Differentiable Knapsack and Top-k Operators via Dynamic Programming

Germain Vivier-Ardisson
Google DeepMind & CERMICS
Paris, France
gvivier@google.com

Michaël E. Sander
Google DeepMind
Paris, France
michaelsander@google.com

Axel Parmentier
CERMICS, CNRS, ENPC, IPP
Marne-la-Vallée, France
axel.parmentier@enpc.fr

Mathieu Blondel
Google DeepMind
Paris, France
mblondel@google.com

Abstract

Knapsack and Top-k operators are useful for selecting discrete subsets of variables. However, their integration into neural networks is challenging as they are piecewise constant, yielding gradients that are zero almost everywhere. In this paper, we propose a unified framework casting these operators as dynamic programs, and derive differentiable relaxations by smoothing the underlying recursions. On the algorithmic side, we develop efficient parallel algorithms supporting both deterministic and stochastic forward passes, and vector-Jacobian products for the backward pass. On the theoretical side, we prove that Shannon's negative entropy is the unique separable regularization choice that yields permutation-equivariant operators, and characterize regularizers inducing sparse selections. On the experimental side, we demonstrate our framework on a benchmark on learning to predict Knapsack solutions, an extension of discrete VAEs, and a constrained dynamic assortment RL problem.

1 Introduction

Many learning tasks rely on selecting a discrete subset of variables, or items, whether to learn sparse latent representations or to optimize resource allocation. The standard Top-k operator selects subsets under a fixed cardinality constraint, and the *Knapsack* problem [Martello and Toth, 1990] generalizes this structure by supporting non-uniform item weights, enforcing a total capacity limit on their sum. While the structure of these constraints provides very useful inductive bias, the resulting operators are piecewise constant, with either zero or undefined Jacobian. As a result, integrating them into neural networks is challenging as they break the differentiable computation graph, preventing the backpropagation of meaningful gradients.

To bridge this gap, current research often relies on continuous relaxations or regularization for gradient estimation. Solver-agnostic, *black-box* approaches, such as perturbation with stochastic noise [Berthet et al., 2020, Niepert et al., 2021] or piecewise affine interpolation [Pogancic et al., 2020], are flexible but computationally expensive, often requiring multiple solver calls. Conversely, specialized differentiable layers for ranking and Top-k operations, often based on optimal transport or regularized linear programming [Cuturi et al., 2019, Blondel et al., 2020b], can be efficient but lack the generality to handle the non-uniform weights inherent to Knapsack constraints. Furthermore, many existing methods restrict the choice of regularization (e.g., to Shannon entropy), limiting control over key properties like sparsity.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

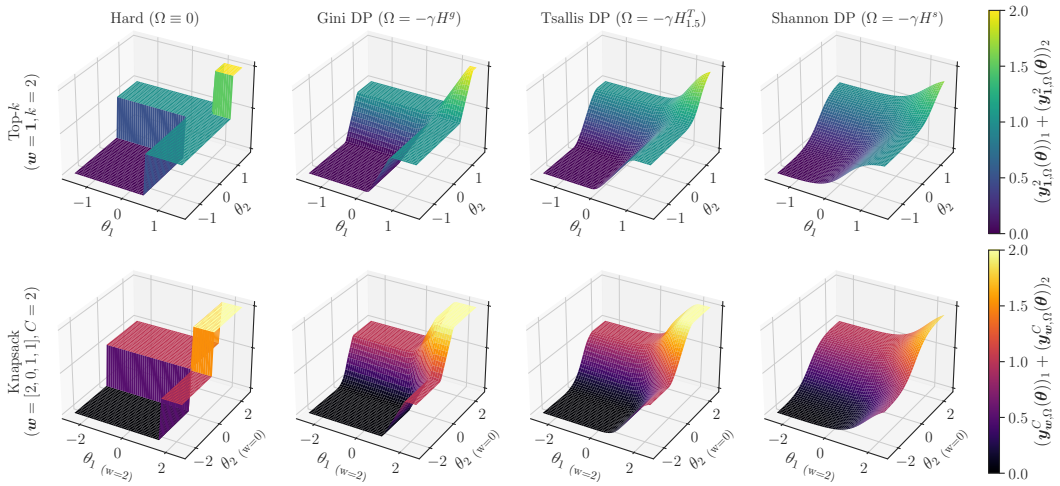


Figure 1: **Illustration of our relaxed Top-k and Knapsack operators.** We plot the sum of the first two coordinates of the operator for $\boldsymbol{\theta} = (\theta_1, \theta_2, \frac{1}{2}, 1)^\top$. In the first row, we use $\boldsymbol{y}^k_{\mathbf{1},\Omega}$ with $k = 2$ and $\boldsymbol{w} = \mathbf{1}$. In the second row, we use $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ with $C = 2$ and $\boldsymbol{w} = (2, 0, 1, 1)^\top$, breaking the symmetry in θ_1 and θ_2. Using $\Omega \equiv 0$ yields piecewise-constant outputs. Gini and Tsallis regularization yield a sparse and differentiable operator, while Shannon regularization yields a dense and differentiable one.

Moreover, most existing relaxations are purely deterministic (yielding a soft, expected selection in the forward pass), whereas stochastic sampling over feasible subsets serves two essential purposes. First, conditional-computation pipelines [Li et al., 2026, Cha et al., 2026] strictly require hard $0/1$ selections, as a computation is either executed or skipped. Second, generative and exploratory paradigms, such as in discrete VAEs [Rolfe, 2016, Ahmed et al., 2023] or structured reinforcement learning [Hoppe et al., 2025], inherently rely on sampling discrete solutions from a tractable distribution.

In this work, we address these limitations by revisiting the structural similarities between Knapsack and Top-k problems, casting both as instances of dynamic programming (DP) and building upon Mensch and Blondel [2018] to propose a unified differentiable formulation. In the Knapsack setting, we introduce the first specialized differentiable layer for this problem: this unlocks the end-to-end training of models that require weighted, capacity-constrained discrete selections, while allowing them to benefit from the full diversity of our framework (supporting both stochastic and deterministic forward passes, as well as sparse and equivariant modes). In the Top-k setting, we provide the first unified framework for differentiable operators that natively jointly combines exact sparsity (or permutation equivariance), exact VJPs, and exact sampling.

Specifically, we make the following technical contributions:

- In Section 4.1, we derive differentiable relaxations of the Knapsack and Top-k problems by regularizing the $\max$ operators within the underlying Bellman DP recursions.
- In Section 4.2, we show that Shannon's entropy is the *unique* separable regularizer yielding *permutation-equivariant* operators, and characterize those inducing *sparsity* (Propositions 1 and 2).
- In Section 5.1, we derive efficient vector-Jacobian product computations, allowing the integration of the proposed operators into differentiable programming pipelines.
- In Section 5.2, we prove the existence of a distribution underlying the proposed operators (Proposition 3), provide an ancestral sampling algorithm for stochastic forward passes, and propose unbiased first-order surrogate gradients to bypass the non-differentiability of sampling.
- We provide a principled supervised learning approach using our operators as output layers in Section 5.3, by computing gradients of associated Fenchel-Young losses [Blondel et al., 2020a].
- In Section 6, we evaluate our DP-based losses on a knapsack problem, benchmark our proposed differentiable Top-k operators on a Fenchel-Young extension of discrete VAEs, and evaluate our Knapsack operators on a constrained dynamic assortment RL problem.

Code is available at `https://github.com/google-deepmind/diff-knapsack`.

2 Background and Related Work

Knapsack operators. The 0/1 Knapsack problem for n items with values given by $\boldsymbol{\theta} \in \mathbb{R}^n$, weights $\boldsymbol{w} \in \mathbb{N}^n$, and capacity $C \in \mathbb{N}$ such that $w_i \leq C \; \forall i \in [n]$, corresponds to finding a subset of items with maximal value and a total weight restricted by C [Martello and Toth, 1990, Kellerer et al., 2004]:

$$\max_{\boldsymbol{y} \in \{0,1\}^n} \sum_{i=1}^{n} \theta_i y_i \quad \text{s.t.} \sum_{i=1}^{n} w_i y_i \leq C \,. \tag{1}$$

We denote by $\mathcal{Y}_{\boldsymbol{w}}^C \triangleq \{\boldsymbol{y} \in \{0,1\}^n \mid \langle \boldsymbol{w}, \boldsymbol{y} \rangle \leq C\}$ the set of feasible item selections. The optimal value of Problem (1) is denoted by $\mathsf{max}_{\boldsymbol{w}}^C(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{y} \in \mathcal{Y}_{\boldsymbol{w}}^C} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$, and $\boldsymbol{y}_{\boldsymbol{w}}^C(\boldsymbol{\theta}) \triangleq \mathsf{argmax}_{\boldsymbol{y} \in \mathcal{Y}_{\boldsymbol{w}}^C} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$ is the corresponding maximizer, or *Knapsack operator*. Danskin's theorem [Danskin, 1966] gives $\nabla \mathsf{max}_{\boldsymbol{w}}^C(\boldsymbol{\theta}) = \boldsymbol{y}_{\boldsymbol{w}}^C(\boldsymbol{\theta})$ when the maximizer is unique. However, since $\boldsymbol{y}_{\boldsymbol{w}}^C$ is piecewise constant, its Jacobian $\nabla_{\boldsymbol{\theta}} \boldsymbol{y}_{\boldsymbol{w}}^C(\boldsymbol{\theta})$ is either zero or undefined, preventing the backpropagation of gradients.

Top-k operators. The variational form of the Top-k selection problem is a notable slight variation of Problem (1), where item weights are given by $\boldsymbol{w} = \mathbf{1}$, and the capacity is $C = k$ for some k such that $1 \leq k < n$. Moreover, the inequality constraint becomes an equality, so that the problem writes:

$$\max_{\boldsymbol{y} \in \{0,1\}^n} \sum_{i=1}^{n} \theta_i y_i \quad \text{s.t.} \sum_{i=1}^{n} y_i = k \,. \tag{2}$$

With a slight abuse of notation (overloading Knapsack notations for $\boldsymbol{w} = \mathbf{1}$ and $C = k$), we respectively denote by $\mathcal{Y}_{\mathbf{1}}^k$, $\mathsf{max}_{\mathbf{1}}^k$, and $\boldsymbol{y}_{\mathbf{1}}^k$ the feasible set, value and operator in this case. Note that $\boldsymbol{y}_{\mathbf{1}}^k(\boldsymbol{\theta}) = (\mathbb{1}_{\{i \in \text{Top-}k(\boldsymbol{\theta})\}})_{i=1}^n$ is also known as the *Top-k mask* of $\boldsymbol{\theta}$.

Smoothed maximum operators. Let $\Omega : \triangle^d \to \mathbb{R}$ be a strictly convex regularization function. The *smoothed maximum operator* $\mathsf{max}_\Omega : \mathbb{R}^d \to \mathbb{R}$ is defined as $\mathsf{max}_\Omega(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{q} \in \triangle^d} \{\langle \boldsymbol{\theta}, \boldsymbol{q} \rangle - \Omega(\boldsymbol{q})\}$. Typical choices for Ω include Shannon's negative entropy, which yields the *log-sum-exp* operator and *softmax* gradients, and quadratic (or Gini's negative entropy) regularization, which leads to the *sparsemax* operator [Martins and Astudillo, 2016]. This choice controls the smoothness and sparsity of the resulting operator: we discuss its consequences in our setting in Section 4.2.

Differentiating through combinatorial solvers. Integrating discrete combinatorial solvers into differentiable pipelines faces the challenge of uninformative gradients, as their output is piecewise constant. A line of research differentiates the Karush-Kuhn-Tucker conditions of linear or mixed integer programs [Mandi and Guns, 2020, Ferber et al., 2020]. One can also treat the solver as a *black-box* oracle. Pogancic et al. [2020] propose a method to compute gradients via piecewise affine interpolation of the solver's output. Alternatively, perturbation-based methods [Berthet et al., 2020, Niepert et al., 2021] smooth the operator by adding noise to inputs and estimate gradients via Monte-Carlo. Cordonnier et al. [2021] apply these techniques specifically to Top-k selection. While flexible, these approaches can be computationally expensive due to the need for multiple solver calls.

Differentiable Top-k and sorting. Existing approaches typically relax Problem (2) using regularized optimal transport [Cuturi et al., 2019, Xie et al., 2020], regularized linear programming [Amos et al., 2019, Blondel et al., 2020b, Qian et al., 2022, Sander et al., 2023], or smoothed sorting networks [Petersen et al., 2021, 2022]. These methods, however, are deterministic and cannot be used as a stochastic layer. Closest to our work, Ahmed et al. [2023] derive a differentiable Top-k operator based on factorizing the entropy of distributions on k-subsets. By adopting a broader DP-based perspective, we strictly generalize that framework: we recover similar algorithms with Shannon entropy-based regularization, and obtain new sparse operators with Gini or Tsallis regularization. Crucially, unlike all aforementioned approaches, our framework naturally handles Knapsack constraints and non-uniform weights. Furthermore, it supports both deterministic and stochastic layers.

DP for structured prediction. In NLP, DP with the standard $(\max, +)$ semiring has been used for k-subset selection [McDonald, 2006, Niculae and Martins, 2020]. Dynamic programs can be smoothed out using second-order expectation semirings [Li and Eisner, 2009] or by replacing $\max$ operators in the Bellman recursion with smoothed max_Ω operators [Mensch and Blondel, 2018].

3 Knapsack and Top-k as Dynamic Programs

Knapsack case. Problem (1) famously exhibits optimal substructure, allowing for a DP solution [Martello and Toth, 1990, Kellerer et al., 2004]. Indeed, let $V^{\boldsymbol{w}}[i, c]$ be the optimal value of the subproblem considering only the first $i \leq n$ items with a capacity $c \leq C$:

$$V^{\boldsymbol{w}}[i, c] \triangleq \max_{\boldsymbol{y} \in \{0,1\}^i} \sum_{l=1}^{i} \theta_l y_l \quad \text{s.t.} \sum_{l=1}^{i} w_l y_l \leq c\,.$$

To compute this, we can compare two possible decisions for item i. We can skip it and set $y_i = 0$, yielding value $V^{\boldsymbol{w}}[i-1, c]$, or we can pick it (if $w_i \leq c$) and set $y_i = 1$, yielding $\theta_i + V^{\boldsymbol{w}}[i-1, c-w_i]$:

$$V^{\boldsymbol{w}}[i, c] = V^{\boldsymbol{w}}[i-1, c] \quad \text{if } w_i > c, \quad \max\big(V^{\boldsymbol{w}}[i-1, c], \theta_i + V^{\boldsymbol{w}}[i-1, c-w_i]\big) \quad \text{else.} \tag{3}$$

The optimal value of Problem (1) is $V^{\boldsymbol{w}}[n, C] = \mathsf{max}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$. This recursion runs in $\mathcal{O}(nC)$ time and space, and is initialized with $V^{\boldsymbol{w}}[0, :\,] = 0$. An illustration is given in Figure 5.

Top-k case. For (2), let $V^{\mathbf{1}}[i, j]$ be the optimal value when selecting *exactly* j items from the first i:

$$V^{\mathbf{1}}[i, j] \triangleq \max_{\boldsymbol{y} \in \{0,1\}^i} \sum_{l=1}^{i} \theta_l y_l \quad \text{s.t.} \sum_{l=1}^{i} y_l = j\,.$$

The equality constraint implies that any subproblem (i, j) with $j > i$ is infeasible. Consequently, if item i is required to reach the target count (i.e., $i = j$), we are forced to pick it, as skipping would lead to an infeasible state. We enforce this behavior by initializing $V^{\mathbf{1}}[0, 0] = 0$ and $V^{\mathbf{1}}[0, 1:\,] = -\infty$, which naturally propagates to infeasible states. With $V^{\mathbf{1}}[:\,, 0] = 0$, for $j \geq 1$ the recursion writes:

$$V^{\mathbf{1}}[i, j] = \max\left(\theta_i + V^{\mathbf{1}}[i-1, j-1],\ V^{\mathbf{1}}[i-1, j]\right).$$

The optimal value of Problem (2) is therefore given by $V^{\mathbf{1}}[n, k] = \mathsf{max}^k_{\mathbf{1}}(\boldsymbol{\theta})$. This recursion runs in $\mathcal{O}(nk)$ time and space, and we illustrate it on an example in Figure 6.

Unified framework. Since in Top-k we have that $w_i = 1 \leq c$ for $c \geq 1$ (and $w_i > 0$ keeps $V^{\mathbf{1}}[:\,, 0] = 0$), Equation (3) serves as a unified recursion for Problems (1) and (2). The Top-k case is then simply a specific instance of the Knapsack with $\boldsymbol{w} = \mathbf{1}$ and $C = k$, the only distinction being that the row $[0, 1:\,]$ of the DP table is initialized with $-\infty$ rather than 0 to enforce the equality constraint.

Backtracking. These recursions yield the optimal values of Problems (1) and (2). To recover the maximizing mask $\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$ from $V^{\boldsymbol{w}}[n, C]$, one can *backtrack* from n down to 1, setting y_i to 0 or 1 based on which transition was active. Since $\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$ is a subgradient of $\mathsf{max}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$, this backtracking procedure is a special case of backpropagation [Blondel and Roulet, 2025] (see B.1 for a derivation).

4 Differentiable Knapsack and Top-k Operators

4.1 Algorithmic smoothing

We now algorithmically smooth the dynamic programs defined in Section 3, by regularizing the $\max$ operator in (3) into max_Ω using a strictly convex function $\Omega : \triangle^2 \to \mathbb{R}$:

$$V^{\boldsymbol{w}}_\Omega[i, c] \triangleq V^{\boldsymbol{w}}_\Omega[i-1, c] \quad \text{if } w_i > c, \quad \mathsf{max}_\Omega\big(\theta_i + V^{\boldsymbol{w}}_\Omega[i-1, c-w_i], V^{\boldsymbol{w}}_\Omega[i-1, c]\big) \quad \text{else.} \tag{4}$$

In Section D.3, we relate this approach to the framework of Mensch and Blondel [2018]. We now define our differentiable Knapsack and Top-k operators using the smoothed recursion in Equation (4).

Definition 1 (Smoothed DP values and operators.)**.** *Let $\boldsymbol{\theta} \in \mathbb{R}^n$. We define the following relaxations of the Knapsack and Top-k problem values and the corresponding operators:*

$$\mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \triangleq V^{\boldsymbol{w}}_\Omega[n, C], \qquad \mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}) \triangleq V^{\mathbf{1}}_\Omega[n, k],$$
$$\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \triangleq \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \in \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}}), \qquad \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}) \triangleq \nabla \mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}) \in \mathsf{conv}(\mathcal{Y}^k_{\mathbf{1}}).$$

Importantly, the smoothed values are convex in $\boldsymbol{\theta}$, allowing us to define convex supervised learning losses in Section 5.3. We prove this and other properties of the relaxed operators in Section D.4.

Algorithm 1 Computation of $\max^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$

input Item values $\boldsymbol{\theta} \in \mathbb{R}^n$, item weights $\boldsymbol{w} \in \mathbb{N}^n$, capacity $C \in \mathbb{N}$, regularization function Ω.
output Smoothed value $\max^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \in \mathbb{R}$.

Initialize $\boldsymbol{V}^{\boldsymbol{w}}_\Omega \in \mathbb{R}^{(n+1)\times(C+1)}$ by setting $V^{\boldsymbol{w}}_\Omega[0,0] \leftarrow 0$, and either $V^{\boldsymbol{w}}_\Omega[0,1:] \leftarrow 0$ (Knapsack) or $V^{\boldsymbol{w}}_\Omega[0,1:] \leftarrow -\infty$ (Top-k).
Initialize $\boldsymbol{Q}^{\boldsymbol{w}}_\Omega \leftarrow \mathbf{0} \in \mathbb{R}^{(n+1)\times(C+1)}$.
for $i = 1$ **to** n **do**
  for $c = 0$ **to** C {parallel loop} **do**

$$V^{\boldsymbol{w}}_\Omega[i,c] \leftarrow \begin{cases} V^{\boldsymbol{w}}_\Omega[i-1,c] & \text{if } w_i > c, \\ \max_\Omega\left(\theta_i + V^{\boldsymbol{w}}_\Omega[i-1,c-w_i]\right), \\ \quad V^{\boldsymbol{w}}_\Omega[i-1,c]\big) & \text{else.} \end{cases}$$

$$Q^{\boldsymbol{w}}_\Omega[i,c] \leftarrow \frac{\partial V^{\boldsymbol{w}}_\Omega[i,c]}{\partial \theta_i}$$

  end for
end for
return $\max^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = V^{\boldsymbol{w}}_\Omega[n,C]$ and $(\boldsymbol{V}^{\boldsymbol{w}}_\Omega, \boldsymbol{Q}^{\boldsymbol{w}}_\Omega)$ (intermediate results for Algorithms 2 to 4).

Algorithm 2 Computation of $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$

input Item values $\boldsymbol{\theta} \in \mathbb{R}^n$, item weights $\boldsymbol{w} \in \mathbb{N}^n$, capacity $C \in \mathbb{N}$, regularization function Ω, outputs $\boldsymbol{V}^{\boldsymbol{w}}_\Omega, \boldsymbol{Q}^{\boldsymbol{w}}_\Omega$ of Algorithm 1.
output Differentiable operator $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$.

Initialize $\boldsymbol{E}^{\boldsymbol{w}}_\Omega \leftarrow \mathbf{0} \in \mathbb{R}^{(n+1)\times(C+1)}$ by setting $E^{\boldsymbol{w}}_\Omega[n,C] \leftarrow 1$.
for $i = n-1$ **to** 1 **do**
  for $c = 0$ **to** C {parallel loop} **do**

$$\begin{aligned} E^{\boldsymbol{w}}_\Omega[i,c] \leftarrow & E^{\boldsymbol{w}}_\Omega[i+1,c] \\ & \times (1 - Q^{\boldsymbol{w}}_\Omega[i+1,c]) \\ & + E^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}] \\ & \times Q^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}] \end{aligned}$$

  end for
end for
$\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \leftarrow (\boldsymbol{E}^{\boldsymbol{w}}_\Omega[1:,:] \circ \boldsymbol{Q}^{\boldsymbol{w}}_\Omega[1:,:]) \cdot \mathbf{1}$
return $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ and $(\boldsymbol{V}^{\boldsymbol{w}}_\Omega, \boldsymbol{Q}^{\boldsymbol{w}}_\Omega, \boldsymbol{E}^{\boldsymbol{w}}_\Omega)$ (intermediate results for Algorithm 4).

Parallel implementation. Observing that values $V^{\boldsymbol{w}}_\Omega[i,\cdot]$ depend only on row $i-1$, we exploit *wavefront parallelism* [Muraoka, 1971] by vectorizing over capacities. This reduces the effective time complexity from $\mathcal{O}(nC)$ to $\mathcal{O}(n)$ parallel steps while incurring an $\mathcal{O}(nC)$ memory footprint (see Limitations and scalability for large-C mitigations). We provide pseudo-code in Algorithms 1 and 2, with detailed derivations given in Section B.2. To achieve high-performance execution without custom CUDA kernels, we provide a just-in-time compiled implementation in `Numba` [Lam et al., 2015], efficiently parallelized over batch and capacity dimensions.

4.2 Regularization choice

We now show how regularization impacts the *equivariance* and *sparsity* of our operators. We discuss three choices for Ω, based on the Shannon, Gini, and 1.5-Tsallis entropies [Blondel et al., 2019]:

$$H^s(\boldsymbol{q}) \triangleq -\left(q_1 \log q_1 + q_2 \log q_2\right), \quad H^g(\boldsymbol{q}) \triangleq \frac{1}{2}\left(1 - q_1^2 - q_2^2\right), \quad H^T_{1.5}(\boldsymbol{q}) \triangleq \frac{4}{3}\left(1 - q_1^{\frac{3}{2}} - q_2^{\frac{3}{2}}\right).$$

We use a regularization strength parameter $\gamma > 0$, and consider $\Omega \in \{-\gamma H^s, -\gamma H^g, -\gamma H^T_{1.5}\}$. Shannon's negative entropy $\Omega = -\gamma H^s$ yields closed forms for $\max^C_{\boldsymbol{w},\Omega}$ and $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ (see Section A.1). Within the α-Tsallis family [Tsallis, 1988], the 1.5-Tsallis entropy can be seen as a trade-off between the Shannon and Gini entropies (recovered as $\alpha \to 1$ and $\alpha = 2$ respectively): it allows us to combine both smoothness and exact sparsity (see Figure 1), while retaining efficient, closed-form expressions for the smoothed maximum and its derivatives (see Section F and Table 3).

In Proposition 2 and Section D.4, we show that the bias induced by regularization (i.e., the distance between hard operators and their smoothed counterparts) decays exponentially with γ under Shannon regularization, and reaches exactly zero under a finite regularization threshold under Gini and Tsallis.

Equivariance. We now show that $\Omega = -\gamma H^s$ is the *only* separable regularization choice ensuring the permutation-equivariance of the smoothed Knapsack and Top-k operators.

Proposition 1 (Characterization of equivariance)**.** *Let $\Omega : \triangle^2 \to \mathbb{R}$ be a convex,* separable *function such that $\Omega(\boldsymbol{e}_1) = \Omega(\boldsymbol{e}_2) = 0$, and let S_n be the group of permutations. We have:*

$$\sigma \circ \boldsymbol{y}^C_{\boldsymbol{w},\Omega} = \boldsymbol{y}^C_{\sigma(\boldsymbol{w}),\Omega} \circ \sigma \quad \forall (\sigma, \boldsymbol{w}) \in S_n \times \mathbb{N}^n \quad \textbf{\textit{if and only if}} \quad \Omega = -\gamma H^s \quad \textit{for some } \gamma \geq 0.$$

Note that in the Top-k case we have $\sigma(\boldsymbol{w}) = \boldsymbol{w} = \mathbf{1}$. The proof of Proposition 1, given in Section A.2, relies on two ingredients: first, the equivalence between the permutation-invariance of a function and the permutation-equivariance of its gradient, and second, the fact that only $\Omega = -\gamma H^s$ yields an *associative* smoothed maximum operator max_Ω.

Sparsity. We now characterize the regularizers that enable the relaxed operators $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ to map to vertices $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$ of the moment polytope $\mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$, i.e., to produce sparse outputs.

Proposition 2 (Characterization of sparsity)**.** *Let $\Omega(\boldsymbol{q}) \triangleq \sum_{i=1}^2 \omega(q_i)$ be a separable, strictly convex regularizer with ω differentiable on $(0,1)$. The following statements are equivalent:*

1. $\lim_{t\to 0^+} \omega'(t) > -\infty$ *and* $\lim_{t\to 1^-} \omega'(t) < +\infty$.
2. $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ *can output hard discrete selections, i.e.,* $\mathcal{Y}^C_{\boldsymbol{w}} \subseteq \mathsf{range}(\boldsymbol{y}^C_{\boldsymbol{w},\Omega})$.
3. Exact recovery: *for any $\boldsymbol{\theta} \in \mathbb{R}^n$ yielding a unique hard maximizer $\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$, there exists a regularization threshold $\gamma^* > 0$ such that $\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$ for all $\gamma \in (0, \gamma^*]$.*

The proof is given in Section A.3. Both Gini ($\omega'(q) = \gamma q$) and 1.5-Tsallis ($\omega'(q) = 2\gamma\sqrt{q}$) satisfy condition (1), whereas even above γ^*, bounded endpoint derivatives allow individual DP transitions to saturate to $\{0,1\}$, placing $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ on boundary faces of $\mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$ (inducing both hard $0/1$ coordinates and soft ones). Conversely, in the Shannon-regularized setting $\Omega = -\gamma H^s$, and more generally for any Legendre-type regularizer [Rockafellar, 1970], whose gradients explode at the boundary of $\triangle^2$, the outputs of $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ are forced to lie in the relative interior $\mathsf{ri}(\mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})) \subseteq (0,1)^n$, and can therefore never be sparse.

We illustrate our operators in Figure 1, and numerically demonstrate Propositions 1 and 2 in Figure 7.

