# Supplementary Material for Learning Optimal Tree Models under Beam Search

**Jingwei Zhuo** [1] **Ziru Xu** [1] **Wei Dai** [1] **Han Zhu** [1] **Han Li** [1] **Jian Xu** [1] **Kun Gai** [1]

This supplementary material consists of 3 sections: Sec. A introduces PLTs and TDMs in details and illustrates that both of them share the same training loss formulation; Sec. B proves Proposition 1 and Proposition 3, and derives the computational complexity of Algorithm 1 in details; Sec. C gives the detailed settings and additional results of experiments.

## A. Detailed Introduction of PLTs and TDMs

### A.1. Probabilistic Label Trees (PLTs)

PLTs formulate tree models $\mathcal{M}(\mathcal{T}, g)$ as hierarchical probability estimators for the marginal distribution $p(y_j|\mathbf{x})$ (Jain et al., 2016; Wydmuch et al., 2018). In PLTs, the pseudo target $z_n$ is defined as $z_n = \mathbb{I}(\sum_{n' \in \mathcal{L}(n)} y_{\pi(n')} \geq 1)$, which implies that $z_n = 1$ if and only if there exists $n' \in \mathcal{C}(n)$ such that $z_{n'} = 1$. In other words, $z_n = 1$ implies $z_{\rho(n)} = 1$. As a result, for any $n \in \mathcal{N}$, corresponding $p(z_n|\mathbf{x})$ can be decomposed as

$$p(z_n = 1|\mathbf{x}) = \prod_{n' \in \text{Path}(n)} p(z_{n'} = 1|z_{\rho(n')} = 1, \mathbf{x}). \tag{A.1}$$

Therefore, $p(y_j|\mathbf{x})$ can be represented as

$$p(y_j|\mathbf{x}) = p(z_{\pi^{-1}(j)}|\mathbf{x}) = \begin{cases} \prod_{n \in \text{Path}(\pi^{-1}(j))} p(z_n = 1|z_{\rho(n)} = 1, \mathbf{x}), & y_j = 1 \\ 1 - \prod_{n \in \text{Path}(\pi^{-1}(j))} p(z_n = 1|z_{\rho(n)} = 1, \mathbf{x}), & y_j = 0 \end{cases}. \tag{A.2}$$

According to Eq. (A.2), $\{p(y_j|\mathbf{x}) : j \in \mathcal{I}\}$ can be decomposed and represented by $\{p(z_n|z_{\rho(n)} = 1, \mathbf{x}) : n \in \mathcal{N}\}$. Leveraging this, PLTs transform the original probability estimation problem for $p(y_j|\mathbf{x})$ to a series of hierarchical estimation problems for $p(z_n|z_{\rho(n)} = 1, \mathbf{x})$, whose corresponding probability estimator is formulated as $p_g(z_n|z_{\rho(n)} = 1, \mathbf{x}) = 1/(1 + \exp(-(2z_n - 1)g(\mathbf{x}, n)))$. In other words, $g(\mathbf{x}, n)$ is trained as a binary classifier for $z_n \sim p(z_n|z_{\rho(n)} = 1, \mathbf{x})$. Given an instance $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{tr}$ where $\mathcal{D}_{tr}$ denotes the training dataset, $g(\mathbf{x}, n)$ is trained only on the node $n \in \mathcal{N}$ whose parent satisfies $z_{\rho(n)} = 1$. As a result, the loss function of PLTs can be denoted as $\sum_{(\mathbf{x},\mathbf{y}) \sim \mathcal{D}_{tr}} L(\mathbf{y}, \mathbf{g}(\mathbf{x}))$ with

$$\begin{aligned} L(\mathbf{y}, \mathbf{g}(\mathbf{x})) &= \sum_{n \in \mathcal{N}} \mathbb{I}(z_{\rho(n)} = 1) \ell_{\text{BCE}}(z_n, g(\mathbf{x}, n)) \\ &= \sum_{h=1}^{H} \sum_{n \in \mathcal{N}_h} \mathbb{I}(z_{\rho(n)} = 1) \ell_{\text{BCE}}(z_n, g(\mathbf{x}, n)) \\ &= \sum_{h=1}^{H} \sum_{n \in \mathcal{C}(n')} \sum_{n' \in \mathcal{N}_{h-1}} \mathbb{I}(z_{n'} = 1) \ell_{\text{BCE}}(z_n, g(\mathbf{x}, n)) \\ &= \sum_{h=1}^{H} \sum_{n \in \mathcal{S}_h(\mathbf{y})} \ell_{\text{BCE}}(z_n, g(\mathbf{x}, n)), \end{aligned} \tag{A.3}$$

where $\mathcal{S}_h(\mathbf{y}) = \{n : n \in \mathcal{C}(n'), n' \in \mathcal{N}_{h-1}, z_{n'} = 1\} = \{n : z_{\rho(n)} = 1, n \in \mathcal{N}_h\}$.

[1]Alibaba Group. Correspondence to: Jingwei Zhuo <zjw169463@alibaba-inc.com>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Vienna, Austria, PMLR 119, 2020. 

### A.2. Tree-based Deep Models (TDMs)

TDMs in the original paper (Zhu et al., 2018) only apply to the restricted case where $|\mathcal{I}_{\mathbf{x}}| = 1$, i.e., there exists only one target relevant to $\mathbf{x}$. For a training instance $(\mathbf{x}, \mathbf{y})$ satisfying $\sum_{j \in \mathcal{I}} y_j = 1$, let $j$ denote the single relevant target (i.e., $y_j = 1$), TDMs assign each $n \in \mathcal{N}$ a pseudo target $z_n = 1$ when $n$ is the ancestor node of $\pi^{-1}(j)$, and $z_n = 0$ when $n$ is not the ancestor node of $\pi^{-1}(j)$. They call $n \in \mathcal{N}$ a positive sample if $z_n = 1$ and $n$ a negative sample if $z_n = 0$.

According to the notations introduced in Sec. 2.1 of the main body, if $n$ is the ancestor node of $\pi^{-1}(j)$, the relationship between $n$ and $\pi^{-1}(j)$ can be represented as

$$\pi^{-1}(j) \in \mathcal{L}(n) \Leftrightarrow \sum_{n' \in \mathcal{L}(n)} y_{\pi(n')} \geq 1, \tag{A.4}$$

and thus the pseudo target $z_n$ defined in TDMs satisfies

$$z_n = \mathbb{I}(\pi^{-1}(j) \in \mathcal{L}(n)) = \mathbb{I}(\sum_{n' \in \mathcal{L}(n)} y_{\pi(n')} \geq 1), \tag{A.5}$$

which coincides with the pseudo target definition of PLTs[1].

