cluster-dependent factor cancels and

$$R^{(N)}_{i,\text{SD-2LS}}(q) \xrightarrow[N\to\infty]{a.s.} c_{\text{SD-2LS}}(q). \tag{45}$$

Applying the same approximation to the mixture expectation in the definition of $c_{\text{SD-2LS}}(q)$ gives

$$c_{\text{SD-2LS}}(q) = 1, \tag{46}$$

hence

$$R^{(N)}_{i,\text{SD-2LS}}(q) \xrightarrow[N\to\infty]{a.s.} 1. \tag{47}$$

This concludes the proof. □

C. Additional Experimental Results and Technical Details

C.1. Technical Details on Hierarchical Softmax Implementation

While the primary goal of our experiments is to compare 2LS with our proposed S-2LS and SD-2LS sampling methods, we also include *hierarchical softmax* (Morin & Bengio, 2005) in the comparisons of Section 5 as a representative tree-based factorization baseline. The textbook variant uses a fully binary tree with one item per leaf, which requires storing roughly N additional internal-node embeddings on top of the dataset itself, effectively doubling the index memory footprint. At the scale of the recommender corpora considered in experiments (N in the tens of millions), this is impractical.

We therefore adopt an intermediate variant: a recursive binary k-means tree in which each internal node stores the mean of its descendant subtree as a routing representative, and recursion terminates at $\ell = 1024$ items per leaf. Within-leaf sampling is performed via an exact softmax over the leaf members. This yields $\mathcal{O}(\log_2(N/\ell))$ routing decisions per query, providing a logarithmic speedup over the two-level case, while keeping memory usage comparable to the 2LS family. We verified that the results are robust to the choice of leaf size over the range $[256, 4096]$.

C.2. Per-Query Latency Across Sampling Methods

Table 2. Per-query latency in milliseconds, averaged over temperatures $\tau \in \{0.05, 0.1, 0.2\}$, with $K = 1024$ clusters and $n_q = 1000$ queries, reported as mean ± 95% confidence intervals. Speed-up gains are computed relative to the exact softmax sampling method, denoted *Exact*. Best results per dataset are in bold, and second-best are underlined.

Dataset	Exact	Top-k	2LS	Hierarchical	S-2LS (ours)	SD-2LS (ours)
GloVe-100	10.30 ± 0.04	0.439 ± 0.004 (23.4×)	0.235 ± 0.003 (43.9×)	**0.221 ± 0.001 (46.7×)**	0.245 ± 0.003 (42.1×)	0.722 ± 0.009 (14.3×)
VK-LSVD	173.0 ± 5.0	2.720 ± 0.019 (63.6×)	2.564 ± 0.728 (67.5×)	**1.383 ± 0.019 (125×)**	2.662 ± 0.751 (65.0×)	2.906 ± 0.529 (59.5×)
YAMBDA	79.28 ± 6.39	2.013 ± 0.018 (39.4×)	1.265 ± 0.063 (62.7×)	**1.185 ± 0.083 (66.9×)**	1.660 ± 0.434 (47.7×)	3.171 ± 1.227 (25.0×)
Synth-Bal	8.543 ± 0.036	0.289 ± 0.002 (29.6×)	0.222 ± 0.004 (38.4×)	0.218 ± 0.001 (39.3×)	**0.213 ± 0.004 (40.2×)**	0.690 ± 0.005 (12.4×)
Synth-Unbal	8.619 ± 0.032	0.317 ± 0.003 (27.2×)	0.213 ± 0.004 (40.5×)	**0.211 ± 0.001 (40.8×)**	0.214 ± 0.004 (40.2×)	0.691 ± 0.005 (12.5×)