5 Layer Integration

In Sections 5.1 to 5.3, we describe three principled scenarios for integration of the proposed operators into a differentiable programming framework: as a deterministic hidden layer, as a stochastic hidden layer, and as an output layer.

We denote by $\boldsymbol{\theta} \in \mathbb{R}^n$ the layer's input, which is typically obtained as $\boldsymbol{\theta} = f_{W_1}(\boldsymbol{x})$ where f_{W_1} is an upstream model with learnable parameters W_1. Similarly, we denote the downstream computation graph by f_{W_2}.

5.1 Deterministic forward passes and VJPs

In the hidden layer case, the most direct approach is to use the relaxed operator $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ as a deterministic, continuous layer $\boldsymbol{\theta} \mapsto \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \in \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$ during training, directly feeding its output to the downstream function f_{W_2}.

To learn the upstream parameters W_1 via backpropagation, we must compute the vector-Jacobian product (VJP) of the layer. Since $\boldsymbol{y}^C_{\boldsymbol{w},\Omega} = \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}$, the Jacobian $\nabla \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ is the Hessian $\nabla^2 \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$, which is symmetric. Thus, for any cotangent vector $\boldsymbol{z} \in \mathbb{R}^n$, the VJP $\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))$ is the transpose of the JVP $(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))\boldsymbol{z}$.

We compute it using a *reverse-over-forward* approach, by backpropagating through the computation of the directional derivative $\langle \nabla_{\boldsymbol{\theta}} \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}), \boldsymbol{z} \rangle$, which we derive in Section B.3. The special structure of our DAG, which has repeated edges and constant ones, leads to important changes compared to the general case of Mensch and Blondel [2018] (see Section D.3). The pseudo-code is given in Algorithm 4, with derivations in Section B.4.

In Section F, we derive explicit formulas for the quantities needed to instantiate Algorithm 4 with Shannon, Gini, and 1.5-Tsallis regularization, which we summarize in Table 3.

5.2 Stochastic forward passes and surrogate gradients

The method described in Section 5.1 is susceptible to a train-test mismatch. Although the network is trained on continuous vectors $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \in \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$, inference often requires hard item selections $\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) \in \mathcal{Y}^C_{\boldsymbol{w}}$. This is particularly true for semantically discrete tasks where passing a soft, averaged vector $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ to f_{W_2} would be nonsensical. We propose to bridge this gap via sampling.

Proposition 3 (Underlying distribution)**.** *Let $\boldsymbol{Q}^{\boldsymbol{w}}_{\Omega}$ be the output of Algorithm 1. Let $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ be the distribution on $\{0,1\}^n$ defined autoregressively by $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(Y_i = 1|Y_{i+1},\dots,Y_n) \triangleq Q^{\boldsymbol{w}}_{\Omega}[i,\, C_i]$ for all $i \in \{n,\dots,1\}$, where $C_i \triangleq C - \sum_{j=i+1}^n Y_j w_j$ is the remaining capacity for items $[i]$.*

The support of $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ is a subset of $\mathcal{Y}^C_{\boldsymbol{w}}$, and its expectation matches the relaxed operator: $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \mathbb{E}_{\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}}[Y] \in \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$. Moreover, in the Shannon-regularized case $\Omega = -\gamma H^s$, we recover the Gibbs distribution $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \propto \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}\rangle/\gamma)$.

Proposition 3, proved in Section A.4, shows the existence of a distribution $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ underlying the proposed operators, and yields a tractable $\mathcal{O}(n)$ ancestral sampling algorithm to sample from it. We give the corresponding pseudo-code in Algorithm 3.

Algorithm 3 Sampling from $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$

input Values $\boldsymbol{\theta} \in \mathbb{R}^n$, weights $\boldsymbol{w} \in \mathbb{N}^n$, capacity $C \in \mathbb{N}$, regularizer Ω, output $\boldsymbol{Q}^{\boldsymbol{w}}_{\Omega}$ of Algorithm 1.
output Sample $\boldsymbol{y} \sim \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$.
  Initialize $\boldsymbol{y} \in \mathbb{R}^n$, $c \leftarrow C$.
  for $i = n$ **to** 1 **do**
    $y_i \sim \text{Bernoulli}\,(Q^{\boldsymbol{w}}_{\Omega}[i, c])$
    $c \leftarrow c - y_i w_i$
  end for
  return $\boldsymbol{y}$

The structure of $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ enables to efficiently compute the probability of any selection $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$ as $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) = \prod_{i=1}^n \left(y_i Q^{\boldsymbol{w}}_{\Omega}[i, c_i] + (1 - y_i)(1 - Q^{\boldsymbol{w}}_{\Omega}[i, c_i])\right)$, where $c_i \triangleq C - \sum_{j=i+1}^n y_j w_j$.This is very useful as it enables assessing the probability of a given Top-k or Knapsack assignment.

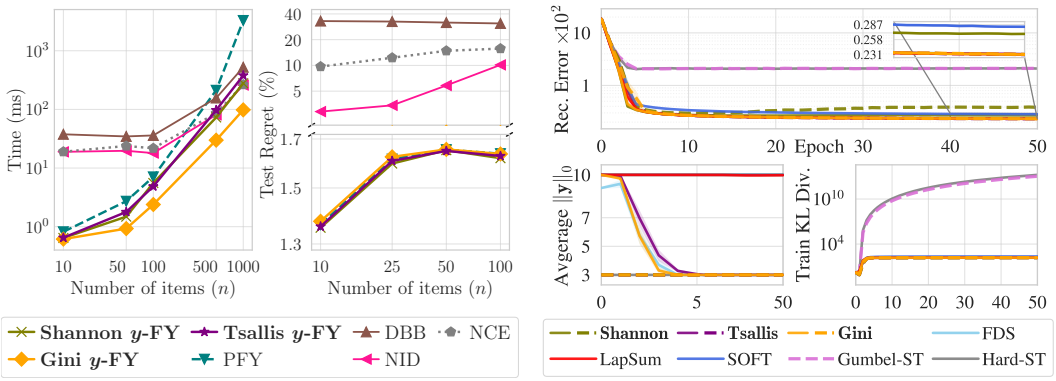
Surrogate gradients. To enforce hard item selections during training, we replace the relaxed output $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \in \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$ with a discrete sample $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$ from $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ in the forward pass. To bypass the non-differentiability of the sampling operation, we adopt the following surrogate gradient approach:

1. During the forward pass, we treat the layer as a stochastic node by sampling $\boldsymbol{y} \sim \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ using Algorithm 3. We can then compute any loss $\mathcal{L}(\boldsymbol{x}, f_{W_2}(\boldsymbol{y}))$.
2. In the backward pass, we treat the layer as the deterministic node $\boldsymbol{\theta} \mapsto \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \mathbb{E}_{\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}}[Y]$, with VJP computed by Algorithm 4. Thus, we use

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, f_{W_2}(\boldsymbol{y})) \triangleq \left(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})\right) \cdot \nabla_{\boldsymbol{y}} \mathcal{L}(\boldsymbol{x}, f_{W_2}(\boldsymbol{y})),$$

   and backpropagate the cotangent $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, f_{W_2}(\boldsymbol{y}))$ to upstream learnable parameters W_1.

A similar approach is also proposed in Ahmed et al. [2023] (although limited to the Shannon-regularized Top-k setting). In Section D.5, we show that this surrogate gradient is an *unbiased first-order estimator* of the expectation of the loss, following the analysis in Liu et al. [2023]. We also show its superiority to alternatives using hard item selections during forward passes in Figure 9.



(a) **Scaling and performance of $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ as an output layer.** Lower computational time and test relative regret are better.

(b) **DVAE training dynamics:** test MSE, sparsity of latent representations $\boldsymbol{y}$ and KL divergence during training (dashed lines indicate stochastic forward passes).

Figure 2: Evaluation of our DP-based operators as output layers on Knapsack prediction (a) and as stochastic and deterministic hidden layers in discrete VAEs (b).

5.3 Output layers and Fenchel-Young losses

We now give principled losses for supervised learning with the proposed operators as output layers.

Moment polytope regularization. For any $\Omega : \triangle^2 \to \mathbb{R}$ used to smooth the DP recursion (3) into Equation (4), define the regularization function $\Omega^C_{\boldsymbol{w}}$ as the Fenchel conjugate of $\mathsf{max}^C_{\boldsymbol{w},\Omega}$ with domain $\mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$. The corresponding *Fenchel-Young loss* [Blondel et al., 2020a] is given by:

$$L_{\Omega^C_{\boldsymbol{w}}} : \mathbb{R}^n \times \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}}) \longrightarrow \mathbb{R}, \quad \text{with} \quad L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}\,;\boldsymbol{y}) \triangleq \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) + \Omega^C_{\boldsymbol{w}}(\boldsymbol{y}) - \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle.$$

This loss function has several desirable properties [Blondel et al., 2020a, Proposition 2]:

- $\boldsymbol{\theta} \mapsto L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}\,;\boldsymbol{y})$ is convex for all $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$,
- $\boldsymbol{\theta} \mapsto L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}\,;\boldsymbol{y})$ is differentiable with gradient given by $\nabla_{\boldsymbol{\theta}} L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}\,;\boldsymbol{y}) = \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) - \boldsymbol{y}$,
- $L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}\,;\boldsymbol{y}) \geq 0$ for all $(\boldsymbol{\theta}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}})$,
- $L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}\,;\boldsymbol{y}) = 0$ if and only if $\boldsymbol{\theta}$ is such that $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}$.

Thus, computing gradients reduces to computing a forward pass of the relaxed layer $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ using Algorithm 2, and backpropagating the difference between its output and the ground-truth target $\boldsymbol{y}$.

6 Experiments

6.1 Learning to predict Knapsack solutions

We compare the losses proposed in Section 5.3 against four established baselines: `PFY` [Berthet et al., 2020], `DBB` [Pogancic et al., 2020], `NCE` [Mulamba et al., 2021], and `NID` [Sahoo et al., 2022] (all are detailed in Section E.1) on the `PyEPO` benchmark [Tang and Khalil, 2024]. Given a dataset $(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)})_{i=1}^N$, the goal is to predict item values $\hat{\boldsymbol{\theta}}^{(i)}$ from features $\boldsymbol{x}^{(i)}$ such that the subsequent Knapsack solution $\boldsymbol{y}^C_{\boldsymbol{w}}(\hat{\boldsymbol{\theta}}^{(i)})$ matches the ground-truth selection $\boldsymbol{y}^{(i)}$.

We measure performance using relative regret on a test set, defined as $\frac{\langle \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)} \rangle - \langle \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^C_{\boldsymbol{w}}(\hat{\boldsymbol{\theta}}^{(i)}) \rangle}{|\langle \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)} \rangle|}$.

Results. We parameterize $\hat{\boldsymbol{\theta}} = f_{W_1}(\boldsymbol{x})$ as a standard feed-forward neural network, and vary the number of items n. We measure the average computation time of each evaluated loss function (forward and backward pass up to $\boldsymbol{\theta}$), and the best test relative regret. The results are gathered in Figure 2a. Our DP-based Fenchel-Young losses consistently outperform the baselines in terms of regret while remaining computationally efficient. Full experimental details are given in Section E.1.

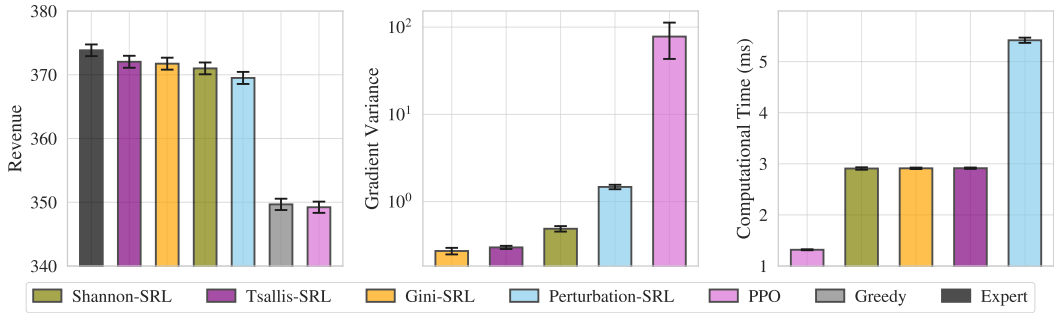


Figure 3: **Constrained dynamic assortment. Left:** Expected revenue on 10^4 test episodes. Our DP framework outperforms realizable baselines and approaches the full-information expert oracle. **Middle:** Expected gradient variance. Sparse regularizers (Gini, Tsallis) significantly reduce variance by inducing a sparse support for $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$. **Right:** Gradient estimation wall-clock time. By avoiding repeated solver calls and reusing intermediate outputs from Algorithm 1 for sampling and differentiation via Algorithms 2 and 3, our approach achieves notable speedups over perturbation-based SRL.

6.2 Fenchel-Young discrete VAE

We now evaluate our relaxed Top-k operators within a discrete variational auto-encoder (DVAE, Rolfe [2016]) framework. The goal is to learn representations that disentangle continuous style attributes from discrete categorical identity. The encoder E_{W_1} maps an input $\boldsymbol{x}$ to latent style parameters $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{R}^{d_z}$ and selection logits $\boldsymbol{\theta} \in \mathbb{R}^n$. The discrete latent selection is obtained via either $\boldsymbol{y} = \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta})$ or $\boldsymbol{y} \sim \pi^{\mathbf{1},k}_{\boldsymbol{\theta},\Omega}$ (following Sections 5.1 and 5.2). The reconstruction $\hat{\boldsymbol{x}}$ is then generated by a shared decoder D_{W_2}, which processes learnable class embeddings $\boldsymbol{e}_i$ conditioned on style variables $\boldsymbol{z}_i$, aggregated by the selection weights $\boldsymbol{y}$, i.e., $\hat{\boldsymbol{x}} \triangleq \frac{1}{k}\sum_{i=1}^n y_i \cdot D_{W_2}([\boldsymbol{e}_i, \boldsymbol{z}_i])$.

To train this model, we minimize the sum of a MSE reconstruction loss on $\hat{\boldsymbol{x}}$, a KL regularization term for the latent style variables $\boldsymbol{z}$, and a Fenchel-Young regularization term $L_{\Omega^k_{\mathbf{1}}}(\mathbf{0}\,; \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}))$ for the latent selection logits $\boldsymbol{\theta}$ using the loss proposed in Section 5.3. The gradient of this regularization term is given by $\nabla_{\boldsymbol{\theta}} L_{\Omega^k_{\mathbf{1}}}(\mathbf{0}\,; \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta})) = (\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta})) \cdot \boldsymbol{\theta}$, i.e., it is the JVP of $\boldsymbol{y}^k_{\mathbf{1},\Omega}$ in direction $\boldsymbol{\theta}$, efficiently computed via Algorithm 4 (see Section E.2 for the derivation).

The term $L_{\Omega^k_{\mathbf{1}}}$ equals the standard KL divergence against a uniform prior in the $\Omega = -H^s$ setting, recovering the framework of SIMPLE [Ahmed et al., 2023], while enabling more general regularizers that induce sparse latent representations, such as Gini and Tsallis entropies. This formulation also extends the Fenchel-Young variational inference framework of Sklaviadis et al. [2025], from continuous VAEs to DVAEs with latent distributions on k-subsets. Table 2 presents an ablation of this regularization term, illustrating the classic rate-distortion trade-off typical of VAEs.

All derivations and experimental details are given in Section E.2.

Experimental setup. We use a stacked version of the MNIST dataset [LeCun and Cortes, 2005], where inputs are pixel-wise averages of $k = 3$ distinct digit images. The model must recover the identities of the constituent digits via the latent k-subset $\boldsymbol{y}$. We compare our deterministic and stochastic Top-k operators against those of Sander et al. [2023] (FDS), Struski et al. [2025] (LapSum), Xie et al. [2020] (SOFT), and to hard and Gumbel-perturbed Top-k baselines [Vieira, 2024, Kool et al., 2019] using straight-through estimators (ST, Bengio et al. [2013]) for the backward pass.

Results. Figure 2b illustrates the training dynamics, highlighting three results. First, our DP-based operators match or outperform baselines in reconstruction quality; in particular, our variants equipped with sparse regularizers (Gini and Tsallis) surpass the Shannon-based SIMPLE framework [Ahmed et al., 2023] across both deterministic and stochastic forward passes. Second, unlike the SIMPLE, LapSum, or SOFT Top-k operators, Gini and Tsallis regularization induce sparse latent representations, quickly converging to hard k-subsets to explain the data. Third, our methods exhibit superior stability compared to ST-based baselines, which lead to a diverging KL regularization term.

6.3 Dynamic assortment with Knapsack constraints

We evaluate our differentiable Knapsack operators on a dynamic assortment problem [Talluri and Van Ryzin, 2004, Chen et al., 2020]. The agent manages a store of $n = 20$ items and selects subsets to display over 80 time steps, subject to capacity and inventory constraints. The environment features endogenous uncertainty driven by a hidden customer choice model, parameterized by unknown preferences $\boldsymbol{\phi}$. Rewards correspond to sold item prices, which deplete inventory and update features.

Method. We employ the structured reinforcement learning (SRL) framework of Hoppe et al. [2025], where the actor maps states to item logits $\boldsymbol{\theta}$. While the original method relies on the perturbation framework of Berthet et al. [2020] for exploration and gradient estimation, our DP-based formulation yields *exact* Fenchel-Young gradients $\nabla_{\boldsymbol{\theta}} L_{\Omega^C_{\boldsymbol{w}}}(\boldsymbol{\theta}; \hat{\boldsymbol{y}}) = \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) - \hat{\boldsymbol{y}}$. Furthermore, we substitute perturbed optimization with ancestral sampling using Algorithm 3 to generate targets $\hat{\boldsymbol{y}} \sim \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$.

Baselines. We compare our method against standard perturbation-based SRL and proximal policy optimization (PPO, Schulman et al. [2017]). We also include a *Greedy* policy, which maximizes the cumulative price of displayed objects, and an *Expert* oracle. This expert "cheats" by accessing the hidden customer model $\boldsymbol{\phi}$ to compute the exact expected immediate revenue of every feasible assortment $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$, and chooses the maximizing one. While this policy is not globally optimal (it ignores inventory and customer choice dynamics), it serves as a strong performance ceiling. Note that it is not computationally scalable, as it requires enumerating all feasible actions at every step, and $|\mathcal{Y}^C_{\boldsymbol{w}}|$ grows exponentially with n.

Results. Figure 3 shows that our proposed operators consistently outperform realizable baselines in expected revenue while reducing gradient variance and runtime over perturbation-based SRL. Although this pipeline blends stochastic sampling and FY gradients, their respective benefits remain clear. First, our sparse regularizers uniquely induce a sparse distribution support for $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ for the sampling phase. Second, our exact and deterministic computation of the FY gradient negates the need for Monte-Carlo sampling, further reducing variance. Third, our regularized DP has the same complexity as a single unregularized call needed by Monte-Carlo sampling for perturbation-based SRL (and reuses Algorithm 1 outputs across Algorithms 2 and 3), which in turn reduces the required computational time. Full experimental details are in Section E.3.

Conclusion and Limitations

We proposed a unified DP framework for differentiable Knapsack and Top-k operators, showing that the choice of regularization dictates key properties like permutation equivariance and sparsity. Our operators seamlessly enforce combinatorial constraints in differentiable pipelines as deterministic or stochastic hidden layers, or as output layers with Fenchel-Young losses for supervised learning.

Limitations and scalability. While the DP formulation has an $\mathcal{O}(nC)$ complexity, wavefront parallelization reduces execution to $\mathcal{O}(n)$ parallel steps, making time complexity competitive on modern hardware. The $\mathcal{O}(nC)$ memory footprint poses no issue for our targeted applications, but exceptionally large C could create bottlenecks. To mitigate this, one can trade compute for memory by storing only a fraction of the DP table and recomputing the rest in the backward. Alternatively, one can reduce the state space by dividing item weights and the capacity by a common factor (this is exact if using their greatest common divisor, or trades slight precision for memory if rounding).

Acknowledgments and Disclosures

This work was fully supported by Google DeepMind, where Germain Vivier-Ardisson is a full-time employee and PhD candidate. The authors have no competing interests or financial relationships outside the submitted work to disclose.

References

K. Ahmed, Z. Zeng, M. Niepert, and G. V. d. Broeck. SIMPLE: A Gradient Estimator for k-Subset Sampling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=GPJVuyX4p_h.

B. Amos, V. Koltun, and J. Z. Kolter. The Limited Multi-Label Projection Layer, Oct. 2019. URL http://arxiv.org/abs/1906.08707. arXiv:1906.08707 [cs.LG].

Y. Bengio, N. Léonard, and A. Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation, Aug. 2013. URL http://arxiv.org/abs/1308.3432. arXiv:1308.3432 [cs].

Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with Differentiable Pertubed Optimizers. In *Advances in Neural Information Processing Systems*, volume 33, pages 9508–9519. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6bb56208f672af0dd65451f869fedfd9-Abstract.html.

M. Blondel and V. Roulet. The Elements of Differentiable Programming, June 2025. URL http://arxiv.org/abs/2403.14606. arXiv:2403.14606 [cs].

M. Blondel, A. Martins, and V. Niculae. Learning Classifiers with Fenchel-Young Losses: Generalized Entropies, Margins, and Algorithms. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 606–615. PMLR, Apr. 2019. URL https://proceedings.mlr.press/v89/blondel19a.html.

M. Blondel, A. F. T. Martins, and V. Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020a. ISSN 1533-7928. URL http://jmlr.org/papers/v21/19-021.html.

M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga. Fast Differentiable Sorting and Ranking. In *Proceedings of the 37th International Conference on Machine Learning*, pages 950–959. PMLR, Nov. 2020b. URL https://proceedings.mlr.press/v119/blondel20a.html.

S. Cha, G. Kim, D. Han, T. Yang, and I. Han. KnapSpec: Self-Speculative Decoding via Adaptive Layer Selection as a Knapsack Problem. In *Proceedings of the 43rd International Conference on Machine Learning*, pages 12280–12293. PMLR, Sept. 2026. URL https://proceedings.mlr.press/v306/cha26a.html.

X. Chen, Y. Wang, and Y. Zhou. Dynamic Assortment Optimization with Changing Contextual Information. *Journal of Machine Learning Research*, 21(216):1–44, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/19-1054.html.

J.-B. Cordonnier, A. Mahendran, A. Dosovitskiy, D. Weissenborn, J. Uszkoreit, and T. Unterthiner. Differentiable Patch Selection for Image Recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2351–2360, June 2021. doi: 10.1109/CVPR46437.2021.00238. URL https://ieeexplore.ieee.org/abstract/document/9578322. ISSN: 2575-7075.

M. Cuturi, O. Teboul, and J.-P. Vert. Differentiable Ranking and Sorting using Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d8c24ca8f23c562a5600876ca2a550ce-Abstract.html.

J. M. Danskin. The Theory of Max-Min, with Applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, July 1966. ISSN 0036-1399. doi: 10.1137/0114053. URL https://epubs.siam.org/doi/abs/10.1137/0114053.

A. Ferber, B. Wilder, B. Dilkina, and M. Tambe. MIPaaL: Mixed Integer Program as a Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1504–1511, Apr. 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i02.5509. URL https://ojs.aaai.org/index.php/AAAI/article/view/5509.

H. Hoppe, L. Baty, L. Bouvier, A. Parmentier, and M. Schiffer. Structured Reinforcement Learning for Combinatorial Decision-Making. In *Advances in Neural Information Processing Systems*, volume 38, pages 173726–173761. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fe100e5e7bca984803f0e4c6e94a6a88-Abstract-Conference.html.

H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Heidelberg, 2004. ISBN 978-3-642-07311-3 978-3-540-24777-7. doi: 10.1007/978-3-540-24777-7. URL http://link.springer.com/10.1007/978-3-540-24777-7.

W. Kool, H. V. Hoof, and M. Welling. Stochastic Beams and Where To Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3499–3508. PMLR, May 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

S. K. Lam, A. Pitrou, and S. Seibert. Numba: a LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, pages 1–6, New York, NY, USA, Nov. 2015. Association for Computing Machinery. ISBN 978-1-4503-4005-2. doi: 10.1145/2833157.2833162. URL https://dl.acm.org/doi/10.1145/2833157.2833162.

Y. LeCun and C. Cortes. The mnist database of handwritten digits, 2005. URL https://www.semanticscholar.org/paper/The-mnist-database-of-handwritten-digits-LeCun-Cortes/dc52d1ede1b90bf9d296bc5b34c9310b7eaa99a2.

Z. Li and J. Eisner. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 40–51, USA, Aug. 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL https://dl.acm.org/doi/10.5555/1699510.1699517.

Z. Li, C. Chen, T. Yang, T. Ding, R. Sun, G. Zhang, W. Huang, and Z.-Q. Luo. Knapsack RL: Compute-Efficient Reinforcement Learning via Heterogeneous Rollout Allocation. In *Proceedings of the 43rd International Conference on Machine Learning*, pages 66340–66368. PMLR, Sept. 2026. URL https://proceedings.mlr.press/v306/li26v.html.

L. Liu, C. Dong, X. Liu, B. Yu, and J. Gao. Bridging Discrete and Backpropagation: Straight-Through and Beyond. In *Advances in Neural Information Processing Systems*, volume 36, pages 12291–12311. Curran Associates, Inc., 2023. doi: 10.52202/075280-0539. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/28b5dfc51e5ae12d84fb7c6172a00df4-Abstract-Conference.html.

J. Mandi and T. Guns. Interior Point Solving for LP-based prediction+optimisation. In *Advances in Neural Information Processing Systems*, volume 33, pages 7272–7282. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/51311013e51adebc3c34d2cc591fefee-Abstract.html.

S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., USA, Oct. 1990. ISBN 978-0-471-92420-3.

A. Martins and R. Astudillo. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1614–1623. PMLR, June 2016. URL https://proceedings.mlr.press/v48/martins16.html.

R. McDonald. Discriminative Sentence Compression with Soft Syntactic Evidence. In D. McCarthy and S. Wintner, editors, *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 297–304, Trento, Italy, Apr. 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1038/.

A. Mensch and M. Blondel. Differentiable Dynamic Programming for Structured Prediction and Attention. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3462–3471. PMLR, July 2018. URL https://proceedings.mlr.press/v80/mensch18a.html.

M. Mulamba, J. Mandi, M. Diligenti, M. Lombardi, V. Bucarey Lopez, and T. Guns. Contrastive Losses and Solution Caching for Predict-and-Optimize: 30th International Joint Conference on Artificial Intelligence (IJCAI-21). *Contrastive Losses and Solution Caching for Predict-and-Optimize*, pages 2833–2840, Aug. 2021. doi: 10.24963/ijcai.2021/390. URL https://www.scopus.com/pages/publications/85122116844.

Y. Muraoka. *Parallelism exposure and exploitation in programs*. phd, University of Illinois at Urbana-Champaign, USA, 1971. AAI7121189.

V. Niculae and A. Martins. LP-SparseMAP: Differentiable Relaxed Optimization for Sparse Structured Prediction. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7348–7359. PMLR, Nov. 2020. URL https://proceedings.mlr.press/v119/niculae20a.html.

M. Niepert, P. Minervini, and L. Franceschi. Implicit MLE: Backpropagating Through Discrete Exponential Family Distributions. In *Advances in Neural Information Processing Systems*, volume 34, pages 14567–14579. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/7a430339c10c642c4b2251756fd1b484-Abstract.html.

F. Petersen, C. Borgelt, H. Kuehne, and O. Deussen. Differentiable Sorting Networks for Scalable Sorting and Ranking Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8546–8555. PMLR, July 2021. URL https://proceedings.mlr.press/v139/petersen21a.html.

F. Petersen, H. Kuehne, C. Borgelt, and O. Deussen. Differentiable Top-k Classification Learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 17656–17668. PMLR, June 2022. URL https://proceedings.mlr.press/v162/petersen22a.html.

M. V. Pogancic, A. Paulus, V. Musil, G. Martius, and M. Rolinek. Differentiation of Blackbox Combinatorial Solvers. In *The Eighth International Conference on Learning Representations*, Apr. 2020. URL https://iclr.cc/virtual_2020/poster_BkevoJSYPB.html.