Unlike PLTs, TDMs formulate tree models $\mathcal{M}(\mathcal{T}, g)$ to estimate $p(z_n|\mathbf{x})$ directly via $p_g(z_n|\mathbf{x}) = 1/(1 + \exp(-(2z_n - 1)g(\mathbf{x}, n))$. More specifically, $g(\mathbf{x}, n)$ is formulated as a binary classifier for $z_n \sim p(z_n|\mathbf{x})$ instead of $z_n \sim p(z_n|z_{\rho(n)} = 1, \mathbf{x})$. To guarantee logarithmic computational complexity in training, TDMs leverage the idea of negative sampling, and constitute the subsample set of negative samples by randomly selecting nodes except the positive one at each level. Let $\mathcal{S}_h^-(\mathbf{y})$ denote the subsample set of negative samples at $h$-th level and $\mathcal{S}_h^+(\mathbf{y}) = \{n : z_n = 1, n \in \mathcal{N}_h\}$ denote the set of the positive sample at $h$-th level, the training loss of TDMs can be denoted as $\sum_{(\mathbf{x},\mathbf{y}) \sim \mathcal{D}_{tr}} L(\mathbf{y}, \mathbf{g}(\mathbf{x}))$ where

$$\begin{aligned} L(\mathbf{y}, \mathbf{g}(\mathbf{x})) &= \sum_{h=1}^{H} \sum_{n \in \mathcal{S}_h^+(\mathbf{y}) \bigcup \mathcal{S}_h^-(\mathbf{y})} \ell_{\text{BCE}}(z_n, g(\mathbf{x}, n)) \\ &= \sum_{h=1}^{H} \sum_{n \in \mathcal{S}_h(\mathbf{y})} \ell_{\text{BCE}}(z_n, g(\mathbf{x}, n)), \end{aligned} \tag{A.6}$$

where $\mathcal{S}_h(\mathbf{y}) = \mathcal{S}_h^+(\mathbf{y}) \bigcup \mathcal{S}_h^-(\mathbf{y})$.

By far, we can find out that Eq. (A.3) and Eq. (A.6) follow the same formulation, which explains Eq. (2) in Sec. 3. 2.1 of the main body. Besides, noticing that both Eq. (A.5) and Eq. (A.6) do not require $|\mathcal{I}_{\mathbf{x}}| = 1$, TDMs can be naturally extended to multiple relevant targets case by removing the original restriction $|\mathcal{I}_{\mathbf{x}}| = 1$ without changing its training loss formulation.

## B. Detailed Derivations

### B.1. Proof of Proposition 1

Given the condition in Proposition 1 that Eq. (9) holds for any $\mathbf{x} \in \mathcal{X}$ and any $n \in \bigcup_{h=1}^{H} \tilde{\mathcal{B}}_h(\mathbf{x})$ with beam size $k$, our proof is divided into two parts: (1) proving $\mathcal{M}(\mathcal{T}, g)$ is top-$m$ Bayes optimal under beam search when $m = k$; (2) proving $\mathcal{M}(\mathcal{T}, g)$ is top-$m$ Bayes optimal under beam search for any $m < k$. Notice that proving $\mathcal{M}(\mathcal{T}, g)$ is Bayes optimal under beam search can be regarded as the beam size $k = M$ case of our proof, since $\tilde{\mathcal{B}}_h(\mathbf{x}) = \mathcal{B}_h(\mathbf{x}) = \mathcal{N}_h$ holds for any $1 \leq h \leq H$ when $k = M$. Besides, our proof does not rely on the assumption that there are no ties[2] among $\{\eta_j(\mathbf{x}) : j \in \mathcal{I}\}$ (that is, for any $i, j \in \mathcal{I}$ with $i \neq j$, $\eta_i(\mathbf{x}) \neq \eta_j(\mathbf{x})$), which makes our proof more general with the price that $\text{argmax}$ and $\text{argTopm}$ operators may have multiple solutions and thus the proof becomes more complex.

[1] In Eq. (A.5), under the restriction $\sum_{j \in \mathcal{I}} y_j = 1$, an equivalent representation of $\sum_{j \in \mathcal{I}} y_j \geq 1$ is $\sum_{j \in \mathcal{I}} y_j = 1$.

[2] This is a common assumption in previous papers, e.g., Lapin et al. (2017).

### B.1.1. PROOF OF THE $m = k$ CASE

Let $\mathcal{L}(\mathcal{B}_h(\mathbf{x})) = \bigcup_{n\in\mathcal{B}_h(\mathbf{x})} \mathcal{L}(n)$ denote the leaf node set of every subtree rooted at $n \in \mathcal{B}_h(\mathbf{x})$, Eq. (8) in Definition 1, i.e., $\{\pi(n) : n \in \mathcal{B}_H(\mathbf{x})\} \in \operatorname{argTopk}_{j\in\mathcal{I}} \eta_j(\mathbf{x})$, can be rewritten as

$$\{\mathcal{B}_H(\mathbf{x})\} = \operatorname*{argTopk}_{n\in\mathcal{B}_H(\mathbf{x})} \eta_{\pi(n)}(\mathbf{x}) = \operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_H(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}) \subset \operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_0(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}) = \operatorname*{argTopk}_{n\in\mathcal{N}_H} \eta_{\pi(n)}(\mathbf{x}). \tag{A.7}$$

In Eq. (A.7), the first equality holds since $|\mathcal{B}_H(\mathbf{x})| \le k$ and thus $\mathcal{B}_H(\mathbf{x})$ is the unique solution of $\operatorname{argTopk}_{n\in\mathcal{B}_H(\mathbf{x})} \eta_{\pi(n)}(\mathbf{x})$, the second equality holds since $\mathcal{L}(\mathcal{B}_H(\mathbf{x})) = \mathcal{B}_H(\mathbf{x})$, the subset relationship holds since $\mathcal{B}_0(\mathbf{x})$ contains ancestor nodes of $\mathcal{B}_H(\mathbf{x})$ and thus $\mathcal{L}(\mathcal{B}_H(\mathbf{x})) \subset \mathcal{L}(\mathcal{B}_0(\mathbf{x}))$, and the last equality holds since $\mathcal{B}_0(\mathbf{x}) = \{r(\mathcal{T})\}$ and $\mathcal{L}(\{r(\mathcal{T})\}) = \mathcal{N}_H$.

As a result, Eq. (A.7) can be proved by showing that pruning $\tilde{\mathcal{B}}_h(\mathbf{x}) \setminus \mathcal{B}_h(\mathbf{x})$ according to Eq. (3) does not lead to the retrieval performance deterioration where the top-$k$ targets w.r.t. $\eta_{\pi(n)}(\mathbf{x})$ among $\mathcal{L}(\mathcal{B}_h(\mathbf{x}))$ are also the top-$k$ targets w.r.t. $\eta_{\pi(n)}(\mathbf{x})$ among $\mathcal{N}_H$ (i.e., $\mathcal{I}$), i.e.,

$$\operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_H(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}) \subset \cdots \subset \operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_h(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}) \subset \cdots \subset \operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_0(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}), \tag{A.8}$$

which corresponds to showing

$$\operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_h(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}) \subset \operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_{h-1}(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}), \tag{A.9}$$

holds for any $\mathbf{x} \in \mathcal{X}$ and any $1 \le h \le H$.

For $h$ such that $|\mathcal{N}_h| \le k$, i.e., the number of nodes at $h$-th level is not larger than $k$, all nodes at $h$-th level are regarded as the beam, which corresponds to $\mathcal{B}_h(\mathbf{x}) = \mathcal{N}_h$. In this case, $|\mathcal{N}_{h-1}| \le k$ also holds, which implies $\mathcal{L}(\mathcal{B}_h(\mathbf{x})) = \mathcal{L}(\mathcal{N}_h) = \mathcal{L}(\mathcal{N}_{h-1}) = \mathcal{L}(\mathcal{B}_{h=1}(\mathbf{x}))$ and thus Eq. (A.9) always holds for any $\mathbf{x} \in \mathcal{X}$.

