MULTI-POSITIVE TOP-K RECOMMENDATION: A DIFFERENTIABLE DYNAMIC PROGRAMMING APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern recommender systems typically optimize for top-K retrieval; however, they largely rely on continuous surrogate loss functions due to the non-differentiable nature of ranking metrics. A fundamental limitation of these conventional losses is that they evaluate positive items independently—contrasting a single positive item against one or more negative items per instance. Consequently, they fail to capture the joint combinatorial structure and mutual associations among a user's multiple positive interactions. To overcome this bottleneck, we adopt a novel framework powered by **Differentiable Dynamic Programming** (DiffDP), which reformulates multi-positive item selection as a dynamic programming process. Specifically, we leverage the Fenchel-Young loss framework—a theoretically rigorous surrogate objective that enables direct optimization over joint multi-positive subsets. By regularizing the multi-positive selection problem with Shannon entropy, our dynamic programming formulation recovers the exact log-partition function of a Conditional Poisson Binomial Distribution (CPBD). Unlike Gumbel-perturbed relaxes or sampling-based approaches, our formulation models simultaneous unordered sets under a strict global capacity constraint. This key property enables the exact, deterministic, and noise-free computation of marginal inclusion probabilities via automatic differentiation. Extensive experiments on real-world benchmark datasets demonstrate that our approach yields competitive top-K recommendation performance. The code is provided anonymously at `https://anonymous.4open.science/r/Differentiable-Dynamic-Programming-1857`.

1 INTRODUCTION

Modern recommender systems are designed to retrieve a top-K list of items tailored to a user's preferences. Because standard ranking metrics like Recall@K and NDCG@K rely on non-differentiable sorting operations, models cannot be optimized directly on these targets. Instead, practitioners rely on continuous surrogate loss functions to train their models, such as pointwise Binary Cross-Entropy (BCE), pairwise Bayesian Personalized Ranking (BPR; Rendle et al. (2012)), and listwise Softmax/InfoNCE (Oord et al., 2018).

While these surrogate losses are formulated differently, they share a restrictive underlying assumption: they all cast the training process as an independent **single-positive selection** task, contrasting only one positive item against negative items. Under this paradigm, the model's target space is implicitly constrained to a single, mutually exclusive choice (i.e., a one-hot vector), where the objective is to score exactly one positive item higher than a set of negative items.

This single-positive assumption introduces a critical *single-positive vs. multi-positive mismatch*. In real-world applications, user feedback is rarely restricted to a single item; users frequently interact with multiple relevant items within the same context (e.g., clicking several related products in a session). Because conventional losses are structurally confined to one-hot targets, they are forced to break down a user's multi-positive interactions into isolated, independent single-positive training events. This forced decomposition is highly problematic for representation learning. First, it completely destroys the joint associations and combinatorial structure among the user's preferred items. Second, it frequently induces "false competition"—when the model strictly optimizes the Softmax

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

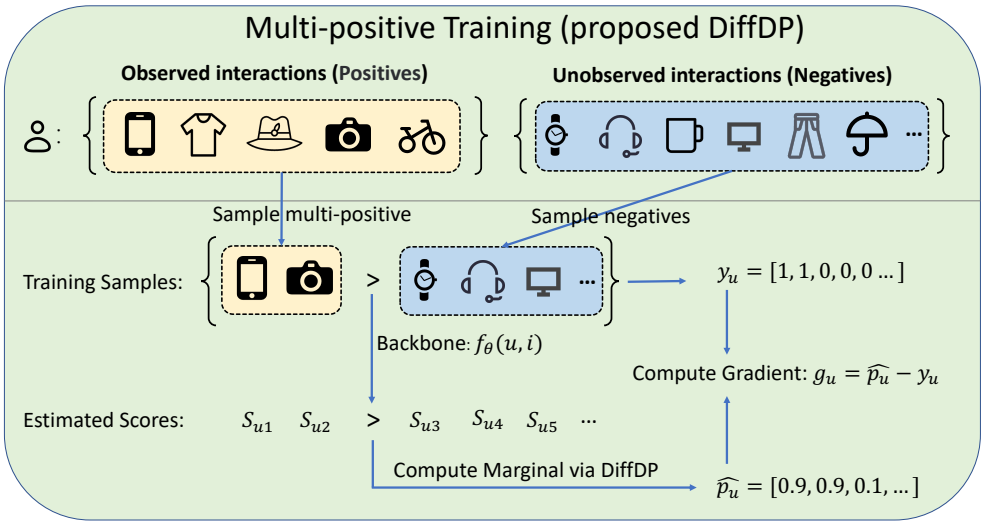


Figure 1: An illustration of the proposed *multi-positive paradigm* for Top-K recommendation.

probability for one positive item, it inadvertently suppresses the predicted probabilities of the user's other relevant items.

To bridge this gap, the underlying selection mechanism must be extended from isolated single-item selection to simultaneous multi-positive selection (i.e., a P-hot vector, where P denotes the number of multiple positive items). However, directly optimizing this joint selection over a discrete space introduces two intrinsic challenges. On the one hand, unlike one-hot boolean space, the size of a P-hot boolean hypercube grows exponentially with the item catalog size M (i.e., $\binom{M}{P}$), making the computation of exact choice probabilities intractable due to **combinatorial explosion**. On the other hand, existing attempts to differentiate P-hot selections typically rely on sampling relaxation (e.g., the Gumbel perturbed trick). These sampling methods map to the Plackett-Luce distribution, which requires $O(M!/(M-P)!)$ complexity for exact marginalization, forcing practitioners to rely on Monte Carlo approximations or biased straight-through estimators that inject significant variance into training (Berthet et al., 2020) .

To overcome these challenges, we propose a **Differentiable Dynamic Programming** (DiffDP) framework for multi-positive Top-K recommendation, inspired by Mensch & Blondel (2018). Instead of relying on noisy stochastic perturbations, we reframe P-hot subset selection as a sequential decision process over the item catalog, which we solve exactly via dynamic programming. By regularizing the DP state transitions with Shannon entropy, we transform the non-differentiable combinatorial subset selection into a convex, smoothly differentiable operation. Figure 1 illustrates the idea of DiffDP for multi-positive paradigm.

Crucially, we show that this entropy-regularized DP mathematically induces a *Conditional Poisson Binomial Distribution (CPBD)*, accurately modeling the joint probability of unordered multi-positive sets. Integrating this continuous relaxation into the Fenchel-Young (FY) loss framework (Blondel et al., 2020) yields two key properties: (i) **Negative Correlation:** It explicitly models item substitution effects across the P available exposure slots. (ii) **Exactness and Efficiency:** Despite operating over an exponentially large combinatorial space, our DP algorithm computes the exact log-partition function in $O(MP)$ time. Consequently, the loss gradient is fully deterministic and reduces to an elegant, noise-free residual between exact DP marginal probabilities and ground-truth targets, executed seamlessly via standard automatic differentiation.

In summary, the main contributions of this work are three-fold:

- We introduce a differentiable dynamic programming formulation to resolve the *single-positive vs. multi-positive mismatch* in top-K recommendation, leveraging Fenchel-Young duality to establish a theoretically sound multi-positive surrogate loss.

- We perform evaluations across multiple real-world benchmark datasets. Results show that directly optimizing over the combinatorial P-hot space with our exact DP formulation yields competitive top-K ranking performance.
- We conduct extensive empirical analyses on the positive target cardinality P, demonstrating that expanding the choice space over a P-hot hypercube consistently unlocks representation capacity and yields steady performance gains across diverse domains.

2 PRELIMINARIES

In this section, we formally define the top-K recommendation task, review the conventional surrogate loss functions, and highlight the systemic limitations of existing paradigms.

2.1 TOP-K RECOMMENDATION

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ denote a set of N users and $\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$ denote a catalog of M items. Historical user-item interactions are represented by a binary matrix $\mathbf{Y} \in \{0,1\}^{N \times M}$, where $y_{ui} = 1$ indicates an observed interaction between user u and item i, and $y_{ui} = 0$ otherwise. For each user u, we denote their observed positive items as $\mathcal{I}_u^+ = \{i \in \mathcal{I} \mid y_{ui} = 1\}$ and unobserved items as $\mathcal{I}_u^- = \mathcal{I} \setminus \mathcal{I}_u^+$. We can also express user u's interaction profile as a binary target vector $\mathbf{y}_u = [y_{u1}, \ldots, y_{uM}]^\top \in \mathcal{C}$, where $\mathcal{C} \subseteq \{0,1\}^M$ represents the discrete target choice space. The objective of a recommendation model is to learn a scoring function $f_\theta : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ that outputs a score vector $\mathbf{s}_u = [s_{u1}, \ldots, s_{uM}]^\top \in \mathbb{R}^M$ for user u. During inference, the system generates a top-K list by retrieving the K unobserved items with the highest predicted scores in $\mathbf{s}_u$.

2.2 CONVENTIONAL SURROGATE LOSS FUNCTIONS

Standard ranking metrics (e.g., Recall@K, NDCG@K) rely on non-differentiable sorting operations. To enable gradient-based optimization, continuous surrogate losses are widely adopted.

Pointwise Loss. Pointwise methods frame recommendation as binary classification. A representative objective is Binary Cross-Entropy (BCE):

$$\mathcal{L}_{\text{BCE}} = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \Big(y_{ui} \log \sigma(s_{ui}) + (1 - y_{ui}) \log(1 - \sigma(s_{ui})) \Big), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function.