Y. Qian, J. Lee, S. M. K. Duddu, Z. Dai, S. Brahma, I. Naim, T. Lei, and V. Y. Zhao. Multi-Vector Retrieval as Sparse Alignment, Nov. 2022. URL http://arxiv.org/abs/2211.01267. arXiv:2211.01267 [cs].

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. ISBN 978-0-691-01586-6. URL https://www.jstor.org/stable/j.ctt14bs1ff.

J. Rolfe. Discrete Variational Autoencoders. *ArXiv*, Sept. 2016. URL https://www.semanticscholar.org/paper/Discrete-Variational-Autoencoders-Rolfe/711683de61ee8d2aa929ec4d5d97d616b8e281d3.

S. S. Sahoo, A. Paulus, M. Vlastelica, V. Musil, V. Kuleshov, and G. Martius. Backpropagation through Combinatorial Algorithms: Identity with Projection Works. In *The Eleventh International Conference on Learning Representations*, Sept. 2022. URL https://openreview.net/forum?id=JZMR727O29.

M. E. Sander, J. Puigcerver, J. Djolonga, G. Peyré, and M. Blondel. Fast, Differentiable and Sparse Top-k: a Convex Analysis Perspective. In *Proceedings of the 40th International Conference on Machine Learning*, pages 29919–29936. PMLR, July 2023. URL https://proceedings.mlr.press/v202/sander23a.html.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms, Aug. 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

S. Sklaviadis, A. Martins, and M. Figueiredo. Fenchel-Young Variational Learning, Aug. 2025. URL http://arxiv.org/abs/2502.10295. arXiv:2502.10295 [cs].

L. Struski, M. B. Bednarczyk, I. T. Podolak, and J. Tabor. LapSum - One Method to Differentiate Them All: Ranking, Sorting and Top-k Selection. In *The Forty-second International Conference on Machine Learning*, June 2025. URL https://openreview.net/forum?id=CXN1Myzsp4&noteId=jsvkH34g07.

K. T. Talluri and G. J. Van Ryzin. *The Theory and Practice of Revenue Management*, volume 68 of *International Series in Operations Research & Management Science*. Springer US, Boston, MA, 2004. ISBN 978-0-387-24376-4 978-0-387-27391-4. doi: 10.1007/b139000. URL http://link.springer.com/10.1007/b139000.

B. Tang and E. B. Khalil. PyEPO: a PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, 16(3):297–335, Sept. 2024. ISSN 1867-2957. doi: 10.1007/s12532-024-00255-x. URL https://doi.org/10.1007/s12532-024-00255-x.

C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52 (1):479–487, July 1988. ISSN 1572-9613. doi: 10.1007/BF01016429. URL https://doi.org/10.1007/BF01016429.

T. Vieira. Gumbel-Max Trick and Weighted Reservoir Sampling — Graduate Descent (blog post)., Aug. 2024. URL https://timvieira.github.io/blog/gumbel-max-trick-and-weighted-reservoir-sampling/.

Y. Xie, H. Dai, M. Chen, B. Dai, T. Zhao, H. Zha, W. Wei, and T. Pfister. Differentiable Top-k with Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 33, pages 20520–20531. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/ec24a54d62ce57ba93a531b460fa8d18-Abstract.html.

Notation. We denote $[n] \triangleq \{1, \dots, n\}$. Vectors and matrices use boldface letters (e.g., $\boldsymbol{v}, \mathbf{1}, \mathbf{0}, \boldsymbol{M}$). The probability simplex in $\mathbb{R}^d$ is $\triangle^d \triangleq \{\boldsymbol{q} \in \mathbb{R}^d_{\geq 0} \mid \sum_i q_i = 1\}$. We use $\langle \cdot, \cdot \rangle$ and $\circ$ for inner and element-wise products. The convex hull and relative interior of a set $\mathcal{S}$ are $\mathsf{conv}(\mathcal{S})$ and $\mathsf{ri}(\mathcal{S})$. For a function f, $\mathsf{range}(f)$ and ∇f are its range and gradient.

A Proofs

A.1 Closed forms in the Shannon-entropy regularized case

For $\Omega = -\gamma H^s$, we have:

$$\begin{aligned}
\mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) &= \gamma \cdot \log \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right) , \\
\mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}) &= \gamma \cdot \log \sum_{\boldsymbol{y} \in \mathcal{Y}^k_{\mathbf{1}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right) , \\
\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) &= \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma\right)} \cdot \boldsymbol{y} , \\
\boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}) &= \sum_{\boldsymbol{y} \in \mathcal{Y}^k_{\mathbf{1}}} \frac{\exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^k_{\mathbf{1}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma\right)} \cdot \boldsymbol{y} .
\end{aligned}$$

Proof. For any integer $d \geq 2$, define $\Omega_d : \triangle^d \to \mathbb{R}$ as $\Omega_d(\boldsymbol{q}) \triangleq \gamma \sum_{i=1}^d q_i \log q_i$. Mensch and Blondel [2018, Proposition 2] gives that this definition of the family $(\Omega_d)_{d \geq 2}$ is such that:

$$\mathsf{max}^C_{\boldsymbol{w},\Omega_2}(\boldsymbol{\theta}) = \mathsf{max}_{\Omega_{|\mathcal{Y}^C_{\boldsymbol{w}}|}}(\boldsymbol{s}_{\boldsymbol{\theta}}),$$

where we define the score vector $\boldsymbol{s}_{\boldsymbol{\theta}} \triangleq (\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}}$. To make the bridges between notations clearer, Mensch and Blondel [2018] use a generic, dimension-agnostic definition of the regularizer Ω, and $\mathsf{max}^C_{\boldsymbol{w},\Omega_2}(\boldsymbol{\theta})$ and $\mathrm{max}_{\Omega_{|\mathcal{Y}^C_{\boldsymbol{w}}|}}(\boldsymbol{s}_{\boldsymbol{\theta}})$ are denoted by $\mathrm{DP}_\Omega(\boldsymbol{\theta})$ and $\mathrm{LP}_\Omega(\boldsymbol{\theta})$, respectively.

Moreover, since we have the closed form $\mathsf{max}_{\Omega_d}(\boldsymbol{q}) = \gamma \log \sum_{i=1}^d \exp(q_i/\gamma)$ for any $d \geq 2$, we get:

$$\begin{aligned}
\boldsymbol{y}^C_{\boldsymbol{w},\Omega_2}(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \mathsf{max}^C_{\boldsymbol{w},\Omega_2}(\boldsymbol{\theta}) \\
&= \nabla_{\boldsymbol{\theta}} \mathsf{max}_{\Omega_{|\mathcal{Y}^C_{\boldsymbol{w}}|}}(\boldsymbol{s}_{\boldsymbol{\theta}}) \\
&= \nabla_{\boldsymbol{\theta}} \left(\gamma \log \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right) \right) \\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma\right)} \cdot \boldsymbol{y} .
\end{aligned}$$

While we wrote the proof only in Knapsack notation, since the differences in the padding values initializing the table $\boldsymbol{V}^{\boldsymbol{w}}_\Omega$ (which is the only distinction between the Knapsack and the Top-k cases) do not impact it, the exact same arguments naturally hold for $\boldsymbol{y}^k_{\mathbf{1},\Omega}$. □

A.2 Proof of Proposition 1 (Characterization of equivariance)

Proof. • ($\Longrightarrow$)

First, we show the equivariance of $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ with respect to permutations of $\boldsymbol{\theta}$ and $\boldsymbol{w}$ for $\Omega = -\gamma H^s$. First, we assume $\gamma > 0$. In this setting, we have the following closed form:

$$\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma\right)} \cdot \boldsymbol{y} .$$

Let $\sigma \in S_n$. We then have:

$$\boldsymbol{y}^C_{\sigma(\boldsymbol{w}),\Omega}\left(\sigma(\boldsymbol{\theta})\right) = \sum_{\boldsymbol{y}\in\mathcal{Y}^C_{\sigma(\boldsymbol{w})}} \frac{\exp\left(\langle\sigma(\boldsymbol{\theta}),\boldsymbol{y}\rangle/\gamma\right)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\sigma(\boldsymbol{w})}}\exp\left(\langle\sigma(\boldsymbol{\theta}),\boldsymbol{y}'\rangle/\gamma\right)}\cdot\boldsymbol{y}\,.$$

Let us recall that the action of the permutation σ on $\mathbb{R}^n$ is that of an orthogonal endomorphism, giving:

$$\forall \boldsymbol{u},\boldsymbol{v}\in\mathbb{R}^n,\quad \langle\sigma(\boldsymbol{u}),\boldsymbol{v}\rangle = \langle\boldsymbol{u},\sigma^{-1}(\boldsymbol{v})\rangle,\quad \text{and}\quad \langle\sigma(\boldsymbol{u}),\sigma(\boldsymbol{v})\rangle = \langle\boldsymbol{u},\boldsymbol{v}\rangle.$$

Then, by definition of the feasible item selection set, we have:

$$\begin{aligned}
\mathcal{Y}^C_{\sigma(\boldsymbol{w})} &= \{\boldsymbol{y}\in\{0,1\}^n \mid \langle\sigma(\boldsymbol{w}),\boldsymbol{y}\rangle \le C\}\\
&= \left\{\boldsymbol{y}\in\{0,1\}^n \mid \langle\boldsymbol{w},\sigma^{-1}(\boldsymbol{y})\rangle \le C\right\}\\
&= \{\sigma(\boldsymbol{y}') \mid \boldsymbol{y}'\in\{0,1\}^n,\ \langle\boldsymbol{w},\boldsymbol{y}'\rangle \le C\} \qquad (\text{since } \sigma^{-1}(\{0,1\}^n)=\{0,1\}^n)\\
&= \left\{\sigma(\boldsymbol{y}') \mid \boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}\right\},
\end{aligned}$$

where we denote $\sigma^{-1}(\boldsymbol{y}) \triangleq \boldsymbol{y}'$ for clarity.

Thus, we have in fact:

$$\begin{aligned}
\boldsymbol{y}^C_{\sigma(\boldsymbol{w}),\Omega}\left(\sigma(\boldsymbol{\theta})\right) &= \sum_{\boldsymbol{y}\in\mathcal{Y}^C_{\sigma(\boldsymbol{w})}} \frac{\exp\left(\langle\sigma(\boldsymbol{\theta}),\boldsymbol{y}\rangle/\gamma\right)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\sigma(\boldsymbol{w})}}\exp\left(\langle\sigma(\boldsymbol{\theta}),\boldsymbol{y}'\rangle/\gamma\right)}\cdot\boldsymbol{y}\\
&= \sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp\left(\langle\sigma(\boldsymbol{\theta}),\sigma(\boldsymbol{y}')\rangle/\gamma\right)}{\sum_{\boldsymbol{y}''\in\mathcal{Y}^C_{\boldsymbol{w}}}\exp\left(\langle\sigma(\boldsymbol{\theta}),\sigma(\boldsymbol{y}'')\rangle/\gamma\right)}\cdot\sigma(\boldsymbol{y}')\\
&= \sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp\left(\langle\boldsymbol{\theta},\boldsymbol{y}'\rangle/\gamma\right)}{\sum_{\boldsymbol{y}''\in\mathcal{Y}^C_{\boldsymbol{w}}}\exp\left(\langle\boldsymbol{\theta},\boldsymbol{y}''\rangle/\gamma\right)}\cdot\sigma(\boldsymbol{y}')\\
&= \sigma\left(\sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp\left(\langle\boldsymbol{\theta},\boldsymbol{y}'\rangle/\gamma\right)}{\sum_{\boldsymbol{y}''\in\mathcal{Y}^C_{\boldsymbol{w}}}\exp\left(\langle\boldsymbol{\theta},\boldsymbol{y}''\rangle/\gamma\right)}\cdot\boldsymbol{y}'\right)\\
&= \sigma\left(\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})\right).
\end{aligned}$$

For the Top-k case, the exact same arguments hold, and the final expression of equivariance is only simplified because $\sigma(\boldsymbol{w}) = \boldsymbol{w}$ as $\boldsymbol{w} = \mathbf{1}$.

Second, for the case $\gamma = 0$, we recover the unregularized operators, i.e., $\boldsymbol{y}^C_{\boldsymbol{w},\Omega} = \boldsymbol{y}^C_{\boldsymbol{w}}$ and $\boldsymbol{y}^k_{\mathbf{1},\Omega} = \boldsymbol{y}^k_{\mathbf{1}}$, so that the equivariance property is simply directly implied by their definition as maximizers in Problems (1) and (2).

• ($\Longleftarrow$)

We now show that Shannon's negative entropy is the only choice of a convex, separable regularizer yielding permutation-equivariant relaxed operators.

Let $\Omega : \triangle^2 \to \mathbb{R}$ be a convex regularization function. We assume it is separable, i.e., such that $\Omega(\boldsymbol{q}) = \omega(q_1) + \omega(q_2)$ for some lower semi-continuous convex function $\omega : [0,1] \to \mathbb{R}$. Moreover, we assume that $\Omega(\boldsymbol{q}) = 0$ for $\boldsymbol{q} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ (i.e., $\omega(0)+\omega(1) = 0$), and that Ω yields the permutation-equivariance property for the relaxed layers, i.e., that we have:

$$\begin{aligned}
\forall(\sigma,\boldsymbol{\theta},\boldsymbol{w})\in S_n\times\mathbb{R}^n\times\mathbb{N}^n,\quad \boldsymbol{y}^C_{\sigma(\boldsymbol{w}),\Omega}(\sigma(\boldsymbol{\theta})) &= \sigma\left(\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})\right),\\
\boldsymbol{y}^k_{\mathbf{1},\Omega}(\sigma(\boldsymbol{\theta})) &= \sigma\left(\boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta})\right).
\end{aligned}$$

We must show that we then necessarily have $\Omega = -\gamma H^s$ for some $\gamma \ge 0$. In fact, we will show that Ω must necessarily yield an *associative* smoothed maximum operator max_Ω: indeed, both are equivalent by Lemma 3 in Mensch and Blondel [2018].

More precisely, we will assume that max_Ω is not associative for the sake of contradiction, and show that the permutation-equivariance property cannot be verified.

To do so, Lemmas 1 and 2 will prove useful.

Lemma 1. *Let $\Omega : \triangle^2 \to \mathbb{R}$ be convex, with $\Omega(\boldsymbol{q}) = \omega(q_1) + \omega(q_2)$, where $\omega : [0, 1] \to \mathbb{R}$ is lower semi-continuous, and such that $\Omega(\boldsymbol{q}) = 0$ for $\boldsymbol{q} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2\}$. We have, for any $x \in \mathbb{R}$:*

$$\lim_{c \to +\infty} \mathsf{max}_\Omega(x + c, 0) - c = x\,.$$

Lemma 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be any differentiable function. Then, we have:*

$$f \circ \sigma = f \quad \textit{for all } \sigma \in S_n \qquad \Longleftrightarrow \qquad \nabla f \circ \sigma = \sigma \circ \nabla f \quad \textit{for all } \sigma \in S_n\,.$$

That is, a differentiable function is permutation-invariant if and only if its gradient is permutation-equivariant.

We give a proof of Lemmas 1 and 2 in Sections A.5 and A.6 for completeness. We now build simple instances of the Top-k and Knapsack problems for which we show that the associativity of max_Ω is implied by the permutation-equivariance assumption for $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}$ and $\boldsymbol{y}^k_{\boldsymbol{1},\Omega}$.

- **Knapsack case.** Let $n = 3$, $\boldsymbol{w} = \boldsymbol{1}$, $C = 1$. The smoothed dynamic program for solving the corresponding Knapsack problem with $\boldsymbol{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ is depicted in Figure 4. Unrolling the smoothed recursion from Equation (4) gives:

$$\begin{aligned}
\mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{x}) &= V^{\boldsymbol{w}}_\Omega[3, 1] \\
&= \mathsf{max}_\Omega\left(x_3 + 0, V^{\boldsymbol{w}}_\Omega[2, 1]\right) \\
&= \mathsf{max}_\Omega\left(x_3, \mathsf{max}_\Omega\left(x_2 + 0, V^{\boldsymbol{w}}_\Omega[1, 1]\right)\right) \\
&= \mathsf{max}_\Omega\left(x_3, \mathsf{max}_\Omega\left(x_2, \mathsf{max}_\Omega\left(x_1, 0\right)\right)\right).
\end{aligned}$$

Moreover, since Ω is separable, it is commutative, so that max_Ω is also commutative. Thus, we get:

$$\mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{x}) = \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(x_1, 0\right), x_2\right), x_3\right).$$

Now let $\sigma \in S_3$ be defined by $\sigma((x_1, x_2, x_3)) = (x_3, x_2, x_1)$ for all $\boldsymbol{x} \in \mathbb{R}^3$. We have, similarly:

$$\begin{aligned}
\mathsf{max}^C_{\sigma(\boldsymbol{w}),\Omega}(\sigma(\boldsymbol{x})) &= \mathsf{max}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{x})) && \text{(since } \boldsymbol{w} = \boldsymbol{1}\text{, so } \sigma(\boldsymbol{w}) = \boldsymbol{w}\text{)} \\
&= \mathsf{max}_\Omega\left(x_1, \mathsf{max}_\Omega\left(x_2, \mathsf{max}_\Omega\left(x_3, 0\right)\right)\right).
\end{aligned}$$

Since we assumed for the sake of contradiction that max_Ω is not associative, we can find $(\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$ such that $\mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\theta_1, \theta_2\right), \theta_3\right) \neq \mathsf{max}_\Omega\left(\theta_1, \mathsf{max}_\Omega\left(\theta_2, \theta_3\right)\right)$. However, from what precedes, and using the distributivity of $+$ over max_Ω, we also get, for any $c \in \mathbb{R}$:

$$\begin{aligned}
\mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta} + c\boldsymbol{1}) &= \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\theta_1 + c, 0\right), \theta_2 + c\right), \theta_3 + c\right) \\
&= c + \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\theta_1 + c, 0\right) - c, \theta_2\right), \theta_3\right), \\
\mathsf{max}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{\theta} + c\boldsymbol{1})) &= \mathsf{max}_\Omega\left(\theta_1 + c, \mathsf{max}_\Omega\left(\theta_2 + c, \mathsf{max}_\Omega\left(\theta_3 + c, 0\right)\right)\right) \\
&= c + \mathsf{max}_\Omega\left(\theta_1, \mathsf{max}_\Omega\left(\theta_2, \mathsf{max}_\Omega\left(\theta_3 + c, 0\right) - c\right)\right).
\end{aligned}$$

Thus, we have, for any $c \in \mathbb{R}$:

$$\begin{aligned}
\mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta} + c\boldsymbol{1}) - c &= \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\theta_1 + c, 0\right) - c, \theta_2\right), \theta_3\right), \\
\mathsf{max}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{\theta} + c\boldsymbol{1})) - c &= \mathsf{max}_\Omega\left(\theta_1, \mathsf{max}_\Omega\left(\theta_2, \mathsf{max}_\Omega\left(\theta_3 + c, 0\right) - c\right)\right).
\end{aligned}$$

By Lemma 1, we have:

$$\begin{aligned}
\mathsf{max}_\Omega\left(\theta_1 + c, 0\right) - c &\xrightarrow[c \to +\infty]{} \theta_1, \\
\mathsf{max}_\Omega\left(\theta_3 + c, 0\right) - c &\xrightarrow[c \to +\infty]{} \theta_3.
\end{aligned}$$

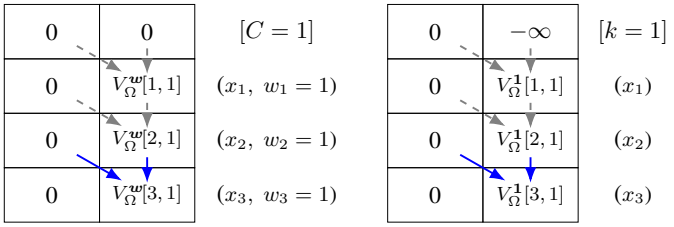


Figure 4: Example DP tables for the smoothed Knapsack (left) and Top-k (right) recursions.

Since max_Ω is continuous, we can take the limit of the previous expressions:

$$\lim_{c\to+\infty} \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta} + c\mathbf{1}) - c = \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\theta_1, \theta_2\right), \theta_3\right),$$
$$\lim_{c\to+\infty} \mathsf{max}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{\theta} + c\mathbf{1})) - c = \mathsf{max}_\Omega\left(\theta_1, \mathsf{max}_\Omega\left(\theta_2, \theta_3\right)\right).$$

Thus, by our definition of $\boldsymbol{\theta}$, we have:

$$\lim_{c\to+\infty} \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta} + c\mathbf{1}) - c \neq \lim_{c\to+\infty} \mathsf{max}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{\theta} + c\mathbf{1})) - c.$$

Thus, we necessarily have:

$$\exists c \in \mathbb{R}: \quad \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta} + c\mathbf{1}) \neq \mathsf{max}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{\theta} + c\mathbf{1})).$$

Further, since $\boldsymbol{y}^C_{\boldsymbol{w},\Omega} = \nabla\mathsf{max}^C_{\boldsymbol{w},\Omega}$, Lemma 2 gives that this prevents us from having $\sigma(\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{x})) = \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\sigma(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathbb{R}^n$, which contradicts the permutation-equivariance assumption (since we have $\boldsymbol{w} = \sigma(\boldsymbol{w}) = \mathbf{1}$).

- **Top-k case.** Let $n = 3$, $k = 1$. The smoothed dynamic program for solving the corresponding Top-k problem with input $\boldsymbol{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ is depicted in Figure 4. Unrolling the smoothed recursion from Equation (4) gives:

$$\begin{aligned}
\mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{x}) &= V^{\mathbf{1}}_\Omega[3,1] \\
&= \mathsf{max}_\Omega\left(x_3 + 0, V^{\mathbf{1}}_\Omega[2,1]\right) \\
&= \mathsf{max}_\Omega\left(x_3, \mathsf{max}_\Omega\left(x_2 + 0, V^{\mathbf{1}}_\Omega[1,1]\right)\right) \\
&= \mathsf{max}_\Omega\left(x_3, \mathsf{max}_\Omega\left(x_2, \mathsf{max}_\Omega\left(x_1, -\infty\right)\right)\right).
\end{aligned}$$

From the definition of smoothed maximum operators, we can easily see that $\mathsf{max}_\Omega(x, -\infty) = x - \Omega(\boldsymbol{e}_1)$ for all $x \in \mathbb{R}$, since the maximizing distribution in the definition of max_Ω is necessarily $\boldsymbol{q} = \boldsymbol{e}_1$. Moreover, since we assumed $\Omega(\boldsymbol{q}) = 0$ when $\boldsymbol{q} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2\}$, we have $\mathsf{max}_\Omega(x, -\infty) = x$. Thus, we get in fact:

$$\begin{aligned}
\mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{x}) &= \mathsf{max}_\Omega\left(x_3, \mathsf{max}_\Omega\left(x_2, x_1\right)\right) \\
&= \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(x_1, x_2\right), x_3\right),
\end{aligned}$$

where we also used commutativity of max_Ω. Now let $\sigma \in S_3$ be defined by $\sigma((x_1, x_2, x_3)) = (x_3, x_2, x_1)$ for all $\boldsymbol{x} \in \mathbb{R}^3$. We have, similarly:

$$\begin{aligned}
\mathsf{max}^k_{\mathbf{1},\Omega}(\sigma(\boldsymbol{x})) &= \mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(x_3, x_2\right), x_1\right) \\
&= \mathsf{max}_\Omega\left(x_1, \mathsf{max}_\Omega\left(x_2, x_3\right)\right).
\end{aligned}$$

Since we assumed for the sake of contradiction that max_Ω is not associative, we can find $(\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$ such that $\mathsf{max}_\Omega\left(\mathsf{max}_\Omega\left(\theta_1, \theta_2\right), \theta_3\right) \neq \mathsf{max}_\Omega\left(\theta_1, \mathsf{max}_\Omega\left(\theta_2, \theta_3\right)\right)$. Thus, from previous calculations, we then have $\mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}) \neq \mathsf{max}^k_{\mathbf{1},\Omega}(\sigma(\boldsymbol{\theta}))$. Further, from Lemma 2, since $\boldsymbol{y}^k_{\mathbf{1},\Omega} = \nabla\mathsf{max}^k_{\mathbf{1},\Omega}$, this prevents us from having $\boldsymbol{y}^k_{\mathbf{1},\Omega}(\sigma(\boldsymbol{\theta})) = \sigma(\boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}))$ for all $\boldsymbol{\theta} \in \mathbb{R}^3$, which contradicts the permutation-equivariance assumption.

□

A.3 Proof of Proposition 2 (Characterization of sparsity)

Proof. We prove the equivalence via the cycle (1) $\implies$ (3) $\implies$ (2) $\implies$ (1) (assuming a non-trivial feasible set $|\mathcal{Y}_{\boldsymbol{w}}^C| \geq 2$, with $1 \leq k < n$ in the Top-k case, as otherwise $\mathsf{conv}(\mathcal{Y}_{\boldsymbol{w}}^C)$ is a singleton and all statements hold trivially). We write the proof in Knapsack notation, noting where the $-\infty$ boundary initialization of the Top-k case ($\boldsymbol{w} = \mathbf{1}, C = k$) plays a role.

• **(1) $\implies$ (3).** Assume $\lim_{t\to 0^+} \omega'(t) > -\infty$ and $\lim_{t\to 1^-} \omega'(t) < +\infty$. By strict convexity of ω, the gap $\tau_\Omega \triangleq \lim_{t\to 1^-} \omega'(t) - \lim_{t\to 0^+} \omega'(t)$ satisfies $0 < \tau_\Omega < \infty$. Moreover, since $\Omega : \triangle^2 \to \mathbb{R}$ is convex and closed (so that max_Ω attains its maximum on $\triangle^2$), it is continuous on the compact simplex $\triangle^2$ and thus bounded: $M_\Omega \triangleq \sup_{\boldsymbol{q}\in\triangle^2} |\Omega(\boldsymbol{q})| < \infty$.

For any $\gamma > 0$, the scaled regularizer $\gamma\Omega$ has saturation threshold $\gamma\tau_\Omega$ and maximum absolute value γM_Ω. By Blondel et al. [2020a, Definition 3, Propositions 7 and 8], for any $a, b \in \mathbb{R}$, the local gradient mapping saturates to a hard decision once $|a - b| \geq \gamma\tau_\Omega$:

$$a - b \geq \gamma\tau_\Omega \implies (\nabla \mathsf{max}_{\gamma\Omega}(a, b))_1 = 1, \qquad b - a \geq \gamma\tau_\Omega \implies (\nabla \mathsf{max}_{\gamma\Omega}(a, b))_1 = 0. \tag{5}$$

Next, we bound the discrepancy between the smoothed and unregularized DP value tables across all feasible recursion states (i, c) (i.e., $0 \leq c \leq C$ for Knapsack, and $0 \leq c \leq \min(i, k)$ for Top-k).

Lemma 3. *Let $M_\Omega \triangleq \sup_{\boldsymbol{q}\in\triangle^2} |\Omega(\boldsymbol{q})| < \infty$. For any $\boldsymbol{\theta} \in \mathbb{R}^n$, $i \in \{0, \ldots, n\}$, and any feasible capacity c at step i, we have:*

$$|V_\Omega^{\boldsymbol{w}}[i, c] - V^{\boldsymbol{w}}[i, c]| \leq iM_\Omega.$$

We prove Lemma 3 by induction in Section A.7.

Applying Lemma 3 to $\gamma\Omega$ gives $|V_{\gamma\Omega}^{\boldsymbol{w}}[i, c] - V^{\boldsymbol{w}}[i, c]| \leq i\gamma M_\Omega$ for every feasible state (i, c).