For $h$ such that $|\mathcal{N}_h| > k$ and thus $|\mathcal{B}_h(\mathbf{x})| = k$, Eq. (A.9) can be proved by contradiction. Let $\mathcal{V}_h(\mathbf{x}) \in \operatorname{argTopk}_{n\in\mathcal{L}(\mathcal{B}_h(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x})$ denote the set of top-$k$ nodes among $\mathcal{L}(\mathcal{B}_h(\mathbf{x}))$, we assume that there exists $\mathbf{x} \in \mathcal{X}$ such that Eq. (A.9) does not holds, i.e.,

$$\operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_h(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}) \not\subseteq \operatorname*{argTopk}_{n\in\mathcal{L}(\mathcal{B}_{h-1}(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x}). \tag{A.10}$$

According to the definition of $\mathcal{L}(\mathcal{B}_h(\mathbf{x}))$, $\mathcal{L}(\mathcal{B}_h(\mathbf{x})) \subset \mathcal{L}(\mathcal{B}_{h-1}(\mathbf{x}))$, which indicates that both $\mathcal{V}_h(\mathbf{x})$ and $\mathcal{V}_{h-1}(\mathbf{x})$ are subsets of $\mathcal{L}(\mathcal{B}_{h-1}(\mathbf{x}))$. As a result, Eq. (A.10) holds if and only if

$$\sum_{n\in\mathcal{V}_{h-1}(\mathbf{x})} \eta_{\pi(n)}(\mathbf{x}) > \sum_{n'\in\mathcal{V}_h(\mathbf{x})} \eta_{\pi(n')}(\mathbf{x}). \tag{A.11}$$

Eq. (A.11) implies that[3] $\mathcal{V}_{h-1}(\mathbf{x}) \setminus \mathcal{L}(\mathcal{B}_h(\mathbf{x})) \neq \emptyset$ and there exists $n_0 \in \mathcal{V}_{h-1}(\mathbf{x}) \setminus \mathcal{L}(\mathcal{B}_h(\mathbf{x}))$ such that

$$\eta_{\pi(n_0)}(\mathbf{x}) > \min_{n'\in\mathcal{V}_h(\mathbf{x})} \eta_{\pi(n')}(\mathbf{x}), \tag{A.12}$$

otherwise $\sum_{n\in\mathcal{V}_h(\mathbf{x})} \eta_{\pi(n)}(\mathbf{x}) = \sum_{n'\in\mathcal{V}'_h(\mathbf{x})} \eta_{\pi(n)}(\mathbf{x})$.

Let $\rho^{H-h}(n_0) = (\rho \circ \cdots \circ \rho)(n_0)$ denote the ancestor node of $n_0$ at $h$-th level, $n_0 \in \mathcal{V}_{h-1}(\mathbf{x}) \setminus \mathcal{L}(\mathcal{B}_h(\mathbf{x}))$ implies that $\rho^{H-h}(n_0) \in \tilde{\mathcal{B}}_h(\mathbf{x}) \setminus \mathcal{B}_h(\mathbf{x})$. According to the definition of $\mathcal{B}_h(\mathbf{x})$ in Eq. (3), we have

$$p_g(z_n = 1|\mathbf{x}) \ge p_g(z_{\rho^{H-h}(n_0))} = 1|\mathbf{x}) \ge \eta_{\pi(n_0)}(\mathbf{x}), \ \ \forall n \in \mathcal{B}_h(\mathbf{x}). \tag{A.13}$$

Recall that $p_g(z_n = 1|\mathbf{x}) = \max_{n'\in\mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x})$ holds for any $n \in \tilde{\mathcal{B}}_h(\mathbf{x})$ according to Eq. (9) of Proposition 1, Eq. (A.13) can be rewritten as

$$\max_{n'\in\mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x}) \ge \max_{n'\in\mathcal{L}(\rho^{H-h}(n_0)))} \eta_{\pi(n')}(\mathbf{x}) \ge \eta_{\pi(n_0)}(\mathbf{x}), \ \ \forall n \in \mathcal{B}_h(\mathbf{x}), \tag{A.14}$$

where the last inequality holds since $n_0 \in \mathcal{L}(\rho^{H-h}(n_0))$.

[3] Otherwise $\mathcal{V}_{h-1}(\mathbf{x}) \subset \mathcal{L}(\mathcal{B}_h(\mathbf{x}))$ which violates that $\mathcal{V}_h(\mathbf{x}) \in \operatorname{argTopk}_{n\in\mathcal{L}(\mathcal{B}_h(\mathbf{x}))} \eta_{\pi(n)}(\mathbf{x})$.

According to Eq. (A.14), for any $n \in \mathcal{B}_h(\mathbf{x})$, there exists at least one $n' \in \mathcal{L}(n)$ such that $\eta_{\pi(n')}(\mathbf{x}) = \max_{n' \in \mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x}) \geq \eta_{\pi(n_0)(\mathbf{x})}$. In other words, let $\mathcal{W}_h(\mathbf{x}) = \{n' : n' \in \operatorname{argmax}_{n' \in \mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x}), n \in \mathcal{B}_h(\mathbf{x})\} \subset \mathcal{L}(\mathcal{B}_h(\mathbf{x}))$ denote such nodes, we have $|\mathcal{W}_h(\mathbf{x})| \geq k$ and $\min_{n' \in \mathcal{W}_h(\mathbf{x})} \eta_{\pi(n')}(\mathbf{x}) \geq \eta_{\pi(n_0)}(\mathbf{x})$.

Since $\mathcal{V}_h(\mathbf{x})$ denotes the top-$k$ nodes among $\mathcal{L}(\mathcal{B}_h(\mathbf{x}))$, we have

$$\min_{n' \in \mathcal{V}_h(\mathbf{x})} \eta_{\pi(n')}(\mathbf{x}) \geq \min_{n' \in \mathcal{W}_h(\mathbf{x})} \eta_{\pi(n')}(\mathbf{x}) \geq \eta_{\pi(n_0)}(\mathbf{x}). \tag{A.15}$$

It is obvious that Eq. (A.15) contradicts with Eq. (A.12). Therefore, the assumption does not hold and Eq. (A.9) always holds. By doing so, we have proven that Eq. (A.11) holds for any $\mathbf{x} \in \mathcal{X}$ and $1 \leq h \leq H$, which indicates $\mathcal{M}(\mathcal{T}, g)$ is top-$m$ Bayes optimal under beam search when $m = k$.

### B.1.2. Proof of The $m < k$ Case

The $m < k$ case can be proved by reusing the proof of the $m = k$ case. To see this, let $\mathcal{B}_h^{(m)}(\mathbf{x}) \in \operatorname{argTopm}_{n \in \mathcal{B}_h(\mathbf{x})} p_g(z_n = 1|\mathbf{x})$ denote the set of top-$m$ nodes w.r.t. $p_g(z_n = 1|\mathbf{x})$ among $\mathcal{B}_h(\mathbf{x})$, which may not be the unique solution since we do not assume there exists no ties, we introduce a lemma as follows:

**Lemma 1.** *Suppose that a tree model $\mathcal{M}(\mathcal{T}, g)$ satisfies Eq. (9) for any $\mathbf{x} \in \mathcal{X}$ and $n \in \bigcup_{h=1}^{H} \tilde{\mathcal{B}}_h(\mathbf{x})$ with beam size $k$. For any $\mathbf{x} \in \mathcal{X}$, $1 \leq h \leq H$, $1 \leq m \leq k$ and $\mathcal{B}_h^{(m)}(\mathbf{x})$, there always exists $\mathcal{B}_{h-1}^{(m)}(\mathbf{x})$ such that*

$$\mathcal{B}_h^{(m)}(\mathbf{x}) \in \operatorname*{argTopm}_{n \in \tilde{\mathcal{B}}_h^{(m)}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}), \quad \tilde{\mathcal{B}}_h^{(m)}(\mathbf{x}) = \bigcup_{n' \in \mathcal{B}_{h-1}^{(m)}(\mathbf{x})} \mathcal{C}(n'). \tag{A.16}$$

*Proof.* Eq. (A.16) follows the same formulation as Eq. (3) with the only difference in replacing $\mathcal{B}_h(\mathbf{x})$ with $\mathcal{B}_h^{(m)}$.