C.3. Cluster Size Distributions Across Datasets

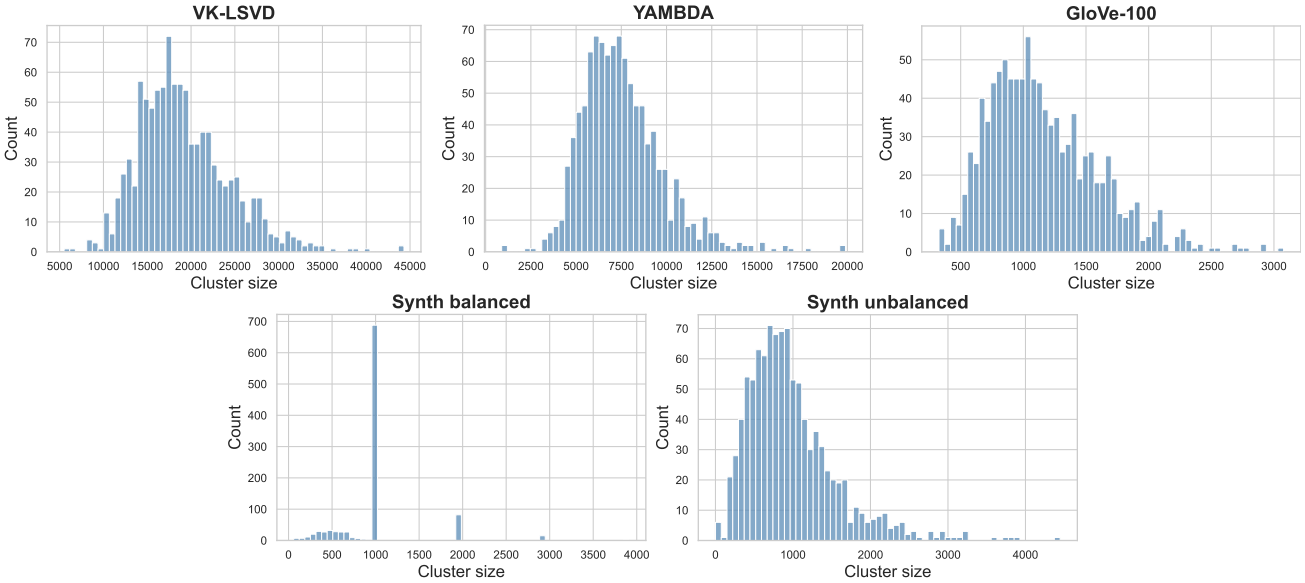

Figure 4. Cluster size distributions at $K = 1024$ across all five datasets, obtained via k-means clustering. The natural recommender corpora (VK-LSVD, YAMBDA) and GloVe-100 exhibit substantial size imbalance, with cluster sizes spanning more than an order of magnitude. The synthetic corpora serve as controlled regimes: Synth-balanced has tightly concentrated cluster sizes, while Synth-unbalanced replicates the heavy-tailed distributions observed in natural corpora. This imbalance is precisely what the size correction in S-2LS and SD-2LS is designed to address. Recall that Synth-balanced was generated from a Gaussian mixture model with an equal number of points per component. However, the k-means clustering used in our experiments does not exactly recover the ground-truth components. As a result, some observed clusters have different sizes even on Synth-balanced.

C.4. Complete Bias Analysis Across All Datasets

Figures 1, 2, and 3 in the main paper present experimental results showing that the 2LS biases theoretically characterized in Section 3 are observable and substantial in practice, and that the proposed S-2LS and SD-2LS methods effectively correct them.

Specifically, Figure 1 reports the sampling ratio $R_i^{(N)}(q)$ as a function of cluster size, Figure 2 reports $R_i^{(N)}(q)$ as a function of intra-cluster similarity variance, and Figure 3 reports the sampling probability per item as a function of query-item similarity.

Due to space constraints, these figures in the main paper focus on the VK-LSVD dataset. For completeness, we provide in this appendix *the full set of analyses*, including results for the four remaining datasets: YAMBDA, GloVe-100, Synth-Balanced, and Synth-Unbalanced. The results lead to conclusions consistent with those reported in the main paper.