Now let $\boldsymbol{\theta} \in \mathbb{R}^n$ admit a unique hard maximizer $\boldsymbol{y}^* \triangleq \boldsymbol{y}_{\boldsymbol{w}}^C(\boldsymbol{\theta})$, and trace the sequence of remaining capacities along its optimal DP path: $c_n \triangleq C$ and $c_{i-1} \triangleq c_i - y_i^* w_i$ for $i = n, \ldots, 1$. Partition $[n]$ into forced steps and active binary steps along this path:

- *Forced steps:* if $w_i > c_i$ (which forces $y_i^* = 0$, and includes $c_i = 0$ in Top-k), Equation (4) sets $V_{\gamma\Omega}^{\boldsymbol{w}}[i, c_i] = V_{\gamma\Omega}^{\boldsymbol{w}}[i - 1, c_i]$, so $Q_{\gamma\Omega}^{\boldsymbol{w}}[i, c_i] = 0 = y_i^*$ for all $\gamma > 0$. Likewise, in Top-k when $c_i = i$ (which forces $y_i^* = 1$), the skip parent $(i - 1, i)$ equals $-\infty$, so $Q_{\gamma\Omega}^{\mathbf{1}}[i, i] = 1 = y_i^*$ for all $\gamma > 0$.
- *Active binary steps ($\mathcal{A}$):* let $\mathcal{A} \subseteq [n]$ be the remaining indices, where both parent states $(i - 1, c_i)$ and $(i - 1, c_i - w_i)$ are feasible. For each $i \in \mathcal{A}$, define the hard local skipping advantage:
$$\Delta_{i,c_i}^{\text{hard}} \triangleq V^{\boldsymbol{w}}[i - 1, c_i] - \big(\theta_i + V^{\boldsymbol{w}}[i - 1, c_i - w_i]\big).$$
Since $\boldsymbol{y}^*$ is the unique global maximizer, every active decision along the optimal path must be strict (otherwise, taking the alternative choice $1 - y_i^*$ and completing items $\{1, \ldots, i - 1\}$ optimally would yield a distinct optimal solution $\tilde{\boldsymbol{y}} \in \mathcal{Y}_{\boldsymbol{w}}^C \setminus \{\boldsymbol{y}^*\}$). Thus $\Delta_{i,c_i}^{\text{hard}} > 0$ if $y_i^* = 0$ and $\Delta_{i,c_i}^{\text{hard}} < 0$ if $y_i^* = 1$.

If $\mathcal{A} = \emptyset$, then $Q_{\gamma\Omega}^{\boldsymbol{w}}[i, c_i] = y_i^*$ already holds for any $\gamma > 0$. Otherwise, define $\Delta^\star \triangleq \min_{i\in\mathcal{A}} |\Delta_{i,c_i}^{\text{hard}}| > 0$ and set:

$$\gamma^* \triangleq \frac{\Delta^\star}{\tau_\Omega + 2(n - 1)M_\Omega} > 0.$$

Fix any $\gamma \in (0, \gamma^*]$ and $i \in \mathcal{A}$, and let $\Delta_{i,c_i}^{\gamma\Omega} \triangleq V_{\gamma\Omega}^{\boldsymbol{w}}[i - 1, c_i] - (\theta_i + V_{\gamma\Omega}^{\boldsymbol{w}}[i - 1, c_i - w_i])$ be the smoothed local advantage. By the triangle inequality and Lemma 3,

$$\begin{aligned}\left|\Delta_{i,c_i}^{\gamma\Omega} - \Delta_{i,c_i}^{\text{hard}}\right| &\leq \left|V_{\gamma\Omega}^{\boldsymbol{w}}[i - 1, c_i] - V^{\boldsymbol{w}}[i - 1, c_i]\right| + \left|V_{\gamma\Omega}^{\boldsymbol{w}}[i - 1, c_i - w_i] - V^{\boldsymbol{w}}[i - 1, c_i - w_i]\right| \\ &\leq 2(i - 1)\gamma M_\Omega \leq 2(n - 1)\gamma M_\Omega.\end{aligned}$$

Because $\gamma \leq \gamma^*$ and $\tau_\Omega > 0$, we have $2(n - 1)\gamma M_\Omega < \gamma(\tau_\Omega + 2(n - 1)M_\Omega) \leq \Delta^\star \leq |\Delta_{i,c_i}^{\text{hard}}|$, ensuring that $\Delta_{i,c_i}^{\gamma\Omega}$ has the same sign as $\Delta_{i,c_i}^{\text{hard}}$. Moreover, by the reverse triangle inequality,

$$\left|\Delta_{i,c_i}^{\gamma\Omega}\right| \geq \left|\Delta_{i,c_i}^{\text{hard}}\right| - 2(n - 1)\gamma M_\Omega \geq \Delta^\star - 2(n - 1)\gamma M_\Omega \geq \gamma\tau_\Omega.$$

By Equation (5), the local gate saturates to $Q^{\boldsymbol{w}}_{\gamma\Omega}[i, c_i] = y^*_i$ for all $i \in \mathcal{A}$, and hence for all $i \in [n]$.

Finally, in the backward pass (Equation (8)), initialized with $E^{\boldsymbol{w}}_{\gamma\Omega}[n, C] = 1$ and 0 elsewhere, downward induction from n to 1 routes all unit mass deterministically along (i, c_i), giving $E^{\boldsymbol{w}}_{\gamma\Omega}[i, c_i] = 1$ and $E^{\boldsymbol{w}}_{\gamma\Omega}[i, c] = 0$ for $c \neq c_i$ (if $c_i = 0$ with positive weights, $y^*_l = 0$ and no mass remains on $c \geq 1$ for all $l \leq i$). Therefore, $(\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}))_i = \sum_c E^{\boldsymbol{w}}_{\gamma\Omega}[i, c] Q^{\boldsymbol{w}}_{\gamma\Omega}[i, c] = y^*_i$ for all $i \in [n]$, i.e., $\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$.

• **(3) $\implies$ (2).** By Property 3 of Section D.4, we have $\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}/\gamma)$ for all $\gamma > 0$ and $\boldsymbol{\theta} \in \mathbb{R}^n$.

Let $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$ be arbitrary and set $\boldsymbol{\theta} \triangleq 2\boldsymbol{y} - \mathbf{1} \in \mathbb{R}^n$. For any $\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}} \setminus \{\boldsymbol{y}\}$, we have

$$\langle \boldsymbol{\theta}, \boldsymbol{y} - \boldsymbol{y}' \rangle = \sum_{i:y_i=1} (1 - y'_i) + \sum_{i:y_i=0} y'_i = \|\boldsymbol{y} - \boldsymbol{y}'\|_1 > 0,$$

so $\boldsymbol{y} = \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$ is the unique hard maximizer. By (3), there exists a threshold $\gamma^* > 0$ such that $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}/\gamma^*) = \boldsymbol{y}^C_{\boldsymbol{w},\gamma^*\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) = \boldsymbol{y}$, and therefore $\boldsymbol{y} \in \mathsf{range}(\boldsymbol{y}^C_{\boldsymbol{w},\Omega})$ and $\mathcal{Y}^C_{\boldsymbol{w}} \subseteq \mathsf{range}(\boldsymbol{y}^C_{\boldsymbol{w},\Omega})$.

• **(2) $\implies$ (1).** We proceed by contraposition. Suppose (1) fails. Since ω is convex on $(0, 1)$, ω' is non-decreasing, so $\omega'(0^+) < +\infty$ and $\omega'(1^-) > -\infty$; hence the negation of (1) implies either $\lim_{t\to 0^+} \omega'(t) = -\infty$ or $\lim_{t\to 1^-} \omega'(t) = +\infty$, and in both cases we have that the difference is $\lim_{q\to 1^-} \omega'(q) - \lim_{q\to 0^+} \omega'(q) = +\infty$.

For any finite $a, b \in \mathbb{R}$, the local objective $f(q) \triangleq qa + (1 - q)b - \omega(q) - \omega(1 - q)$ has derivative $f'(q) = (a - b) - (\omega'(q) - \omega'(1 - q))$ on $(0, 1)$, satisfying $\lim_{q\to 0^+} f'(q) = +\infty$ and $\lim_{q\to 1^-} f'(q) = -\infty$. Consequently, its unique maximizer $q^* = (\nabla \mathsf{max}_\Omega(a, b))_1$ lies strictly in $(0, 1)$ whenever $a, b \in \mathbb{R}$ are finite.

Now fix any $\boldsymbol{\theta} \in \mathbb{R}^n$:

- In the Top-k case ($1 \leq k < n$), both parent states $(n-1, k-1)$ and $(n-1, k)$ of the sink (n, k) are feasible ($0 \leq k - 1 < k \leq n - 1$), so their smoothed DP values are finite. Thus $Q^{\boldsymbol{w}}_\Omega[n, k] \in (0, 1)$, and since $E^{\boldsymbol{w}}_\Omega[n, k] = 1$ (with $E^{\boldsymbol{w}}_\Omega[n, c] = 0$ for $c \neq k$), we get $(\boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}))_n = Q^{\boldsymbol{w}}_\Omega[n, k] \in (0, 1)$.
- In the Knapsack case ($|\mathcal{Y}^C_{\boldsymbol{w}}| \geq 2$), since $\mathbf{0} \in \mathcal{Y}^C_{\boldsymbol{w}}$, there exists $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}} \setminus \{\mathbf{0}\}$, so $\{i \in [n] \mid w_i \leq C\}$ is non-empty; let $i^\star \triangleq \max\{i \in [n] \mid w_i \leq C\}$. For all $j > i^\star$, $w_j > C$ implies $Q^{\boldsymbol{w}}_\Omega[j, C] = 0$, so the backward recursion (Equation (8)) preserves $E^{\boldsymbol{w}}_\Omega[i^\star, C] = 1$ and $E^{\boldsymbol{w}}_\Omega[i^\star, c] = 0$ for $c \neq C$. At $(i^\star, C)$, $w_{i^\star} \leq C$ and both parent values $V^{\boldsymbol{w}}_\Omega[i^\star - 1, C - w_{i^\star}]$ and $V^{\boldsymbol{w}}_\Omega[i^\star - 1, C]$ are finite, giving $Q^{\boldsymbol{w}}_\Omega[i^\star, C] \in (0, 1)$ and $(\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))_{i^\star} = Q^{\boldsymbol{w}}_\Omega[i^\star, C] \in (0, 1)$.

In both cases, $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) \notin \{0, 1\}^n \ \forall \boldsymbol{\theta} \in \mathbb{R}^n$, so $\mathsf{range}(\boldsymbol{y}^C_{\boldsymbol{w},\Omega}) \cap \mathcal{Y}^C_{\boldsymbol{w}} = \emptyset$, and (2) cannot hold. □

A.4 Proof of Proposition 3 (Underlying distribution)

Proof. We prove the three claims of the proposition sequentially. We adopt the Knapsack notation ($\mathsf{max}^C_{\boldsymbol{w},\Omega}$, $\boldsymbol{V}^{\boldsymbol{w}}_\Omega$, etc.), but the proof holds identically for the Top-k case.

Support of the distribution. We wish to show that for any $\boldsymbol{y}$ sampled from $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$, we have $\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}$ (i.e., $\langle \boldsymbol{w}, \boldsymbol{y} \rangle \leq C$). The sampling procedure in Algorithm 3 samples y_i sequentially from $i = n$ down to 1. Note that the remaining capacity C_i available for items $\{1, \ldots, i\}$ is initialized at $C_n = C$ and updated as $C_{i-1} = C_i - y_i w_i$.

Consider the step for item i with current remaining capacity C_i. If $w_i > C_i$, the forward DP recursion in Equation (4) is defined via the first case:

$$V^{\boldsymbol{w}}_\Omega[i, C_i] = V^{\boldsymbol{w}}_\Omega[i - 1, C_i].$$

Consequently, the derivative with respect to θ_i is zero:

$$Q^{\boldsymbol{w}}_\Omega[i, C_i] = \frac{\partial V^{\boldsymbol{w}}_\Omega[i, C_i]}{\partial \theta_i} = 0.$$

By definition of $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$, the probability of selecting item i is given by $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_i = 1 \mid C_i) = Q^{\boldsymbol{w}}_{\Omega}[i, C_i]$. Thus, if $w_i > C_i$, we have $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_i = 1 \mid C_i) = 0$, forcing $y_i = 0$. This ensures that the capacity constraint is never violated at any step. Thus, $\sum_{i=1}^{n} y_i w_i \leq C$, and the support of $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ is a subset of $\mathcal{Y}^{C}_{\boldsymbol{w}}$.

Expectation as the relaxed operator. We now show that $\mathbb{E}_{Y\sim\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}}[Y] = \boldsymbol{y}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$. Since $\boldsymbol{y}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \nabla\mathsf{max}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$, and recalling from Section B.2 that the gradient is computed as:

$$(\boldsymbol{y}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))_i = (\nabla\mathsf{max}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))_i = \sum_{c=0}^{C} E^{\boldsymbol{w}}_{\Omega}[i, c] Q^{\boldsymbol{w}}_{\Omega}[i, c],$$

we must show that the marginal probability of selecting item i under $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$, denoted μ_i, is equal to $\sum_{c=0}^{C} E^{\boldsymbol{w}}_{\Omega}[i, c] Q^{\boldsymbol{w}}_{\Omega}[i, c]$.

Let C_i be the random variable representing the remaining capacity for items $\{1 \dots i\}$ during the ancestral sampling process. The probability of selecting item i is:

$$\mu_i = \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_i = 1) = \sum_{c=0}^{C} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_i = 1 \mid C_i = c)\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_i = c).$$

By definition of the sampling distribution, $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_i = 1 \mid C_i = c) = Q^{\boldsymbol{w}}_{\Omega}[i, c]$. Thus, it suffices to show that $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_i = c) = E^{\boldsymbol{w}}_{\Omega}[i, c]$ holds for all $i \in [n]$ and $c \in \{0, \dots, C\}$.

We proceed by induction on i, moving backwards from n to 1.

- **Base case ($i = n$):** The sampling always starts with capacity C. Thus $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_n = C) = 1$ and $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_n = c) = 0$ for $c \neq C$. Matching this, the backward pass in Algorithm 2 initializes $E^{\boldsymbol{w}}_{\Omega}[n, C] = 1$ and $E^{\boldsymbol{w}}_{\Omega}[n, c] = 0$ otherwise. The base case holds.
- **Inductive step ($i < n$):** Assume $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_{i+1} = c) = E^{\boldsymbol{w}}_{\Omega}[i + 1, c]$ for all c. We express the probability of the capacity C_i at step i. The capacity c at step i can be reached from step $i + 1$ in two ways:

  1. We had capacity c at $i + 1$ and chose $y_{i+1} = 0$ (skip).
  2. We had capacity $c + w_{i+1}$ at $i + 1$ and chose $y_{i+1} = 1$ (pick).

  Formally:

  $$\begin{aligned}\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_i = c) = \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_{i+1} = c) \times \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_{i+1} = 0 \mid C_{i+1} = c) \\ + \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_{i+1} = c + w_{i+1}) \times \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_{i+1} = 1 \mid C_{i+1} = c + w_{i+1}).\end{aligned}$$

  Plugging the definition of the sampling probabilities as $\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(y_k = 1|C_k = \cdot) \triangleq Q^{\boldsymbol{w}}_{\Omega}[k, \cdot]$ and the induction hypothesis, we get:

  $$\begin{aligned}\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(C_i = c) = E^{\boldsymbol{w}}_{\Omega}[i + 1, c](1 - Q^{\boldsymbol{w}}_{\Omega}[i + 1, c]) \\ + E^{\boldsymbol{w}}_{\Omega}[i + 1, c + w_{i+1}]Q^{\boldsymbol{w}}_{\Omega}[i + 1, c + w_{i+1}].\end{aligned}$$

  This is exactly the backward recursion derived for $E^{\boldsymbol{w}}_{\Omega}$ in Equation (8).

Thus, $E^{\boldsymbol{w}}_{\Omega}[i, c]$ represents the marginal probability that the remaining capacity is c at step i. We therefore have:

$$\mathbb{E}_{Y\sim\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}}[Y_i] = \mu_i = \sum_{c=0}^{C} E^{\boldsymbol{w}}_{\Omega}[i, c] Q^{\boldsymbol{w}}_{\Omega}[i, c] = (\boldsymbol{y}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))_i\,,$$

which finally gives:

$$\mathbb{E}_{Y\sim\pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}}[Y] = \boldsymbol{y}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}).$$

Recovery of the Gibbs distribution. Let $\Omega = -\gamma H^s$. The smoothed maximum becomes the *log-sum-exp* function:

$$\mathsf{max}_\Omega(a, b) = \gamma \log(\exp(a/\gamma) + \exp(b/\gamma)).$$

The forward recursion becomes:

$$V_\Omega^{\boldsymbol{w}}[i, c] = \gamma \log\left(\exp(V_\Omega^{\boldsymbol{w}}[i-1, c]/\gamma) + \exp((\theta_i + V_\Omega^{\boldsymbol{w}}[i-1, c-w_i])/\gamma)\right).$$

The local probability $Q_\Omega^{\boldsymbol{w}}[i, c]$ becomes the standard softmax probability:

$$Q_\Omega^{\boldsymbol{w}}[i, c] = \frac{\exp((\theta_i + V_\Omega^{\boldsymbol{w}}[i-1, c-w_i])/\gamma)}{\exp(V_\Omega^{\boldsymbol{w}}[i, c]/\gamma)},$$
$$1 - Q_\Omega^{\boldsymbol{w}}[i, c] = \frac{\exp(V_\Omega^{\boldsymbol{w}}[i-1, c]/\gamma)}{\exp(V_\Omega^{\boldsymbol{w}}[i, c]/\gamma)}.$$

Now consider the probability of a full sampled vector $\boldsymbol{y}$. Let c_i be the sequence of remaining capacities realized during sampling (with $c_n = C$ and $c_{i-1} = c_i - y_i w_i$).

$$\begin{aligned}
\pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}(\boldsymbol{y}) &= \prod_{i=n}^{1} \pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}(y_i \mid c_i) \\
&= \prod_{i=n}^{1} \begin{cases} Q_\Omega^{\boldsymbol{w}}[i, c_i] & \text{if } y_i = 1 \\ 1 - Q_\Omega^{\boldsymbol{w}}[i, c_i] & \text{if } y_i = 0 \end{cases} \\
&= \prod_{i=n}^{1} \frac{\exp\left((y_i\theta_i + V_\Omega^{\boldsymbol{w}}[i-1, c_{i-1}])/\gamma\right)}{\exp(V_\Omega^{\boldsymbol{w}}[i, c_i]/\gamma)}.
\end{aligned}$$

Note that the term $V_\Omega^{\boldsymbol{w}}[i-1, c_{i-1}]$ in the numerator of step i cancels with the denominator term $\exp(V_\Omega^{\boldsymbol{w}}[i-1, c_{i-1}]/\gamma)$ of step $i-1$. This telescoping product yields:

$$\begin{aligned}
\pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}(\boldsymbol{y}) &= \frac{\exp(V_\Omega^{\boldsymbol{w}}[0, c_0]/\gamma)}{\exp(V_\Omega^{\boldsymbol{w}}[n, C]/\gamma)} \prod_{i=n}^{1} \exp(y_i\theta_i/\gamma) \\
&\propto \exp\left(\frac{\sum_{i=1}^{n} y_i\theta_i}{\gamma}\right) = \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle/\gamma).
\end{aligned}$$

□

A.5 Proof of Lemma 1

Proof. Let $x, c \in \mathbb{R}$. We have:

$$\begin{aligned}
\mathsf{max}_\Omega(x + c, 0) - c &= \max_{p \in [0,1]} \left\langle \begin{pmatrix} x + c \\ 0 \end{pmatrix}, \begin{pmatrix} p \\ 1 - p \end{pmatrix} \right\rangle - \Omega\left(\begin{pmatrix} p \\ 1 - p \end{pmatrix}\right) - c \\
&= \max_{p \in [0,1]} (x + c)p - \omega(p) - \omega(1 - p) - c \\
&= \max_{p \in [0,1]} xp + c(p - 1) - \omega(p) - \omega(1 - p).
\end{aligned}$$

Define:

$$h_c(p) \triangleq xp + c(p - 1) - \omega(p) - \omega(1 - p).$$

Since ω is lower semi-continuous, $p \mapsto \omega(p) + \omega(1 - p)$ also is lower semi-continuous. Thus, h_c is upper semi-continuous, as the difference between a continuous function and a lower semi-continuous one. The quantity $\max_{p \in [0,1]} h_c(p)$ is thus well-defined by Weierstrass' extreme value theorem, and h_c attains its maximum on $[0, 1]$. We are computing $\ell \triangleq \lim_{c \to +\infty} \max_{p \in [0,1]} h_c(p)$.

Lower bound. By definition of the maximum, we have:

$$\begin{aligned}
\max_{p \in [0,1]} h_c(p) &\geq h_c(1) \\
&= x - \omega(1) - \omega(0) \\
&= x - \Omega(\boldsymbol{e}_1) \\
&= x.
\end{aligned}$$

This gives us a lower bound $\ell \geq x$ for the limit.

Upper bound. We now show that $\limsup_{c\to+\infty} \max_{p\in[0,1]} h_c(p) \leq x$. For any $c \in \mathbb{R}$, let $p_c^\star \in \mathsf{argmax}_{p\in[0,1]} h_c(p)$, which is well-defined by upper semi-continuity of h_c. First, we show that $p_c^\star \xrightarrow[c\to+\infty]{} 1$. From the lower bound, we know $h_c(p_c^\star) \geq h_c(1) = x$. Substituting the definition of h_c, we have:

$$\begin{aligned} & xp_c^\star + c(p_c^\star - 1) - \omega(p_c^\star) - \omega(1 - p_c^\star) \geq x \\ \iff & c(p_c^\star - 1) \geq x(1 - p_c^\star) + \omega(p_c^\star) + \omega(1 - p_c^\star). \end{aligned}$$

Since the function $\alpha(p) \triangleq x(1-p) + \omega(p) + \omega(1-p)$ is lower semi-continuous on the compact set $[0,1]$, it is bounded below and attains its minimum. Thus, we have $c(p_c^\star - 1) \geq K$, where $K \triangleq \min_{p\in[0,1]} \alpha(p) > -\infty$.

Since $p_c^\star \in [0,1]$, we have $p_c^\star - 1 \leq 0$, giving:

$$\frac{K}{c} \leq p_c^\star - 1 \leq 0, \quad \text{(for } c > 0\text{)}$$

which implies $p_c^\star - 1 \xrightarrow[c\to+\infty]{} 0$, i.e., $\lim_{c\to+\infty} p_c^\star = 1$.

We now find the limit of $h_c(p_c^\star)$. Define $\beta(p) \triangleq xp - \omega(p) - \omega(1-p)$. Since ω is lower semi-continuous, β is upper semi-continuous. We can write $h_c(p) = \beta(p) + c(p-1)$. Since $p_c^\star \in [0,1]$, we have $c(p_c^\star - 1) \leq 0$ for $c \geq 0$, giving the inequality:

$$h_c(p_c^\star) = \beta(p_c^\star) + c(p_c^\star - 1) \leq \beta(p_c^\star).$$

Now we can take the $\limsup$:

$$\limsup_{c\to+\infty} h_c(p_c^\star) \leq \limsup_{c\to+\infty} \beta(p_c^\star).$$

By upper semi-continuity of β, since $\lim_{c\to+\infty} p_c^\star = 1$, we have:

$$\limsup_{c\to+\infty} \beta(p_c^\star) \leq \beta(1) = x - \omega(1) - \omega(0) = x - \Omega(\boldsymbol{e}_1) = x.$$

Since we have $\liminf_{c\to+\infty} h_c(p_c^\star) \geq x$ from the lower bound and $\limsup_{c\to+\infty} h_c(p_c^\star) \leq x$, we conclude that the limit exists and is equal to x:

$$\ell = \lim_{c\to+\infty} \max_{p\in[0,1]} h_c(p) = \lim_{c\to+\infty} h_c(p_c^\star) = x.$$

□

A.6 Proof of Lemma 2

Proof. • ($\implies$) Assume that $f \circ \sigma = f$ for all $\sigma \in S_n$. Fix $\sigma \in S_n$ and $\boldsymbol{x} \in \mathbb{R}^n$. For any direction $\boldsymbol{v} \in \mathbb{R}^n$, by the chain rule we have:

$$\langle \nabla(f \circ \sigma)(\boldsymbol{x}), \boldsymbol{v} \rangle = \langle \nabla f(\sigma(\boldsymbol{x})), \nabla\sigma(\boldsymbol{x})\boldsymbol{v} \rangle.$$

Since σ acts as a linear operator, $\nabla\sigma(\boldsymbol{x})$ acts simply as the permutation σ itself, giving:

$$\langle \nabla(f \circ \sigma)(\boldsymbol{x}), \boldsymbol{v} \rangle = \langle \nabla f(\sigma(\boldsymbol{x})), \sigma(\boldsymbol{v}) \rangle.$$

Since by assumption $f \circ \sigma = f$, we have $\nabla(f \circ \sigma)(\boldsymbol{x}) = \nabla f(\boldsymbol{x})$. Therefore:

$$\begin{aligned} \langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle &= \langle \nabla f(\sigma(\boldsymbol{x})), \sigma(\boldsymbol{v}) \rangle \\ &= \langle \sigma^{-1}(\nabla f(\sigma(\boldsymbol{x}))), \boldsymbol{v} \rangle. \end{aligned}$$

Since this holds for all $\boldsymbol{v} \in \mathbb{R}^n$, we get in fact:

$$\begin{aligned} & \forall \boldsymbol{x} \in \mathbb{R}^n, \nabla f(\boldsymbol{x}) = \sigma^{-1}(\nabla f(\sigma(\boldsymbol{x}))) \\ \iff & \forall \boldsymbol{x} \in \mathbb{R}^n, \sigma(\nabla f(\boldsymbol{x})) = \nabla f(\sigma(\boldsymbol{x})). \end{aligned}$$

Thus, $\nabla f \circ \sigma = \sigma \circ \nabla f$ for all $\sigma \in S_n$.

• ($\Longleftarrow$) Conversely, assume that $\nabla f \circ \sigma = \sigma \circ \nabla f$ for all $\sigma \in S_n$. Fix $\sigma \in S_n$ and define:

$$g(\boldsymbol{x}) \triangleq (f \circ \sigma)(\boldsymbol{x}) - f(\boldsymbol{x}).$$

Then, by the chain rule:

$$\nabla g(\boldsymbol{x}) = \sigma^{-1}(\nabla f(\sigma(\boldsymbol{x}))) - \nabla f(\boldsymbol{x}).$$

Using the equivariance assumption $\nabla f(\sigma(\boldsymbol{x})) = \sigma(\nabla f(\boldsymbol{x}))$, we have:

$$\begin{aligned}\nabla g(\boldsymbol{x}) &= \sigma^{-1}(\nabla f(\sigma(\boldsymbol{x}))) - \nabla f(\boldsymbol{x}) \\ &= \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}) = 0.\end{aligned}$$

Hence, $\nabla g(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$, which implies that g is constant. To determine this constant, note that we clearly have $g(\mathbf{0}) = 0$, since $\sigma(\mathbf{0}) = \mathbf{0}$. Therefore:

$$f(\sigma(\boldsymbol{x})) = f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^n.$$

Thus, $f \circ \sigma = f$ for all $\sigma \in S_n$. □

A.7 Proof of Lemma 3

Proof. We proceed by induction on i. For $i = 0$, the feasible states are $c \in \{0, \dots, C\}$ (Knapsack) or $c = 0$ (Top-k), where $V_\Omega^{\boldsymbol{w}}[0, c] = V^{\boldsymbol{w}}[0, c] = 0$. Assume the bound holds at $i - 1 \in \{0, \dots, n - 1\}$, and consider a feasible state (i, c).

If $w_i > c$ (which includes $c = 0$ in Top-k), $V_\Omega^{\boldsymbol{w}}[i, c] = V_\Omega^{\boldsymbol{w}}[i - 1, c]$ and $V^{\boldsymbol{w}}[i, c] = V^{\boldsymbol{w}}[i - 1, c]$, so the error remains bounded by $(i - 1)M_\Omega \leq iM_\Omega$. In Top-$k$ when $c = i$, the skip parent $(i - 1, i)$ equals $-\infty$, so $V_\Omega^{\mathbf{1}}[i, i] = \theta_i + V_\Omega^{\mathbf{1}}[i - 1, i - 1] - \Omega(\boldsymbol{e}_1)$ while $V^{\mathbf{1}}[i, i] = \theta_i + V^{\mathbf{1}}[i - 1, i - 1]$, giving $|V_\Omega^{\mathbf{1}}[i, i] - V^{\mathbf{1}}[i, i]| \leq (i - 1)M_\Omega + |\Omega(\boldsymbol{e}_1)| \leq iM_\Omega$.

