



Calibrated Tree-Structured Speculative Decoding: Mitigating Overconfidence via Path-Level Calibration

Zixian Li Xiaohui Song Chi Xie Ke Chen Chen Chen Haonan Lu[†]

OPPO AI Center

{lizixian, songxiaohui, xiechi, chenke2, chenchen4, luhaonan}@oppo.com

Abstract

Tree-structured speculative decoding accelerates Large Language Model inference by drafting multiple candidate tokens in parallel and verifying them against a target model. Dynamic tree expansion strategies further improve efficiency by using draft model confidence to prune unlikely branches and stop expansion early. However, we find that existing draft models exhibit severe overconfidence: they assign high probabilities to tokens that will be rejected, causing unreliable early stopping and wasted computation on incorrect branches.

We propose **Calibrated Tree-Structured Speculative Decoding (CATS-SD)**, a training framework for tree-based confidence calibration. Our approach calibrates draft model confidence at two complementary levels: (1) *token-level calibration* that aligns draft-target token ranking, and (2) *path-level calibration* that supervises confidence over tree-structured decoding paths. We also introduce a calibration metric using Pearson correlation to quantify confidence-acceptance alignment. On Qwen2.5-3B-Instruct, CATS-SD achieves up to 11.3% higher acceptance length and 4.0% better confidence-acceptance correlation compared to the baseline, with end-to-end speedup improving from $2.51\times$ to $2.71\times$ on GSM8K. We validate these results across mathematical reasoning (GSM8K), multi-turn conversation (MT-bench), and code generation (HumanEval).

1 Introduction

The inference latency of Large Language Models (LLMs) remains a major bottleneck for deployment, particularly when generating long sequences. Tree-structured speculative decoding (Miao et al., 2024; Li et al., 2024b) addresses this by using a lightweight draft model to propose multiple candidate tokens along a tree structure and verify-

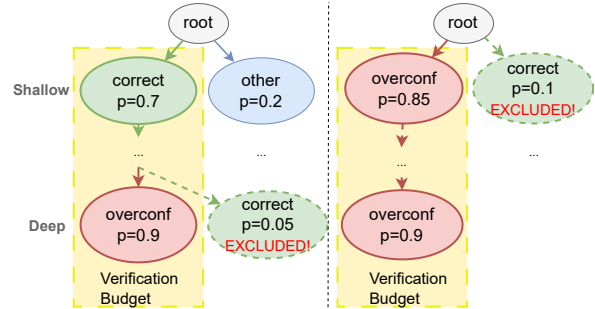


Figure 1: Two failure modes of draft model overconfidence (green: correct path; red: overconfident incorrect tokens). (Left) Overconfidence at deep layers pushes the correct path out of the top- N budget. (Right) Overconfidence at shallow layers triggers wasteful subtree expansion.

ing them in parallel with the target model. Dynamic tree expansion strategies (Li et al., 2024a; Wang et al., 2025) further improve efficiency by using the draft model’s confidence to prune unlikely branches and terminate expansion early. In these systems, confidence estimation is a key signal for candidate selection, early stopping, and dynamic control of tree depth.

However, we observe that existing draft models suffer from severe *overconfidence*: they assign high probabilities to tokens that the target model will reject. This overconfidence manifests in two failure modes (Figure 1). First, at deep layers, when verification is constrained to a fixed budget (e.g., 32) of candidate tokens, overconfident predictions on incorrect tokens push the correct acceptance path out of the top-ranked candidates. Second, at shallow layers, overconfident tokens that are ultimately rejected trigger wasteful expansion of deep subtrees that are eventually discarded. In both cases, computation is wasted without improving the acceptance length τ .

This raises a natural question: *how can we calibrate draft model confidence to suppress over-*

[†] Corresponding author.

confidence while preserving acceptance length?

Rather than relying on inference-time heuristics, we propose **Calibrated Tree-Structured Speculative Decoding (CATS-SD)**, which calibrates draft model confidence at two complementary levels during training: **(1) Token-level calibration** via BiLD loss (Li et al., 2025a), which aligns the relative ranking of candidate tokens between draft and target models, ensuring the draft model’s top predictions match the target’s preferences; and **(2) Path-level calibration** via a novel tree-based loss that constructs decoding trees during training, supervises the model to assign higher probabilities to accepted paths than rejected ones, and generates negative examples via top- k sampling. These two levels complement each other and benefit both fixed-depth expansion and dynamic methods like OPT-Tree (Wang et al., 2025).