For $h$ such that $|\mathcal{B}_h(\mathbf{x})| \leq m$, we have $\mathcal{B}_h^{(m)}(\mathbf{x}) = \mathcal{B}_h(\mathbf{x})$ and thus there always exists $\mathcal{B}_{h-1}^{(m)}(\mathbf{x}) = \mathcal{B}_{h-1}(\mathbf{x})$ such that Eq. (A.16) holds.

For $h$ such that $|\mathcal{B}_h(\mathbf{x})| > m$, since Eq. (9) holds for any $n \in \bigcup_{h=1}^{H} \tilde{\mathcal{B}}_h(\mathbf{x})$, we have

$$p_g(z_n = 1|\mathbf{x}) = \max_{n' \in \mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x}) = \max_{n' \in \mathcal{C}(n)} \max_{n'' \in \mathcal{L}(n')} \eta_{\pi(n'')}(\mathbf{x}) = \max_{n' \in \mathcal{C}(n)} p_g(z_{n'} = 1|\mathbf{x}), \quad \forall n \in \bigcup_{h=1}^{H} \mathcal{B}_h(\mathbf{x}). \tag{A.17}$$

For any $\mathcal{B}_h^{(m)}(\mathbf{x}) \in \operatorname{argTopm}_{n \in \mathcal{B}_h(\mathbf{x})} p_g(z_n = 1|\mathbf{x})$, since $\mathcal{B}_h(\mathbf{x}) \in \operatorname{argTopk}_{n \in \tilde{\mathcal{B}}_h(\mathbf{x})} p_g(z_n = 1|\mathbf{x})$, we have $\mathcal{B}_h^{(m)}(\mathbf{x}) \in \operatorname{argTopm}_{n \in \tilde{\mathcal{B}}_h(\mathbf{x})} p_g(z_n = 1|\mathbf{x})$, i.e., $\mathcal{B}_h^{(m)}(\mathbf{x})$ is also the set of top-$m$ nodes among $\tilde{\mathcal{B}}_h(\mathbf{x})$, which is equivalent to

$$\min_{n \in \mathcal{B}_h^{(m)}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}) \geq \max_{n \in \tilde{\mathcal{B}}_h(\mathbf{x}) \setminus \mathcal{B}_h^{(m)}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}). \tag{A.18}$$

As a result, let $\mathcal{A}_{h-1}(\mathbf{x}) = \{\rho(n) : n \in \mathcal{B}_h^{(m)}(\mathbf{x})\} \subset \mathcal{B}_{h-1}(\mathbf{x})$ denote the parent node set of $\mathcal{B}_h^{(m)}(\mathbf{x})$, we have

$$\begin{aligned}
\min_{n \in \mathcal{A}_{h-1}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}) &= \min_{n \in \mathcal{A}_{h-1}(\mathbf{x})} \max_{n' \in \mathcal{C}(n)} p_g(z_{n'} = 1|\mathbf{x}) \\
&\geq \min_{n' \in \mathcal{B}_h^{(m)}(\mathbf{x})} p_g(z_{n'} = 1|\mathbf{x}) \\
&\geq \max_{n' \in \tilde{\mathcal{B}}_h(\mathbf{x}) \setminus \mathcal{B}_h^{(m)}(\mathbf{x})} p_g(z_{n'} = 1|\mathbf{x}) \\
&\geq \max_{n \in \mathcal{B}_{h-1}(\mathbf{x}) \setminus \mathcal{A}_{h-1}(\mathbf{x})} \max_{n' \in \mathcal{C}(n)} p_g(z_{n'} = 1|\mathbf{x}) \\
&= \max_{n' \in \mathcal{B}_{h-1}(\mathbf{x}) \setminus \mathcal{A}_{h-1}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}).
\end{aligned} \tag{A.19}$$

In Eq. (A.19), the first equality and the last equality holds because of Eq. (A.17), the first inequality holds since[4] $\operatorname{argmax}_{n' \in \mathcal{C}(n)} p_g(z_{n'} = 1|\mathbf{x}) \bigcap \mathcal{B}_h^{(m)}(\mathbf{x}) \neq \emptyset$ always holds for any $n \in \mathcal{A}_{h-1}(\mathbf{x})$, the second inequality holds because of Eq. (A.18), and the last inequality holds since $\{n' : n' \in \mathcal{C}(n),\ n \in \mathcal{B}_{h-1}(\mathbf{x}) \setminus \mathcal{A}_{h-1}(\mathbf{x})\} \subset \tilde{\mathcal{B}}_h(\mathbf{x}) \setminus \mathcal{B}_h^{(m)}(\mathbf{x})$.

[4] Otherwise there exists $n \in \mathcal{A}_{h-1}(\mathbf{x})$ such that for any $n' \in \operatorname{argmax}_{n' \in \mathcal{C}(n)} p_g(z_{n'} = 1|\mathbf{x})$, $p_g(z_{n'} = 1|\mathbf{x}) > p_g(z_{n''} = 1|\mathbf{x})$ always holds for any $n'' \in \mathcal{C}(n) \bigcap \mathcal{B}_h^{(m)}(\mathbf{x})$, which violates Eq. (A.18).

Now, suppose that $\mathcal{A}'_h(\mathbf{x})$ denotes the top-$(m - |\mathcal{A}_h(\mathbf{x})|)$ nodes among $\mathcal{B}_{h-1}(\mathbf{x}) \setminus \mathcal{A}_{h-1}(\mathbf{x})$, i.e.,

$$\min_{n \in \mathcal{A}'_{h-1}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}) \geq \max_{n' \in \mathcal{B}_{h-1}(\mathbf{x}) \setminus \mathcal{A}_{h-1}(\mathbf{x}) \setminus \mathcal{A}'_{h-1}(\mathbf{x})} p_g(z_{n'} = 1|\mathbf{x}), \tag{A.20}$$

we have $|\mathcal{A}_{h-1}(\mathbf{x}) \bigcup \mathcal{A}'_{h-1}(\mathbf{x})| = m$ and

$$\min_{n \in \mathcal{A}_{h-1}(\mathbf{x}) \bigcup \mathcal{A}'_{h-1}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}) \geq \max_{n' \in \mathcal{B}_{h-1}(\mathbf{x}) \setminus \mathcal{A}_{h-1}(\mathbf{x}) \setminus \mathcal{A}'_{h-1}(\mathbf{x})} p_g(z_{n'} = 1|\mathbf{x}), \tag{A.21}$$

which is equivalent to

$$\mathcal{A}_{h-1}(\mathbf{x}) \bigcup \mathcal{A}'_{h-1}(\mathbf{x}) \in \operatorname*{argTopm}_{n \in \mathcal{B}_{h-1}(\mathbf{x})} p_g(z_n = 1|\mathbf{x}). \tag{A.22}$$

In other words, there always exists $\mathcal{B}_h^{(m)}(\mathbf{x}) = \mathcal{A}_{h-1}(\mathbf{x}) \bigcup \mathcal{A}'_{h-1}(\mathbf{x})$ such that Eq. (A.16) holds. Therefore Lemma 1 has been proved. □