Otherwise, both $a = \theta_i + V^{\boldsymbol{w}}[i - 1, c - w_i]$ and $b = V^{\boldsymbol{w}}[i - 1, c]$ are finite, and we have that $V_\Omega^{\boldsymbol{w}}[i, c] = \mathsf{max}_\Omega(a + \epsilon_a, b + \epsilon_b)$ with $|\epsilon_a|, |\epsilon_b| \leq (i - 1)M_\Omega$ by the induction hypothesis. By the triangle inequality, $|\mathsf{max}_\Omega(\boldsymbol{x}) - \max(\boldsymbol{x})| \leq M_\Omega$, and 1-Lipschitz continuity of $\max$ in $\|\cdot\|_\infty$:

$$\begin{aligned}|V_\Omega^{\boldsymbol{w}}[i, c] - V^{\boldsymbol{w}}[i, c]| &\leq |\mathsf{max}_\Omega(a + \epsilon_a, b + \epsilon_b) - \max(a + \epsilon_a, b + \epsilon_b)| \\ &\quad + |\max(a + \epsilon_a, b + \epsilon_b) - \max(a, b)| \\ &\leq M_\Omega + \max(|\epsilon_a|, |\epsilon_b|) \\ &\leq M_\Omega + (i - 1)M_\Omega \\ &= iM_\Omega.\end{aligned}$$

□

B Algorithmic Derivations

B.1 Backtracking as a backward pass

We now turn to the derivation of the backward pass, which enables to compute $\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) = \nabla\mathsf{max}^C_{\boldsymbol{w}}(\boldsymbol{\theta})$ and $\boldsymbol{y}^k_{\mathbf{1}}(\boldsymbol{\theta}) = \nabla\mathsf{max}^k_{\mathbf{1}}(\boldsymbol{\theta})$. In this unregularized case, we recover the idea of backtracking. We will write everything in the Knapsack case, as everything holds for the Top-k case by letting $C = k$ and $\boldsymbol{w} = \mathbf{1}$ (the slight change due to the difference in padding values of the DP table does not impact differentiation).

Let $E^{\boldsymbol{w}}[i, c] \triangleq \frac{\partial V^{\boldsymbol{w}}[n,C]}{\partial V^{\boldsymbol{w}}[i,c]}$ be the main object of interest in this derivation, and define:

$$\begin{aligned} Q^{\boldsymbol{w}}[i, c] &\triangleq \frac{\partial V^{\boldsymbol{w}}[i, c]}{\partial \theta_i} \\ &= \begin{cases} 0 \quad \text{if } w_i > c, \\ \mathbb{1}\left(\{\max\left(\theta_i + V^{\boldsymbol{w}}[i-1, c-w_i]\right., \left. V^{\boldsymbol{w}}[i-1, c]\right) = \theta_i + V^{\boldsymbol{w}}[i-1, c-w_i]\}\right) \quad \text{else,} \end{cases} \end{aligned}$$

so that the matrix $\boldsymbol{Q}^{\boldsymbol{w}}$ stores a track of the maximizers in Equation (3) during the forward pass (we have $Q^{\boldsymbol{w}}[i, c] = 1 \iff$ item i is needed to reach value $V^{\boldsymbol{w}}[i, c]$). As $V^{\boldsymbol{w}}[i, c]$ only directly influences $V^{\boldsymbol{w}}[i+1, c]$ and $V^{\boldsymbol{w}}[i+1, c+w_{i+1}]$, we have:

$$\begin{aligned} E^{\boldsymbol{w}}[i, c] &= \frac{\partial V^{\boldsymbol{w}}[n, C]}{\partial V^{\boldsymbol{w}}[i+1, c]} \times \frac{\partial V^{\boldsymbol{w}}[i+1, c]}{\partial V^{\boldsymbol{w}}[i, c]} + \frac{\partial V^{\boldsymbol{w}}[n, C]}{\partial V^{\boldsymbol{w}}[i+1, c+w_{i+1}]} \times \frac{\partial V^{\boldsymbol{w}}[i+1, c+w_{i+1}]}{\partial V^{\boldsymbol{w}}[i, c]} \\ &= E^{\boldsymbol{w}}[i+1, c] \times \frac{\partial V^{\boldsymbol{w}}[i+1, c]}{\partial V^{\boldsymbol{w}}[i, c]} + E^{\boldsymbol{w}}[i+1, c+w_{i+1}] \times \frac{\partial V^{\boldsymbol{w}}[i+1, c+w_{i+1}]}{\partial V^{\boldsymbol{w}}[i, c]}. \end{aligned}$$

Then, from the recursion in Equation (3) (for $c \geq 0$, with $c + w_{i+1} \leq C$ for the second line), we have:

$$\begin{aligned} V^{\boldsymbol{w}}[i+1, c] &= \begin{cases} V^{\boldsymbol{w}}[i, c] & \text{if } w_{i+1} > c, \\ \max\left(\theta_{i+1} + V^{\boldsymbol{w}}[i, c-w_{i+1}]\right., \left. V^{\boldsymbol{w}}[i, c]\right) & \text{else,} \end{cases} \\ V^{\boldsymbol{w}}[i+1, c+w_{i+1}] &= \max\left(\theta_{i+1} + V^{\boldsymbol{w}}[i, c]\right., \left. V^{\boldsymbol{w}}[i, c+w_{i+1}]\right), \end{aligned}$$

which further gives:

$$\begin{aligned} \frac{\partial V^{\boldsymbol{w}}[i+1, c]}{\partial V^{\boldsymbol{w}}[i, c]} &= \begin{cases} 1 \quad \text{if } w_{i+1} > c, \\ \mathbb{1}\left(\{\max\left(\theta_{i+1} + V^{\boldsymbol{w}}[i, c-w_{i+1}], V^{\boldsymbol{w}}[i, c]\right) = V^{\boldsymbol{w}}[i, c]\}\right) \quad \text{else,} \end{cases} \\ &= 1 - Q^{\boldsymbol{w}}[i+1, c], \\ \frac{\partial V^{\boldsymbol{w}}[i+1, c+w_{i+1}]}{\partial V^{\boldsymbol{w}}[i, c]} &= \mathbb{1}\left(\{\max\left(\theta_{i+1} + V^{\boldsymbol{w}}[i, c], V^{\boldsymbol{w}}[i, c+w_{i+1}]\right) = \theta_{i+1} + V^{\boldsymbol{w}}[i, c]\}\right) \\ &= Q^{\boldsymbol{w}}[i+1, c+w_{i+1}]. \end{aligned}$$

Thus, we finally have the following backward recursion:

$$E^{\boldsymbol{w}}[i, c] = E^{\boldsymbol{w}}[i+1, c] \times (1 - Q^{\boldsymbol{w}}[i+1, c]) + E^{\boldsymbol{w}}[i+1, c+w_{i+1}] \times Q^{\boldsymbol{w}}[i+1, c+w_{i+1}].$$

The recursion is initialized with $E^{\boldsymbol{w}}[n, C] = \frac{\partial V^{\boldsymbol{w}}[n,C]}{\partial V^{\boldsymbol{w}}[n,C]} = 1$, and $\forall c < C,\ E^{\boldsymbol{w}}[n, c] = 0$.

The hard maximizing mask $\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) = \nabla\mathsf{max}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) = \frac{\partial V^{\boldsymbol{w}}[n,C]}{\partial \boldsymbol{\theta}}$ is then recovered by noting that since θ_i only directly influences the row $V^{\boldsymbol{w}}[i, :]$, we have:

$$\begin{aligned} \frac{\partial V^{\boldsymbol{w}}[n, C]}{\partial \theta_i} &= \sum_{c=0}^{C} \frac{\partial V^{\boldsymbol{w}}[n, C]}{\partial V^{\boldsymbol{w}}[i, c]} \times \frac{\partial V^{\boldsymbol{w}}[i, c]}{\partial \theta_i} \\ &= \sum_{c=0}^{C} E^{\boldsymbol{w}}[i, c] \times Q^{\boldsymbol{w}}[i, c], \end{aligned}$$

giving the following compact expression for the maximizing mask:

$$\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) = (\boldsymbol{E}^{\boldsymbol{w}}[1\!:\!,:] \circ \boldsymbol{Q}^{\boldsymbol{w}}[1\!:\!,:]) \cdot \mathbf{1},$$

where $\circ$ denotes the Hadamard product and $\mathbf{1} \in \mathbb{R}^{C+1}$ is the all-ones vector.

B.2 Derivation of the layer

We now turn to the derivation of the backward pass to compute the relaxed layers. We will write everything in the Knapsack case, with $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$. All results also hold for the Top-k case, by setting $\boldsymbol{w} = \mathbf{1}$ and $C = k$ (the only slight change due to the difference in the padding values for $\boldsymbol{V}^{\boldsymbol{w}}_\Omega$ does not impact differentiation).

We first recall here the smoothed recursion given in Equation (4):

$$V^{\boldsymbol{w}}_\Omega[i,c] = \begin{cases} V^{\boldsymbol{w}}_\Omega[i-1,c] & \text{if } w_i > c, \\ \mathsf{max}_\Omega\left(\theta_i + V^{\boldsymbol{w}}_\Omega[i-1,c-w_i],\ V^{\boldsymbol{w}}_\Omega[i-1,c]\right) & \text{else.} \end{cases}$$

Define the following objects:

$$E^{\boldsymbol{w}}_\Omega[i,c] \triangleq \frac{\partial V^{\boldsymbol{w}}_\Omega[n,C]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]},$$
$$Q^{\boldsymbol{w}}_\Omega[i,c] \triangleq \frac{\partial V^{\boldsymbol{w}}_\Omega[i,c]}{\partial \theta_i}.$$

The matrix $\boldsymbol{Q}^{\boldsymbol{w}}_\Omega$ is computed and stored during the forward pass (see Algorithm 1). As we have

$$\nabla_{\boldsymbol{x}} \mathsf{max}_\Omega(x_1,x_2) \in \triangle^2, \quad \text{i.e.,} \quad \frac{\partial \mathsf{max}_\Omega(x_1,x_2)}{\partial x_1} + \frac{\partial \mathsf{max}_\Omega(x_1,x_2)}{\partial x_2} = 1,$$

one can easily check that we have:

$$\frac{\partial V^{\boldsymbol{w}}_\Omega[i,c]}{\partial V^{\boldsymbol{w}}_\Omega[i-1,c]} = 1 - Q^{\boldsymbol{w}}_\Omega[i,c]\,, \tag{6}$$

$$\frac{\partial V^{\boldsymbol{w}}_\Omega[i,c]}{\partial V^{\boldsymbol{w}}_\Omega[i-1,c-w_i]} = \frac{\partial V^{\boldsymbol{w}}_\Omega[i,c]}{\partial \theta_i} = Q^{\boldsymbol{w}}_\Omega[i,c]. \tag{7}$$

These identities will prove useful in the following. We can now derive the computation of $E^{\boldsymbol{w}}_\Omega[i,c]$. As $V^{\boldsymbol{w}}_\Omega[i,c]$ only directly influences $V^{\boldsymbol{w}}_\Omega[i+1,c]$ and $V^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]$, we have (similarly to the unregularized case, see Section B.1):

$$E^{\boldsymbol{w}}_\Omega[i,c] = E^{\boldsymbol{w}}_\Omega[i+1,c] \times \frac{\partial V^{\boldsymbol{w}}_\Omega[i+1,c]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]} + E^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}] \times \frac{\partial V^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]}.$$

Moreover, using Equations (6) and (7), we have:

$$\frac{\partial V^{\boldsymbol{w}}_\Omega[i+1,c]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]} = 1 - Q^{\boldsymbol{w}}_\Omega[i+1,c]\,,$$
$$\frac{\partial V^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]} = Q^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]\,.$$

Thus, we have the following backward recursion:

$$E^{\boldsymbol{w}}_\Omega[i,c] = E^{\boldsymbol{w}}_\Omega[i+1,c] \times (1 - Q^{\boldsymbol{w}}_\Omega[i+1,c]) + E^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}] \times Q^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]. \tag{8}$$

The recursion is initialized with $E^{\boldsymbol{w}}_\Omega[n,C] = \frac{\partial V^{\boldsymbol{w}}_\Omega[n,C]}{\partial V^{\boldsymbol{w}}_\Omega[n,C]} = 1$ and $\forall c < C,\ E^{\boldsymbol{w}}_\Omega[n,c] = 0$. The gradient $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \frac{\partial V^{\boldsymbol{w}}_\Omega[n,C]}{\partial \boldsymbol{\theta}}$ is then recovered by noting that, since θ_i only directly influences the row $V^{\boldsymbol{w}}_\Omega[i,:]$, we have:

$$\frac{\partial V^{\boldsymbol{w}}_\Omega[n,C]}{\partial \theta_i} = \sum_{c=0}^{C} \frac{\partial V^{\boldsymbol{w}}_\Omega[n,C]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]} \times \frac{\partial V^{\boldsymbol{w}}_\Omega[i,c]}{\partial \theta_i} = \sum_{c=0}^{C} E^{\boldsymbol{w}}_\Omega[i,c] \times Q^{\boldsymbol{w}}_\Omega[i,c].$$

Finally, we get the following compact expression:

$$\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = (\boldsymbol{E}^{\boldsymbol{w}}_{\Omega}[1:,:] \circ \boldsymbol{Q}^{\boldsymbol{w}}_{\Omega}[1:,:]) \cdot \mathbf{1},$$

where $\circ$ denotes the Hadamard product and $\mathbf{1} \in \mathbb{R}^{C+1}$ is the all-ones vector.

B.3 Derivation of the directional derivative

We now turn to the computation of the directional derivative $\langle \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}),\, \boldsymbol{z}\rangle = \langle \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}),\, \boldsymbol{z}\rangle$ of the soft value in direction $\boldsymbol{z} \in \mathbb{R}^n$ (i.e., the inner product between the layer and a cotangent vector $\boldsymbol{z}$). We will write everything in the Knapsack case, with $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$. All results also hold for the Top-k case, by setting $\boldsymbol{w} = \mathbf{1}$ and $C = k$ (the only slight change due to the difference in the padding values for $\boldsymbol{V}^{\boldsymbol{w}}_{\Omega}$ does not impact differentiation).

Let $\boldsymbol{z} \in \mathbb{R}^n$ be a direction. We define here:

$$\dot{V}^{\boldsymbol{w}}_{\Omega}[i,c] \triangleq \left\langle \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial \boldsymbol{\theta}}, \boldsymbol{z} \right\rangle.$$

Since $V^{\boldsymbol{w}}_{\Omega}[i,c]$ only directly depends on $V^{\boldsymbol{w}}_{\Omega}[i-1, c-w_i]$, $V^{\boldsymbol{w}}_{\Omega}[i-1,c]$ and θ_i, we have:

$$\begin{aligned}
\dot{V}^{\boldsymbol{w}}_{\Omega}[i,c] &= \left\langle \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial \boldsymbol{\theta}}, \boldsymbol{z} \right\rangle \\
&= \left\langle \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial V^{\boldsymbol{w}}_{\Omega}[i-1,c-w_i]} \times \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i-1,c-w_i]}{\partial \boldsymbol{\theta}} \right. \\
&\quad \left. + \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial V^{\boldsymbol{w}}_{\Omega}[i-1,c]} \times \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i-1,c]}{\partial \boldsymbol{\theta}} + \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial \theta_i} \times \frac{\partial \theta_i}{\partial \boldsymbol{\theta}}\,,\, \boldsymbol{z} \right\rangle \\
&= \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial V^{\boldsymbol{w}}_{\Omega}[i-1,c-w_i]} \times \dot{V}^{\boldsymbol{w}}_{\Omega}[i-1,c-w_i] \\
&\quad + \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial V^{\boldsymbol{w}}_{\Omega}[i-1,c]} \times \dot{V}^{\boldsymbol{w}}_{\Omega}[i-1,c] + \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial \theta_i} \times z_i.
\end{aligned}$$

Then, using Equations (6) and (7) gives the following forward recursion:

$$\dot{V}^{\boldsymbol{w}}_{\Omega}[i,c] = \left(\dot{V}^{\boldsymbol{w}}_{\Omega}[i-1,c-w_i] + z_i\right) \times Q^{\boldsymbol{w}}_{\Omega}[i,c] + \dot{V}^{\boldsymbol{w}}_{\Omega}[i-1,c] \times (1 - Q^{\boldsymbol{w}}_{\Omega}[i,c]), \tag{9}$$

which is initialized with $\dot{V}^{\boldsymbol{w}}_{\Omega}[0,:] = 0$, and ends at $\dot{V}^{\boldsymbol{w}}_{\Omega}[n,C] = \left\langle \frac{\partial V^{\boldsymbol{w}}_{\Omega}[n,C]}{\partial \boldsymbol{\theta}}, \boldsymbol{z} \right\rangle = \langle \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}), \boldsymbol{z}\rangle$.

B.4 Derivation of the vector-Jacobian product

Here, we derive the computation of the vector-Jacobian product $\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))$ for a given cotangent vector $\boldsymbol{z} \in \mathbb{R}^n$. As stated in Section 5.1, since we have $\boldsymbol{y}^C_{\boldsymbol{w},\Omega} = \nabla \mathsf{max}^C_{\boldsymbol{w},\Omega}$, the Jacobian $\nabla \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ is equal to the Hessian $\nabla^2 \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$, so it is in fact symmetric for all $\boldsymbol{\theta} \in \mathbb{R}^n$. Thus, we can view the VJP $\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))$ as the corresponding JVP $(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))\boldsymbol{z}$, for any tangent/cotangent vector $\boldsymbol{z} \in \mathbb{R}^n$. Further, we use the fact that $(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))\boldsymbol{z} = \nabla_{\boldsymbol{\theta}} \langle \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}), \boldsymbol{z}\rangle$, and focus on the differentiation of the directional derivative $\langle \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}), \boldsymbol{z}\rangle = \dot{V}^{\boldsymbol{w}}_{\Omega}[n,C]$. Thus, our goal is to compute $\frac{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[n,C]}{\partial \boldsymbol{\theta}}$.

As θ_i only directly influences the rows $V^{\boldsymbol{w}}_{\Omega}[i,:]$ and $\dot{V}^{\boldsymbol{w}}_{\Omega}[i,:]$, we have:

$$\frac{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[n,C]}{\partial \theta_i} = \sum_{c=0}^{C} \frac{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[n,C]}{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]} \times \frac{\partial V^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial \theta_i} + \frac{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[n,C]}{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[i,c]} \times \frac{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[i,c]}{\partial \theta_i}. \tag{10}$$

We now turn to the derivation of every element in Equation (10) needed to compute $\frac{\partial \dot{V}^{\boldsymbol{w}}_{\Omega}[n,C]}{\partial \theta_i}$.

To begin with, the second term is already computed as we simply have $\frac{\partial V_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i} = Q_\Omega^{\boldsymbol{w}}[i,c]$.

Now, we compute the third term $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]}$. As $\dot{V}_\Omega^{\boldsymbol{w}}[i,c]$ only directly influences $\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]$ and $\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]$, we have the recursion:

$$\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]} = \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]} \times \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]} + \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]} \times \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]}.$$

However, Equation (9) gives:

$$\begin{aligned}\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c] = &\left(\dot{V}_\Omega^{\boldsymbol{w}}[i,c-w_{i+1}] + z_{i+1}\right) \times Q_\Omega^{\boldsymbol{w}}[i+1,c] \\ &+ \dot{V}_\Omega^{\boldsymbol{w}}[i,c] \times (1 - Q_\Omega^{\boldsymbol{w}}[i+1,c]),\end{aligned} \tag{11}$$

$$\begin{aligned}\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] = &\left(\dot{V}_\Omega^{\boldsymbol{w}}[i,c] + z_{i+1}\right) \times Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \\ &+ \dot{V}_\Omega^{\boldsymbol{w}}[i,c+w_{i+1}] \times (1 - Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]).\end{aligned} \tag{12}$$

Thus, as $\dot{V}_\Omega^{\boldsymbol{w}}[i,c]$ only directly influences $\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]$ and $\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]$, we get, by differentiating Equations (11) and (12):

$$\begin{aligned}\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]} &= 1 - Q_\Omega^{\boldsymbol{w}}[i+1,c], \\ \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]} &= Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}].\end{aligned}$$

Thus, we have in fact the following recursion:

$$\begin{aligned}\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]} = &\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]} \times (1 - Q_\Omega^{\boldsymbol{w}}[i+1,c]) \\ &+ \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]} \times Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}],\end{aligned}$$

and $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]}$ is defined by the exact same recursion as $E_\Omega^{\boldsymbol{w}}[i,c]$, which is given by Equation (8). As they are both initialized by $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]} = E_\Omega^{\boldsymbol{w}}[n,C] = 1$ and $\forall c < C$, $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,c]} = E_\Omega^{\boldsymbol{w}}[n,c] = 0$, the two sequences are in fact equal:

$$\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]} = E_\Omega^{\boldsymbol{w}}[i,c].$$

Now, we derive the red term $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i}$. As θ_i does not influence $\dot{V}_\Omega^{\boldsymbol{w}}[i-1,c-w_i]$, $\dot{V}_\Omega^{\boldsymbol{w}}[i-1,c]$, $V_\Omega^{\boldsymbol{w}}[i-1,c-w_i]$, or $V_\Omega^{\boldsymbol{w}}[i-1,c]$, the differentiation of the forward recursion in Equation (9) gives:

$$\begin{aligned}\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i} &= \left(\dot{V}_\Omega^{\boldsymbol{w}}[i-1,c-w_i] + z_i\right) \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i} - \dot{V}_\Omega^{\boldsymbol{w}}[i-1,c] \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i} \\ &= \left(\dot{V}_\Omega^{\boldsymbol{w}}[i-1,c-w_i] - \dot{V}_\Omega^{\boldsymbol{w}}[i-1,c] + z_i\right) \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i}.\end{aligned}$$

Finally, for the orange term, let us define:

$$\dot{E}_\Omega^{\boldsymbol{w}}[i,c] \triangleq \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}.$$

As $V_\Omega^{\boldsymbol{w}}[i,c]$ only directly influences $V_\Omega^{\boldsymbol{w}}[i+1,c]$, $V_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]$, $\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]$ and $\dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]$, we have:

$$\begin{aligned}
\dot{E}_\Omega^{\boldsymbol{w}}[i,c] &= \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} \\
&= \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial V_\Omega^{\boldsymbol{w}}[i+1,c]} \times \frac{\partial V_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} + \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial V_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]} \times \frac{\partial V_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} \\
&+ \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]} \times \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} + \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]} \times \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} \\
&= \dot{E}_\Omega^{\boldsymbol{w}}[i+1,c] \times (1 - Q_\Omega^{\boldsymbol{w}}[i+1,c]) + \dot{E}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \times Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \\
&+ E_\Omega^{\boldsymbol{w}}[i+1,c] \times \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} + E_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \times \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}.
\end{aligned}$$

Thus, we need to compute the two terms $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}$ and $\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}$ in order to get a recursive formulation of $\dot{E}_\Omega^{\boldsymbol{w}}[i,c] = \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}$.

As $V_\Omega^{\boldsymbol{w}}[i,c]$ does not influence $\dot{V}_\Omega^{\boldsymbol{w}}[i,c-w_{i+1}]$, $\dot{V}_\Omega^{\boldsymbol{w}}[i,c]$ or z_{i+1}, differentiating Equation (11) gives on the one hand:

$$\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} = \left(\dot{V}_\Omega^{\boldsymbol{w}}[i,c-w_{i+1}] - \dot{V}_\Omega^{\boldsymbol{w}}[i,c] + z_{i+1}\right) \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}.$$

On the other hand, as $V_\Omega^{\boldsymbol{w}}[i,c]$ does not influence $\dot{V}_\Omega^{\boldsymbol{w}}[i,c]$ or $\dot{V}_\Omega^{\boldsymbol{w}}[i,c+w_{i+1}]$, differentiating Equation (12) gives:

$$\frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} = \left(\dot{V}_\Omega^{\boldsymbol{w}}[i,c] - \dot{V}_\Omega^{\boldsymbol{w}}[i,c+w_{i+1}] + z_{i+1}\right) \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}.$$

Thus, $\dot{E}_\Omega^{\boldsymbol{w}}[i,c] = \frac{\partial \dot{V}_\Omega^{\boldsymbol{w}}[n,C]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}$ is given by the following backward recursion:

$$\begin{aligned}
\dot{E}_\Omega^{\boldsymbol{w}}[i,c] &= \dot{E}_\Omega^{\boldsymbol{w}}[i+1,c] \times (1 - Q_\Omega^{\boldsymbol{w}}[i+1,c]) + \dot{E}_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \times Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \\
&+ E_\Omega^{\boldsymbol{w}}[i+1,c] \times \left(\dot{V}_\Omega^{\boldsymbol{w}}[i,c-w_{i+1}] - \dot{V}_\Omega^{\boldsymbol{w}}[i,c] + z_{i+1}\right) \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i+1,c]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]} \\
&+ E_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}] \times \left(\dot{V}_\Omega^{\boldsymbol{w}}[i,c] - \dot{V}_\Omega^{\boldsymbol{w}}[i,c+w_{i+1}] + z_{i+1}\right) \times \frac{\partial Q_\Omega^{\boldsymbol{w}}[i+1,c+w_{i+1}]}{\partial V_\Omega^{\boldsymbol{w}}[i,c]}.
\end{aligned}$$

Thus, to summarize our results, we have (for any choice of Ω):

$$
\begin{aligned}
\frac{\partial \dot{V}_{\Omega}^{\boldsymbol{w}}[n, C]}{\partial V_{\Omega}^{\boldsymbol{w}}[i, c]} &= \dot{E}_{\Omega}^{\boldsymbol{w}}[i, c] \\
&= \dot{E}_{\Omega}^{\boldsymbol{w}}[i+1, c] \times (1 - Q_{\Omega}^{\boldsymbol{w}}[i+1, c]) + \dot{E}_{\Omega}^{\boldsymbol{w}}[i+1, c+w_{i+1}] \times Q_{\Omega}^{\boldsymbol{w}}[i+1, c+w_{i+1}] \\
&+ E_{\Omega}^{\boldsymbol{w}}[i+1, c] \times \left(\dot{V}_{\Omega}^{\boldsymbol{w}}[i, c-w_{i+1}] - \dot{V}_{\Omega}^{\boldsymbol{w}}[i, c] + z_{i+1} \right) \times \frac{\partial Q_{\Omega}^{\boldsymbol{w}}[i+1, c]}{\partial V_{\Omega}^{\boldsymbol{w}}[i, c]} \\
&+ E_{\Omega}^{\boldsymbol{w}}[i+1, c+w_{i+1}] \times \left(\dot{V}_{\Omega}^{\boldsymbol{w}}[i, c] - \dot{V}_{\Omega}^{\boldsymbol{w}}[i, c+w_{i+1}] + z_{i+1} \right) \\
&\qquad \times \frac{\partial Q_{\Omega}^{\boldsymbol{w}}[i+1, c+w_{i+1}]}{\partial V_{\Omega}^{\boldsymbol{w}}[i, c]} \,, \\
\frac{\partial V_{\Omega}^{\boldsymbol{w}}[i, c]}{\partial \theta_i} &= Q_{\Omega}^{\boldsymbol{w}}[i, c] \,, \\
\frac{\partial \dot{V}_{\Omega}^{\boldsymbol{w}}[n, C]}{\partial \dot{V}_{\Omega}^{\boldsymbol{w}}[i, c]} &= E_{\Omega}^{\boldsymbol{w}}[i, c] \,, \\
\frac{\partial \dot{V}_{\Omega}^{\boldsymbol{w}}[i, c]}{\partial \theta_i} &= \left(\dot{V}_{\Omega}^{\boldsymbol{w}}[i-1, c-w_i] - \dot{V}_{\Omega}^{\boldsymbol{w}}[i-1, c] + z_i \right) \times \frac{\partial Q_{\Omega}^{\boldsymbol{w}}[i, c]}{\partial \theta_i} \,.
\end{aligned}
$$

C Pseudo-code

Algorithm 4 Computation of the VJP $\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}}\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))$, for general Ω

input Item values $\boldsymbol{\theta} \in \mathbb{R}^n$, item weights $\boldsymbol{w} \in \mathbb{N}^n$, capacity $C \in \mathbb{N}$, regularization function Ω, cotangent vector $\boldsymbol{z} \in \mathbb{R}^n$, outputs $\boldsymbol{V}^{\boldsymbol{w}}_\Omega, \boldsymbol{Q}^{\boldsymbol{w}}_\Omega, \boldsymbol{E}^{\boldsymbol{w}}_\Omega$ of Algorithm 2.
output Vector-Jacobian product $\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}}\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))$.