To evaluate calibration quality, we also introduce the Accepted Path Confidence (APC) metric for tree-structured decoding, which quantifies confidence–acceptance alignment via Pearson correlation.

Our contributions are as follows:

- **Path-level calibration via tree-based loss:** A training objective that supervises decoding paths via top- k sampled negative examples (Section 3.2).
- **Calibration metric for tree-structured decoding:** A metric quantifying confidence–acceptance alignment via Pearson correlation (Section 4).
- **Empirical validation:** Experiments on three benchmarks showing gains in acceptance length, calibration, and end-to-end speedup under fixed-depth and dynamic settings.

2 Related Work

2.1 Speculative Decoding and Early Stopping

Xia et al. (2023) and Leviathan et al. (2023) introduced the idea of using a lightweight draft model to propose candidates for parallel verification by a heavier target model. Crucially, the verification step guarantees that the final output distribution is identical to that of the target model alone, making speculative decoding a *lossless* acceleration technique. Since classical approaches generate a single candidate path, the rejection of any intermediate token causes all subsequent tokens to be

discarded, limiting efficiency gains. This limitation motivated the transition to tree-structured decoding, pioneered by SpecInfer (Miao et al., 2024), where the draft model expands a tree of candidate tokens and the target model evaluates all nodes in parallel, thus reducing the rejection probability.

Subsequent work has improved acceptance rates through better draft model design. EAGLE (Li et al., 2024b) showed that using the target model’s hidden states as input features substantially improves acceptance rates. EAGLE-2 (Li et al., 2024a) further introduced dynamic sampling trees with top- k reranking to avoid expanding low-confidence paths. HASS (Zhang et al., 2024) addressed a critical training–inference mismatch by harmonizing objective distillation and context alignment. More recently, EAGLE-3 (Li et al., 2025b) achieved state-of-the-art acceptance rates by fusing hidden states from multiple layers of the target model. Across these methods, a consistent finding is that draft model performance scales with training data volume.

While most methods focus on acceptance length without considering tree depth, OPT-Tree (Wang et al., 2025) enables dynamic early stopping using the expectation of classification confidence, based on the implicit assumption that confidence approximates acceptance rate. However, the draft model’s own confidence is not calibrated and can be unreliable. In this work, we focus on *training-time calibration* to strengthen this assumption.

2.2 Confidence Estimation and Calibration

Confidence calibration has been extensively studied in classification, where post-hoc techniques such as expected calibration error (ECE) (Naeini et al., 2015) are commonly used to assess predictive probabilities (Guo et al., 2017). However, ECE suffers from well-documented limitations (Pavlovic, 2025): low ECE does not guarantee high accuracy, binning sensitivity creates a bias-variance trade-off, and it only considers maximum probabilities rather than the full distribution—particularly problematic for tree-structured decoding, where multiple candidate paths must be evaluated simultaneously. The calibration challenge extends to NLP as well: Ott et al. (2018) evaluated token- and sequence-level confidence in machine translation, while Park and Caragea (2022) addressed calibration in natural language understanding via data augmentation. Despite these efforts, LLMs often display marked overconfidence (Geng et al., 2024), which directly

undermines tree construction and early-stopping decisions in speculative decoding.

3 Training-Time Confidence Calibration

3.1 Preliminaries

EAGLE-style speculative decoding methods (Li et al., 2024b,a; Zhang et al., 2024; Li et al., 2025b) share a common formulation for draft model training. Let $\mathcal{M}_T$ denote the target model and $\mathcal{M}_D$ the draft model. Given an input sequence, the target model produces hidden states $h_T^{(t)} \in \mathbb{R}^d$ at each position t . The draft model takes the concatenation of the next token embedding e_{t+1} and the target model’s hidden state $h_T^{(t)}$ as input, and outputs a predicted hidden state $h_D^{(t+1)} \in \mathbb{R}^d$. The key objective is to align $h_D^{(t+1)}$ with $h_T^{(t+1)}$ through distillation, which enables the draft model to autoregressively generate multiple tokens by feeding its own predicted hidden states back as input. Since tree construction and verification both rely on probability ordering, the probability estimates of the draft model directly determine decoding quality.