Lemma 1 indicates a nice property of $\mathcal{M}(\mathcal{T}, g)$ in Proposition 1: The top-$m$ nodes among $\mathcal{B}_h(\mathbf{x})$, i.e., $\mathcal{B}_h^{(m)}(\mathbf{x})$, can be regarded as the generated beam of the top-$m$ nodes among $\mathcal{B}_{h-1}(\mathbf{x})$, i.e., $\mathcal{B}_{h-1}^{(m)}(\mathbf{x})$. Besides, according to the definition of $\mathcal{B}_h^{(m)}(\mathbf{x})$, $\mathcal{B}_h^{(m)}(\mathbf{x}) \subset \mathcal{B}_h(\mathbf{x})$ always holds, which implies that Eq. (9) also holds for any $n \in \bigcup_{h=1}^H \tilde{\mathcal{B}}_h^{(m)}(\mathbf{x})$ given the condition of Proposition 1. Combining these two together, for any $\mathbf{x} \in \mathcal{X}$, there exists $\{\mathcal{B}_h^{(m)}(\mathbf{x})\}_{h=1}^H$ satisfying Eq. (A.16) such that Eq. (9) also holds for any $n \in \bigcup_{h=1}^H \tilde{\mathcal{B}}_h^{(m)}(\mathbf{x})$. Therefore, the top-$m$ Bayes optimality under beam search of $\mathcal{M}(\mathcal{T}, g)$ for any $m < k$ can be proved by reusing the proof in Sec. B.1.1 to show that $\{\pi(n) : n \in \mathcal{B}_H^{(m)}(\mathbf{x})\} \subset \operatorname{argTopm}_{j \in \mathcal{I}} \eta_j(\mathbf{x})$.

Combining the proof of the $m = k$ case and the $m < k$ case, we have proven Proposition 1.

### B.2. Proof of Proposition 3

Let $p_{g_{\boldsymbol{\theta}}}(z_n|\mathbf{x}) = 1/(1 + \exp(-(2z_n - 1)g_{\boldsymbol{\theta}}(\mathbf{x}, n)))$, the loss function in Eq. (18) of Proposition 3 can be rewritten as

$$\begin{aligned}
& \mathbb{E}_{p(\mathbf{x},\mathbf{y})} \left[ L^*_{\boldsymbol{\theta}_t}(\mathbf{y}, \mathbf{g}(\mathbf{x}); \boldsymbol{\theta}) \right] \\
= \; & \mathbb{E}_{p(\mathbf{x},\mathbf{y})} \left[ \sum_{h=1}^H \sum_{n \in \mathcal{N}_h} w_n(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}_t) \Big( - \hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t) \log p_{g_{\boldsymbol{\theta}}}(z_n = 1|\mathbf{x}) - (1 - \hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t)) \log p_{g_{\boldsymbol{\theta}}}(z_n = 0|\mathbf{x}) \Big) \right] \\
= \; & \mathbb{E}_{p(\mathbf{x})} \left[ \sum_{h=1}^H \sum_{n \in \mathcal{N}_h} w_n(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}_t) \Big( - \mathbb{E}_{p(\mathbf{y}|\mathbf{x})} \left[ \hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t) \right] \log p_{g_{\boldsymbol{\theta}}}(z_n = 1|\mathbf{x}) - (1 - \mathbb{E}_{p(\mathbf{y}|\mathbf{x})} \left[ \hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t) \right]) \log p_{g_{\boldsymbol{\theta}}}(z_n = 0|\mathbf{x}) \Big) \right] \\
= \; & \mathbb{E}_{p(\mathbf{x})} \left[ \sum_{h=1}^H \sum_{n \in \mathcal{N}_h} w_n(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}_t) \Big( \mathrm{KL} \left( \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x}) \| p_{g_{\boldsymbol{\theta}}}(z_n|\mathbf{x}) \right) + \mathrm{H}(\tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x})) \Big) \right],
\end{aligned} \tag{A.23}$$

where

$$\tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x}) = \begin{cases} \mathbb{E}_{p(\mathbf{y}|\mathbf{x})} \left[ \hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t) \right], & z_n = 1 \\ 1 - \mathbb{E}_{p(\mathbf{y}|\mathbf{x})} \left[ \hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t) \right], & z_n = 0 \end{cases}, \tag{A.24}$$

and $\mathrm{H}(\tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x})) = -\tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) \log \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) - \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n = 0|\mathbf{x}) \log \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n = 0|\mathbf{x})$ denotes the entropy of $p_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x})$. Since $\mathrm{H}(\tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x}))$ can be regarded as a constant term with respect to $\boldsymbol{\theta}$, Eq. (18) can be rewritten as

$$\boldsymbol{\theta}_t \in \operatorname*{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{p(\mathbf{x})} \left[ \sum_{h=1}^H \sum_{n \in \mathcal{N}_h} w_n(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}_t) \mathrm{KL} \left( \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x}) \| p_{g_{\boldsymbol{\theta}}}(z_n|\mathbf{x}) \right) \right]. \tag{A.25}$$

According to the definition of KL divergence, the minimizer of $\mathrm{KL} \left( \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x}) \| p_{g_{\boldsymbol{\theta}}}(z_n|\mathbf{x}) \right)$ satisfies $p_{g_{\boldsymbol{\theta}}}(z_n|\mathbf{x}) = \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x})$. Since $\mathcal{G}$ has enough capacity (e.g., infinite capacity with non-parametric limit), such a minimizer can be obtained in $\boldsymbol{\Theta}$. Therefore, if Eq. (A.25) holds, we have $p_{g_{\boldsymbol{\theta}}}(z_n|\mathbf{x}) = \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x})$ when $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ for any $\mathbf{x} \in \mathcal{X}$ and $n \in \mathcal{N}$.

For any $n \in \mathcal{N} \setminus \mathcal{N}_H$, we have

$$\begin{aligned} p_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) &= \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) \\ &= \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}\left[\hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t)\right] \\ &= \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}\left[\hat{z}_{n'}(\mathbf{x}; \boldsymbol{\theta}_t)\right],\ n' \in \operatorname*{argmax}_{n' \in \mathcal{C}(n)} p_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}) \\ &= \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}),\ n' \in \operatorname*{argmax}_{n' \in \mathcal{C}(n)} p_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}) \\ &= p_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}),\ n' \in \operatorname*{argmax}_{n' \in \mathcal{C}(n)} p_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}) \\ &= \max_{n' \in \mathcal{C}(n)} p_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}),\ \forall \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{A.26}$$

For any $n \in \mathcal{N}_H$, $\hat{z}_n(\mathbf{x}; \boldsymbol{\theta}_t) = y_{\pi(n)}$ according to Eq. (15), and thus we have

$$p_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) = \tilde{p}_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}\left[y_{\pi(n)}\right] = \eta_{\pi(n)}(\mathbf{x}),\ \forall \mathbf{x} \in \mathcal{X}. \tag{A.27}$$

Now, assuming that $p_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) = \max_{n' \in \mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x})$ holds for any $n \in \mathcal{N}_{h+1}$, for any $n \in \mathcal{N}_h$, we have

$$\begin{aligned} p_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) &= \max_{n' \in \mathcal{C}(n)} p_{g_{\boldsymbol{\theta}_t}}(z_{n'} = 1|\mathbf{x}) \\ &= \max_{n' \in \mathcal{C}(n)} \max_{n'' \in \mathcal{L}(n')} \eta_{\pi(n'')}(\mathbf{x}) \\ &= \max_{n'' \in \mathcal{L}(n)} \eta_{\pi(n'')(\mathbf{x})}. \end{aligned} \tag{A.28}$$

By doing so, we have proven that $p_{g_{\boldsymbol{\theta}_t}}(z_n = 1|\mathbf{x}) = \max_{n' \in \mathcal{L}(n)} \eta_{\pi(n')}(\mathbf{x})$ holds for any $\mathbf{x} \in \mathcal{X}$ and $n \in \mathcal{N}$, i.e., $p_{g_{\boldsymbol{\theta}_t}}(z_n|\mathbf{x}) = \tilde{p}(z_n|\mathbf{x})$ holds for any $\mathbf{x} \in \mathcal{X}$ and $n \in \mathcal{N}$. According to Proposition 1, we can conclude that $\mathcal{M}(\mathcal{T}, g_{\boldsymbol{\theta}_t})$ is Bayes optimal under beam search.