Initialize $\dot{\boldsymbol{V}}^{\boldsymbol{w}}_\Omega \leftarrow \mathbf{0} \in \mathbb{R}^{(n+1)\times(C+1)}$, $\dot{\boldsymbol{E}}^{\boldsymbol{w}}_\Omega \leftarrow \mathbf{0} \in \mathbb{R}^{(n+1)\times(C+1)}$, and VJP $\leftarrow \mathbf{0} \in \mathbb{R}^n$.
{forward recursion to compute $\dot{\boldsymbol{V}}^{\boldsymbol{w}}_\Omega$}
for $i = 1$ **to** n **do**
  {the loop on c is parallelizable (wavefront computation)}
  for $c = 0$ **to** C **do**
    $$\dot{V}^{\boldsymbol{w}}_\Omega[i,c] \leftarrow \left(\dot{V}^{\boldsymbol{w}}_\Omega[i-1, c-w_i] + z_i\right) \times Q^{\boldsymbol{w}}_\Omega[i,c] + \dot{V}^{\boldsymbol{w}}_\Omega[i-1,c] \times (1 - Q^{\boldsymbol{w}}_\Omega[i,c])$$
  end for
end for
{backward recursion to compute $\dot{\boldsymbol{E}}^{\boldsymbol{w}}_\Omega$}
for $i = n-1$ **to** 1 **do**
  {the loop on c is parallelizable (wavefront computation)}
  for $c = 0$ **to** C **do**
    $$\begin{aligned}\dot{E}^{\boldsymbol{w}}_\Omega[i,c] \leftarrow &\left(\dot{E}^{\boldsymbol{w}}_\Omega[i+1,c] \times (1 - Q^{\boldsymbol{w}}_\Omega[i+1,c])\right)\\ &+ \left(\dot{E}^{\boldsymbol{w}}_\Omega[i+1, c+w_{i+1}] \times Q^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]\right)\\ &+ \left(E^{\boldsymbol{w}}_\Omega[i+1,c] \times \left(\dot{V}^{\boldsymbol{w}}_\Omega[i, c-w_{i+1}] - \dot{V}^{\boldsymbol{w}}_\Omega[i,c] + z_{i+1}\right) \times \frac{\partial Q^{\boldsymbol{w}}_\Omega[i+1,c]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]}\right)\\ &+ \left(E^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}] \times \left(\dot{V}^{\boldsymbol{w}}_\Omega[i,c] - \dot{V}^{\boldsymbol{w}}_\Omega[i,c+w_{i+1}] + z_{i+1}\right)\right.\\ &\qquad \left.\times \frac{\partial Q^{\boldsymbol{w}}_\Omega[i+1,c+w_{i+1}]}{\partial V^{\boldsymbol{w}}_\Omega[i,c]}\right)\end{aligned}$$
  end for
end for
{Final VJP assembly (both the loops on i and c are parallelizable)}
for $i = 1$ **to** n **do**
  for $c = 0$ **to** C **do**
    {colors match the ones used in Section B.4}
    $$\begin{aligned}\mathrm{VJP}[i] \leftarrow \mathrm{VJP}[i] &+ \left(\dot{E}^{\boldsymbol{w}}_\Omega[i,c] \times Q^{\boldsymbol{w}}_\Omega[i,c]\right)\\ &+ \left(E^{\boldsymbol{w}}_\Omega[i,c] \times \left(\dot{V}^{\boldsymbol{w}}_\Omega[i-1,c-w_i] - \dot{V}^{\boldsymbol{w}}_\Omega[i-1,c] + z_i\right) \times \frac{\partial Q^{\boldsymbol{w}}_\Omega[i,c]}{\partial \theta_i}\right)\end{aligned}$$
  end for
end for
$\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}}\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})) \leftarrow \mathrm{VJP}$
return $\boldsymbol{z}^\top(\nabla_{\boldsymbol{\theta}}\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}))$

D Additional Material

D.1 Visualization of the DP recursions

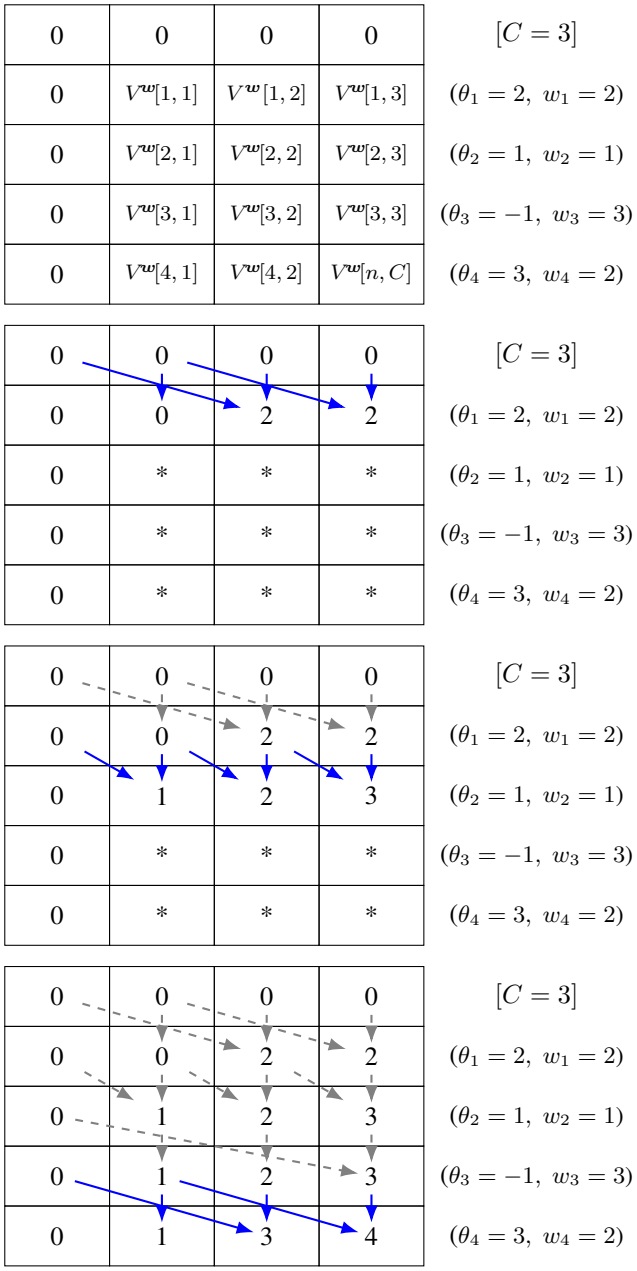


Figure 5: Illustration of the DP recursion table for the Knapsack problem, with $\boldsymbol{\theta} = (2, 1, -1, 3)^\top$, $\boldsymbol{w} = (2, 1, 3, 2)^\top$ and $C = 3$. Arrows indicate the influence of values on others, illustrating the DAG structure underlying the recursion. See Section D.3 for more details.

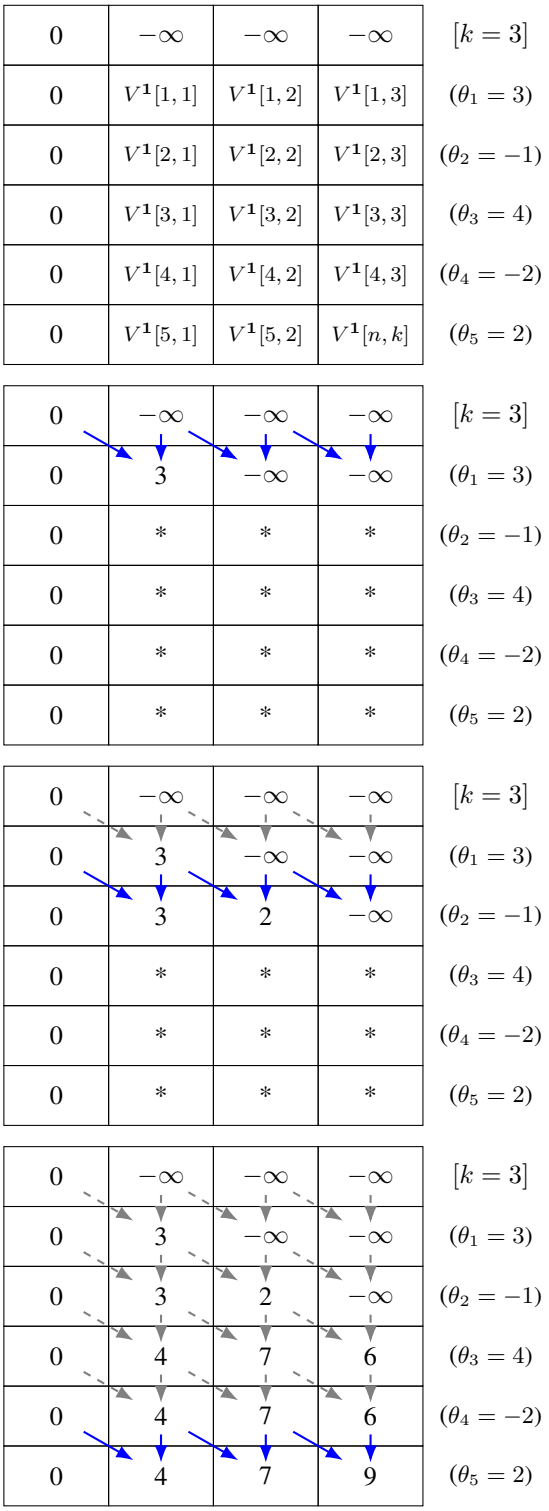


Figure 6: Illustration of the DP recursion table for the Top-k problem, with $\boldsymbol{\theta} = (3, -1, 4, -2, 2)^\top$ and $k = 3$. Arrows indicate the influence of values on others, illustrating the DAG structure underlying the recursion. See Section D.3 for more details.

D.2 Visualization of Propositions 1 and 2

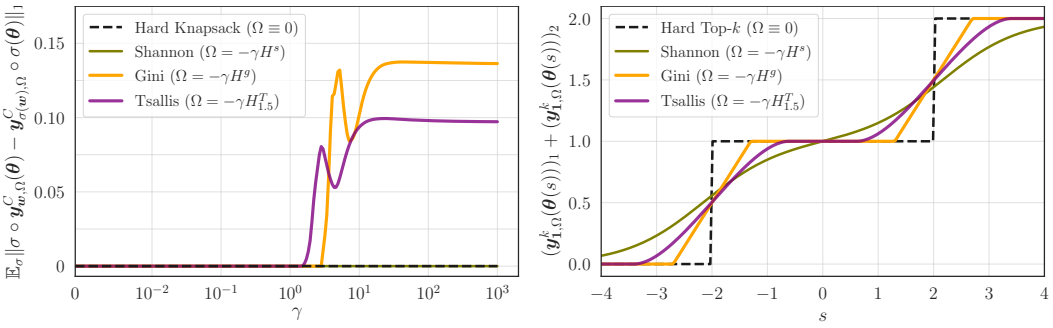


Figure 7: **Illustration of Propositions 1 and 2**. On the **left**, we measure the ℓ_1 distance between $\sigma \circ \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ and $\boldsymbol{y}^C_{\sigma(\boldsymbol{w}),\Omega} \circ \sigma(\boldsymbol{\theta})$, averaged on every possible permutation $\sigma \in S_n$. We use $\boldsymbol{\theta} = (1,2,3,4,5,6)^\top$, $\boldsymbol{w} = (6,5,4,3,2,1)^\top$, and $C = 10$, for varying regularization weight γ. The Shannon entropy is the only one yielding a permutation-equivariant operator with $\gamma > 0$.
On the **right**, we use $\gamma = 0.7$, $\boldsymbol{w} = \mathbf{1}$ and $k = 2$, and define $\boldsymbol{\theta}(s) \triangleq (s-1, s, 1, -2)^\top$ for $s \in [-4, 4]$. We plot the sum of the first two components of the relaxed operator $\boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}(s))$. The hard Top-$k$ mask ($\Omega \equiv 0$) is piecewise-constant. Gini and Tsallis regularization yield a sparse and differentiable (*a.e.* for Gini) operator $\boldsymbol{y}^k_{\mathbf{1},\Omega}$, while Shannon regularization yields a dense and differentiable $\boldsymbol{y}^k_{\mathbf{1},\Omega}$.

D.3 Mapping with the Differentiable Dynamic Programming framework

In this section, we formally cast our proposed smoothed recursions as instances of the differentiable dynamic programming (DDP) framework introduced by Mensch and Blondel [2018].

DDP framework. The DDP framework considers optimization problems framed as finding a highest-scoring path on a Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The edge scores are parameterized by $\boldsymbol{\Theta} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, where $\Theta_{u,v}$ is the score of edge $(u, v) \in \mathcal{E}$. The value of a node v, denoted $V_v(\boldsymbol{\Theta})$, is defined recursively by the Bellman equation:

$$V_v(\boldsymbol{\Theta}) \triangleq \max_{u \in \mathcal{P}_v} \left(\Theta_{u,v} + V_u(\boldsymbol{\Theta})\right),$$

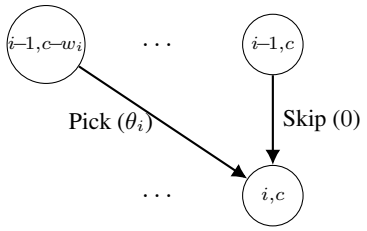
where $\mathcal{P}_v$ denotes the set of parents of node v. The smoothed recursion is obtained by replacing the $\max$ operator with $\max_\Omega$.

Knapsack DAG construction. To map the Knapsack problem defined by weights $\boldsymbol{w}$ and capacity C to this framework, we construct a DAG where nodes represent the states of the dynamic program.

- **Nodes ($\mathcal{V}$).** The set of nodes is the grid of DP states $\mathcal{V} = \{(i, c) \mid 0 \le i \le n,\ 0 \le c \le C\}$. The source node is $(0, 0)$ and the sink node is (n, C).
- **Edges ($\mathcal{E}$) and Scores ($\boldsymbol{\Theta}$).** The transitions from row $i-1$ to row i correspond to the decision of selecting item i or not. For each node (i, c) with $i \in \{1, \dots, n\}$ and $c \in \{0, \dots, C\}$:
  1. **Skip item i:** An edge exists from $(i-1, c)$ to (i, c) representing the decision $y_i = 0$. The score is $\Theta_{(i-1,c),(i,c)} = 0$.
  2. **Pick item i:** If $c \ge w_i$, an edge exists from $(i-1, c-w_i)$ to (i, c) representing the decision $y_i = 1$. The score is $\Theta_{(i-1,c-w_i),(i,c)} = \theta_i$.

  This local construction is illustrated in Figure 8.
- **Boundary conditions.** As explained in Section 3, boundary conditions of the form $V_{(0,c)}(\boldsymbol{\Theta}) = 0$ for all $c \in \{0, \dots, C\}$ are imposed on the first row to initialize the recursion (note that whenever $w_i \ge 1$ for all $i \in [n]$, only the skip edge exists at $c = 0$, which automatically propagates $V_{(i,0)}(\boldsymbol{\Theta}) = 0$ down the first column as well).

  In order to formally cast this as an instance of DDP (where only the source node $(0, 0)$ is assigned a fixed value of 0), we add artificial edges between the source node $(0, 0)$ and every boundary node $(0, c)$ for $c \in \{1, \dots, C\}$, with edge scores $\boldsymbol{\Theta}_{(0,0),(0,c)} = 0$.



Transitions to (i, c):
• From $(i-1, c)$: item i not selected.
Edge score: 0
• From $(i-1, c-w_i)$: item i selected.
Edge score: θ_i

Figure 8: Local DAG structure for the Knapsack DDP formulation. The value $V_{(i,c)}(\boldsymbol{\Theta})$ is computed by taking the maximum (or smoothed maximum) over incoming edges.

Under this construction, the DDP recursion exactly matches the Knapsack recursion. Indeed, for $i \in \{1, \dots, n\}$, $c \in \{0, \dots, C\}$ (and $c' \in \{1, \dots, C\}$ on row 0), we have:

$$V_{(i,c)}(\boldsymbol{\Theta}) = \max\left(\underbrace{\underbrace{0}_{\boldsymbol{\Theta}_{(i-1,c),(i,c)}} + V_{(i-1,c)}(\boldsymbol{\Theta})}_{\text{Skip}}, \underbrace{\underbrace{\theta_i}_{\boldsymbol{\Theta}_{(i-1,c-w_i),(i,c)}} + V_{(i-1,c-w_i)}(\boldsymbol{\Theta})}_{\text{Pick (if } c \geq w_i)}\right),$$

$$V_{(0,c')}(\boldsymbol{\Theta}) = \max\left(\underbrace{0}_{\boldsymbol{\Theta}_{(0,0),(0,c')}} + V_{(0,0)}(\boldsymbol{\Theta})\right) = V_{(0,0)}(\boldsymbol{\Theta}) = 0.$$

A full example of the recursion with corresponding edges is drawn in Figure 5.

Top-k adaptation. The Top-k problem ($\boldsymbol{w} = \mathbf{1}, C = k$) follows the same graph topology. As explained in Section 3, the constraint that exactly k items must be chosen is enforced via the boundary conditions $V_{(0,0)}(\boldsymbol{\Theta}) = 0$ and $V_{(0,j)}(\boldsymbol{\Theta}) = -\infty$ for all $j \in \{1, \dots, k\}$ (while $w_i = 1 > 0$ ensures only the skip edge exists at $j = 0$, propagating $V_{(i,0)}(\boldsymbol{\Theta}) = 0$ for all $i \in [n]$). An example showing how these boundary conditions effectively propagate and enforce the Top-k constraint is depicted in Figure 6.

In order to formally cast this as an instance of DDP, we add artificial edges between the source node $(0, 0)$ and every boundary node $(0, j)$ for $j \in [k]$ with edge scores $\boldsymbol{\Theta}_{(0,0),(0,j)} = -\infty$.

Under this construction, the DDP recursion exactly matches the Top-k recursion. Indeed, for $i \in \{1, \dots, n\}, j \in \{1, \dots, k\}$, we have:

$$V_{(i,j)}(\boldsymbol{\Theta}) = \max\left(\underbrace{\underbrace{0}_{\boldsymbol{\Theta}_{(i-1,j),(i,j)}} + V_{(i-1,j)}(\boldsymbol{\Theta})}_{\text{Skip}}, \underbrace{\underbrace{\theta_i}_{\boldsymbol{\Theta}_{(i-1,j-1),(i,j)}} + V_{(i-1,j-1)}(\boldsymbol{\Theta})}_{\text{Pick}}\right),$$

$$V_{(i,0)}(\boldsymbol{\Theta}) = \max\left(\underbrace{0}_{\boldsymbol{\Theta}_{(i-1,0),(i,0)}} + V_{(i-1,0)}(\boldsymbol{\Theta})\right) = V_{(i-1,0)}(\boldsymbol{\Theta}) = 0,$$

$$V_{(0,j)}(\boldsymbol{\Theta}) = \max\left(\underbrace{-\infty}_{\boldsymbol{\Theta}_{(0,0),(0,j)}} + V_{(0,0)}(\boldsymbol{\Theta})\right) = -\infty.$$

D.4 Properties of the relaxed values and operators

The relaxed values and operators have the following properties:

1. The smoothed output values $\boldsymbol{\theta} \mapsto \mathsf{max}^{C}_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})$ and $\boldsymbol{\theta} \mapsto \mathsf{max}^{k}_{\mathbf{1},\Omega}(\boldsymbol{\theta})$ are convex in $\boldsymbol{\theta}$.

2. Define $\Omega_d : \triangle^d \to \mathbb{R}$ as $\Omega_d(\boldsymbol{q}) \triangleq \sum_{i=1}^d \omega(q_i)$ for any integer d. Then, the difference between the smoothed and the unregularized values are bounded above and below, for all $\boldsymbol{\theta} \in \mathbb{R}^n$:

$$(N_C - 1)L_{N_C} \leq \mathsf{max}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) - \mathsf{max}^C_{\boldsymbol{w},\Omega_2}(\boldsymbol{\theta}) \leq (N_C - 1)U_{N_C},$$
$$(N_k - 1)L_{N_k} \leq \mathsf{max}^k_{\mathbf{1}}(\boldsymbol{\theta}) - \mathsf{max}^k_{\mathbf{1},\Omega_2}(\boldsymbol{\theta}) \leq (N_k - 1)U_{N_k},$$

where L_d and U_d are lower and upper bounds of Ω_d on $\triangle^d$, and with $N_C \triangleq (C+1)(n+1)$, $N_k \triangleq (k+1)(n+1)$.

3. Scaling the regularizer by $\gamma > 0$ acts as a temperature parameter on the input scores $\boldsymbol{\theta} \in \mathbb{R}^n$:

$$\mathsf{max}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \gamma\, \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}/\gamma), \qquad \boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}/\gamma),$$
$$\mathsf{max}^k_{\mathbf{1},\gamma\Omega}(\boldsymbol{\theta}) = \gamma\, \mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}/\gamma), \qquad \boldsymbol{y}^k_{\mathbf{1},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}/\gamma).$$

4. The proposed layers converge to the original, hard operators for vanishing regularization:

$$\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) \xrightarrow[\gamma\to 0^+]{} \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}).$$

5. For a fixed $\gamma > 0$ and Shannon regularization $\Omega = -H^s$, we can bound the bias induced by regularization on the forward pass of the operators in ℓ_1 distance as:

$$\|\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\|_1 \leq n|\mathcal{Y}^C_{\boldsymbol{w}}| \exp(-\delta/\gamma),$$

where $\delta \triangleq \max_{\boldsymbol{y}\in\mathcal{Y}^C_{\boldsymbol{w}}} \langle \boldsymbol{\theta}, \boldsymbol{y}\rangle - \max_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}\setminus\{\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\}} \langle \boldsymbol{\theta}, \boldsymbol{y}'\rangle$ is the hard optimality gap between the best and second best solutions (note that we have $\delta > 0$ almost everywhere with respect to $\boldsymbol{\theta} \in \mathbb{R}^n$). Thus, this forward bias decays exponentially with the regularization strength γ.

Proof. **Properties 1, 2, and 4.** Properties 1, 2, and 4 stem from a direct application of results in Mensch and Blondel [2018, Proposition 2] to our special DAG and edge weights, after mapping our framework to the DDP framework as described in Section D.3. The constants N_C and N_k emerge from the size of the corresponding DAG.

Property 3. For any $\gamma > 0$ and $\boldsymbol{x} \in \mathbb{R}^2$, we have:

$$\mathsf{max}_{\gamma\Omega}(\boldsymbol{x}) = \max_{\boldsymbol{q}\in\triangle^2} \{\langle \boldsymbol{x}, \boldsymbol{q}\rangle - \gamma\Omega(\boldsymbol{q})\} = \gamma \max_{\boldsymbol{q}\in\triangle^2} \left\{ \left\langle \frac{\boldsymbol{x}}{\gamma}, \boldsymbol{q} \right\rangle - \Omega(\boldsymbol{q}) \right\} = \gamma\, \mathsf{max}_{\Omega}(\boldsymbol{x}/\gamma).$$

Since the DP boundary values (0 and $-\infty$) are invariant under multiplication by $\gamma > 0$, induction on i in Equation (4) shows that evaluating the $\gamma\Omega$-smoothed DP table at $\boldsymbol{\theta}$ multiplies every entry of the Ω-smoothed DP table at $\boldsymbol{\theta}/\gamma$ by γ, yielding the equalities $\mathsf{max}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \gamma\, \mathsf{max}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}/\gamma)$ and $\mathsf{max}^k_{\mathbf{1},\gamma\Omega}(\boldsymbol{\theta}) = \gamma\, \mathsf{max}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}/\gamma)$. Differentiating both sides with respect to $\boldsymbol{\theta}$ via the chain rule further gives $\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}/\gamma)$ and $\boldsymbol{y}^k_{\mathbf{1},\gamma\Omega}(\boldsymbol{\theta}) = \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}/\gamma)$.

Property 5. For the fifth property, we can write:

$$\begin{aligned}
\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) &= \sum_{\boldsymbol{y}\in\mathcal{Y}^C_{\boldsymbol{w}}} \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y}\rangle/\gamma)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'\rangle/\gamma)} \boldsymbol{y} - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}^C_{\boldsymbol{w}}\setminus\{\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\}} \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y}\rangle/\gamma)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'\rangle/\gamma)} (\boldsymbol{y} - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta}))
\end{aligned}$$

and then use the triangular inequality to get:

$$\begin{aligned}
\|\boldsymbol{y}^C_{\boldsymbol{w},\gamma\Omega}(\boldsymbol{\theta}) - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\|_1 &\leq \sum_{\boldsymbol{y}\in\mathcal{Y}^C_{\boldsymbol{w}}\setminus\{\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\}} \|\boldsymbol{y} - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\|_1 \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y}\rangle/\gamma)}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'\rangle/\gamma)} \\
&\leq \sum_{\boldsymbol{y}\in\mathcal{Y}^C_{\boldsymbol{w}}\setminus\{\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\}} \|\boldsymbol{y} - \boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\|_1 \frac{\exp(\max_{\boldsymbol{y}''\in\mathcal{Y}^C_{\boldsymbol{w}}\setminus\{\boldsymbol{y}^C_{\boldsymbol{w}}(\boldsymbol{\theta})\}} \langle \boldsymbol{\theta}, \boldsymbol{y}''\rangle/\gamma)}{\exp(\max_{\boldsymbol{y}'\in\mathcal{Y}^C_{\boldsymbol{w}}} \langle \boldsymbol{\theta}, \boldsymbol{y}'\rangle/\gamma)} \\
&\leq n|\mathcal{Y}^C_{\boldsymbol{w}}| \exp(-\delta/\gamma).
\end{aligned}$$

□

D.5 Properties of the surrogate gradients for stochastic forward passes

Unbiased first-order estimator. Let $\mathcal{L} : \mathsf{conv}(\mathcal{Y}^C_{\boldsymbol{w}}) \to \mathbb{R}$ denote the composition of the downstream computation graph f_{W_2} with any loss function. The sampling operation $\boldsymbol{\theta} \mapsto \boldsymbol{y} \sim \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$ performed in the stochastic forward pass of Section 5.2 is inherently non-differentiable. Thus, there is no "exact gradient" to compare against. However, following Liu et al. [2023], we can show that our surrogate gradient is still principled as it is an *unbiased first-order estimator* of the gradient of the expectation of the loss, $J(\boldsymbol{\theta}) \triangleq \mathbb{E}_{Y \sim \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}}[\mathcal{L}(Y)]$.