Token-Level Distillation Losses. Our training objective builds upon the HASS framework (Zhang et al., 2024). We use four token-level losses: (1) L_{MSE} : mean-squared error between draft and target hidden states for representation alignment (replacing HASS’s smooth L1 loss); (2) L_{CE} : cross-entropy between target and draft distributions for probability matching; (3) L_{topk} : cross-entropy restricted to top- k tokens, emphasizing high-probability regions critical for tree construction; and (4) L_{BiLD} (Li et al., 2025a): bi-directional logits difference loss that aligns the *relative ordering* of high-probability tokens. While HASS evaluated these distillation losses separately, we find L_{topk} and L_{BiLD} complementary: the former matches probability values while the latter aligns relative token rankings over the same top- k candidates used during inference-time tree construction. The combined token-level objective is

$$L_{\text{base}} = w_c L_{\text{CE}} + w_t L_{\text{topk}} + w_m L_{\text{MSE}} + w_b L_{\text{BiLD}}, \quad (1)$$

where the weights w_c, w_t, w_m, w_b control relative contributions.

3.2 Tree-Based Confidence Calibration Loss

While the token-level losses align the draft model with the target at each token position, they operate

Ground Truth

CATS-SD is a method to calibrate draft models and suppress over-confidence ...

Random Index Selection

[CATS-SD] [is] [a] [method] [to] [calibrate]
[draft] [models] [and] ...
0 1 2 3 4 5 6 7
Start expansion from random token "method"

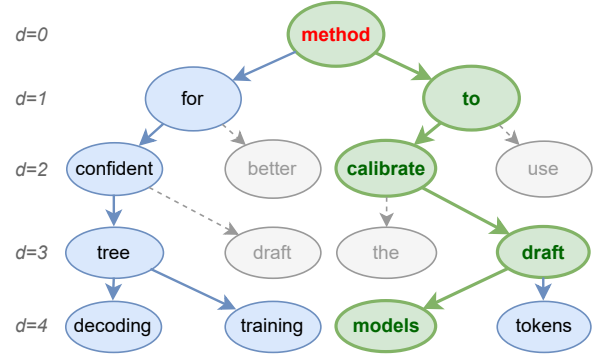


Figure 2: Illustration of tree-based confidence calibration loss (example with depth $D = 4$, top- $k = 2$). **Blue**: the top- k nodes selected for expansion; **gray**: pruned nodes still participating in CE loss; **green**: teacher acceptance path. CE is computed independently per depth and averaged across depths.

on individual tokens and do not directly constrain confidence along entire decoding paths. Existing methods construct decoding trees only at inference time; we bridge this gap by building the same tree structures during training and supervising the model over entire paths. This training-inference consistency ensures the draft model learns to estimate probabilities under the same tree-expansion conditions it encounters at test time.

Per-Depth Cross-Entropy. For each training sample, we first randomly select m positions from the label sequence as tree roots. For each selected root position, we build a tree of depth D with branching factor k , yielding m independent trees per sample (Figure 2 shows a concrete example with $D = 4$ and $k = 2$). During tree construction, we use top- k expansion: at each depth, top- k nodes are selected for further expansion, matching the inference-time process. We ensure the ground-truth token is always included at each depth—if it falls outside the candidates, we replace the lowest-probability one with the ground-truth token, guaranteeing that the teacher acceptance path exists in every constructed tree. For loss computation, all nodes at each depth, including those not selected for further expansion, participate in the cross-

entropy calculation.

At depth d , let $c_*^{(d)}$ be the teacher-path node among all n_d candidates $\{c_j^{(d)}\}$. Since these candidates form a subset of the vocabulary, we re-normalize their probabilities within the candidate set and compute cross-entropy as

$$L_{\text{layer}}^{(d)} = -\log \frac{p_{c_*^{(d)}}}{\sum_{j=1}^{n_d} p_{c_j^{(d)}}}. \quad (2)$$

Depth-Dependent Loss Weighting. In tree-structured speculative decoding, prediction errors at different depths have asymmetric consequences. If the draft model makes an incorrect prediction at a shallow layer, the entire downstream subtree rooted at that node becomes invalid, wasting all computation spent on deeper expansion. In contrast, an error at a deeper layer affects only a small number of remaining candidates. To reflect this asymmetry, we introduce depth-dependent weights that assign higher penalties to shallower layers, as

$$w_d = \exp\left(-\frac{d-1}{\gamma}\right), \quad (3)$$

where $\gamma > 0$ controls the decay rate. At the shallowest layer ($d = 1$), $w_1 = 1$, assigning the maximum penalty; as depth increases, the weight decays exponentially. We ablate the effect of this weighting scheme in Section 5.7.