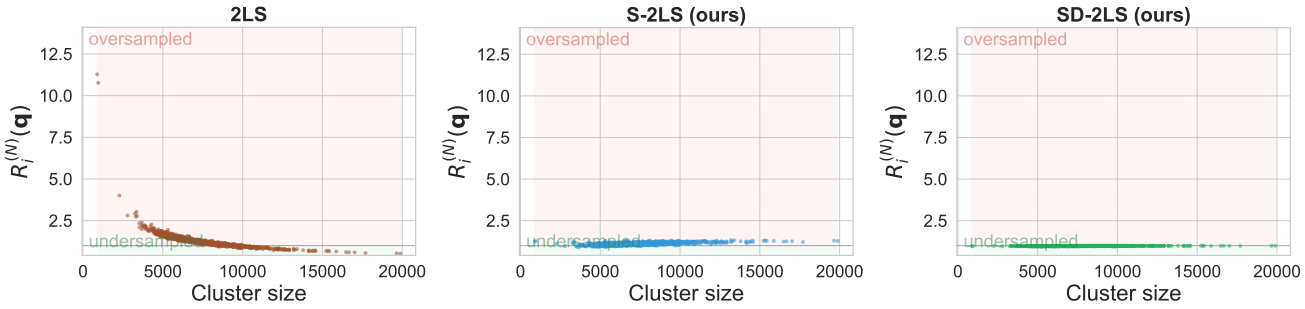

Figure 5. Sampling ratio on **YAMBDA** ($\tau = 0.1$) as a function of cluster size, averaged over queries. Standard 2LS undersamples large clusters and oversamples small ones relative to exact softmax ($R_i^{(N)}(q) \neq 1$, p-value $p < 0.01$), while S-2LS and SD-2LS correct these biases.

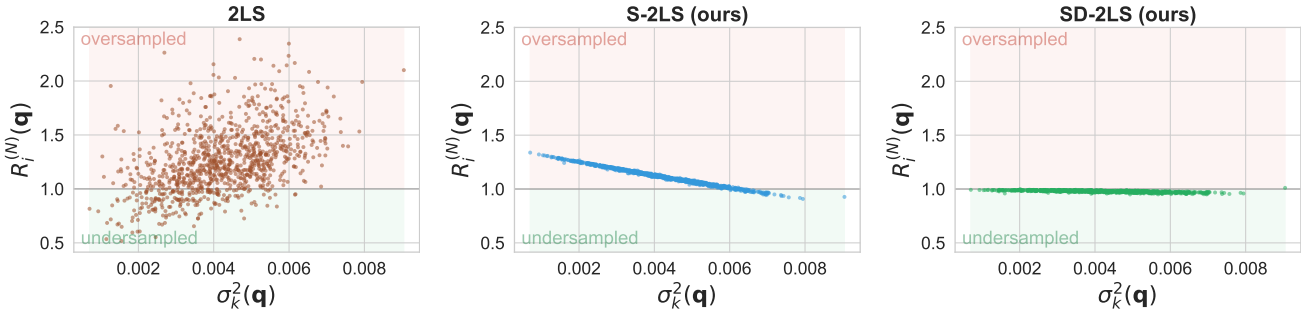

Figure 6. Sampling ratio on **YAMBDA** ($\tau = 0.1$) as a function of intra-cluster similarity variance, averaged over queries. The effect is entangled with cluster-size bias for 2LS; S-2LS corrects the cluster-size bias but not the variance bias, isolating the latter and revealing that high- (respectively, low-) variance clusters are undersampled (resp., oversampled) relative to exact softmax ($R_i^{(N)}(q) \neq 1, p < 0.01$). SD-2LS corrects these biases.

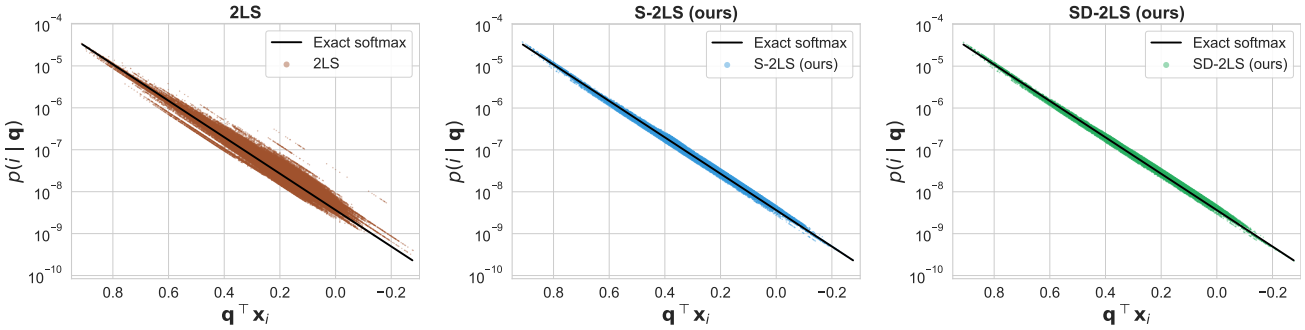

Figure 7. Sampling probability on **YAMBDA** ($\tau = 0.1$) per item as a function of query-item dot-product similarity for a fixed query. Standard 2LS assigns widely different probabilities to items with identical similarity, while S-2LS and SD-2LS markedly reduce this dispersion.