Let $\hat{\nabla}_{\text{surr}} \triangleq (\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta})) \cdot \nabla \mathcal{L}(Y)$ denote the proposed surrogate gradient, where $Y \sim \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}$. First, notice that we have $\sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) = \nabla_{\boldsymbol{\theta}} \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) = \nabla_{\boldsymbol{\theta}}(1) = 0$. Thus, we have:

$$\nabla J(\boldsymbol{\theta}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \mathcal{L}(\boldsymbol{y}) \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \left(\mathcal{L}(\boldsymbol{y}) - J(\boldsymbol{\theta})\right) \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}).$$

By expanding the expected loss $J(\boldsymbol{\theta}) = \sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}') \mathcal{L}(\boldsymbol{y}')$, we obtain:

$$\nabla J(\boldsymbol{\theta}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}') \left(\mathcal{L}(\boldsymbol{y}) - \mathcal{L}(\boldsymbol{y}')\right) \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y})$$

At this point, we define a first-order approximation of the true gradient, $\hat{\nabla}_{\text{1st-order}}$, by replacing the exact loss difference $\mathcal{L}(\boldsymbol{y}) - \mathcal{L}(\boldsymbol{y}')$ with its first-order Taylor expansion $\nabla \mathcal{L}(\boldsymbol{y}')^\top (\boldsymbol{y} - \boldsymbol{y}')$:

$$\hat{\nabla}_{\text{1st-order}} \triangleq \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}') \left(\nabla \mathcal{L}(\boldsymbol{y}')^\top (\boldsymbol{y} - \boldsymbol{y}')\right) \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y})$$

To show that our surrogate gradient estimator $\hat{\nabla}_{\text{surr}}$ is an unbiased estimator of this first-order approximation, we rearrange the terms in $\hat{\nabla}_{\text{1st-order}}$ to isolate the sum over $\boldsymbol{y}$:

$$\hat{\nabla}_{\text{1st-order}} = \sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}') \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \left((\boldsymbol{y} - \boldsymbol{y}')^\top \nabla \mathcal{L}(\boldsymbol{y}')\right)$$

Distributing the dot product allows us to split the inner summation into two separate terms:

$$\hat{\nabla}_{\text{1st-order}} = \sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}') \left[\left(\sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \boldsymbol{y}^\top \right) \nabla \mathcal{L}(\boldsymbol{y}') - \left(\sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \right) \boldsymbol{y}'^\top \nabla \mathcal{L}(\boldsymbol{y}') \right]$$

Recall that the sum of the gradients of the probabilities is zero ($\sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) = 0$). This eliminates the entire second term inside the bracket, simplifying the expression to:

$$\hat{\nabla}_{\text{1st-order}} = \sum_{\boldsymbol{y}' \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}') \left(\sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \boldsymbol{y}^\top \right) \nabla \mathcal{L}(\boldsymbol{y}')$$

Next, we recognize that the expected value of the sample under the distribution is $\boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \boldsymbol{y}$. Its Jacobian with respect to $\boldsymbol{\theta}$ corresponds exactly to the inner summation:

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{y}^C_{\boldsymbol{w},\Omega}(\boldsymbol{\theta}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^C_{\boldsymbol{w}}} \nabla_{\boldsymbol{\theta}} \pi^{\boldsymbol{w},C}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) \boldsymbol{y}^\top$$

Substituting this Jacobian back into our equation yields:

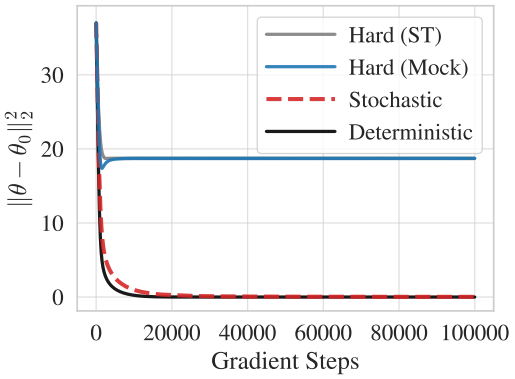


Figure 9: Comparison of our stochastic forward pass and surrogate gradient against a hard forward pass with straight-through backward, a hard forward pass with relaxed VJP backward (mock gradient), and the relaxed deterministic forward with relaxed VJP backward (deterministic layer from Section 5.1).

$$\hat{\nabla}_{\text{1st-order}} = \sum_{\boldsymbol{y}' \in \mathcal{Y}_{\boldsymbol{w}}^C} \pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}(\boldsymbol{y}') \left(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}_{\boldsymbol{w},\Omega}^C(\boldsymbol{\theta})\right) \cdot \nabla \mathcal{L}(\boldsymbol{y}')$$

Finally, rewriting this sum over $\boldsymbol{y}'$ weighted by $\pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}(\boldsymbol{y}')$ as an expectation directly recovers the expected value of our surrogate gradient:

$$\hat{\nabla}_{\text{1st-order}} = \mathbb{E}_{Y \sim \pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}} \left[(\nabla_{\boldsymbol{\theta}} \boldsymbol{y}_{\boldsymbol{w},\Omega}^C(\boldsymbol{\theta})) \cdot \nabla \mathcal{L}(Y) \right] = \mathbb{E}_{Y \sim \pi_{\boldsymbol{\theta},\Omega}^{\boldsymbol{w},C}} \left[\hat{\nabla}_{\text{surr}} \right].$$

This concludes the proof that our surrogate gradient $\hat{\nabla}_{\text{surr}}$ is a principled, unbiased first-order estimate of $\nabla J(\boldsymbol{\theta})$.

Comparison to straight-through and mock gradient approaches. In order to further validate our proposed surrogate gradient estimator, we compare it to two alternatives for gradient propagation when using hard item selections in the forward passes:

1. a hard forward pass $\boldsymbol{\theta} \mapsto \boldsymbol{y}_{\boldsymbol{w}}^C(\boldsymbol{\theta})$ with a straight-through estimator in the backward (treating the layer as if it were the identity),
2. a hard forward pass $\boldsymbol{\theta} \mapsto \boldsymbol{y}_{\boldsymbol{w}}^C(\boldsymbol{\theta})$ using the relaxed layer's VJP $\nabla_{\boldsymbol{\theta}} \boldsymbol{y}_{\boldsymbol{w},\Omega}^C(\boldsymbol{\theta})$ in the backward (a "mock gradient" approach).

We evaluate the layers on the following simple task: we generate a hidden, ground-truth parameter $\boldsymbol{\theta}_0 \in \mathbb{R}^n$ and initialize a learnable one $\boldsymbol{\theta}$. Then, we minimize the MSE between $\hat{\boldsymbol{y}}(\boldsymbol{\theta})$ (where $\hat{\boldsymbol{y}}$ is the evaluated layer) and $\boldsymbol{y}_{\boldsymbol{w},\Omega}^C(\boldsymbol{\theta}_0)$ using gradient descent (where the gradient computation is custom and defined differently for each layer as described above), and record the squared ℓ_2 distance between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_0$ along the optimization trajectory.

The results are gathered in Figure 9. While the alternatives using hard forward passes exhibit a strong bias and do not converge to the true $\boldsymbol{\theta}_0$, the proposed stochastic forward pass from Section 5.2 almost matches the exact, deterministic forward pass from Section 5.1.

E Additional Details on Numerical Experiments

E.1 Experimental details for Section 6.1

Relative regret metric. For a data point $(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)})$, the relative regret of a predicted solution $\boldsymbol{y}_{\boldsymbol{w}}^C(\hat{\boldsymbol{\theta}}^{(i)})$ is defined as the relative difference between its objective value and the optimal one under

the true item values $\boldsymbol{\theta}^{(i)}$, i.e.:

$$\frac{\langle \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)} \rangle - \langle \boldsymbol{\theta}^{(i)}, \boldsymbol{y}_{\boldsymbol{w}}^{C}(\hat{\boldsymbol{\theta}}^{(i)}) \rangle}{|\langle \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)} \rangle|}.$$

Data generation. We generate synthetic Knapsack instances using the `genData` protocol from the `PyEPO` library. For each instance, item weights $\boldsymbol{w} \in \mathbb{N}^n$ are sampled uniformly from the interval $\{3, \dots, 8\}$. The item values $\boldsymbol{\theta}$ are generated dependent on input features $\boldsymbol{x} \in \mathbb{R}^p$ (with $\boldsymbol{x} \sim \mathcal{N}(0, 1)$) according to a polynomial relation with degree deg $= 3$:

$$\theta_j = \left\lceil \left(\left(\frac{1}{\sqrt{p}} (\boldsymbol{B}\boldsymbol{x})_j + 3 \right)^3 + 1 \right) \cdot 5 \cdot 3.5^{-3} \cdot \epsilon_j \right\rceil,$$

where $\boldsymbol{B}_{ij} \sim$ Bernoulli(0.5) and $\epsilon_j \sim \mathcal{U}(0.7, 1.3)$ is a multiplicative noise term. The Knapsack capacity C is set to 50% of the total weight of all items: $C \triangleq \lfloor \frac{1}{2} \sum_{i=1}^{n} w_i \rfloor$. We generate a total of $18,000$ samples, split into $10,000$ for training, $4,000$ for validation, and $4,000$ for testing.

Architecture and optimization. We utilize a standard feed-forward neural network with two hidden layers of dimension 64 and ReLU activations. All models are trained for 30 epochs using the Adam optimizer with a learning rate of 2×10^{-3} and a batch size of 32. All experiments were executed on the CPU of an Apple M1 Max processor with 64 GB of RAM.

Statistical significance. We quantify the computational overhead by measuring the wall-clock time required for a single gradient step (encompassing both forward and backward passes), averaged over 10^4 repetitions to mitigate variance. We monitor performance of the methods on distinct datasets generated with $n \in \{10, 25, 50, 100\}$, each generated with the protocol described earlier. Each method is used to train a model on the training set, then the best model iteration is selected according to its relative regret on the validation set, and performance is finally reported using the relative regret on the test set. To ensure the robustness of our results, we perform a systematic evaluation across 10 independent random seeds for every method.

Baselines details. We compare against the following methods provided in the `PyEPO` benchmark:

- `PFY`: The *perturbed Fenchel-Young* loss [Berthet et al., 2020], which relies on Monte-Carlo sampling of perturbed costs to estimate gradients.
- `DBB`: The *differentiable black-box* [Pogancic et al., 2020] method, which computes gradients via piecewise affine interpolation.
- `NCE`: A surrogate loss based on *noise-contrastive estimation* [Mulamba et al., 2021].
- `NID`: Treats the optimal solution mapping of combinatorial minimization problems as a *negative identity* function during the backward pass [Sahoo et al., 2022].

In order to enable a fair comparison, we use the same Numba DP-based implementation of the unregularized knapsack operator $\boldsymbol{y}_{\boldsymbol{w}}^{C}$ as a solver for every baseline.

For `DBB` and `NID`, which are differentiation methods and not loss functions, we minimize the mean squared error on solutions $\frac{1}{2n} \|\boldsymbol{y}_{\boldsymbol{w}}^{C}(\hat{\boldsymbol{\theta}}^{(i)}) - \boldsymbol{y}^{(i)}\|_2^2$.

Hyperparameters. For the baselines and our method, we use the following hyperparameters:

- DP-based Fenchel-Young losses (ours): We use a regularization strength $\gamma = 5.0$.
- `PFY`: We use 10 Monte-Carlo samples and a perturbation noise $\sigma = 5.0$.
- `DBB`: We use a smoothing parameter $\lambda = 0.1$.

These hyperparameters were tuned using a grid search on the validation set.

E.2 Derivations and experimental details for Section 6.2

Derivation of the FY-DVAE objective. Standard DVAEs maximize the ELBO:

$$\log p_{W_2}(\boldsymbol{x}) \geq \mathbb{E}_{Y \sim q_{W_1}(\cdot|\boldsymbol{x})}[\log p_{W_2}(\boldsymbol{x} \mid Y)] - D_{\mathrm{KL}}(q_{W_1}(\cdot \mid \boldsymbol{x}) \,||\, p(\cdot)).$$

In the standard DVAE setting of, e.g., Ahmed et al. [2023], where the approximate posterior is parameterized as a Gibbs distribution $q_{W_1}(\boldsymbol{y}|\boldsymbol{x}) = \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \propto \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)$ with $\boldsymbol{\theta} = E_{W_1}(\boldsymbol{x})$, and the prior p is the uniform distribution on k-subsets, the KL divergence regularization term writes:

$$
\begin{aligned}
D_{\mathrm{KL}}(\pi_{\boldsymbol{\theta}} \,||\, p) &\triangleq \sum_{\boldsymbol{y} \in \mathcal{Y}_{\mathbf{1}}^k} \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \log \left(\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y})}{p(\boldsymbol{y})} \right) \\
&= - \sum_{\boldsymbol{y} \in \mathcal{Y}_{\mathbf{1}}^k} \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \log \left(\frac{1}{|\mathcal{Y}_{\mathbf{1}}^k|} \right) + \sum_{\boldsymbol{y} \in \mathcal{Y}_{\mathbf{1}}^k} \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \log \left(\pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \right) \\
&= \log \left(|\mathcal{Y}_{\mathbf{1}}^k| \right) + \sum_{\boldsymbol{y} \in \mathcal{Y}_{\mathbf{1}}^k} \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}_{\mathbf{1}}^k} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle)} \left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle - \log \sum_{\boldsymbol{y}' \in \mathcal{Y}_{\mathbf{1}}^k} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle) \right) \\
&= \log \left(|\mathcal{Y}_{\mathbf{1}}^k| \right) - \log \sum_{\boldsymbol{y}' \in \mathcal{Y}_{\mathbf{1}}^k} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle) + \left\langle \sum_{\boldsymbol{y} \in \mathcal{Y}_{\mathbf{1}}^k} \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}_{\mathbf{1}}^k} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle)} \boldsymbol{y},\ \boldsymbol{\theta} \right\rangle \\
&= A(\mathbf{0}) - A(\boldsymbol{\theta}) + \langle \nabla A(\boldsymbol{\theta}), \boldsymbol{\theta} \rangle,
\end{aligned}
$$

where $A(\boldsymbol{\theta}) = \log \sum_{\boldsymbol{y} \in \mathcal{Y}_{\mathbf{1}}^k} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)$ is the cumulant function of $\pi_{\boldsymbol{\theta}}$. We recognize this expression as a Fenchel-Young loss generated by the convex conjugate of the cumulant, A^*. Indeed, we have:

$$
\begin{aligned}
L_{A^*}(\mathbf{0}\,; \nabla A(\boldsymbol{\theta})) &\triangleq A^{**}(\mathbf{0}) + A^*(\nabla A(\boldsymbol{\theta})) - \langle \mathbf{0}, \nabla A(\boldsymbol{\theta}) \rangle \quad \text{[Blondel et al., 2020a]} \\
&= A(\mathbf{0}) + (\langle \boldsymbol{\theta}, \nabla A(\boldsymbol{\theta}) \rangle - A(\boldsymbol{\theta})) \quad (A^{**} = A \text{ and Fenchel's equality}) \\
&= A(\mathbf{0}) - A(\boldsymbol{\theta}) + \langle \boldsymbol{\theta}, \nabla A(\boldsymbol{\theta}) \rangle \\
&= D_{\mathrm{KL}}(\pi_{\boldsymbol{\theta}} \,||\, p).
\end{aligned}
$$

Here, we used the fact that A is convex and lower semi-continuous to apply the Fenchel-Moreau biconjugation theorem, and Fenchel's equality $A^*(\nabla A(\boldsymbol{\theta})) + A(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \nabla A(\boldsymbol{\theta}) \rangle$.

We generalize this to our smoothed DP setting by using a generalized approximate posterior $q_{W_1}(\boldsymbol{y}|\boldsymbol{x}) = \pi_{\boldsymbol{\theta},\Omega}^{\mathbf{1},k}(\boldsymbol{y})$, and by replacing A^* with the generalized moment polytope regularization function $\Omega_{\mathbf{1}}^k$, defined in Section 5.3 as the conjugate of the smoothed value $\mathsf{max}_{\mathbf{1},\Omega}^k$. The resulting regularization term is the Fenchel-Young loss between the uniform prior parameters $\mathbf{0}$ and the relaxed item selection $\boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})$:

$$
\begin{aligned}
L_{\Omega_{\mathbf{1}}^k}(\mathbf{0}\,; \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})) &\triangleq (\Omega_{\mathbf{1}}^k)^*(\mathbf{0}) + \Omega_{\mathbf{1}}^k(\boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})) - \langle \mathbf{0}, \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta}) \rangle \\
&= (\langle \boldsymbol{\theta}, \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta}) \rangle - \mathsf{max}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})) + \mathsf{max}_{\mathbf{1},\Omega}^k(\mathbf{0}) \\
&= \mathsf{max}_{\mathbf{1},\Omega}^k(\mathbf{0}) - \mathsf{max}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta}) + \langle \boldsymbol{\theta}, \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta}) \rangle.
\end{aligned}
$$

By construction, as $\Omega = -H^s$ yields $\mathsf{max}_{\mathbf{1},\Omega}^k = A$ and $\pi_{\boldsymbol{\theta},\Omega}^{\mathbf{1},k} = \pi_{\boldsymbol{\theta}}$ (see Section A.1 and Proposition 3), this formulation recovers the standard KL divergence objective used in Ahmed et al. [2023] as a special case, while enabling the use of regularizers that incentivize sparse latent representations (e.g., Gini or 1.5-Tsallis). This framework also extends the Fenchel-Young variational inference framework of Sklaviadis et al. [2025], which introduced Fenchel-Young regularization for learning continuous VAEs, to the discrete setting of latent distributions on k-subsets. Moreover, the gradient of this Fenchel-Young regularization term is given by:

$$
\nabla_{\boldsymbol{\theta}} L_{\Omega_{\mathbf{1}}^k}(\mathbf{0}\,; \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})) = (\nabla_{\boldsymbol{\theta}} \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})) \cdot \boldsymbol{\theta},
$$

that is, it is simply the JVP of the relaxed layer $\boldsymbol{y}_{\mathbf{1},\Omega}^k$ in the direction of its input $\boldsymbol{\theta}$, efficiently computed via Algorithm 4.

Dataset generation. We construct each data point in our Stacked MNIST dataset by sampling $k = 3$ distinct digit classes $\{c_1, \ldots, c_k\} \subset \{0, \ldots, 9\}$ without replacement. For each class c_j, we sample a random image $\boldsymbol{x}^{(j)}$ from the standard MNIST dataset. The input $\boldsymbol{x}$ is the pixel-wise average: $\boldsymbol{x} = \frac{1}{k} \sum_{j=1}^k \boldsymbol{x}^{(j)}$. The ground truth is the k-hot vector $\boldsymbol{y} \in \{0, 1\}^{10}$ indicating the present digits. We generate $120,000$ training examples and $20,000$ test examples.

Architecture. The model architecture captures both the presence and the style of the digits using a decomposed latent space with dimensions $n = 10$ for the discrete part and $d_z = 32$ for the continuous part. The encoder E_{W_1} is a convolutional neural network consisting of a stack of strided convolution layers (kernel size 3×3, stride 2, ReLU activation), which progressively downsample the spatial dimensions while increasing channel depth. The resulting flattened feature map is fed into two separate dense projection heads: one outputting the selection logits $\boldsymbol{\theta} \in \mathbb{R}^{10}$ for the discrete mask $\boldsymbol{y}$, and another outputting the parameters $[\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2] \in \mathbb{R}^{10 \times 2d_z}$ for the $n = 10$ independent Gaussian style posteriors $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2))$.

The mapping from selection logits $\boldsymbol{\theta}$ to mixing coefficients $\boldsymbol{y}$ in the discrete bottleneck depends on the considered method:

- for the hard Top-k baseline, the mapping is simply $\boldsymbol{y} = \boldsymbol{y}_{\mathbf{1}}^k(\boldsymbol{\theta}) \in \mathcal{Y}_{\mathbf{1}}^k$,
- for the Gumbel Top-k baseline, we sample i.i.d. Gumbel noise $Z_i \sim \mathrm{Gumbel}(0, 1)$ for $i \in [10]$ and take the Top-k of the perturbed values $\boldsymbol{y} = \boldsymbol{y}_{\mathbf{1}}^k(\boldsymbol{\theta} + \tau \boldsymbol{Z}) \in \mathcal{Y}_{\mathbf{1}}^k$,
- for the specialized deterministic Top-k baselines, we apply the operators of Sander et al. [2023] (FDS), Struski et al. [2025] (LapSum), and Xie et al. [2020] (SOFT, using optimal transport regularization strength $\epsilon = 0.005$ selected via a hyperparameter sweep),
- for our deterministic DP-based layers, we directly use the regularized Top-k operator $\boldsymbol{y} = \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta}) \in \mathsf{conv}(\mathcal{Y}_{\mathbf{1}}^k)$ using Algorithm 2,
- and for our stochastic DP-based layers, we sample from the underlying distribution via $\boldsymbol{y} \sim \pi_{\boldsymbol{\theta},\Omega}^{\mathbf{1},k}$ using Algorithm 3.

We maintain a set of learned embeddings $\mathbf{E} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{10}\}$ with $\boldsymbol{e}_i \in \mathbb{R}^{d_z}$, representing the prototypes for each digit class. The decoder D_{W_2} is a *shared* de-convolutional network applied independently to each expert. For each item i, it receives the concatenation $[\boldsymbol{z}_i, \boldsymbol{e}_i] \in \mathbb{R}^{2d_z}$ of the sampled style code and the prototype embedding. This input is mapped via a dense layer to a low-resolution feature map, then upsampled via a stack of transposed convolution layers (kernel size 3×3, stride 2, ReLU activation) to reach the target image resolution. A final convolution layer followed by a sigmoid activation produces the candidate reconstruction $\hat{\boldsymbol{x}}_i \in [0, 1]^{H \times W}$. The final output is the weighted average of these candidates:

$$\hat{\boldsymbol{x}} = \frac{1}{k} \sum_{i=1}^{10} y_i \cdot \sigma\left(D_{W_2}([\boldsymbol{z}_i, \boldsymbol{e}_i])\right) = \frac{1}{k} \sum_{i=1}^{10} y_i \cdot \hat{\boldsymbol{x}}_i.$$

Training details. The models are trained using the Adam optimizer with a learning rate of 3×10^{-4} and a batch size of 64 for 50 epochs. The full training objective is given by:

$$\mathcal{L}(\boldsymbol{x}\,; W_1, W_2) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 + \beta_{\mathrm{KL}} \sum_{i=1}^{10} D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)) || \mathcal{N}(\mathbf{0}, \boldsymbol{I})) + \beta_{\mathrm{FY}} \cdot L_{\Omega_{\mathbf{1}}^k}(\mathbf{0}\,; \boldsymbol{y}_{\mathbf{1},\Omega}^k(\boldsymbol{\theta})).$$

We use a fixed weight $\beta_{\mathrm{KL}} = 10^{-5}$ for the continuous latent regularization. We anneal the discrete Fenchel-Young regularization strength β_{FY} linearly from 0 to a base value of 10^{-5} over the first two-thirds of the training epochs. All displayed metrics are averaged over 30 trainings with different random seeds. All experiments were executed on TPUv2 with one device.

Final test metrics. To complement the training curves in Figure 2b with exact precision, we report the final test reconstruction error and KL divergence for all evaluated methods in Table 1.

Ablation on the Fenchel-Young regularization weight β_{FY}. On the considered Stacked MNIST task, setting $\beta_{\mathrm{FY}} = 0$ (no discrete regularization) yields test metrics that are nearly identical to our default hyperparameter ($\beta_{\mathrm{FY}} = 10^{-5}$). This reflects a deliberate hyperparameter choice governed by the usual rate-distortion trade-off in VAEs: to ensure a fair comparison against purely deterministic, unregularized Top-k baselines such as FDS and LapSum, we tuned β_{FY} to a small value so that the model could prioritize reconstruction. Table 2 illustrates this dynamic across increasing values of β_{FY}: larger weights enforce the uniform prior over k-subsets more strongly, which opposes latent sparsity (increasing the average support size $\|\boldsymbol{y}\|_0$ toward $n = 10$) and degrades test reconstruction error.

Table 1: **Final DVAE metrics** (test reconstruction error $\times 10^2$ and continuous latent KL divergence) averaged across 30 random seeds, sorted by reconstruction error in ascending order.

Method	**Test Reconstruction Error** $\times 10^2$	**KL Divergence**
Stochastic Tsallis	0.2315	155.14
Deterministic Gini	0.2315	155.18
LapSum	0.2316	155.24
Stochastic Gini	0.2319	155.05
Deterministic Tsallis	0.2321	155.06
FDS	0.2323	155.15
Deterministic Shannon	0.2684	149.45
SOFT ($\epsilon = 0.005$)	0.2824	222.72
Stochastic Shannon	0.3830	156.11
Hard-ST	2.0949	1.6741×10^{13}
Gumbel-ST	2.1203	1.0777×10^{13}

Consequently, the primary role of the Fenchel-Young regularization term $L_{\Omega_1^k}(\mathbf{0}\,; \boldsymbol{y}^k_{\mathbf{1},\Omega}(\boldsymbol{\theta}))$ here is theoretical completeness rather than an empirical gain on this specific benchmark: as shown above, it generalizes the standard discrete KL divergence against a uniform prior used in Ahmed et al. [2023] and extends Sklaviadis et al. [2025] to discrete k-subset bottlenecks, thereby grounding the model within a proper DVAE framework rather than an unregularized discrete-bottleneck autoencoder.

Table 2: **Ablation on the Fenchel-Young regularization weight** $\beta_{\mathbf{FY}}$ in the FY-DVAE objective.

β_{FY}	**Test Reconstruction Error** $\times 10^2$	**Fenchel-Young Loss**	**Average** $\|\boldsymbol{y}\|_0$
0.0 (Ablated)	0.231	1.31	3.0
10^{-5} (Ours)	0.232	1.31	3.0
3×10^{-5}	0.236	1.26	3.3
5×10^{-5}	0.260	0.81	6.3
7×10^{-5}	0.279	0.31	8.8
10^{-4}	0.284	0.09	9.6

E.3 Experimental details for Section 6.3

Problem formulation. The agent manages a store containing $n = 20$ items over a finite horizon of $T = 80$ time steps. Each item i is assigned a static weight w_i drawn uniformly from $[5]$, a static price $s_i > 0$, and possesses a dynamic feature vector $\boldsymbol{x}^{(t,i)}$. At each time step t, the agent selects a subset of items (an assortment) $\boldsymbol{y}^{(t)} \in \{0,1\}^n$ to display. This selection is subject to two operational constraints:

1. **Knapsack constraint:** the total weight of displayed items must not exceed the store's capacity, i.e.:
$$\sum_{i=1}^{n} w_i \, y_i^{(t)} \leq C \triangleq 40.$$
2. **Inventory constraint:** an item i may only be displayed if it remains in stock ($I_i^{(t)} \geq 1$). All items are initialized with a uniform inventory of $I_i^{(0)} \triangleq I^{(0)} = 5$.

In our precise setting, there are exactly $|\mathcal{Y}^C_{\boldsymbol{w}}| = 989034$ feasible assortments when every item is in stock.

Customer choice model. Customer behavior is simulated using a multinomial logit model. Given an assortment $\boldsymbol{y}^{(t)}$, the customer purchases item $i \in \{j \in [n] \mid y_j^{(t)} = 1\}$ or selects the "no-purchase"

option ($i = 0$) according to the following probabilities:

$$P(i|\boldsymbol{y}^{(t)}) = \frac{\exp(u_i^{(t)})}{1 + \sum_{j:y_j^{(t)}=1} \exp(u_j^{(t)})}, \tag{13}$$

where $u_i^{(t)} = \langle \boldsymbol{\phi}, \boldsymbol{x}^{(t,i)} \rangle$ represents the utility of item i derived from a hidden linear customer model $\boldsymbol{\phi}$.

Endogenous uncertainty. The environment features strong endogeneity in both feature evolution and state transitions. First, the inventory evolves deterministically: $I_i^{(t+1)} = I_i^{(t)} - 1$ if item i is purchased, creating a coupling between immediate rewards and future availability. Second, item features evolve based on customer decisions. Specifically, if item i is purchased, its "satisfaction" feature (part of $\boldsymbol{x}^{(t,i)}$) increases by 1%. Furthermore, a "hype" feature evolves based on a rolling window of the last 5 purchases: the most recently purchased item receives a hype boost of $+2 \times 10^{-2}$, while items purchased 2 to 5 steps ago suffer a decay of -5×10^{-3}. Finally, a time feature increments linearly by $9/T$ at each step.