Pairwise Loss. Pairwise approaches maximize the score margin between observed and unobserved items. Bayesian Personalized Ranking (BPR; Rendle et al. (2012)) is the canonical baseline:

$$\mathcal{L}_{\text{BPR}} = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u^+} \sum_{j \in \mathcal{I}_u^-} \log \sigma(s_{ui} - s_{uj}). \tag{2}$$

Listwise Loss. Listwise methods, such as Softmax Cross-Entropy (or InfoNCE (Oord et al., 2018)), optimize the target item's score relative to negative samples:

$$\mathcal{L}_{\text{Softmax}} = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u^+} \log \frac{\exp(s_{ui})}{\exp(s_{ui}) + \sum_{j \in \mathcal{I}_u^-} \exp(s_{uj})}. \tag{3}$$

2.3 SINGLE-POSITIVE VS. MULTI-POSITIVE MISMATCH

Despite differences in formulation, conventional surrogate losses evaluate items independently or pair-wise—contrasting a *single* positive item against one or more negative items per instance. These losses implicitly restrict the target choice space $\mathcal{C}$ to one-hot encodings (i.e., $\mathbf{c} \in \{0,1\}^M$ with $\sum_i c_i = 1$), assuming user interactions are single, mutually exclusive events. In real-world recommendation, however, users naturally interact with multiple positive items within the same context. Decomposing holistic user histories into isolated single-positive training instances induces a fundamental *single-positive vs. multi-positive mismatch*, which inherently destroys joint item associations and triggers false competition among positive candidates. To bridge this gap, we expand

the choice space to simultaneous multi-positive subset selections via a P-hot combinatorial space $\mathcal{C}_P = \{\mathbf{c} \in \{0,1\}^M \mid \sum_i c_i = P\}$, enabling smooth and direct optimization over joint user preferences.

3 Differentiable Dynamic Programming Approach

Building upon entropy-regularized dynamic programming (DP) formulation, we leverage a theoretically grounded surrogate loss tailored for multi-positive recommendation. In this section, we derive the P-hot Fenchel-Young loss, demonstrate its exact and deterministic optimization process via DP, and showcase its integration into top-K recommendation frameworks.

3.1 Exact Marginals via Differentiable Dynamic Programming

To resolve the single-positive mismatch, we explicitly extend the target choice space to the P-hot boolean hypercube $\mathcal{C}_P = \{\mathbf{c} \in \{0,1\}^M \mid \sum_{i=1}^M c_i = P\}$. Hard top-$P$ selection over candidate score vector $\mathbf{s}_u = [s_{u1}, \ldots, s_{uM}]^\top$ corresponds to finding a binary subset vector $\mathbf{c}^*$ that maximizes the total score:

$$\max_{\mathbf{c}\in\{0,1\}^M} \sum_{i=1}^{M} c_i s_{ui} \quad \text{s.t.} \quad \sum_{i=1}^{M} c_i = P. \tag{4}$$

In standard combinatorial optimization, Eq. (4) can be solved via dynamic programming (DP) using the recurrence relation $V_{\text{hard}}[i,p] = \max\{V_{\text{hard}}[i-1,p], V_{\text{hard}}[i-1,p-1] + s_{ui}\}$. However, because the discrete $\max$ operator has zero gradients almost everywhere, this exact combinatorial formulation cannot be directly integrated into end-to-end neural network architectures.

While traditional continuous relaxations (e.g., Plackett-Luce models parameterized via Gumbel noise) relax the discrete choices, they suffer from an $O(M!/(M-P)!)$ combinatorial sampling complexity. We circumvent this limitation entirely by introducing Shannon entropy regularization into the DP recursion, enabling polynomial-time evaluation of the exact analytical log-partition function.

Specifically, at each step i, the binary selection decision ($c_i = 1$ vs. $c_i = 0$) is regularized using Shannon Entropy $\mathcal{H}_(\pi) = -\pi \log \pi - (1-\pi)\log(1-\pi)$. By Fenchel duality, regularizing the state transitions with Shannon entropy transforms the non-differentiable $\max$ operator into a smooth LogSumExp (LSE) soft-maximum transition scaled by temperature $\tau > 0$. Let $V[i,p]$ denote the smoothed value function (log-partition function) of selecting exactly p items from the first i candidate items. The forward DP recursion is formulated as:

$$V[i,p] = \tau \log \left[\exp\left(\frac{V[i-1,p]}{\tau}\right) + \exp\left(\frac{V[i-1,p-1] + s_{ui}}{\tau}\right) \right], \tag{5}$$

subject to boundary conditions $V[0,0] = 0$ and $V[i,p] = -\infty$ for $p < 0$ or $p > i$. The final scalar state $W_{P,\tau}(\mathbf{s}_u) \triangleq V[M,P]$ serves as our global *P-hot surplus function*.

Theorem 1 (Exact Deterministic Marginals)**.** *Under the Shannon-entropy regularized DP framework, the exact marginal inclusion probability vector $\mathbf{p}_{u,P}(\mathbf{s}_u) \in (0,1)^M$ is deterministically given by the gradient of the final DP surplus function:*

$$\mathbf{p}_{u,P}(\mathbf{s}_u) = \nabla_{\mathbf{s}_u} V[M,P]. \tag{6}$$

Because the DP grid forms a Directed Acyclic Graph (DAG) composed strictly of standard differentiable operations, evaluating $\mathbf{p}_{u,P}(\mathbf{s}_u)$ via reverse-mode automatic differentiation requires strictly $O(MP)$ time and space, completely eliminating the variance and computational cost of stochastic sampling.

3.2 P-hot Fenchel-Young Loss Formulation

We leverage the Fenchel-Young loss framework (Blondel et al., 2020) to construct a convex surrogate objective. Let Ω_P denote the negative Shannon entropy regularization over the marginal polytope of $\mathcal{C}_P$. Its convex conjugate $\Omega_P^*(\mathbf{s}_u) = \max_{\mathbf{y}} \langle \mathbf{s}_u, \mathbf{y} \rangle - \Omega_P(\mathbf{y})$ mathematically corresponds to the log-partition function, which exactly aligns with our DP surplus $V[M,P]$.

Algorithm 1 End-to-End Training via DiffDP P-hot Selection

Require: Recommender f_θ, Training Set $\mathcal{D}$, Positive Cardinality P, Temperature τ, Learning rate η.
Ensure: Optimized parameters θ.

while not converged **do**
  Sample a mini-batch of users $\mathcal{B} \subset \mathcal{D}$
  $\mathcal{L}_{\text{batch}} \leftarrow 0$
  for each user $u \in \mathcal{B}$ **do**
    Compute scores $\mathbf{s}_u \leftarrow f_\theta(u, \mathcal{I})$
    Initialize DP table $V \in \mathbb{R}^{(M+1)\times(P+1)}$ with $-\infty$, and set $V[0,0] \leftarrow 0$
    for $i = 1 \dots M$ **do**
      for $p = 0 \dots \min(i, P)$ **do**
        $V[i,p] \leftarrow \tau \log\left(\exp\left(\frac{V[i-1,p]}{\tau}\right) + \exp\left(\frac{V[i-1,p-1]+s_{ui}}{\tau}\right)\right)$
      end for
    end for
    $\mathcal{L}_u \leftarrow V[M,P] - \langle \mathbf{s}_u, \mathbf{y}_u \rangle$ ▷ Fenchel-Young Surrogate Loss
    $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}_u$
  end for
  $\mathcal{L}_{\text{batch}} \leftarrow \frac{1}{|\mathcal{B}|}\mathcal{L}_{\text{batch}}$
  Compute $\nabla_\theta \mathcal{L}_{\text{batch}}$ via `Autograd.backward()` ▷ Yields exact $\mathbf{p}_{u,P} - \mathbf{y}_u$ internally
  Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{batch}}$
end while
return θ

Definition 1 (DiffDP Fenchel-Young Loss)**.** *Let $\mathbf{y}_u \in \mathcal{C}_P$ be the ground-truth multi-hot target for user u, and $\mathbf{s}_u \in \mathbb{R}^M$ be the predicted score vector. The **DiffDP Fenchel-Young loss** $\mathcal{L}^P_{FY}$ is defined as:*

$$\mathcal{L}^P_{FY}(\mathbf{s}_u, \mathbf{y}_u) = V[M,P] - \langle \mathbf{s}_u, \mathbf{y}_u \rangle + \Omega_P(\mathbf{y}_u), \tag{7}$$

where $\Omega_P(\mathbf{y}_u)$ is a constant w.r.t. model parameters and can be omitted during training.

Proposition 1 (Properties of DiffDP Fenchel-Young loss)**.** *For any $\tau > 0$, the loss $\mathcal{L}^P_{FY}(\mathbf{s}_u, \mathbf{y}_u)$ satisfies:*

- ***Non-negativity & Convexity:*** *$\mathcal{L}^P_{FY}(\mathbf{s}_u, \mathbf{y}_u) \geq 0$, and is convex w.r.t. $\mathbf{s}_u$.*
- ***Gradients:*** *$\mathcal{L}^P_{FY}$ is continuously differentiable, with its exact gradient reducing to the residual between marginal probabilities and ground-truth targets:*

$$\nabla_{\mathbf{s}_u} \mathcal{L}^P_{FY}(\mathbf{s}_u, \mathbf{y}_u) = \mathbf{p}_{u,P}(\mathbf{s}_u) - \mathbf{y}_u. \tag{8}$$

Eq. (8) shows that optimizing the recommender model corresponds to minimizing the discrepancy between predicted marginals $\mathbf{p}_{u,P}$ and observed interactions $\mathbf{y}_u$. Modern autograd engines allow us to optimize this loss without manually deriving backward passes. The full training algorithm is summarized in Algorithm 1.