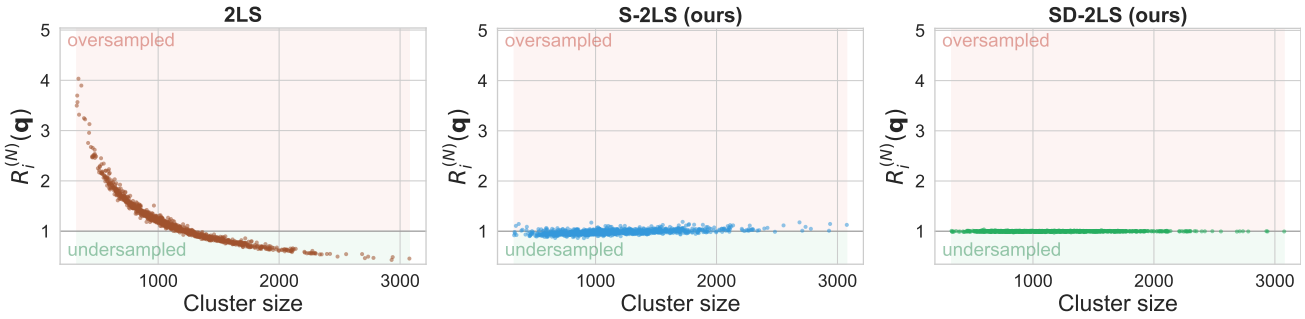

Figure 8. Sampling ratio on **GloVe-100** ($\tau = 0.1$) as a function of cluster size, averaged over queries. Standard 2LS undersamples large clusters and oversamples small ones relative to exact softmax ($R_i^{(N)}(q) \neq 1$, p-value $p < 0.01$), while S-2LS and SD-2LS correct these biases.

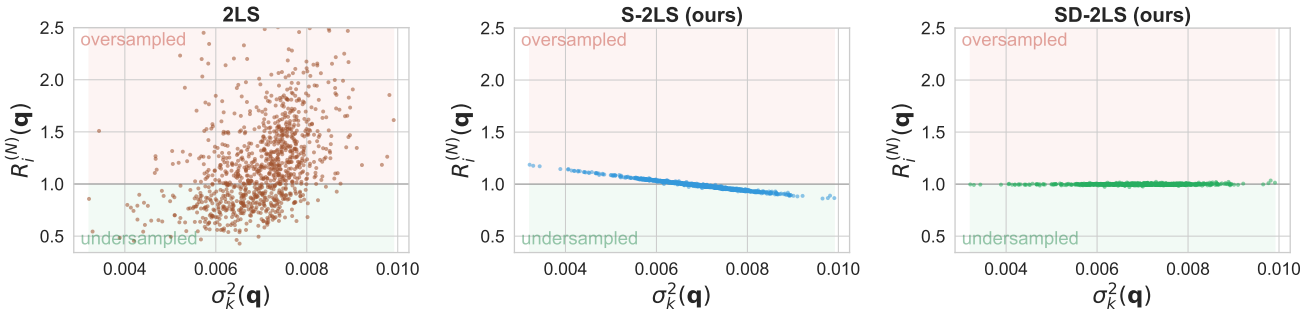

Figure 9. Sampling ratio on **GloVe-100** ($\tau = 0.1$) as a function of intra-cluster similarity variance, averaged over queries. The effect is entangled with cluster-size bias for 2LS; S-2LS corrects the cluster-size bias but not the variance bias, isolating the latter and revealing that high- (respectively, low-) variance clusters are undersampled (resp., oversampled) relative to exact softmax ($R_i^{(N)}(q) \neq 1, p < 0.01$). SD-2LS corrects these biases.