### B.3. Computational Complexity of Algorithm 1

In Algorithm 1, the computational complexity depends on step 4 and step 5, where the former retrieves nodes according to beam search and the latter estimates the optimal pseudo target for each node retrieved by beam search. Recall that $(\mathbf{x}, \mathbf{y})$ denotes an instance and $\mathbf{y} \in \{0, 1\}^M$ is an equivalent representation of $\mathcal{I}_\mathbf{x}$ (the relevant target subset), we analyze the complexity per instance as follows.

For step 4, there are at most $bk$ nodes needed to be queried at each level according to Eq. (3) and the tree has $H$ levels. Therefore, its complexity is $\mathrm{O}(Hbk)$.

For step 5, according to the definition in Eq. (15), $\hat{z}_n(x; \theta) = 0$ always holds if $\mathcal{L}(n) \bigcap \mathcal{I}_\mathbf{x} = \emptyset$, i.e., $n \notin \mathcal{S}_h^+(\mathbf{y})$ for any $h \in \{1, ..., H\}$. Therefore, at the $h$-th level, we only need to compute $\hat{z}_n(\mathbf{x}; \boldsymbol{\theta})$ for $n \in \tilde{\mathcal{B}}_h(\mathbf{x}; \boldsymbol{\theta}) \bigcap \mathcal{S}_h^+(\mathbf{y})$ and set $\hat{z}_n(\mathbf{x}; \boldsymbol{\theta}) = 0$ directly for $n \in \tilde{\mathcal{B}}_h(\mathbf{x}; \boldsymbol{\theta}) \setminus \mathcal{S}_h^+(\mathbf{y})$. In the worst case, we need to compute $\hat{z}_n(\mathbf{x}; \boldsymbol{\theta})$ for each $n \in \mathcal{S}_h^+(\mathbf{y})$ at the $h$-th level, which can be computed recursively in a bottom-up manner. According to Eq. (15), computing $\hat{z}_n(\mathbf{x}; \boldsymbol{\theta})$ needs to query the children set $\mathcal{C}(n)$ of $n \in \mathcal{S}_h^+(\mathbf{y})$, where $\hat{z}_{n'}(\mathbf{x}; \boldsymbol{\theta})$ for $n' \in \mathcal{C}(n) \bigcap \mathcal{S}_{h+1}^+(\mathbf{y})$ has been computed while that for $n' \in \mathcal{C}(n) \setminus \mathcal{S}_{h+1}^+(\mathbf{y})$ is always zero. As a result, it needs to query $|\mathcal{C}(n)| \leq b$ nodes. Since $|\mathcal{S}_h^+(\mathbf{y})| \leq |\mathcal{I}_\mathbf{x}|$ and the tree has $H$ levels, the complexity of step 5 is $\mathrm{O}(Hb|\mathcal{I}_\mathbf{x}|)$.

To conclude, the computational complexity of Algorithm 1 is $\mathrm{O}(Hbk + Hb|\mathcal{I}_\mathbf{x}|)$.

## C. Experiments

### C.1. Toy Example

The toy example in Sec. 4.1 investigates the retrieval performance of a tree model $\mathcal{M}(\mathcal{T}, g)$ whose pseudo targets are defined in Eq. (1). Given the training dataset $\mathcal{D}_{tr} = \{\mathbf{y}^{(i)}\}_{i=1}^N$, $\mathcal{M}(\mathcal{T}, g)$ is trained to estimate the node-wise probability of $z_n$ directly via $p_g(z_n = 1) = \sum_{i=1}^N z_n^{(i)}/N$, where $z_n^{(i)} = \mathbb{I}(\sum_{n' \in \mathcal{L}(n)} y_{\pi(n)}^{(i)} \geq 1)$. Table 1 shows $\mathcal{M}(\mathcal{T}, g)$ with such $p_g(z_n = 1)$ have non-zero regret in general, which corresponds to the retrieval performance deterioration.

*Table A.1.* Results for the toy experiment with $M = 1000$, $b = 2$. The reported number is $\text{reg}_{p@m}(\mathcal{M})$ under different hyperparameter settings of $m$, $k$ and $N$, and is averaged over 100 runs with random initialization over $\mathcal{T}$ and $\eta_j$.

| k | m | DirEst | | | | HierEst | | | | OptEst | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 100 | 1000 | 10000 | $\infty$ | 100 | 1000 | 10000 | $\infty$ | 100 | 1000 | 10000 | $\infty$ |
| 1 | 1 | 0.088 | 0.083 | 0.079 | 0.059 | 0.093 | 0.078 | 0.081 | 0.059 | 0.009 | 0.000 | 0.000 | 0.000 |
| 5 | 1 | 0.023 | 0.013 | 0.012 | 0.007 | 0.021 | 0.012 | 0.011 | 0.007 | 0.009 | 0.000 | 0.000 | 0.000 |
| 5 | 5 | 0.073 | 0.052 | 0.044 | 0.032 | 0.071 | 0.051 | 0.046 | 0.032 | 0.007 | 0.000 | 0.000 | 0.000 |
| 10 | 1 | 0.014 | 0.006 | 0.004 | 0.002 | 0.014 | 0.006 | 0.005 | 0.002 | 0.008 | 0.001 | 0.000 | 0.000 |
| 10 | 5 | 0.031 | 0.019 | 0.015 | 0.008 | 0.031 | 0.018 | 0.016 | 0.008 | 0.007 | 0.000 | 0.000 | 0.000 |
| 10 | 10 | 0.064 | 0.046 | 0.039 | 0.023 | 0.063 | 0.045 | 0.039 | 0.023 | 0.005 | 0.001 | 0.000 | 0.000 |
| 20 | 1 | 0.010 | 0.003 | 0.002 | 0.001 | 0.011 | 0.003 | 0.002 | 0.001 | 0.009 | 0.001 | 0.000 | 0.000 |
| 20 | 5 | 0.017 | 0.008 | 0.006 | 0.002 | 0.017 | 0.008 | 0.006 | 0.002 | 0.007 | 0.000 | 0.000 | 0.000 |
| 20 | 10 | 0.028 | 0.015 | 0.013 | 0.006 | 0.028 | 0.016 | 0.013 | 0.006 | 0.005 | 0.001 | 0.000 | 0.000 |
| 20 | 20 | 0.059 | 0.038 | 0.033 | 0.020 | 0.060 | 0.039 | 0.033 | 0.020 | 0.005 | 0.000 | 0.000 | 0.000 |
| 50 | 1 | 0.009 | 0.001 | 0.000 | 0.000 | 0.009 | 0.001 | 0.000 | 0.000 | 0.009 | 0.001 | 0.000 | 0.000 |
| 50 | 5 | 0.008 | 0.001 | 0.001 | 0.000 | 0.009 | 0.002 | 0.001 | 0.000 | 0.007 | 0.000 | 0.000 | 0.000 |
| 50 | 10 | 0.011 | 0.002 | 0.001 | 0.000 | 0.011 | 0.003 | 0.001 | 0.000 | 0.005 | 0.001 | 0.000 | 0.000 |
| 50 | 20 | 0.017 | 0.005 | 0.003 | 0.001 | 0.017 | 0.005 | 0.003 | 0.001 | 0.005 | 0.000 | 0.000 | 0.000 |
| 50 | 50 | 0.042 | 0.021 | 0.016 | 0.011 | 0.042 | 0.021 | 0.016 | 0.011 | 0.005 | 0.001 | 0.000 | 0.000 |