3.3 Probabilistic Interpretation

By regularizing state transitions with Shannon entropy, our model directly evaluates the exact marginal inclusion probabilities of a *Conditional Poisson Binomial Distribution (CPBD)*. Conditioning M independent Bernoulli trials (with logits s_{ui}/τ) on selecting exactly P items induces a Gibbs measure over the P-hot hypercube $\mathcal{C}_P$, defined as $\mathbb{P}(\mathbf{c} \mid \sum_{i=1}^M c_i = P) = \frac{1}{Z_\tau(M,P)} \exp(\frac{1}{\tau}\sum_{i=1}^M c_i s_{ui})$, where $Z_\tau(M,P) = \sum_{\mathbf{c}' \in \mathcal{C}_P} \exp(\frac{1}{\tau}\langle \mathbf{s}_u, \mathbf{c}' \rangle)$ is the partition function. Our smoothed DP value $V[M,P]$ evaluates $\tau \log Z_\tau(M,P)$ exactly, allowing deterministic gradient computation for exact CPBD marginals. A crucial property of this formulation is its intrinsic *negatively correlated* (Chen, 2000) under the global budget constraint $\sum_{i=1}^M p_{u,P}(\mathbf{s}_u)_i = P$. This forces active catalog competition: elevating an item's score s_{ui} automatically suppresses competing candidates, providing implicit probability calibration without heuristic negative sampling.

Compared to sequential Gumbel-Top-P (Plackett-Luce) approaches, DiffDP offers two major theoretical advantages. First, while Plackett-Luce models impose an artificial sequential ordering over sampled sequences, DiffDP models simultaneous choice over an *unordered* subset, perfectly aligning with set-based top-K evaluation metrics like Recall@K. Second, marginalizing over all $P!$ sequence permutations in Plackett-Luce is computationally intractable, forcing Gumbel approaches to rely on high-variance stochastic approximations. In contrast, DiffDP computes noise-free, exact marginal probabilities deterministically in $O(MP)$ time via dynamic programming over a combinatorial graph.

4 EXPERIMENTS

In this section, we evaluate the proposed DiffDP Fenchel-Young loss against state-of-the-art surrogate loss baselines. Specifically, our experiments are designed to address the following key research questions: (1) How does our loss compare against existing baselines across diverse backbones? and (2) How does the positive target cardinality P impact overall recommendation performance?

4.1 EXPERIMENTAL SETUP

In our training process, we utilize P-hot vectors to represent user-item interactions. The term "multi-hot" indicates that multiple positive items are considered for a given user, where P serves as a fixed hyperparameter denoting the exact number of positive samples drawn during each training step.

Datasets. We conduct experiments on four widely-used public benchmark datasets that vary across domains and sparsity levels: Beauty, VideoGames, Gowalla, and Pinterest. Following standard procedures, all datasets are preprocessed via 5-core filtering.

Evaluation Metrics. To assess the top-K ranking performance, we employ standard evaluation metrics: Precision@K (P@K), Recall@K (R@K), NDCG@K (N@K) and Mean Reciprocal Rank (MRR). We rank all unobserved items in the catalog for each user and report the average performance and set the truncation length $K = 20$ for all experiments.

Backbone Architectures. To demonstrate the universality and model-agnostic nature of our proposed loss, we evaluate its performance across three representative recommendation backbones of varying complexity: Matrix Factorization (MF) (Rendle et al., 2012), LightGCN (He et al., 2020), and XSimGCL (Yu et al., 2023).

Baselines (Loss Functions). We compare our PFY against a comprehensive suite of recommendation losses, which can be categorized into: (1) *Conventional* 1*-hot oriented losses*, including the pairwise BCE, BPR (Rendle et al., 2012) and the listwise Softmax (Wu et al., 2024a). (2) *Advanced Softmax variants*, including BSL (Wu et al., 2024b) and PSL (Yang et al., 2024) . (3) *Metric-driven and smoothed ranking losses*, including SmoothI (Thonet et al., 2022), LLPAUC (Shi et al., 2024) and Talos (Zhang et al., 2026).

Implementation Details. All models and loss functions are implemented in PyTorch. To ensure a fair comparison, the latent embedding dimension is fixed at $d = 64$ across all backbone architectures. We optimize the models using the Adam optimizer, tuning the learning rate based on validation performance over a predefined search space.

To ensure a fair comparison, the number of negative samples is fixed at 128 per instance across all baselines and DiffDP. Multi-positive samples are drawn randomly with replacement per user. We search the positive cardinality $P \in \{1, 2, \ldots, 10\}$ and the entropy temperature $\tau \in \{0.05, 0.10, 0.20\}$. Further implementation details and hyperparameter settings are available in the supplementary material.

4.2 OVERALL PERFORMANCE

Tables 1 and 2 report the top-20 recommendation performance across four benchmarks and three backbone architectures. Based on the empirical results, our proposed DiffDP achieves competitive performance across different backbone architectures. On modern representation models (LightGCN and XSimGCL), DiffDP achieves comprehensive superiority over all baselines on all datasets. Even

Table 1: Overall recommendation performance in terms of Precision@20 and Recall@20. All reported results are test means over five formal seeds.

Model	Loss	Beauty		VideoGames		Gowalla		Pinterest	
		P@20	R@20	P@20	R@20	P@20	R@20	P@20	R@20
MF	BPR	0.0124	0.1085	0.0044	0.0886	0.0298	0.1039	0.0285	0.1080
	BSL	0.0148	0.1236	0.0063	0.1264	0.0439	0.1448	0.0375	0.1425
	Softmax	0.0139	0.1223	0.0062	0.1239	0.0384	0.1280	0.0372	0.1415
	SmoothI	0.0144	0.1240	0.0057	0.1134	0.0428	0.1402	0.0379	0.1439
	PSL	0.0151	0.1260	0.0063	0.1256	0.0442	0.1443	0.0289	0.1097
	LLPAUC	0.0134	0.1071	0.0057	0.1140	0.0472	**0.1575**	0.0325	0.1235
	BCE	0.0122	0.1087	0.0043	0.0856	0.0311	0.1084	0.0282	0.1065
	Talos	0.0157	0.1343	0.0064	0.1280	0.0452	0.1504	0.0398	0.1514
	DiffDP(ours)	**0.0170**	**0.1429**	**0.0068**	**0.1360**	**0.0476**	0.1566	**0.0412**	**0.1566**
LightGCN	BPR	0.0124	0.1116	0.0042	0.0843	0.0266	0.0874	0.0251	0.0951
	BSL	0.0154	0.1308	0.0062	0.1248	0.0462	0.1503	0.0390	0.1480
	Softmax	0.0156	0.1329	0.0064	0.1278	0.0374	0.1239	0.0348	0.1325
	SmoothI	0.0152	0.1308	0.0048	0.0951	0.0448	0.1448	0.0395	0.1501
	PSL	0.0154	0.1292	0.0063	0.1263	0.0419	0.1380	0.0306	0.1157
	LLPAUC	0.0148	0.1317	0.0053	0.1066	0.0298	0.1026	0.0206	0.0773
	BCE	0.0123	0.1110	0.0041	0.0828	0.0239	0.0858	0.0252	0.0949
	Talos	0.0164	0.1405	0.0067	0.1331	0.0464	0.1537	0.0409	0.1553
	DiffDP(ours)	**0.0170**	**0.1430**	**0.0068**	**0.1367**	**0.0482**	**0.1580**	**0.0413**	**0.1572**
XSimGCL	BPR	0.0131	0.1080	0.0052	0.1048	0.0310	0.1081	0.0335	0.1274
	BSL	0.0140	0.1125	0.0063	0.1268	0.0458	0.1490	0.0416	0.1581
	Softmax	0.0153	0.1295	0.0063	0.1263	0.0393	0.1311	0.0417	0.1584
	SmoothI	0.0127	0.1005	0.0048	0.0960	0.0393	0.1253	0.0327	0.1244
	PSL	0.0153	0.1266	0.0063	0.1269	0.0419	0.1383	0.0297	0.1114
	LLPAUC	0.0156	0.1331	0.0061	0.1213	0.0378	0.1307	0.0375	0.1428
	BCE	0.0146	0.1216	0.0060	0.1190	0.0359	0.1253	0.0352	0.1336
	Talos	0.0162	0.1380	0.0066	0.1327	0.0454	0.1506	0.0421	0.1597
	DiffDP(ours)	**0.0168**	**0.1399**	**0.0068**	**0.1358**	**0.0471**	**0.1543**	**0.0422**	**0.1600**

on the simpler MF backbone, DiffDP achieves top performance across almost all metrics, outperforming the strongest baseline (Talos) by a clear margin.

4.3 Impact of Positive Cardinality P

To address RQ2, we investigate the impact of the positive target cardinality P on recommendation performance. Specifically, we vary $P \in \{1, 2, \ldots, 10\}$ and evaluate our proposed DiffDP loss across three representative backbone architectures (MF, LightGCN, and XSimGCL). Figure 2 illustrates the general sensitivity analysis on Precision@20 across all four datasets. Based on the empirical curves, we highlight two key observations:

1. Superiority of Multi-Positive Modeling. Across all datasets and backbones, setting $P > 1$ consistently yields substantial performance gains compared to the traditional single-positive baseline ($P = 1$). This empirically validates our core premise: expanding the target choice space to accommodate multiple positive items enables the model to effectively capture joint item associations that conventional single-hot losses inherently miss.

2. Monotonic Growth and Saturation Dynamics. As P increases from 1 to 10, Precision@20 exhibits a steady, monotonic upward trend on Beauty, VideoGames, and Gowalla. On Pinterest, performance rises sharply as P grows from 1 to 5, after which the marginal gains gradually saturate. These highly consistent trajectories across diverse dataset domains and sparsity levels underscore the exceptional robustness and model-agnostic generality of DiffDP. More discussion can be found in Appendix B.2 .

Table 2: Overall recommendation performance in terms of MRR and NDCG@20. All reported results are test means over five formal seeds.