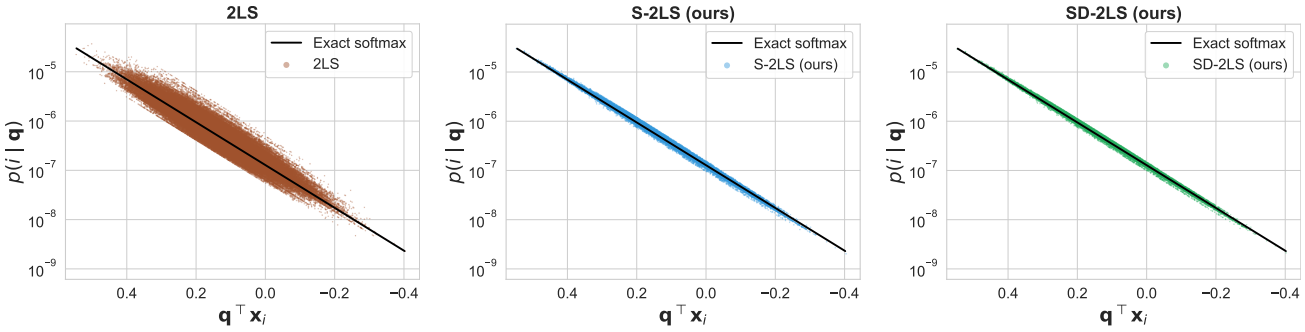

Figure 10. Sampling probability on **GloVe-100** ($\tau = 0.1$) per item as a function of query-item dot-product similarity for a fixed query. Standard 2LS assigns widely different probabilities to items with identical similarity, while S-2LS and SD-2LS markedly reduce this dispersion.

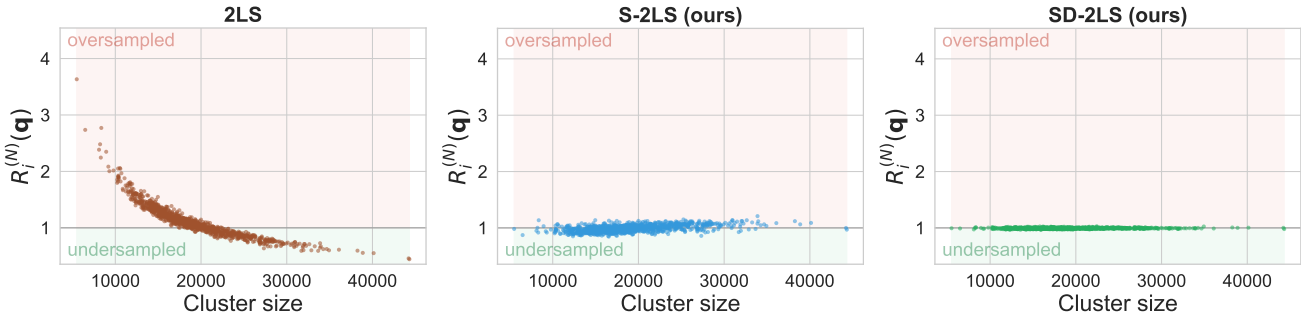

Figure 11. Sampling ratio on **VK-LSVD** ($\tau = 0.1$) as a function of cluster size, averaged over queries. Standard 2LS undersamples large clusters and oversamples small ones relative to exact softmax ($R_i^{(N)}(q) \neq 1$, p-value $p < 0.01$), while S-2LS and SD-2LS correct these biases.

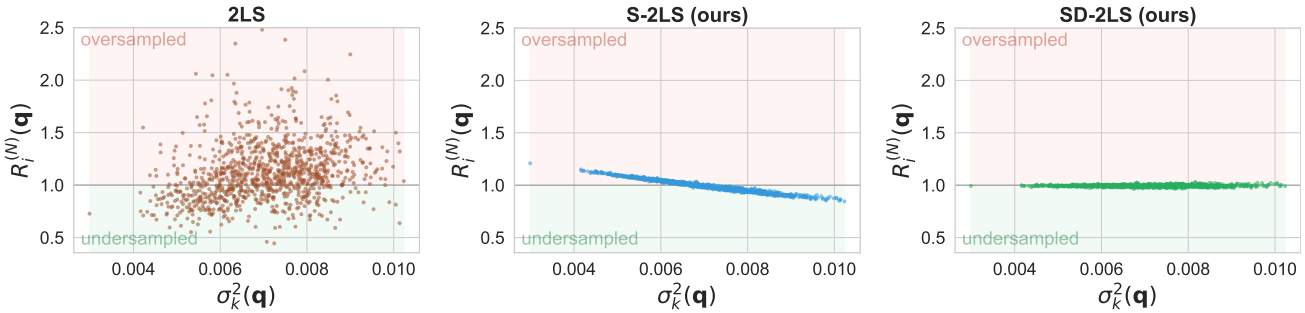

Figure 12. Sampling ratio on **VK-LSVD** ($\tau = 0.1$) as a function of intra-cluster similarity variance, averaged over queries. The effect is entangled with cluster-size bias for 2LS; S-2LS corrects the cluster-size bias but not the variance bias, isolating the latter and revealing that high- (respectively, low-) variance clusters are undersampled (resp., oversampled) relative to exact softmax ($R_i^{(N)}(q) \neq 1, p < 0.01$). SD-2LS corrects these biases.