In this subsection, we provide additional experimental results for this toy example. More specifically, we consider three different methods for building $p_g(z_n = 1)$, i.e.,

- Direct Estimator (DirEst)[5]: $p_g(z_n = 1) = \sum_{i=1}^N z_n^{(i)}/N$, where $z_n^{(i)} = \mathbb{I}(\sum_{n' \in \mathcal{L}(n)} y_{\pi(n')}^{(i)} \geq 1)$ (i.e., Eq. (1));
- Hierarchical Estimator (HierEst): $p_g(z_n = 1) = \prod_{n' \in \text{Path}(n)} p_g(z_{n'} = 1 | z_{\rho(n')} = 1)$ with $p_g(z_n = 1 | z_{\rho(n)} = 1) = \sum_{i=1}^N z_n^{(i)} z_{\rho(n)}^{(i)} / \sum_{i=1}^N z_{\rho(n)}^{(i)}$, where $z_n^{(i)} = \mathbb{I}(\sum_{n' \in \mathcal{L}(n)} y_{\pi(n')}^{(i)} \geq 1)$ and $z_{\rho(n)}^{(i)} = \mathbb{I}(\sum_{n' \in \mathcal{L}(\rho(n))} y_{\pi(n')}^{(i)} \geq 1)$ (i.e., Eq. (1));
- Optimal Estimator (OptEst): $p_g(z_n = 1) = \sum_{i=1}^N z_n^{(i)}/N$, where $z_n = z_{n'}$ with $n' \in \text{argmax}_{n' \in \mathcal{C}(n)} p_g(z_{n'} = 1)$ (i.e., Eq. (15)).

We use the abbreviation DirEst, HierEst and OptEst to denote these three methods. Table A.1 shows corresponding experimental results. Comparing DirEst and HierEst, we can find that both of them produce similar results for any choices of $k$, $m$ and $N$, which verifies the rationality of only providing results of DirEst in Table 1. Besides, $\text{reg}_{p@m}(\mathcal{M})$ of both DirEst and HierEst is non-zero in general[6], which is consistent with the results in Table 1. Compared to both DirEst and HierEst, OptEst achieves much smaller $\text{reg}_{p@m}(\mathcal{M})$ for any choices of $k$, $m$ and $N$. In the ideal case when $N = \infty$, OptEst achieves zero regret. These findings verify the correctness of the optimal pseudo target definition in Eq. (13) and the rationality of its recursive estimation in Eq. (15). For these three methods, a common phenomenon is that given the same $m$ and $N$, increasing $k$ leads to smaller $\text{reg}_{p@m}(\mathcal{M})$, i.e., better retrieval performance. The reason is that a larger beam size $k$ corresponds to a larger leaf level beam set $\mathcal{B}_H$ on which the $\text{argTopm}$ operator is used to retrieve targets[7].

### C.2. Synthetic Data

Recall that in Sec. 5.1, we set $c = -5$ to simulate the practical case when the number of relevant targets is much smaller than the target set size. In other words, such a $c$ can be regarded as a sparsity controller for the target set. In this subsection,

[5]The method used in Table 1 corresponds to DirEst.

[6]$\text{reg}_{p@m}(\mathcal{M})$ seems to be zero for cases when $N = \infty$, $k = 50$ and $m = 1$, 5 or 10. This is because the reported number is rounded to three decimal places.

[7]An extreme case is that $k = M$ and $N = \infty$ such that $\mathcal{B}_H = \mathcal{N}_H$, $p_g(z_n = 1) = \eta_{\pi(n)}$, respectively. In this case, no matter which $p_g(z_n = 1)$ is used, $\text{reg}_{p@m}(\mathcal{M})$ always equals zero.

*Table A.2.* A comparison of $\widehat{\text{reg}}_{p@m}(\mathcal{M})$ averaged by 5 runs with random initialization under hyperparameter settings $M = 1000$, $d = 10$, $|\mathcal{D}_{tr}| = 10000$, $|\mathcal{D}_{te}| = 1000$, $k = 50$ and various $c$.

(a) $c = 0$.

| $m$ | 1 | 10 | 20 | 50 |
|---|---|---|---|---|
| PLT | 0.0008 | 0.0024 | 0.0041 | 0.0150 |
| TDM | 0.0005 | 0.0021 | 0.0037 | 0.0148 |
| OTM | 0.0006 | 0.0015 | 0.0026 | **0.0088** |
| OTM (-BS) | **0.0001** | **0.0007** | **0.0019** | 0.0110 |
| OTM (-OptEst) | 0.0007 | 0.0025 | 0.0048 | 0.0139 |

(b) $c = -1$.

| $m$ | 1 | 10 | 20 | 50 |
|---|---|---|---|---|
| PLT | 0.0022 | 0.0056 | 0.0091 | 0.0302 |
| TDM | 0.0004 | 0.0024 | 0.0057 | 0.0280 |
| OTM | 0.0006 | 0.0021 | 0.0044 | **0.0182** |
| OTM (-BS) | **0.0002** | **0.0016** | **0.0042** | 0.0242 |
| OTM (-OptEst) | 0.0005 | 0.0023 | 0.0054 | 0.0262 |

(c) $c = -2$.

| $m$ | 1 | 10 | 20 | 50 |
|---|---|---|---|---|
| PLT | 0.0055 | 0.0124 | 0.0189 | 0.0564 |
| TDM | 0.0008 | 0.0045 | 0.0111 | 0.0538 |
| OTM | 0.0008 | 0.0036 | **0.0082** | **0.0369** |
| OTM (-BS) | **0.0004** | **0.0033** | 0.0091 | 0.0502 |
| OTM (-OptEst) | 0.0007 | 0.0039 | 0.0097 | 0.0481 |

(d) $c = -3$.

| $m$ | 1 | 10 | 20 | 50 |
|---|---|---|---|---|
| PLT | 0.0145 | 0.0278 | 0.0394 | 0.0978 |
| TDM | 0.0013 | 0.0087 | 0.0214 | 0.0919 |
| OTM | **0.0008** | **0.0061** | **0.0149** | **0.0648** |
| OTM (-BS) | 0.0008 | 0.0070 | 0.0183 | 0.0878 |
| OTM (-OptEst) | 0.0011 | 0.0076 | 0.0187 | 0.0826 |

(e) $c = -4$.

| $m$ | 1 | 10 | 20 | 50 |
|---|---|---|---|---|
| PLT | 0.0281 | 0.0542 | 0.0693 | 0.1335 |
| TDM | 0.0022 | 0.0139 | 0.0319 | 0.1224 |
| OTM | **0.0012** | **0.0093** | **0.0225** | **0.0906** |
| OTM (-BS) | 0.0012 | 0.0106 | 0.0278 | 0.1183 |
| OTM (-OptEst) | 0.0014 | 0.0116 | 0.0278 | 0.1090 |

(f) $c = -5$.

| $m$ | 1 | 10 | 20 | 50 |
|---|---|---|---|---|
| PLT | 0.0444 | 0.0778 | 0.0955 | 0.1492 |
| TDM | 0.0033 | 0.0205 | 0.0453 | 0.1363 |
| OTM | **0.0024** | **0.0163** | **0.0349** | **0.1083** |
| OTM (-BS) | 0.0048 | 0.0201 | 0.0421 | 0.1313 |
| OTM (-OptEst) | 0.0033 | 0.0198 | 0.0418 | 0.1218 |

we provide a thorough comparison of PLT, TDM and OTM on the synthetic data with various sparsity of relevant targets. More specifically, we choose $c$ in $\{0, -1, -2, -3, -4, -5\}$, which results in that the ratio of relevant targets per instance becomes $50.07\%, 38.57\%, 28.26\%, 19.38\%, 12.92\%, 8.14\%$, respectively.