Model	Loss	Beauty		VideoGames		Gowalla		Pinterest	
		MRR	N@20	MRR	N@20	MRR	N@20	MRR	N@20
MF	BPR	0.0312	0.0591	0.0212	0.0357	0.0213	0.0760	0.0239	0.0720
	BSL	0.0399	0.0703	0.0314	0.0519	0.0330	0.1100	0.0329	0.0970
	Softmax	0.0352	0.0665	0.0307	0.0508	0.0278	0.0956	0.0320	0.0954
	SmoothI	0.0401	0.0712	0.0302	0.0482	0.0328	0.1078	0.0334	0.0981
	PSL	0.0399	0.0710	0.0316	0.0519	0.0339	0.1109	0.0246	0.0737
	LLPAUC	0.0353	0.0614	0.0305	0.0486	**0.0427**	**0.1312**	0.0280	0.0832
	BCE	0.0309	0.0587	0.0206	0.0345	0.0231	0.0812	0.0245	0.0723
	Talos	0.0425	0.0763	0.0322	0.0528	0.0356	0.1175	0.0348	0.1029
	DiffDP(ours)	**0.0454**	**0.0808**	**0.0341**	**0.0561**	0.0373	0.1218	**0.0358**	**0.1060**
LightGCN	BPR	0.0307	0.0596	0.0197	0.0335	0.0180	0.0630	0.0214	0.0639
	BSL	0.0406	0.0730	0.0311	0.0513	0.0349	0.1149	0.0344	0.1010
	Softmax	0.0415	0.0746	0.0317	0.0524	0.0261	0.0911	0.0287	0.0874
	SmoothI	0.0416	0.0742	0.0232	0.0386	0.0338	0.1108	0.0351	0.1028
	PSL	0.0413	0.0731	0.0317	0.0521	0.0304	0.1034	0.0263	0.0781
	LLPAUC	0.0381	0.0717	0.0253	0.0427	0.0209	0.0746	0.0151	0.0485
	BCE	0.0309	0.0596	0.0194	0.0330	0.0168	0.0617	0.0217	0.0641
	Talos	0.0440	0.0795	0.0334	0.0549	0.0365	0.1202	0.0358	0.1056
	DiffDP(ours)	**0.0454**	**0.0808**	**0.0340**	**0.0561**	**0.0377**	**0.1230**	**0.0358**	**0.1064**
XSimGCL	BPR	0.0326	0.0589	0.0247	0.0419	0.0199	0.0746	0.0277	0.0842
	BSL	0.0375	0.0645	0.0313	0.0519	0.0343	0.1135	0.0364	0.1075
	Softmax	0.0394	0.0716	0.0311	0.0516	0.0272	0.0953	0.0363	0.1074
	SmoothI	0.0332	0.0570	0.0235	0.0391	0.0303	0.0981	0.0274	0.0828
	PSL	0.0403	0.0713	0.0315	0.0520	0.0301	0.1027	0.0257	0.0757
	LLPAUC	0.0419	0.0754	0.0302	0.0498	0.0300	0.1012	0.0315	0.0951
	BCE	0.0370	0.0671	0.0289	0.0483	0.0244	0.0890	0.0295	0.0888
	Talos	0.0438	0.0782	0.0329	0.0543	0.0346	0.1154	0.0372	0.1091
	DiffDP(ours)	**0.0447**	**0.0792**	**0.0340**	**0.0559**	**0.0362**	**0.1191**	**0.0372**	**0.1093**

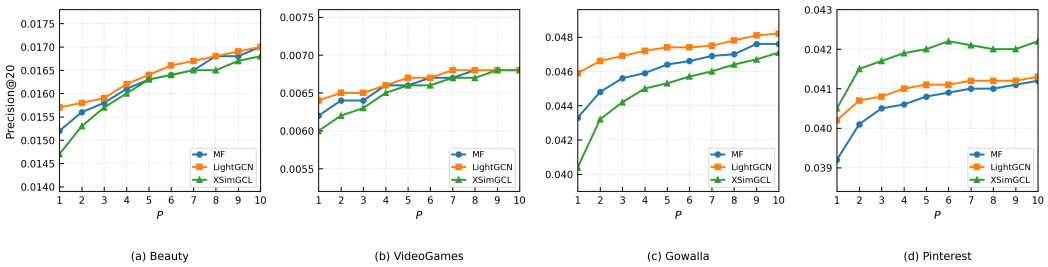


Figure 2: Sensitivity analysis of positive cardinality P on Precision@20.

We also compare our proposal against a strong surrogate loss *Talos* (Zhang et al., 2026). Figure 3 presents a detailed head-to-head comparison under the LightGCN backbone. While DiffDP yields comparable or slightly lower performance than Talos at $P = 1$, expanding the positive cardinality P rapidly enables DiffDP to surpass Talos across all four benchmarks. Specifically, DiffDP consistently outperforms Talos once $P \geq 2$ on Gowalla, $P \geq 4$ on Pinterest, and $P \geq 5$ on Beauty and VideoGames, achieving substantial margins at $P = 10$. This demonstrates that smooth Fenchel-Young optimization over the P-hot hypercube more effectively unlocks the representation capacity of recommendation backbones. More details of comparison can be found in Appendix B.3.

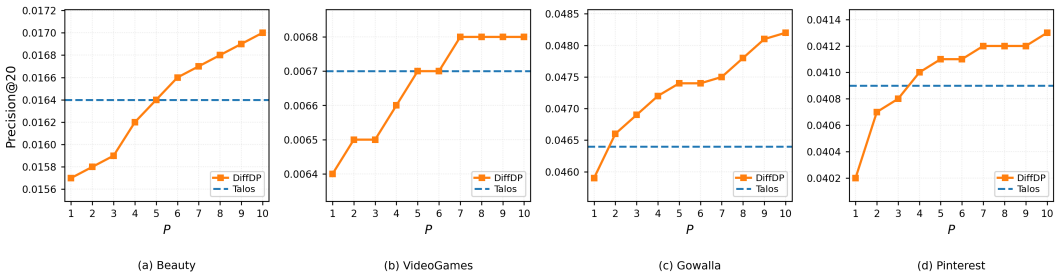


Figure 3: Precision@20 with respect to positive cardinality P for DiffDP versus Talos on LightGCN.

5 Related Work

For a comprehensive overview of the evolution of modern recommender systems, we refer readers to recent surveys (Zang et al., 2022; Balog & Zhai, 2023). Our work specifically intersects two core domain areas: surrogate loss design in recommendation systems and differentiable multi-item subset selection.

Conventional Surrogate Losses. Due to the non-differentiable nature of discrete item sorting, training recommendation models fundamentally relies on continuous surrogate losses. Early pairwise methods, such as BPR (Rendle et al., 2012), maximize the score margin between observed and unobserved items. Subsequently, listwise approaches—particularly Softmax Cross-Entropy (InfoNCE) (Oord et al., 2018)—have become the de facto standard, spurring variants to mitigate popularity bias (Wu et al., 2024b), enhance hard negative mining (Zhang et al., 2023; Yang et al., 2024), and optimize sampling efficiency (Wu et al., 2024a). However, standard Softmax-based losses implicitly assume a mutually exclusive, single-positive choice space. By ignoring joint item dependencies among multiple user interactions, single-positive losses suffer from a structural mismatch. To mitigate this limitations, recent efforts attempt to optimize top-K ranking directly—either via continuous relaxations and metric smoothing (e.g., NeuralNDCG (Pobrotyn & Białobrzeski, 2021), Smooth Rank Indicators (Thonet et al., 2022), and Recall@K surrogates (Patel et al., 2022)) or by concentrating model capacity on top-K decision boundaries (Boyd et al., 2012; Shi et al., 2024; Yang et al., 2025; Zhang et al., 2026). Another paradigm for multi-item modeling relies on Determinantal Point Processes (DPPs) (Kulesza & Taskar, 2012; 2011; Mariet & Sra, 2015), which capture repulsiveness and subset diversity through kernel determinants. However, while DPPs effectively mitigate item redundancy, their model parameters and kernel matrices scale quadratically, introducing severe computational and statistical bottlenecks during optimization (Gartrell et al., 2020; Liu et al., 2022).

Differentiable Multi-Item Selection. Beyond recommendation, resolving the non-differentiability of discrete combinatorial subset selection remains a fundamental challenge in machine learning. Existing differentiable top-k mechanisms generally fall into three main technical paradigms: *stochastic perturbations*, which utilize Gumbel-max relaxations or continuous reparameterizations to sample top-k subsets (Xie & Ermon, 2019; Struminsky et al., 2021); *implicit or relaxed gradient estimators*, which approximate inclusion probabilities via implicit maximum likelihood (I-MLE (Niepert et al., 2021)), recursive variable definitions (SIMPLE (Ahmed et al., 2022)), rescaled sigmoids (NCPSS (Pervez et al., 2023)), or score function estimators (SFESS (Wijk et al., 2025)); and *exact algorithmic differentiation*, which leverages dynamic programming over combinatorial graphs to evaluate deterministic, noise-free gradients (Vivier-Ardisson et al., 2026).

6 Conclusion

In this paper, we addressed the fundamental single-positive versus multi-positive mismatch inherent to traditional surrogate losses in recommender systems. To solve this, we proposed DiffDP, a novel differentiable dynamic programming loss framework that elevates the underlying decision space from a simple one-hot target to a P-hot multi-positive hypercube. By regularizing state transitions with Shannon entropy, DiffDP evaluates exact marginal probabilities, unifying multi-item choice modeling with the Fenchel-Young loss framework. Extensive experiments across four benchmark datasets and three representative backbone architectures demonstrate that DiffDP outperforms state-of-the-art surrogate loss baselines in most cases while exhibiting model-agnostic generality.

AI USE STATEMENT

In this work, we used generative AI tools for proofreading and typographical error correction. Additionally, we used generative AI tools for minor Python code generation. LLM-generated code was verified and tested for correctness by the authors. We have not used generative AI tools for research idea generation or theoretical formulations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research does not raise any potential violations of the ICLR Code of Ethics. In particular, it does not involve human subjects, dataset release practices, or potentially harmful insights, methodologies, or applications. It also raises no concerns regarding conflicts of interest or sponsorship, discrimination, bias, or fairness, nor does it present issues related to privacy, security, legal compliance, or research integrity (e.g., IRB approval, documentation, or research ethics).