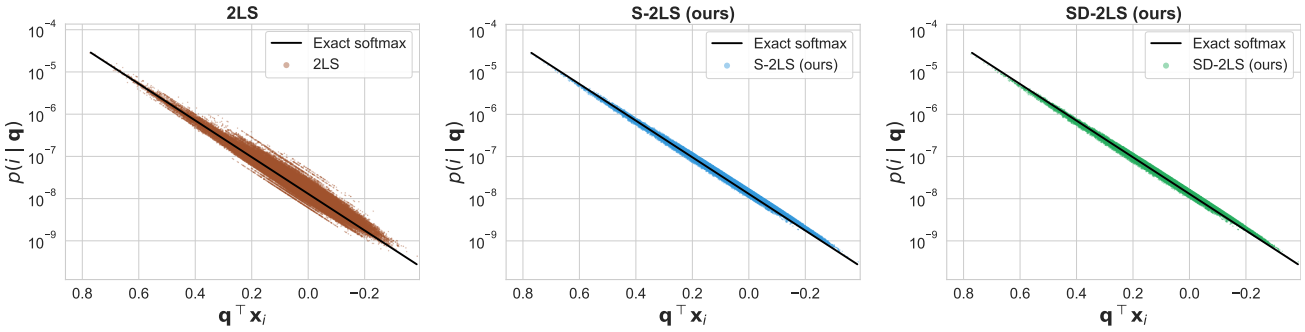

Figure 13. Sampling probability on **VK-LSVD** ($\tau = 0.1$) per item as a function of query-item dot-product similarity for a fixed query. Standard 2LS assigns widely different probabilities to items with identical similarity, while S-2LS and SD-2LS markedly reduce this dispersion.

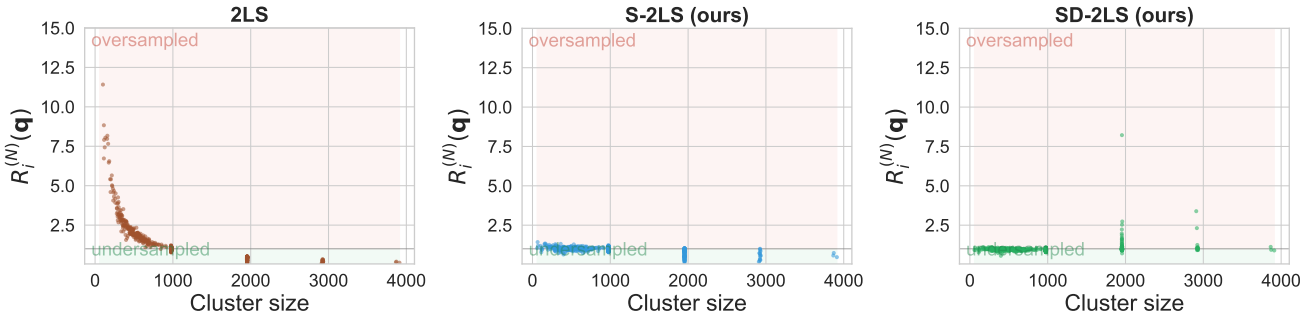

Figure 14. Sampling ratio on **Synth-Balanced** ($\tau = 0.1$) as a function of cluster size, averaged over queries. Standard 2LS undersamples large clusters and oversamples small ones relative to exact softmax ($R_i^{(N)}(q) \neq 1$, p-value $p < 0.01$), while S-2LS and SD-2LS correct these biases.