State space. The state s_t provided to the agent is a concatenation of static and dynamic information resulting in a feature dimension of $12 \times n$. Specifically, the 12-dimensional vector $\boldsymbol{x}^{(t,i)}$ for item i is constructed as follows: indices $x_0^{(t,i)}, x_1^{(t,i)}$ contain static latent attributes; $x_2^{(t,i)}$ is the dynamic hype; $x_3^{(t,i)}$ is the dynamic satisfaction; $x_4^{(t,i)}$ is the static price s_i; and $x_5^{(t,i)}$ is the current time step. These base features are augmented by their dynamics: $x_6^{(t,i)}, x_7^{(t,i)}$ contain the one-step difference in hype and satisfaction, while $x_8^{(t,i)}, x_9^{(t,i)}$ contain their cumulative change since $t = 0$. Finally, $x_{10}^{(t,i)}$ contains the normalized inventory $I_i^{(t)}/I^{(0)}$ and $x_{11}^{(t,i)}$ contains the normalized weight w_i/C. The reward r_t is defined as the price s_i of the purchased item i, or 0 if no purchase occurs. The objective is to maximize cumulative revenue $\sum_{t=1}^{T} r_t$.

Greedy policy. As a simple baseline, we include a greedy actor, which always displays available items that maximize the total price:

$$\begin{aligned} \boldsymbol{y}_{\text{greedy}}^{(t)} \triangleq \underset{\boldsymbol{y} \in \{0,1\}^n}{\mathsf{argmax}} \quad & \sum_{i=1}^{n} s_i y_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i y_i \leq C, \\ \text{and} \quad & \forall i \in [n] \mid y_i = 1,\ I_i^{(t)} \geq 1. \end{aligned}$$

This is effectively implemented using a regular Knapsack operator, by using an artificial weights vector $\hat{\boldsymbol{w}}^{(t)}$ such that $\hat{w}_i^{(t)} = w_i$ if $I_i^{(t)} \geq 1$, and $\hat{w}_i^{(t)} = C + 1$ if $I_i^{(t)} = 0$. We then have $\boldsymbol{y}_{\text{greedy}}^{(t)} = \boldsymbol{y}_{\hat{\boldsymbol{w}}^{(t)}}^{C}(\boldsymbol{s})$.

Expert policy. Finding the optimal policy is computationally intractable. We define a myopic expert policy baseline as follows: at each time step t, the expert computes the assortment $\boldsymbol{y}^{(t)}$ that maximizes the expected immediate revenue, subject to current constraints:

$$\begin{aligned} \boldsymbol{y}_{\text{expert}}^{(t)} \triangleq \underset{\boldsymbol{y} \in \{0,1\}^n}{\mathsf{argmax}} \quad & \sum_{i \in [n] | y_i = 1} s_i \cdot P(i|\boldsymbol{y}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i y_i \leq C, \\ \text{and} \quad & \forall i \in [n] \mid y_i = 1,\ I_i^{(t)} \geq 1. \end{aligned}$$

To compute $\boldsymbol{y}_{\text{expert}}^{(t)}$, we enumerate every feasible item selection (there are 989034 of them in our configuration), and compute the corresponding purchase probabilities. Naturally, this baseline requires access to hidden information (the customer model $\boldsymbol{\phi}$ defining purchase probabilities $P(i|\boldsymbol{y})$), and is computationally costly.

SRL actors. The actors in the SRL-based methods are parameterized by a feed-forward neural network f_{W_1} mapping states $\boldsymbol{x}^{(t)} \in \mathbb{R}^{12\times n}$ to item logits $\boldsymbol{\theta} \in \mathbb{R}^n$, which are then mapped to feasible assortments by the hard Knapsack operator $\boldsymbol{y}^C_{\hat{\boldsymbol{w}}^{(t)}}(\boldsymbol{\theta})$, where $\hat{w}_i^{(t)} = w_i$ if $I_i^{(t)} \geq 1$, and $\hat{w}_i^{(t)} = C+1$ if $I_i^{(t)} = 0$ (effectively enforcing the inventory constraint).

Exploration and gradient computation. In standard SRL [Hoppe et al., 2025], K exploration targets are generated by perturbing the predicted logits $\boldsymbol{\theta}$ with a random variable Z (which follows, e.g., a centered Gaussian distribution in $\mathbb{R}^n$) to get samples:

$$\boldsymbol{y}^{(k)} \triangleq \boldsymbol{y}^C_{\hat{\boldsymbol{w}}^{(t)}}(\boldsymbol{\theta} + \varepsilon Z^{(k)}) \in \mathcal{Y}^C_{\hat{\boldsymbol{w}}^{(t)}} \ ,$$

where $\varepsilon > 0$ is a hyperparameter and the $Z^{(k)}$ are i.i.d. samples. Then, the target $\hat{\boldsymbol{y}}$ is defined by aggregating these samples via a softmax operation using Q-values predicted by a critic network ψ_{W_2}:

$$\hat{\boldsymbol{y}} \triangleq \sum_{k=1}^{K} \frac{\exp\left(\frac{\psi_{W_2}(\boldsymbol{x},\boldsymbol{y}^{(k)})}{\tau}\right)}{\sum_{k'=1}^{K}\exp\left(\frac{\psi_{W_2}(\boldsymbol{x},\boldsymbol{y}^{(k')})}{\tau}\right)} \cdot \boldsymbol{y}^{(k)} \in \mathsf{conv}\left(\mathcal{Y}^C_{\hat{\boldsymbol{w}}^{(t)}}\right).$$

Finally, a perturbation-based Fenchel-Young loss [Berthet et al., 2020] is used to compute gradients with respect to $\boldsymbol{\theta}$, which are then backpropagated to upstream weights W_1. This gradient is estimated via Monte-Carlo as:

$$\nabla_{\boldsymbol{\theta}} L_\varepsilon(\boldsymbol{\theta}\,;\hat{\boldsymbol{y}}) \approx \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{y}^C_{\hat{\boldsymbol{w}}^{(t)}}(\boldsymbol{\theta} + \varepsilon Z^{(m)}) - \hat{\boldsymbol{y}},$$

where the $Z^{(m)}$ are again i.i.d. samples.

What we propose for our DP-based SRL agents instead, is to use Algorithm 3 to produce K exploration actions:

$$\boldsymbol{y}^{(k)} \sim \pi^{\hat{\boldsymbol{w}}^{(t)},C}_{\boldsymbol{\theta},\Omega} \in \mathcal{Y}^C_{\hat{\boldsymbol{w}}^{(t)}} \ ,$$

and to use our DP-based Fenchel-Young losses to compute exact gradients

$$\nabla_{\boldsymbol{\theta}} L_{\Omega^C_{\hat{\boldsymbol{w}}^{(t)}}}(\boldsymbol{\theta}\,;\hat{\boldsymbol{y}}) = \boldsymbol{y}^C_{\hat{\boldsymbol{w}}^{(t)},\Omega}(\boldsymbol{\theta}) - \hat{\boldsymbol{y}},$$

using Algorithm 2. Importantly, we only need to call Algorithm 1 once to produce intermediate outputs for Algorithms 2 and 3, yielding computational gains.

PPO baseline. We adapt proximal policy optimization (PPO) [Schulman et al., 2017] to the combinatorial setting following the COaML-pipeline framework detailed in Hoppe et al. [2025, Section C.2.2]. In this formulation, the PPO agent does not directly output discrete actions. Instead, the actor network f_{W_1} predicts a latent score vector $\boldsymbol{\theta} \in \mathbb{R}^n$, which parameterizes a multivariate Gaussian policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta} \mid \boldsymbol{\theta}, \sigma^2\boldsymbol{I})$. During the rollout, a continuous score vector $\boldsymbol{\eta}$ is sampled from this distribution and passed to the hard Knapsack solver to produce the discrete item selection $\boldsymbol{y}^{(t)} = \boldsymbol{y}^C_{\hat{\boldsymbol{w}}^{(t)}}(\boldsymbol{\eta})$. From the perspective of the PPO algorithm, the actor's action is the sampled continuous vector $\boldsymbol{\eta}$, the combinatorial solver is treated as a deterministic transition function within the environment dynamics, and the optimization objective is to adjust the mean $\boldsymbol{\theta}$ of the score distribution to maximize returns. The policy ratio and clipping are computed on the continuous densities of the perturbed scores $\boldsymbol{\eta}$. We anneal the exploration standard deviation σ linearly during training.

Hyperparameters. We train all agents for 10^4 episodes, performing 10 update iterations with a batch size of 32 at the end of each episode for SRL, and 100 iterations for PPO to compensate for its on-policy data efficiency. We maintain a replay buffer with a capacity of 10^4 transitions and use the Adam optimizer with a learning rate of 8×10^{-5} for all networks. For the SRL agents, we set the perturbation noise scale ε and the regularization strength γ to the same value, annealing it linearly from 5.0 to 1.0 over the course of training. We use $K = 32$ exploration samples for target action creation, and the softmax aggregation temperature τ is annealed linearly from 1.0 to 0.5. For PPO, we use a clipping ratio of $\epsilon = 0.2$, and the exploration standard deviation σ is annealed linearly from 2.0 to 1.0. All experiments were executed on the CPU of an Apple M1 Max processor with 64 GB of RAM.

F Explicit Derivations

In this section, we provide explicit derivations for the quantities $Q_\Omega^{\boldsymbol{w}}[i,c]$, $V_\Omega^{\boldsymbol{w}}[i,c]$, and the partial derivatives required to instantiate Algorithms 1 to 4. The obtained closed forms are summarized in Table 3.

Recall the definition of the smoothed value function $V_\Omega^{\boldsymbol{w}}[i,c]$ from Equation (4):

$$V_\Omega^{\boldsymbol{w}}[i,c] \triangleq \begin{cases} V_\Omega^{\boldsymbol{w}}[i-1,c] & \text{if } w_i > c, \\ \mathsf{max}_\Omega\big(\theta_i + V_\Omega^{\boldsymbol{w}}[i-1,c-w_i],\ V_\Omega^{\boldsymbol{w}}[i-1,c]\big) & \text{else.} \end{cases}$$

We focus on the non-trivial case where $w_i \leq c$. We define the local inputs a and b as:

$$\begin{aligned} a &\triangleq \theta_i + V_\Omega^{\boldsymbol{w}}[i-1,c-w_i], \\ b &\triangleq V_\Omega^{\boldsymbol{w}}[i-1,c]. \end{aligned}$$

The smoothed maximum is defined as the optimal value of the regularized objective:

$$\mathsf{max}_\Omega(a,b) \triangleq \max_{q\in[0,1]} J(q\,;a,b), \quad \text{where } J(q\,;a,b) \triangleq qa + (1-q)b - \Omega(\boldsymbol{q}).$$

Here, $\boldsymbol{q} = (q, 1-q)^\top$ and $\Omega(\boldsymbol{q}) = \omega(q) + \omega(1-q)$ is a separable strictly convex regularization function. We denote the unique maximizer by $q^*(a,b)$. The partial derivatives required for the algorithms are given by Danskin's theorem and the chain rule:

$$\begin{aligned} Q_\Omega^{\boldsymbol{w}}[i,c] &= q^*(a,b), \\ \frac{\partial Q_\Omega^{\boldsymbol{w}}[i,c]}{\partial \theta_i} &= \frac{\partial q^*}{\partial a}(a,b). \end{aligned}$$

In the following, we derive q^* and $\frac{\partial q^*}{\partial a}$ for specific choices of Ω. The resulting explicit expressions are summarized in Table 3, together with the corresponding expression of $V_\Omega^{\boldsymbol{w}}[i,c] = \mathsf{max}_\Omega(a,b) = J(q^*(a,b)\,;a,b)$.

F.1 General derivative computation

The optimization problem $\max_{q\in[0,1]} J(q\,;a,b)$ is equivalent to the minimization problem:

$$\begin{aligned} \min_{q\in\mathbb{R}} \quad & \Phi(q) \triangleq \omega(q) + \omega(1-q) - qa - (1-q)b \\ \text{s.t.} \quad & g_1(q) \triangleq -q \leq 0, \\ & g_2(q) \triangleq q - 1 \leq 0. \end{aligned}$$

Let $\mu, \lambda \geq 0$ be the KKT multipliers associated with g_1 and g_2 respectively. The KKT stationarity condition necessitates:

$$\omega'(q^*) - \omega'(1-q^*) - (a-b) - \mu + \lambda = 0. \tag{14}$$

We now distinguish two regimes for the sensitivity $\frac{\partial q^*}{\partial a}$.

Case 1: interior solution. Assume the optimal solution lies in the strict interior $q^* \in (0,1)$. The box constraints are inactive and we have $\mu = \lambda = 0$. The stationarity condition Equation (14) simplifies to:

$$\omega'(q^*) - \omega'(1-q^*) = a - b.$$

Differentiating with respect to a yields:

$$\frac{\partial q^*}{\partial a} = \frac{1}{\omega''(q^*) + \omega''(1-q^*)}. \tag{15}$$

Case 2: boundary solution. We now consider the case where the solution saturates at a boundary. This occurs if and only if the regularizer has bounded gradients at the endpoints. Assume $\lim_{q\to 1^-}\omega'(q) = L < \infty$ and $\lim_{q\to 0^+}\omega'(q) = l > -\infty$. Let $\tau_\Omega \triangleq L - l$. We prove that if $a - b \geq \tau_\Omega$, then $q^* = 1$ and $\frac{\partial q^*}{\partial a} = 0$.

Let $q^* = 1$. The active constraint is $g_2(q) \leq 0$, so $\mu = 0$ and $\lambda \geq 0$. The stationarity condition Equation (14) becomes:

$$\begin{aligned}\lambda &= (a - b) - (\omega'(1) - \omega'(0)) \\ &= (a - b) - \tau_\Omega.\end{aligned}$$

Dual feasibility requires $\lambda \geq 0$, which holds if and only if $a - b \geq \tau_\Omega$. Since the problem is strictly convex, the KKT conditions are sufficient for optimality. Thus, for all (a, b) such that $a - b > \tau_\Omega$, the unique minimizer is $q^* = 1$. In this open region, q^* is constant with respect to a, implying:

$$\frac{\partial q^*}{\partial a} = 0.$$

A symmetric argument shows that if $a - b \leq -\tau_\Omega$, then $q^* = 0$ and the derivative is zero.

F.2 Shannon entropy

We consider $\Omega(\boldsymbol{q}) = -\gamma H^s(\boldsymbol{q}) \triangleq \gamma \sum_{i=1}^2 q_i \log q_i$, which implies $\omega(q) = \gamma q \log q$. The gradients of ω become unbounded at the boundaries ($\lim_{q\to 0}\omega'(q) = -\infty$), forcing the solution to be strictly interior for any $a, b \in \mathbb{R}$.

Maximizer. The stationarity condition Equation (14) yields:

$$\gamma \log q^* - \gamma \log(1 - q^*) = a - b \iff \gamma \log\left(\frac{q^*}{1 - q^*}\right) = a - b.$$

Solving for q^* yields the logistic function $q^* = \sigma(\frac{a-b}{\gamma}) \triangleq (1 + \exp((b - a)/\gamma))^{-1}$.

Partial derivative. The derivative is computed via Equation (15) derived in Case 1:

$$\frac{\partial q^*}{\partial a} = \frac{1}{\omega''(q^*) + \omega''(1 - q^*)} = \left(\frac{\gamma}{q^*} + \frac{\gamma}{1 - q^*}\right)^{-1} = \frac{q^*(1 - q^*)}{\gamma}.$$

Smoothed maximum. Substituting the optimal probability q^* back into the objective function J yields the standard *log-sum-exp* as expected:

$$\mathsf{max}_\Omega(a, b) = \gamma \log\left(\exp(a/\gamma) + \exp(b/\gamma)\right).$$

F.3 Gini entropy

We consider $\Omega(\boldsymbol{q}) = -\gamma H^g(\boldsymbol{q}) \triangleq \frac{\gamma}{2}(\|\boldsymbol{q}\|_2^2 - 1)$, which implies $\omega(q) = \frac{\gamma}{2}q^2 - \frac{\gamma}{4}$.

Maximizer. In this case, we have:

$$\omega'(q) - \omega'(1 - q) = \gamma q - \gamma(1 - q) = 2\gamma q - \gamma,$$

so that Equation (14) becomes:

$$2\gamma q^* - \gamma - (a - b) - \mu + \lambda = 0 \implies q^* = \frac{a - b + \gamma + \mu - \lambda}{2\gamma}.$$

We analyze the solution based on the active constraints using the complementary slackness conditions $\mu q^* = 0$ and $\lambda(q^* - 1) = 0$.

1. **Inactive constraints ($\mu = 0, \lambda = 0$):** we have $q^* = \frac{a-b+\gamma}{2\gamma}$. Primal feasibility $0 < q^* < 1$ requires $0 < \frac{a-b+\gamma}{2\gamma} < 1$, which simplifies to $-\gamma < a - b < \gamma$. In this region, $\frac{\partial q^*}{\partial a} = \frac{1}{2\gamma}$.

2. **Active lower bound ($q^* = 0$):** we have $\lambda = 0$ and $\mu \geq 0$. The stationarity condition yields $0 = \frac{a-b+\gamma+\mu}{2\gamma}$, which implies $\mu = -(a - b + \gamma)$. Dual feasibility $\mu \geq 0$ requires $a - b + \gamma \leq 0 \iff a - b \leq -\gamma$. In this region, q^* is constant, so $\frac{\partial q^*}{\partial a} = 0$.
3. **Active upper bound ($q^* = 1$):** we have $\mu = 0$ and $\lambda \geq 0$. The stationarity condition yields $1 = \frac{a-b+\gamma-\lambda}{2\gamma}$, which implies $2\gamma = a - b + \gamma - \lambda$, or $\lambda = a - b - \gamma$. Dual feasibility $\lambda \geq 0$ requires $a - b - \gamma \geq 0 \iff a - b \geq \gamma$. In this region, q^* is constant, so $\frac{\partial q^*}{\partial a} = 0$.

Combining these three cases, we recover the projection onto $[0, 1]$:

$$q^* = \mathsf{clip}_{[0,1]}\left(\frac{a - b + \gamma}{2\gamma}\right).$$

Partial derivative. The partial derivative is constant and only non-zero on the non-clipped region:

$$\frac{\partial q^*}{\partial a} = \frac{1}{2\gamma} \cdot \mathbb{1}_{\{|a-b|<\gamma\}}.$$

Smoothed maximum. The regularization term simplifies to $\Omega(\boldsymbol{q}) = \gamma(q^2 - q)$ for any $\boldsymbol{q} = (q, (1 - q))^\top$. Substituting the optimal q^* into the objective function J yields:

$$\begin{aligned}\mathsf{max}_\Omega(a, b) &= q^* a + (1 - q^*)b - \gamma((q^*)^2 - q^*) \\ &= (a - b)q^* + b - \gamma q^*(q^* - 1).\end{aligned}$$

When saturation occurs (i.e., $|a - b| \geq \gamma$), we have $q^* \in \{0, 1\}$. In this case, the regularization term vanishes and we recover the hard maximum $\max(a, b)$. In the interior region (i.e., $|a - b| < \gamma$), substituting $q^* = \frac{a-b+\gamma}{2\gamma}$ yields:

$$\begin{aligned}\mathsf{max}_\Omega(a, b) &= (a - b)\left(\frac{a - b + \gamma}{2\gamma}\right) + b - \gamma\left(\frac{a - b + \gamma}{2\gamma}\right)\left(\frac{a - b - \gamma}{2\gamma}\right) \\ &= \frac{(a - b)^2 + \gamma(a - b)}{2\gamma} + b - \gamma\left(\frac{(a - b)^2 - \gamma^2}{4\gamma^2}\right) \\ &= \frac{(a - b)^2}{2\gamma} + \frac{a - b}{2} + b - \frac{(a - b)^2}{4\gamma} + \frac{\gamma^2}{4\gamma} \\ &= \frac{(a - b)^2}{4\gamma} + \frac{a + b}{2} + \frac{\gamma}{4}.\end{aligned}$$

F.4 1.5-Tsallis Entropy

We consider $\Omega(\boldsymbol{q}) = -\gamma H^T_{1.5}(\boldsymbol{q}) \triangleq \gamma \sum_{i=1}^2 \frac{q_i^{\frac{3}{2}} - q_i}{\frac{3}{2}(\frac{3}{2}-1)}$, i.e., $\omega(q) = \frac{4\gamma}{3} q^{\frac{3}{2}} - \frac{2\gamma}{3}$ and $\omega'(q) = 2\gamma\sqrt{q}$.

Maximizer. The stationarity condition for an interior solution ($q^* \in (0, 1)$) requires:

$$2\gamma\sqrt{q^*} - 2\gamma\sqrt{1 - q^*} = a - b.$$

Let $C \triangleq (a - b)/2\gamma$. We seek to solve $\sqrt{q^*} - \sqrt{1 - q^*} = C$. The function $f(q) = \sqrt{q} - \sqrt{1 - q}$ is strictly increasing on $[0, 1]$ with range $[-1, 1]$.

- If $|C| > 1$ (i.e., $|a - b| > 2\gamma$), no interior solution exists. The monotonicity of the objective forces the solution to the boundary: $q^* = 1$ if $C > 1$, and $q^* = 0$ if $C < -1$.
- If $|C| \leq 1$, we square the stationarity equation: $q^* - 2\sqrt{q^*(1 - q^*)} + (1 - q^*) = C^2$, which rearranges to $2\sqrt{q^*(1 - q^*)} = 1 - C^2$. Squaring again yields the quadratic equation

Table 3: Instantiations of the quantities required for Algorithm 1 and Algorithm 4.
We use $\Delta_{i,c} \triangleq V_\Omega^{\boldsymbol{w}}[i-1,c]-(\theta_i+V_\Omega^{\boldsymbol{w}}[i-1,c-w_i])$, and $\sigma(t) \triangleq (1+\exp(-t))^{-1}$ denotes the logistic function. For the 1.5-Tsallis case, we use the intermediate variable $\bar{C} \triangleq \mathsf{clip}_{[-1,1]}\left(-\Delta_{i,c}/2\gamma\right)$.

	Shannon ($\Omega = -\gamma H^s$)	**Gini** ($\Omega = -\gamma H^g$)	**1.5-Tsallis** ($\Omega = -\gamma H^T_{1.5}$)
$Q_\Omega^{\boldsymbol{w}}[i,c]$	$\sigma(-\Delta_{i,c}/\gamma)$	$\mathsf{clip}_{[0,1]}\left(\frac{-\Delta_{i,c}+\gamma}{2\gamma}\right)$	$\frac{1}{2}\left(1+\bar{C}\sqrt{2-\bar{C}^2}\right)$
$V_\Omega^{\boldsymbol{w}}[i,c]$	$V_\Omega^{\boldsymbol{w}}[i-1,c]$ $+\gamma\log\left(1+\exp(-\Delta_{i,c}/\gamma)\right)$	$V_\Omega^{\boldsymbol{w}}[i-1,c]-\Delta_{i,c}Q_\Omega^{\boldsymbol{w}}[i,c]$ $+\gamma Q_\Omega^{\boldsymbol{w}}[i,c](1-Q_\Omega^{\boldsymbol{w}}[i,c])$	$V_\Omega^{\boldsymbol{w}}[i-1,c]-\Delta_{i,c}Q_\Omega^{\boldsymbol{w}}[i,c]$ $+\frac{4\gamma}{3}\Big(1-Q_\Omega^{\boldsymbol{w}}[i,c]^{\frac{3}{2}}$ $-(1-Q_\Omega^{\boldsymbol{w}}[i,c])^{\frac{3}{2}}\Big)$
$\frac{\partial Q_\Omega^{\boldsymbol{w}}[i,c]}{\partial\theta_i}$	$\frac{1}{\gamma}Q_\Omega^{\boldsymbol{w}}[i,c](1-Q_\Omega^{\boldsymbol{w}}[i,c])$	$\frac{1}{2\gamma}\mathbb{1}_{\{0<Q_\Omega^{\boldsymbol{w}}[i,c]<1\}}$	$\frac{1}{\gamma}\left(\frac{1}{\sqrt{Q_\Omega^{\boldsymbol{w}}[i,c]}}+\frac{1}{\sqrt{1-Q_\Omega^{\boldsymbol{w}}[i,c]}}\right)^{-1}$
$\frac{\partial Q_\Omega^{\boldsymbol{w}}[i,c]}{\partial V_\Omega^{\boldsymbol{w}}[i-1,c]}$	$-\frac{1}{\gamma}Q_\Omega^{\boldsymbol{w}}[i,c](1-Q_\Omega^{\boldsymbol{w}}[i,c])$	$-\frac{1}{2\gamma}\mathbb{1}_{\{0<Q_\Omega^{\boldsymbol{w}}[i,c]<1\}}$	$-\frac{1}{\gamma}\left(\frac{1}{\sqrt{Q_\Omega^{\boldsymbol{w}}[i,c]}}+\frac{1}{\sqrt{1-Q_\Omega^{\boldsymbol{w}}[i,c]}}\right)^{-1}$

$4(q^*)^2-4q^*+(1-C^2)^2=0$. The roots are $q_\pm=(1\pm|C|\sqrt{2-C^2})/2$. To satisfy the sign of the original equation $\sqrt{q^*}-\sqrt{1-q^*}=C$, we must select the root such that $\operatorname{sgn}(\sqrt{q^*}-\sqrt{1-q^*})=\operatorname{sgn}(q^*-\frac{1}{2})=\operatorname{sgn}(C)$. Since $q_+\geq\frac{1}{2}$ and $q_-\leq\frac{1}{2}$, we must then choose q_+ when $C\geq 0$ and q_- when $C\leq 0$. This yields $q^*=\frac{1+C\sqrt{2-C^2}}{2}$.

Combining the boundary and interior cases and denoting $\bar{C}\triangleq\mathsf{clip}_{[-1,1]}(C)$, the closed-form solution is $q^*=\frac{1}{2}\left(1+\bar{C}\sqrt{2-\bar{C}^2}\right)$.

Partial derivative. Using $\omega''(q)=\gamma q^{-\frac{1}{2}}$, the derivative in the interior (where $|\bar{C}|<1$) is:

$$\frac{\partial q^*}{\partial a}=\left(\frac{\gamma}{\sqrt{q^*}}+\frac{\gamma}{\sqrt{1-q^*}}\right)^{-1}.$$

If $|\bar{C}|=1$, the derivative is 0.

Smoothed maximum. The value is obtained by substituting the optimal q^* into the objective. Using the definition $\Omega(\boldsymbol{q})=\frac{4\gamma}{3}(q^{\frac{3}{2}}+(1-q)^{\frac{3}{2}}-1)$, we obtain:

$$\mathsf{max}_\Omega(a,b)=q^*a+(1-q^*)b-\Omega(\boldsymbol{q}^*)=q^*a+(1-q^*)b+\frac{4\gamma}{3}\left(1-(q^*)^{\frac{3}{2}}-(1-q^*)^{\frac{3}{2}}\right).$$

Unlike the Gini case, this expression does not admit a simple polynomial form in terms of a and b, and is computed using the closed-form solution for q^* derived above.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly state the claims and contributions of the paper on the algorithmic, theoretical, and experimental levels.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a dedicated section in the conclusion of the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All proofs are included in the appendix, and provide the full set of needed assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main practical contributions of the paper are Algorithms 1 to 3, which are described in detail. The numerical experiments are carefully documented to ensure they are as reproducible as possible.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All data used is either public or synthetic with a clearly described generation process, and an open-source implementation of our differentiable operators (Algorithms 1 to 4) is available at https://github.com/google-deepmind/diff-knapsack.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We present the experimental settings with sufficient level of detail to fully appreciate the results in the core of the paper, and give full details in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We average experiments over multiple runs with varying seeds and report error bars where relevant in consequence, with statistical significance statements and precise details in appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include sufficient relevant information on the compute resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and ensured that all aspects of the research comply with its guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: As this is a foundational research paper, we do not foresee direct negative societal impacts in its current form. Potential negative impacts would depend on downstream applications, which are beyond the scope of this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: this work does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit original papers [Tang and Khalil, 2024, LeCun and Cortes, 2005] for the datasets we use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.