REPRODUCIBILITY STATEMENT

The code for reproducing our results is provided anonymously at `https://anonymous.4open.science/r/Differentiable-Dynamic-Programming-1857`.

REFERENCES

Kareem Ahmed, Zhe Zeng, Mathias Niepert, and Guy Van den Broeck. Simple: A gradient estimator for k-subset sampling. *arXiv preprint arXiv:2210.01941*, 2022.

Krisztian Balog and ChengXiang Zhai. User simulation for evaluating information access systems. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 302–305, 2023.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

Mathieu Blondel, André FT Martins, and Vlad Niculae. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. Accuracy at the top. *Advances in neural information processing systems*, 25, 2012.

Sean X Chen. General properties and estimation of conditional bernoulli models. *Journal of Multivariate Analysis*, 74(1):69–87, 2000.

Sean X Chen and Jun S Liu. Statistical applications of the poisson-binomial and conditional bernoulli distributions. *Statistica Sinica*, pp. 875–892, 1997.

Mike Gartrell, Insu Han, Elvis Dohmatob, Jennifer Gillenwater, and Victor-Emmanuel Brunel. Scalable learning and map inference for nonsymmetric determinantal point processes. *arXiv preprint arXiv:2006.09862*, 2020.

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

Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 1193–1200, 2011.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Yuli Liu, Christian Walder, and Lexing Xie. Determinantal point process likelihoods for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 1653–1663, 2022.

Zelda Mariet and Suvrit Sra. Fixed-point algorithms for learning determinantal point processes. In *International Conference on Machine Learning*, pp. 2389–2397. PMLR, 2015.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pp. 3462–3471. PMLR, 2018.

Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit mle: backpropagating through discrete exponential family distributions. *Advances in Neural Information Processing Systems*, 34:14567–14579, 2021.

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

Yash Patel, Giorgos Tolias, and Jiří Matas. Recall@ k surrogate loss with large batches and similarity mixup. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7502–7511, 2022.

Adeel Pervez, Phillip Lippe, and Efstratios Gavves. Scalable subset sampling with neural conditional poisson networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Przemysław Pobrotyn and Radosław Białobrzeski. Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. *arXiv preprint arXiv:2102.07831*, 2021.

Yuanhao Pu, Defu Lian, and Enhong Chen. Is softmax loss all you need? a principled analysis of softmax-family loss. *arXiv preprint arXiv:2601.22745*, 2026.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

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

Kirill Struminsky, Artyom Gadetsky, Denis Rakitin, Danil Karpushkin, and Dmitry P Vetrov. Leveraging recursive gumbel-max trick for approximate inference in combinatorial spaces. *Advances in Neural Information Processing Systems*, 34:10999–11011, 2021.

Xiangqian Sun, Chaoqun Wang, and Wei Zhang. Perturbed dynamic time warping: A probabilistic framework and generalized variants. In *The Fourteenth International Conference on Learning Representations*, 2026.

Thibaut Thonet, Yagmur Gizem Cinar, Eric Gaussier, Minghan Li, and Jean-Michel Renders. Listwise learning to rank based on approximate rank indicators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8494–8502, 2022.

Germain Vivier-Ardisson, Michaël E Sander, Axel Parmentier, and Mathieu Blondel. Differentiable knapsack and top-k operators via dynamic programming. *arXiv preprint arXiv:2601.21775*, 2026.

Klas Wijk, Ricardo Vinuesa, and Hossein Azizpour. Sfess: Score function estimators for k-subset sampling. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. *arXiv preprint arXiv:1901.10517*, 2019.

Weiqin Yang, Jiawei Chen, Xin Xin, Sheng Zhou, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. Psl: Rethinking and improving softmax loss from pairwise perspective for recommendation. *Advances in Neural Information Processing Systems*, 37:120974–121006, 2024.

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

Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):913–926, 2023.

Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. A survey on cross-domain recommendation: taxonomies, methods, and future directions. *ACM Transactions on Information Systems*, 41(2):1–39, 2022.

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

Shengjia Zhang, Weiqin Yang, Jiawei Chen, Peng Wu, Yuegang Sun, Gang Wang, Qihao Shi, and Can Wang. Talos: Optimizing top-k accuracy in recommender systems. *arXiv preprint arXiv:2601.19276*, 2026.

A Derivation of Differentiable Dynamic Programming

Here, we provide a self-contained derivation showing that introducing Shannon entropy regularization into the exact dynamic programming (DP) formulation transforms the non-differentiable hard-max operator into the smooth LogSumExp (LSE) soft-maximum recursion. Related results can also be found in Mensch & Blondel (2018); Sun et al. (2026); Vivier-Ardisson et al. (2026)

A.1 Hard Top-P Selection

Let $\mathcal{C}_P = \{\mathbf{c} \in \{0,1\}^M \mid \sum_{i=1}^M c_i = P\}$ denote the set of all P-hot selection vectors over M candidate items. The hard Top-P subset selection problem seeks a binary vector $\mathbf{c}^*$ that maximizes total predicted score:

$$V_{\text{hard}}[M, P] = \max_{\mathbf{c} \in \mathcal{C}_P} \sum_{i=1}^{M} c_i s_{ui}. \tag{9}$$

By defining $V_{\text{hard}}[i, p]$ as the maximum total score achievable by selecting exactly p items from the first i candidates, Eq. (9) decomposes into the classical DP recursion:

$$V_{\text{hard}}[i, p] = \max\Big\{V_{\text{hard}}[i-1, p],\ V_{\text{hard}}[i-1, p-1] + s_{ui}\Big\}, \tag{10}$$

subject to boundary conditions $V_{\text{hard}}[0, 0] = 0$ and $V_{\text{hard}}[i, p] = -\infty$ for $p < 0$ or $p > i$.

A.2 Entropy Regularization via Fenchel Duality

At state (i, p), the choice whether to exclude ($c_i = 0$) or include ($c_i = 1$) item i can be represented as a discrete decision over a Bernoulli probability distribution $\boldsymbol{\pi} = [\pi_0, \pi_1]^\top \in \Delta_1$, where $\pi_1 = \mathbb{P}(c_i = 1)$, $\pi_0 = 1 - \pi_1$, and Δ_1 is the 1-simplex.

Rewriting the discrete $\max$ operator in Eq. (10) as a linear optimization problem over Δ_1 yields:

$$V_{\text{hard}}[i, p] = \max_{\boldsymbol{\pi} \in \Delta_1} \left(\pi_0 \cdot V_{\text{hard}}[i-1, p] + \pi_1 \cdot \big(V_{\text{hard}}[i-1, p-1] + s_{ui}\big)\right). \tag{11}$$

To construct a continuous relaxation, we regularize the local transition distribution $\boldsymbol{\pi}$ using Shannon entropy $\mathcal{H}(\boldsymbol{\pi}) = -\sum_{k \in \{0,1\}} \pi_k \log \pi_k$, scaled by temperature $\tau > 0$:

$$V[i, p] = \max_{\boldsymbol{\pi} \in \Delta_1} \left\{\pi_0 V[i-1, p] + \pi_1 \big(V[i-1, p-1] + s_{ui}\big) + \tau \mathcal{H}(\boldsymbol{\pi})\right\}. \tag{12}$$

To solve the constrained optimization problem in Eq. (12), let $a_0 = V[i-1, p]$ and $a_1 = V[i-1, p-1] + s_{ui}$. The objective function becomes:

$$\mathcal{L}(\pi_1) = (1 - \pi_1) a_0 + \pi_1 a_1 - \tau \left(\pi_1 \log \pi_1 + (1 - \pi_1) \log(1 - \pi_1)\right). \tag{13}$$

Taking the derivative with respect to π_1 and setting it to zero:

$$\frac{\partial \mathcal{L}}{\partial \pi_1} = a_1 - a_0 - \tau \left(\log \pi_1 + 1 - \log(1 - \pi_1) - 1\right) = 0 \tag{14}$$

$$\implies \log\left(\frac{\pi_1}{1 - \pi_1}\right) = \frac{a_1 - a_0}{\tau}. \tag{15}$$

Solving for the optimal transition probability π_1^*:

$$\pi_1^* = \frac{\exp(a_1/\tau)}{\exp(a_0/\tau) + \exp(a_1/\tau)}, \quad \pi_0^* = \frac{\exp(a_0/\tau)}{\exp(a_0/\tau) + \exp(a_1/\tau)}. \tag{16}$$

Substituting π_0^* and π_1^* back into the regularized objective in Eq. (12):

$$V[i, p] = \pi_0^* a_0 + \pi_1^* a_1 - \tau \left(\pi_0^* \log \pi_0^* + \pi_1^* \log \pi_1^*\right) \tag{17}$$

$$= \sum_{k \in \{0,1\}} \pi_k^* \left(a_k - \tau \log \pi_k^*\right). \tag{18}$$

Using Eq. (16), notice that $\log \pi_k^* = \frac{a_k}{\tau} - \log\left(\exp(a_0/\tau) + \exp(a_1/\tau)\right)$. Substituting this identity into Eq. (18) simplifies the expression directly:

$$V[i,p] = \sum_{k\in\{0,1\}} \pi_k^* \left(a_k - \tau\left[\frac{a_k}{\tau} - \log\left(\exp\left(\frac{a_0}{\tau}\right) + \exp\left(\frac{a_1}{\tau}\right)\right)\right]\right) \tag{19}$$

$$= \sum_{k\in\{0,1\}} \pi_k^* \cdot \tau \log\left(\exp\left(\frac{a_0}{\tau}\right) + \exp\left(\frac{a_1}{\tau}\right)\right) \tag{20}$$

$$= \tau \log\left(\exp\left(\frac{a_0}{\tau}\right) + \exp\left(\frac{a_1}{\tau}\right)\right) \cdot \underbrace{(\pi_0^* + \pi_1^*)}_{=1}. \tag{21}$$

Substituting back $a_0 = V[i-1,p]$ and $a_1 = V[i-1,p-1] + s_{ui}$ yields the exact LogSumExp DP recursion:

$$V[i,p] = \tau \log\left[\exp\left(\frac{V[i-1,p]}{\tau}\right) + \exp\left(\frac{V[i-1,p-1] + s_{ui}}{\tau}\right)\right]. \tag{22}$$

This completes the derivation.