The loss for a single tree aggregates weighted contributions from all depths:

$$L_{\text{layer}}^{(\text{tree})} = \frac{1}{\sum_{d=1}^D w_d} \sum_{d=1}^D w_d \cdot L_{\text{layer}}^{(d)}. \quad (4)$$

The final loss averages over all m trees per sample:

$$L_{\text{tree}} = \frac{1}{m} \sum_{i=1}^m L_{\text{layer}}^{(\text{tree}_i)}. \quad (5)$$

This formulation encourages the draft model to assign the highest probability to the teacher-path node at every layer, with stronger emphasis on shallow layers where errors are most costly. By computing cross-entropy over all candidate nodes at each depth (including pruned ones), the model also learns to suppress overconfident predictions on incorrect paths.

Combined with the token-level losses (Eq. 1), the complete training objective is

$$L = L_{\text{base}} + L_{\text{tree}}. \quad (6)$$

Efficient Tree Construction. Introducing tree construction into the training loop raises two practical challenges.

The first challenge is **exponential expansion cost**. Naively expanding a tree of depth D with branching factor k would require $\mathcal{O}(k^D)$ forward passes, which is computationally prohibitive. We address this with three optimizations: (1) reusing KV-cache across branches sharing the same parent node, (2) using tree attention masks for batched expansion at each depth, and (3) retaining gradients only for per-layer log-probabilities rather than the full computation graph. Together, these reduce forward passes from exponential to linear in depth.

The second challenge is **additional training time**. Even with the above optimizations, tree-based training still requires multiple forward passes per sample to construct the full tree structure, increasing per-iteration cost compared to standard token-level distillation. To manage this overhead, we support two training modes: (1) training from scratch for full calibration (suitable for small datasets), or (2) fine-tuning a pre-trained draft model with fewer epochs for faster convergence (suitable for large datasets).

Note that this tree-based loss is especially useful when training data is limited. As discussed in Section 2.1, draft model performance typically scales with training data volume (Li et al., 2025b). Our tree loss circumvents this dependency by dynamically generating high-quality negative examples during training. Each tree constructed from a random root produces multiple counterfactual paths that the model considers plausible but that diverge from the ground truth. These synthetic negatives amplify the supervision signal: from a single sample, the model learns not only to increase the probability of the correct path but also to suppress confidence on the most confusing alternatives, i.e., the very paths it would otherwise explore during inference.

4 Calibration Evaluation

Although confidence estimation plays a central role in speculative decoding and early stopping strategies like OPT-Tree, no existing method explicitly evaluates calibration in this setting. We fill this gap by proposing the Accepted Path Confidence metric, which collects confidence–acceptance pairs along candidate paths and quantifies their alignment via Pearson correlation.

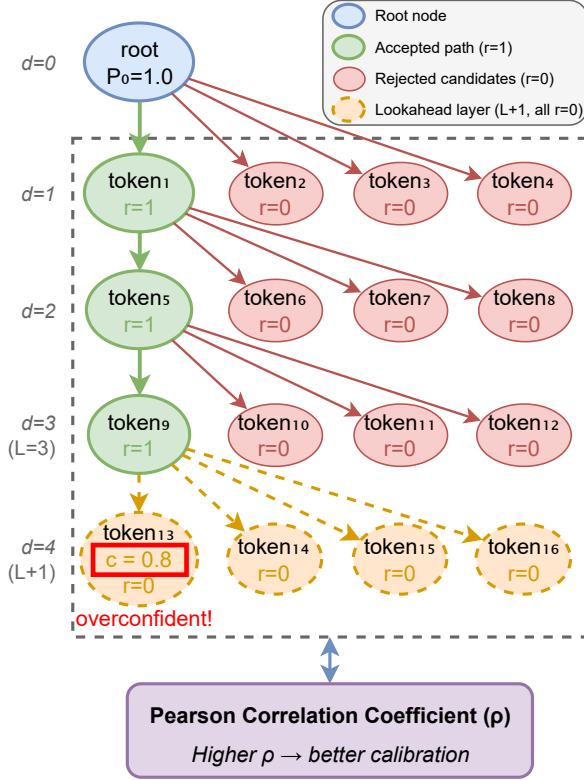


Figure 3: APC calculation example ($L = 3$). The lookahead layer ($d = 4$) is entirely rejected ($r = 0$), yet token₁₃ receives $c = 0.8$ from the draft model—a clear case of overconfidence that APC is designed to capture.

4.1 Definition of APC

Here, draft model *confidence* refers to its conditional probability. APC measures how well these confidence estimates along the accepted path align with actual acceptance behavior, incorporating a one-layer lookahead mechanism. Figure 3 illustrates the calculation process with an example where 3 draft tokens are accepted.