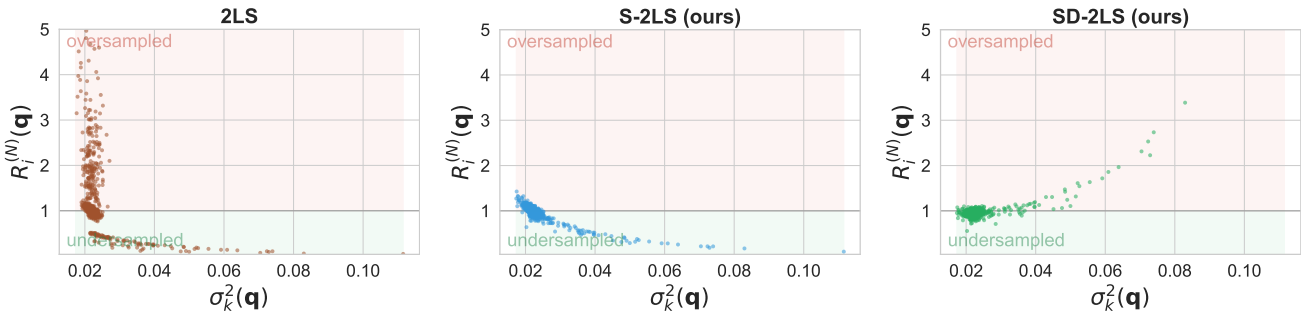

Figure 15. Sampling ratio on **Synth-Balanced** ($\tau = 0.1$) as a function of intra-cluster similarity variance, averaged over queries. The effect is entangled with cluster-size bias for 2LS; S-2LS corrects the cluster-size bias but not the variance bias, isolating the latter and revealing that high- (respectively, low-) variance clusters are undersampled (resp., oversampled) relative to exact softmax ($R_i^{(N)}(q) \neq 1, p < 0.01$). SD-2LS corrects these biases for the vast majority of points.

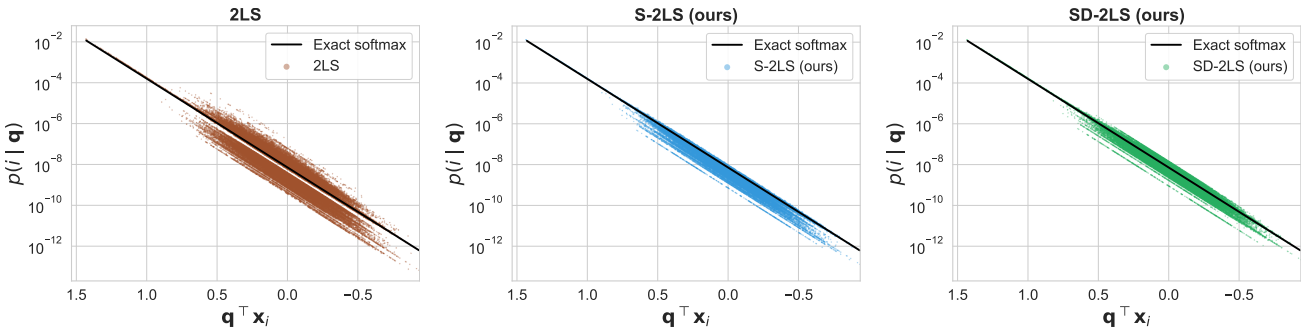

Figure 16. Sampling probability on **Synth-Balanced** ($\tau = 0.1$) per item as a function of query-item dot-product similarity for a fixed query. Standard 2LS assigns widely different probabilities to items with identical similarity, while S-2LS and SD-2LS markedly reduce this dispersion.