A.3 Proof of Theorem 1

In this section, we provide the formal mathematical proof for Theorem 1, establishing that taking the gradient of the smoothed surplus function $V[M,P]$ with respect to the input score vector $\mathbf{s}_u$ yields the exact marginal inclusion probabilities $\mathbf{p}_{u,P}(\mathbf{s}_u)$.

It is known that conditioning M independent Bernoulli trials (with logits s_{ui}/τ) on selecting exactly P items induces a Gibbs measure over the P-hot boolean hypercube $\mathcal{C}_P = \{\mathbf{c} \in \{0,1\}^M \mid \sum_{i=1}^M c_i = P\}$ (Chen & Liu, 1997; Ahmed et al., 2022). The joint probability of selecting a specific subset $\mathbf{c} \in \mathcal{C}_P$ is given by:

$$\mathbb{P}(\mathbf{c} \mid \mathbf{s}_u, P) = \frac{1}{Z_\tau(M,P)} \exp\left(\frac{\langle \mathbf{s}_u, \mathbf{c}\rangle}{\tau}\right), \tag{23}$$

where $Z_\tau(M,P) = \sum_{\mathbf{c}'\in\mathcal{C}_P} \exp\left(\frac{\langle \mathbf{s}_u, \mathbf{c}'\rangle}{\tau}\right)$ is the partition function summing over all $\binom{M}{P}$ valid subsets.

By induction on the forward LogSumExp dynamic programming recursion derived in Appendix A, the smoothed value function $V[M,P]$ computes the scaled log-partition function of the Conditional Poisson Binomial Distribution (CPBD) exactly:

$$V[M,P] = \tau \log Z_\tau(M,P) = \tau \log\left(\sum_{\mathbf{c}\in\mathcal{C}_P} \exp\left(\frac{\langle \mathbf{s}_u, \mathbf{c}\rangle}{\tau}\right)\right). \tag{24}$$

To evaluate the i-th component of the gradient $\nabla_{\mathbf{s}_u} V[M,P]$, we compute the partial derivative of $V[M,P]$ with respect to s_{ui}:

$$\frac{\partial V[M,P]}{\partial s_{ui}} = \frac{\partial}{\partial s_{ui}}\left[\tau \log\left(\sum_{\mathbf{c}\in\mathcal{C}_P} \exp\left(\frac{1}{\tau}\sum_{j=1}^M c_j s_{uj}\right)\right)\right] \tag{25}$$

$$= \tau \cdot \frac{1}{\sum_{\mathbf{c}'\in\mathcal{C}_P} \exp\left(\frac{\langle \mathbf{s}_u, \mathbf{c}'\rangle}{\tau}\right)} \cdot \frac{\partial}{\partial s_{ui}}\left(\sum_{\mathbf{c}\in\mathcal{C}_P} \exp\left(\frac{1}{\tau}\sum_{j=1}^M c_j s_{uj}\right)\right) \tag{26}$$

$$= \tau \cdot \frac{1}{Z_\tau(M,P)} \sum_{\mathbf{c}\in\mathcal{C}_P} \left[\exp\left(\frac{\langle \mathbf{s}_u, \mathbf{c}\rangle}{\tau}\right) \cdot \frac{c_i}{\tau}\right]. \tag{27}$$

Canceling out the temperature parameter τ:

$$\frac{\partial V[M,P]}{\partial s_{ui}} = \sum_{\mathbf{c}\in\mathcal{C}_P} c_i \cdot \frac{\exp\left(\frac{\langle \mathbf{s}_u, \mathbf{c}\rangle}{\tau}\right)}{Z_\tau(M,P)} \tag{28}$$

$$= \sum_{\mathbf{c}\in\mathcal{C}_P} c_i \cdot \mathbb{P}(\mathbf{c} \mid \mathbf{s}_u, P) \tag{29}$$

$$= \mathbb{E}_{\mathbf{c}\sim\text{CPBD}(\mathbf{s}_u,P)}[c_i]. \tag{30}$$

Since each component c_i of the indicator vector $\mathbf{c} \in \{0,1\}^M$ is a binary random variable, its expected value under the conditioned Gibbs measure corresponds exactly to the marginal probability that item i is included in the chosen P-element subset:

$$\mathbb{E}_{\mathbf{c}\sim\text{CPBD}(\mathbf{s}_u,P)}[c_i] = \mathbb{P}(i \in \text{Top-}P \text{ subset}) = p_{u,P}(\mathbf{s}_u)_i. \tag{31}$$

Thus, vectorizing over all M items gives:

$$\nabla_{\mathbf{s}_u} V[M,P] = \mathbf{p}_{u,P}(\mathbf{s}_u). \tag{32}$$

B Experimental Details

To ensure fair comparison and reproducibility, we evaluate all models across four benchmark datasets spanning diverse domains and density profiles. Table 3 summarizes the key statistics of these datasets. Furthermore, we summarize the specific hyperparameter search space for each baseline and our proposed framework in Table 4.

Table 3: Statistics of the datasets used in our experiments.

Dataset	Users	Items	Interactions	Density
Beauty	15,603	8,693	139,554	0.001029
VideoGames	24,303	10,672	231,780	0.000894
Gowalla	29,858	40,981	1,027,370	0.000840
Pinterest	55,187	9,916	1,028,553	0.001879

Table 4: Hyperparameters to be searched for each method

Method	Other hyperparameters
BPR	no other hyperparameters
Softmax	$\{\tau\}$
BSL	$\{\tau_+, \tau_-\}$
SmoothI	$\{\tau, \delta\}$
PSL	$\{\tau_*\}$
LLPAUC	$\{\alpha, \beta\}$
BCE	no other hyperparameters
Talos	$\{K, \tau_{\text{T}}\}$
DiffDP	$\{P, \tau\}$

Implementation Details. All models and loss functions are implemented using the PyTorch framework. To ensure a fair comparison, the latent embedding dimension is fixed to $d = 64$ for all backbone architectures. We optimize all models using the Adam optimizer, and the learning rate is selected based on validation performance from a predefined candidate set.

For loss functions requiring negative sampling, the number of negative samples is uniformly fixed to 128 for each positive instance.

- **BPR:** No additional hyperparameters are introduced. The learning rate η is searched in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.
- **BCE:** No additional hyperparameters are introduced.
- **Softmax:** The temperature τ is searched in $\{0.2, 0.5, 1.0\}$.
- **BSL:** The temperatures τ_+ and τ_- for positive and negative terms are searched in $\{0.05, 0.10, 0.15, 0.20\}$.
- **SmoothI:** The target rank is fixed to $K = 20$. The temperature τ and offset δ are searched in $\{0.02, 0.05, 0.10, 0.20\}$ and $\{0.10, 0.20\}$, respectively.
- **PSL:** The loss temperature τ_* is searched in $\{0.025, 0.05, 0.10\}$.
- **LLPAUC:** The optimization-region parameters α and β are searched in $\{0.7, 0.8, 0.9\}$ and $\{0.01, 0.02, 0.05\}$, respectively.
- **Talos:** The target cutoff K and temperature τ_{T} are searched in $\{5, 10, 20\}$ and $\{0.05, 0.10, 0.20\}$, respectively.
- **DiffDP:** The positive cardinality P is searched in $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, and the entropy temperature τ is searched in $\{0.05, 0.10, 0.20\}$.

The optimal hyperparameter settings of all datasets can be found in Tables 5 and 6 .

Table 5: Optimal hyperparameters of Beauty and VideoGames.