Results are shown in Table A.2. We can find that OTM and its variants perform better than PLT and TDM in general. An interesting phenomenon is the different behaviors of the beam search aware subsampling (BS) and the estimating optimal pseudo targets (OptEst) on various $c$. OTM (-BS) has lower $\widehat{\text{reg}}_{p@m}(\mathcal{M})$ compared to PLT and TDM for all choices of $c$, which reflects that OptEst contributes to better performance consistently. By comparing the results between OTM (-OptEst) and OTM (-BS), we can find that OTM (-OptEst) performs worse than OTM (-BS) when $c$ is large (e.g., $c = 0$) and OTM (-OptEst) performs better than OTM (-BS) when $c$ is small (e.g., $c = -5$), which implies that BS plays different roles according to the sparsity of relevant targets: When relevant targets are sparse, BS contributes to better performance; When relevant targets are not sparse, BS negatively affect retrieval performance. Besides, another interesting phenomenon is that OTM performs the best when $m = k$, while for the $m < k$ case, OTM may perform worse than OTM (-BS). Since $k = 50$ is fixed, the retrieved beam $\mathcal{B}_H(\mathbf{x})$ is also fixed and the performance for varying $m$ only depends on $\{g(\mathbf{x}, n) : n \in \mathcal{B}_H(\mathbf{x})\}$. Therefore, such a phenomenon implies that $g(\mathbf{x}, n)$ on the leaf level may not be well trained to preserve the order information among $\{\eta_{\pi(n)}(\mathbf{x}) : n \in \mathcal{B}_H(\mathbf{x})\}$. We'd like to analyze this phenomenon in details for future work.

### C.3. Real Data

Table A.3 summarizes the statistics of Amazon Books and UserBehavior datasets.

**Preprocessing:** In the main body we briefly introduce how to preprocess these two datasets.

**Implementation:** The node-wise scorer $g(\mathbf{x}, n)$ is built as follows: After preprocessing, for each instance $(\mathbf{x}, \mathbf{y})$, $\mathbf{x} \in \mathcal{I}^{69}$ is a 69-dimensional vector where $x_t$ $(1 \leq t \leq 69)$ denotes the user's behavior (i.e., the interacted item) at $t$-th nearest time

*Table A.3.* Statistics of Amazon Books and UserBehavior datasets.

| | Amazon Books | UserBehavior |
|---|---|---|
| Num. of users | 294,739 | 969,529 |
| Num. of items | 1,477,922 | 4,162,024 |
| Num. of records | 8,654,619 | 100,020,395 |

*Table A.4.* Precision@$m$, Recall@$m$ and F-Measure@$m$ comparison between Zhu et al. (2019) and our implementation, with beam size $k = 400$ and $m = 200$. The percent sign (%) is omitted for each number.

| Method | | Amazon Books | | | UserBehavior | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| HSM | Zhu et al. (2019) | 0.42 | 6.22 | 0.72 | 1.80 | 8.62 | 2.71 |
| | Our implementation | 0.54 | 8.04 | 0.95 | 2.01 | 9.52 | 3.03 |
| JTM | Zhu et al. (2019) | 0.79 | 12.45 | 1.38 | 3.11 | 14.71 | 4.68 |
| | Our implementation | 0.80 | 12.60 | 1.40 | 3.12 | 14.75 | 4.70 |

w.r.t. the earliest interaction time of item in $\mathbf{y} \in \{0,1\}^{|\mathcal{I}_\mathbf{x}|}$. For any $n \in \mathcal{N}_h$, $\mathbf{x}$ is transformed[8] to a level-wise representation $\mathbf{x}(n)$ by replacing $x_t$ with $x_t(n) = \rho^{H-h}(\pi^{-1}(x_t))$. Both $x_t(n)$ and $n$ are embeded to be 24-dimensional continuous vectors denoted by $\mathrm{emb}(x_t(n))$ and $\mathrm{emb}(n)$, respectively. According to the time order from near to far, $\{\mathrm{emb}(x_t(n))\}_{t=1}^{69}$ is further split into 10 windows with window size 1, 1, 1, 2, 2, 2, 10, 10, 20, 20, respectively. Then, average pooling is applied to produce a 24-dimensional vector for each window. Finally, these 10 vectors are concatenated with the node embedding vector and produces a 264-dimensional vector as the input to the following neural network, which consists of three fully-connected layers, with 128, 64 and 24 hidden units and Parametric ReLU as the activation function. The tree hierarchy $\mathcal{T}$ is chosen to be the one produced by JTM[9] and is shared by all the tree models, including HSM, PLT, JTM and OTM. By doing so, all the tree models have the same tree hierarchy and the same formulation of node-wise scorers, the difference of retrieval performance of these models can be attributed to the difference of training algorithms of them, and thus different tree models can be compared fairly under this setting.

**Results:** In Table 3 and Table 4 of the main body, the results for YouTube product-DNN and HSM are produced using codes in `https://github.com/alibaba/x-deeplearning/tree/master/xdl-algorithm-solution/TDM/` and the results for JTM are produced using codes provided by the supplemental of `https://papers.nips.cc/paper/8652-joint-optimization-of-tree-based-index-and-deep-model-for-recommender-systems`.

We also provide additional experimental results for comparing the retrieval performance reported in the original JTM paper (Zhu et al., 2019) and that in our implementation, which is shown in Table A.4. For JTM, we can find that our implementation achieves similar results to that reported in the original paper, which verifies the rationality of our implementation. An interesting observation is that our implemented HSM achieves better results compared to the original one. The reason is that in our experiment settings, HSM uses the same $\mathcal{T}$ and $g(\mathbf{x}, n)$ formulation as JTM. While in the original paper, HSM uses a different $\mathcal{T}$ which is built according to category information of raw dataset and $g(\mathbf{x}, n)$ does not use the hierarchical user preference representation.

## References

Jain, H., Prabhu, Y., and Varma, M. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–944, 2016.

Lapin, M., Hein, M., and Schiele, B. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1533–1554, 2017.

[8] This is called the hierarchical user preference representation, which is proposed in Zhu et al. (2019).

[9] Recall that JTM optimizes $\mathcal{T}$ and $g(\mathbf{x}, n)$ jointlty.

Wydmuch, M., Jasinska, K., Kuznetsov, M., Busa-Fekete, R., and Dembczynski, K. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pp. 6355–6366, 2018.

Zhu, H., Li, X., Zhang, P., Li, G., He, J., Li, H., and Gai, K. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1079–1088. ACM, 2018.

Zhu, H., Chang, D., Xu, Z., Zhang, P., Li, X., He, J., Li, H., Xu, J., and Gai, K. Joint optimization of tree-based index and deep model for recommender systems. In *Advances in Neural Information Processing Systems*, pp. 3973–3982, 2019.