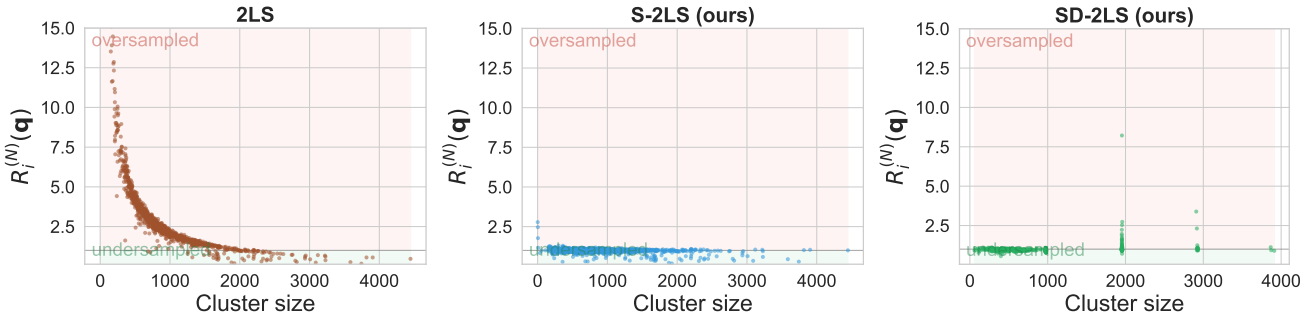

Figure 17. Sampling ratio on **Synth-Unbalanced** ($\tau = 0.1$) as a function of cluster size, averaged over queries. Standard 2LS undersamples large clusters and oversamples small ones relative to exact softmax ($R_i^{(N)}(q) \neq 1$, p-value $p < 0.01$), while S-2LS and SD-2LS correct these biases.

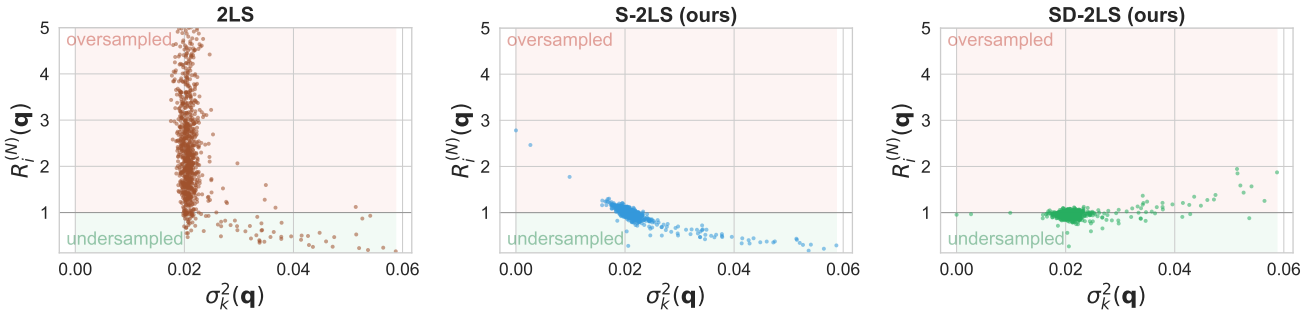

Figure 18. Sampling ratio on **Synth-Unbalanced** ($\tau = 0.1$) as a function of intra-cluster similarity variance, averaged over queries. The effect is entangled with cluster-size bias for 2LS; S-2LS corrects the cluster-size bias but not the variance bias, isolating the latter and revealing that high- (respectively, low-) variance clusters are undersampled (resp., oversampled) relative to exact softmax ($R_i^{(N)}(q) \neq 1, p < 0.01$). SD-2LS corrects these biases for the vast majority of points.

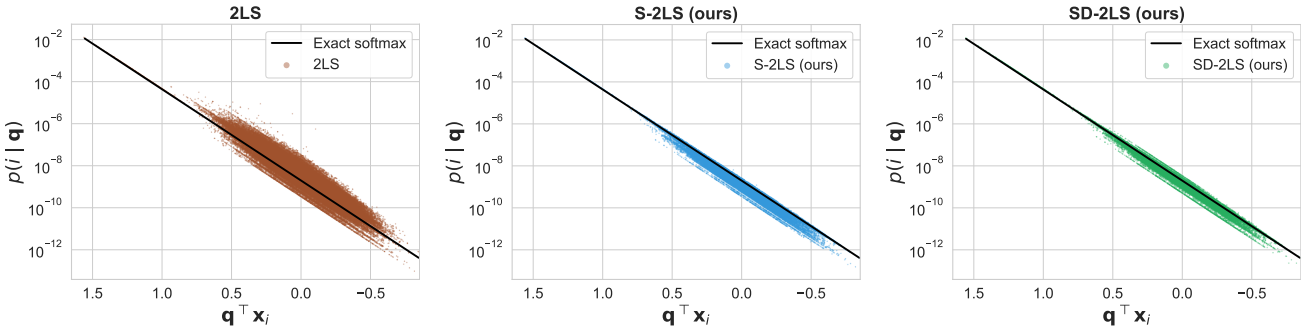

Figure 19. Sampling probability on **Synth-Unbalanced** ($\tau = 0.1$) per item as a function of query-item dot-product similarity for a fixed query. Standard 2LS assigns widely different probabilities to items with identical similarity, while S-2LS and SD-2LS markedly reduce this dispersion.