Model	Loss	Beauty				VideoGames			
		lr	wd	wl	others	lr	wd	wl	others
MF	BPR	10^{-2}	0	N/A	N/A	10^{-2}	0	N/A	N/A
	Softmax	10^{-2}	0	N/A	$\{0.5\}$	10^{-2}	0	N/A	$\{0.2\}$
	BSL	10^{-1}	0	N/A	$\{0.05, 0.15\}$	10^{-1}	0	N/A	$\{0.10, 0.20\}$
	SmoothI	10^{-1}	0	N/A	$\{0.20, 0.10\}$	10^{-1}	0	N/A	$\{0.20, 0.10\}$
	PSL	10^{-1}	0	N/A	$\{0.10\}$	10^{-1}	0	N/A	$\{0.10\}$
	LLPAUC	10^{-3}	0	N/A	$\{0.7, 0.05\}$	10^{-3}	0	N/A	$\{0.9, 0.02\}$
	BCE	10^{-2}	0	N/A	N/A	10^{-2}	0	N/A	N/A
	Talos	10^{-2}	0	N/A	$\{20, 0.10\}$	10^{-2}	0	N/A	$\{20, 0.10\}$
	DiffDP	10^{-2}	0	N/A	$\{10, 0.20\}$	10^{-2}	0	N/A	$\{9, 0.20\}$
LightGCN	BPR	10^{-2}	0	N/A	N/A	10^{-2}	0	N/A	N/A
	Softmax	10^{-2}	0	N/A	$\{0.2\}$	10^{-2}	0	N/A	$\{0.2\}$
	BSL	10^{-1}	0	N/A	$\{0.15, 0.20\}$	10^{-1}	10^{-8}	N/A	$\{0.05, 0.20\}$
	SmoothI	10^{-1}	0	N/A	$\{0.20, 0.10\}$	10^{-1}	10^{-8}	N/A	$\{0.20, 0.10\}$
	PSL	10^{-2}	0	N/A	$\{0.10\}$	10^{-1}	0	N/A	$\{0.10\}$
	LLPAUC	10^{-2}	0	N/A	$\{0.7, 0.05\}$	10^{-2}	0	N/A	$\{0.7, 0.05\}$
	BCE	10^{-2}	0	N/A	N/A	10^{-2}	0	N/A	N/A
	Talos	10^{-2}	0	N/A	$\{20, 0.10\}$	10^{-2}	0	N/A	$\{20, 0.10\}$
	DiffDP	10^{-2}	0	N/A	$\{10, 0.20\}$	10^{-2}	0	N/A	$\{10, 0.20\}$
XSimGCL	BPR	10^{-2}	0	0.05	N/A	10^{-2}	0	0.05	N/A
	Softmax	10^{-2}	0	0.05	$\{0.5\}$	10^{-3}	0	0.05	$\{0.2\}$
	BSL	10^{-3}	10^{-8}	0.20	$\{0.10, 0.10\}$	10^{-2}	0	0.20	$\{0.05, 0.20\}$
	SmoothI	10^{-3}	10^{-8}	0.02	$\{0.10, 0.10\}$	10^{-2}	0	0.02	$\{0.10, 0.10\}$
	PSL	10^{-1}	0	0.05	$\{0.10\}$	10^{-2}	0	0.05	$\{0.10\}$
	LLPAUC	10^{-4}	0	0.05	$\{0.7, 0.05\}$	10^{-4}	0	0.05	$\{0.7, 0.05\}$
	BCE	10^{-2}	0	0.05	N/A	10^{-2}	0	0.05	N/A
	Talos	10^{-2}	0	0.05	$\{20, 0.10\}$	10^{-2}	0	0.10	$\{20, 0.10\}$
	DiffDP	10^{-2}	0	0.05	$\{10, 0.20\}$	10^{-3}	0	0.05	$\{10, 0.20\}$

Table 6: Optimal hyperparameters of Gowalla and Pinterest.

Model	Loss	Gowalla				Pinterest			
		lr	wd	wl	others	lr	wd	wl	others
MF	BPR	10^{-2}	0	N/A	N/A	10^{-3}	0	N/A	N/A
	Softmax	10^{-2}	0	N/A	$\{0.2\}$	10^{-3}	0	N/A	$\{0.2\}$
	BSL	10^{-1}	0	N/A	$\{0.10, 0.10\}$	10^{-1}	0	N/A	$\{0.10, 0.10\}$
	SmoothI	10^{-1}	0	N/A	$\{0.10, 0.10\}$	10^{-1}	0	N/A	$\{0.10, 0.10\}$
	PSL	10^{-2}	0	N/A	$\{0.05\}$	10^{-2}	0	N/A	$\{0.025\}$
	LLPAUC	10^{-3}	0	N/A	$\{0.9, 0.01\}$	10^{-2}	0	N/A	$\{0.8, 0.01\}$
	BCE	10^{-2}	0	N/A	N/A	10^{-3}	0	N/A	N/A
	Talos	10^{-2}	0	N/A	$\{20, 0.05\}$	10^{-2}	0	N/A	$\{20, 0.10\}$
	DiffDP	10^{-2}	0	N/A	$\{10, 0.10\}$	10^{-3}	0	N/A	$\{10, 0.10\}$
LightGCN	BPR	10^{-3}	0	N/A	N/A	10^{-3}	0	N/A	N/A
	Softmax	10^{-3}	0	N/A	$\{0.2\}$	10^{-3}	0	N/A	$\{0.2\}$
	BSL	10^{-1}	0	N/A	$\{0.10, 0.10\}$	10^{-1}	0	N/A	$\{0.10, 0.10\}$
	SmoothI	10^{-1}	0	N/A	$\{0.10, 0.10\}$	10^{-1}	0	N/A	$\{0.10, 0.10\}$
	PSL	10^{-2}	0	N/A	$\{0.10\}$	10^{-2}	0	N/A	$\{0.025\}$
	LLPAUC	10^{-2}	0	N/A	$\{0.7, 0.05\}$	10^{-4}	0	N/A	$\{0.8, 0.01\}$
	BCE	10^{-3}	0	N/A	N/A	10^{-3}	0	N/A	N/A
	Talos	10^{-2}	0	N/A	$\{20, 0.05\}$	10^{-2}	0	N/A	$\{20, 0.10\}$
	DiffDP	10^{-2}	0	N/A	$\{10, 0.10\}$	10^{-3}	0	N/A	$\{10, 0.10\}$
XSimGCL	BPR	10^{-3}	0	0.05	N/A	10^{-3}	0	0.10	N/A
	Softmax	10^{-3}	0	0.20	$\{0.2\}$	10^{-3}	0	0.05	$\{0.2\}$
	BSL	10^{-2}	0	0.20	$\{0.05, 0.10\}$	10^{-2}	0	0.20	$\{0.05, 0.20\}$
	SmoothI	10^{-2}	0	0.10	$\{0.05, 0.10\}$	10^{-1}	0	0.10	$\{0.10, 0.10\}$
	PSL	10^{-2}	0	0.10	$\{0.10\}$	10^{-2}	0	0.10	$\{0.025\}$
	LLPAUC	10^{-4}	0	0.20	$\{0.7, 0.05\}$	10^{-3}	0	0.10	$\{0.8, 0.01\}$
	BCE	10^{-3}	0	0.05	N/A	10^{-3}	0	0.10	N/A
	Talos	10^{-2}	0	0.20	$\{20, 0.05\}$	10^{-2}	0	0.10	$\{20, 0.10\}$
	DiffDP	10^{-2}	0	0.05	$\{10, 0.10\}$	10^{-3}	0	0.05	$\{10, 0.20\}$

B.1 Running Time Analysis

To evaluate the computational efficiency of our approach, Table 7 reports the average total wall-clock training time (in seconds) on an NVIDIA GeForce RTX 4090 GPU across all four datasets and three backbone architectures.

DiffDP ($P = 10$) introduces additional compute time compared to conventional single-positive surrogate losses (e.g., BPR and Softmax). This cost directly aligns with our theoretical complexity analysis: computing the exact P-hot marginal inclusion probabilities via dynamic programming over an $(M+1) \times (P+1)$ state grid incurs a deterministic $O(MP)$ time complexity per user-item slice, whereas standard one-hot losses require only $O(1)$ softmax or sigmoid reductions.

Crucially, the empirical wall-clock increase remains well within a practical range for realistic deployment. On larger datasets such as Pinterest, the runtime overhead of DiffDP is remarkably modest—requiring only $\sim 1.5\times$ to $2.5\times$ the training time of simple baselines like BPR (e.g., 1131.3s vs. 610.4s under LightGCN). On datasets with higher positive cardinalities per user like Gowalla, the exact DP forward-backward passes take longer ($\sim$5000s), reflecting the computational trade-off required for exact joint multi-positive modeling.

In summary, given the substantial performance gains demonstrated in Section 4, DiffDP achieves a highly favorable balance between empirical accuracy and offline computational overhead.

Table 7: Average wall-clock training time (in seconds) per run over five random seeds on an NVIDIA GeForce RTX 4090.

Dataset	Backbone	BPR	Softmax	BSL	PSL	SmoothI	LLPAUC	BCE	Talos	DiffDP-P10
Beauty	MF	128.6	130.1	111.1	103.0	138.1	132.6	118.7	146.2	723.2
	LightGCN	146.1	184.7	79.5	186.2	154.1	181.4	188.5	205.8	605.3
	XSimGCL	141.9	139.9	92.1	191.4	119.2	183.9	147.3	141.5	762.1
VideoGames	MF	174.2	276.9	214.9	173.4	227.1	194.2	210.8	205.6	1217.8
	LightGCN	115.5	305.8	200.3	359.7	128.0	320.7	268.5	334.3	1449.0
	XSimGCL	188.7	328.5	202.0	343.4	269.3	249.7	189.2	302.7	1261.2
Gowalla	MF	183.6	346.0	321.3	259.7	359.5	273.3	276.5	316.3	5103.7
	LightGCN	242.0	315.6	388.5	405.2	430.1	412.7	212.3	479.5	5265.2
	XSimGCL	368.6	421.8	458.9	482.6	448.1	505.6	436.7	466.0	5371.4
Pinterest	MF	420.5	468.3	495.7	510.4	512.8	548.6	522.1	530.5	1234.8
	LightGCN	610.4	672.6	716.8	742.5	738.9	781.5	765.4	679.4	1131.3
	XSimGCL	742.8	801.5	836.2	861.7	872.4	918.7	895.6	773.3	1267.4

B.2 Sensitivity Analysis of Positive Target Cardinality P

We conduct a comprehensive empirical investigation into how the choice of positive target cardinality P impacts recommendation performance. Specifically, we vary $P \in \{1, 2, \ldots, 10\}$ and evaluate our proposed DiffDP framework across three representative backbone architectures: Matrix Factorization (MF), LightGCN, and XSimGCL. Figure 4 illustrates the complete sensitivity trajectories across all four benchmark datasets and four primary metrics (Precision@20, Recall@20, MRR, and NDCG@20).

The empirical curves reveal two core insights regarding the behavior of multi-positive surrogate loss optimization.

Advantage of Multi-Positive Choice Space. Across all metrics, datasets, and backbone architectures, expanding the positive cardinality ($P > 1$) yields substantial and consistent performance improvements over the traditional single-positive setting ($P = 1$). This uniform gain empirically validates our central hypothesis: by relaxing the rigid one-hot target assumption and optimizing over a P-hot multi-positive hypercube, DiffDP successfully enables recommendation backbones to capture underlying joint item associations that conventional single-positive losses inherently fail to represent.

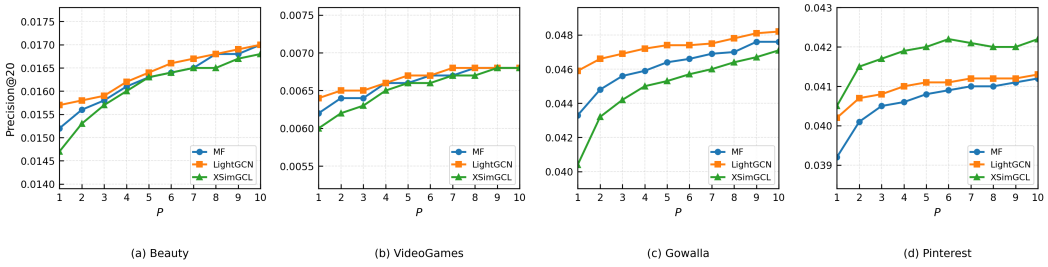
Domain-Specific Growth and Robustness. Across Beauty, VideoGames, and Gowalla, performance across all metrics exhibits a sustained, monotonic upward trend as P increases from 1 to 10, underscoring that larger choice capacities continually unlock stronger model representations. On Pinterest, all backbones experience a steep, rapid performance surge as P grows from 1 to 5, beyond which marginal gains gradually saturate. Remarkably, these multi-metric trajectories remain virtually identical whether applied to MF, LightGCN, or XSimGCL, confirming the exceptional model-agnostic generality of DiffDP.

B.3 Comparison with Baseline (Talos)

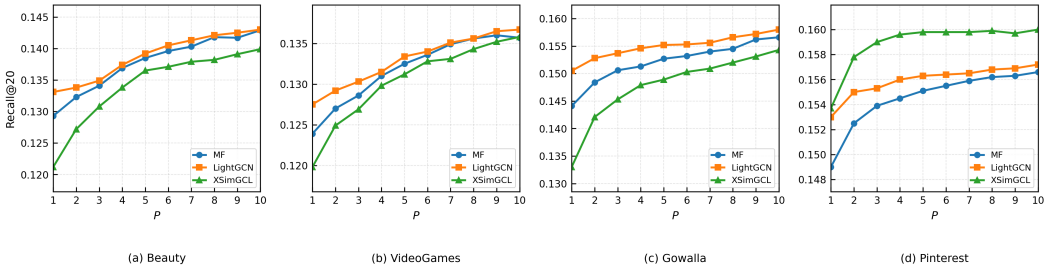
To further validate the intrinsic advantage of our proposed DiffDP framework, we presents a detailed performance comparison against the state-of-the-art loss Talos (Zhang et al., 2026) across varying positive target cardinalities P under the LightGCN backbone in Figure 5 .

As illustrated, Talos (represented by the dashed horizontal line) maintains a static benchmark performance across datasets. In contrast, our DiffDP loss demonstrates a continuous, monotonic performance improvement as the choice space expands ($P \geq 1$). Specifically, we highlight two core takeaways:

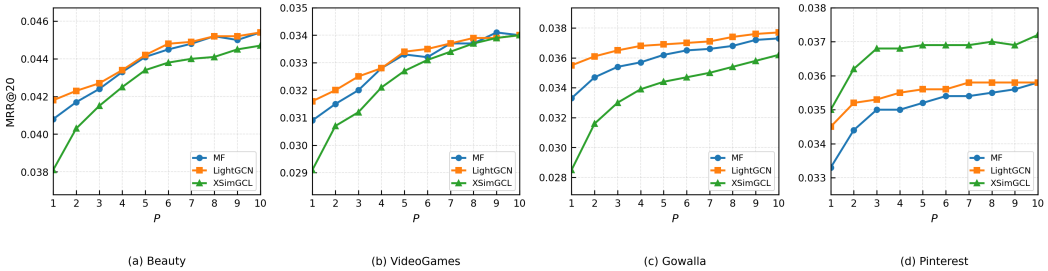
1. Clear Superiority via Choice Space Expansion. While DiffDP is comparable or slightly lower than Talos at $P = 1$, expanding P quickly enables DiffDP to surpass Talos across all four benchmarks. Specifically, DiffDP consistently outperforms Talos when $P \geq 2$ on Gowalla, $P \geq 4$ on



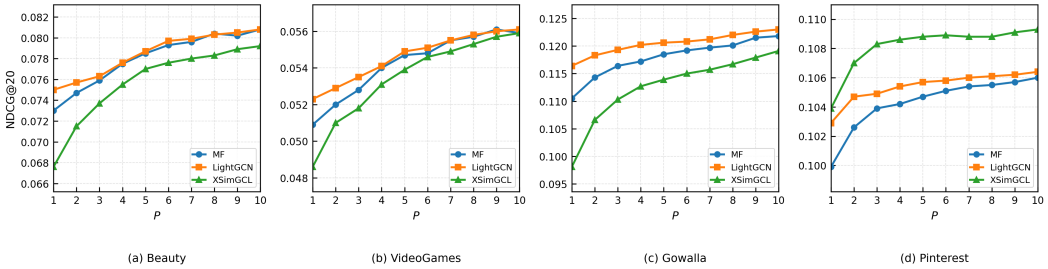
(a) Precision@20



(b) Recall@20



(c) MRR

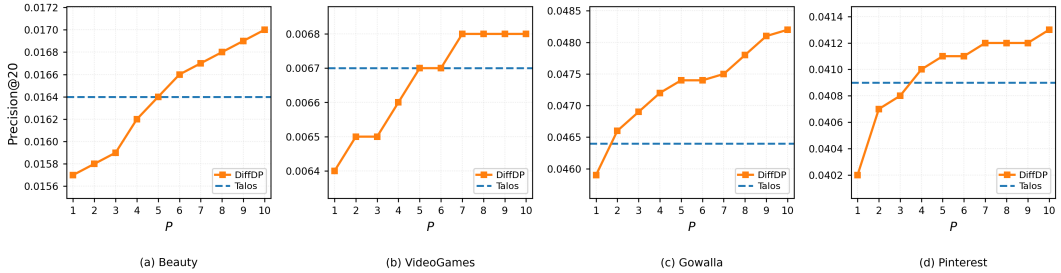


(d) NDCG@20

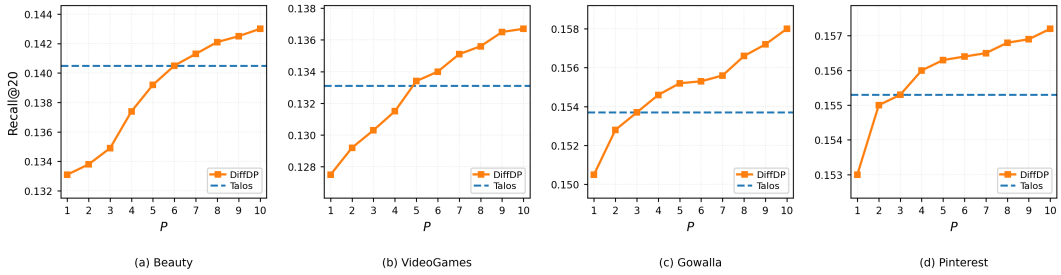
Figure 4: Sensitivity analysis of positive target cardinality P across four evaluation metrics (Precision@20, Recall@20, MRR and NDCG@20) on four benchmark datasets.

Pinterest, and $P \geq 5$ on Beauty and VideoGames, achieving substantial performance margins at $P = 10$.

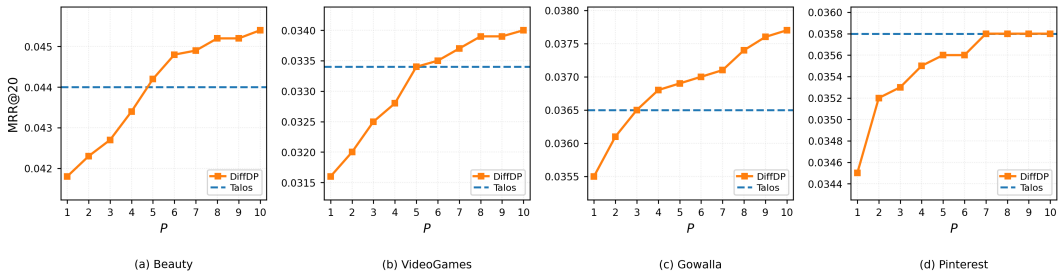
2. Effective Capacity Utilization. This comparison reveals that DiffDP's smooth Fenchel-Young optimization over the P-hot hypercube more effectively unlocks the representation power of recommendation backbones. By explicitly leveraging larger positive subsets, DiffDP surpasses existing strong baselines, confirming that proper multi-positive subset modeling is crucial for achieving state-of-the-art recommendation quality.



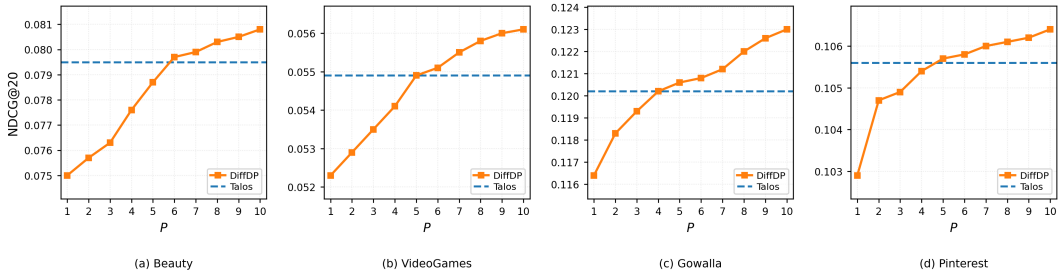
(a) Precision@20



(b) Recall@20



(c) MRR



(d) NDCG@20

Figure 5: Sensitivity analysis of positive target cardinality P comparing DiffDP and Talos on LightGCN across Precision@20, Recall@20, MRR, and NDCG@20.