Consider a decoding step where L tokens are accepted. As illustrated in Figure 3, we exclude the root node and examine each layer d from depth 1 to depth $L + 1$ (the lookahead layer). At each depth d , we identify all candidate child nodes $\{c_1^{(d)}, c_2^{(d)}, \dots, c_k^{(d)}\}$ that are sampled via top- k expansion from the parent node on the accepted path. For each candidate $c_j^{(d)}$, we compute its conditional probability:

$$p(c_j^{(d)} \mid x_{<d}) = \frac{P(c_j^{(d)})}{P(x_{<d})}, \quad (7)$$

where $P(\cdot)$ denotes joint probability, and $x_{<d}$ is the complete prefix to depth $d - 1$. We then assign a binary label $r_j^{(d)} \in \{0, 1\}$: $r_j^{(d)} = 1$ if the

candidate lies on the accepted path, and $r_j^{(d)} = 0$ otherwise. A well-calibrated model should assign high probabilities to accepted tokens ($r = 1$) and low probabilities to rejected ones ($r = 0$). The lookahead layer ($L + 1$) is particularly useful for detecting overconfidence: since all tokens at this depth are rejected, high probabilities paired with $r = 0$ expose unreliable confidence estimates and reduce the Pearson correlation.

4.2 Quantification via Pearson Correlation

We pair each candidate’s confidence $c_i = p(c_j^{(d)} \mid x_{<d})$ with its acceptance label $r_i = r_j^{(d)}$, yielding a dataset $\mathcal{D} = \{(c_i, r_i)\}_{i=1}^N$. Following the bin-wise analysis in Naeini et al. (2015), we partition the probability range $[0, 1]$ into fixed intervals and compute, for each bin, the average confidence and the average acceptance rate. These bin-level statistics yield empirical calibration curves that reveal how acceptance frequency varies as a function of predicted probability (see Figure 5).

We compute the Pearson correlation coefficient between conditional probabilities and acceptance labels as our primary calibration metric:

$$\rho = \frac{\sum_i (c_i - \bar{c})(r_i - \bar{r})}{\sqrt{\sum_i (c_i - \bar{c})^2} \cdot \sqrt{\sum_i (r_i - \bar{r})^2}}. \quad (8)$$

A larger ρ indicates that higher conditional probabilities are more strongly associated with actual acceptance, meaning that the draft model’s probability estimates are better aligned with the target model’s verification behavior. We also compute ECE (Naeini et al., 2015), but observe limited variation across configurations in tree-structured decoding, consistent with the ECE limitations discussed in Section 2.2. Therefore, we use Pearson correlation as our primary metric, as it directly measures the linear relationship between probabilities and acceptance outcomes without binning artifacts. See Appendix A for detailed ECE results and discussion.

5 Experiments

We evaluate acceptance length under fixed-depth expansion (Section 5.3), confidence calibration via our APC metric (Section 5.4), dynamic early stopping with OPT-Tree (Section 5.5), generalization to a reasoning-oriented target model with EAGLE-3 (Section 5.6), and ablation of individual loss components (Section 5.7).

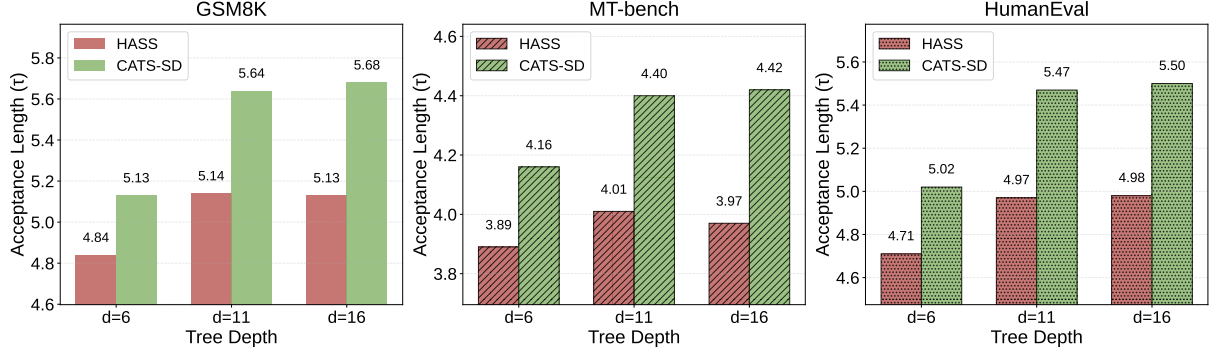


Figure 4: Acceptance length (τ) at different fixed tree depths on GSM8K, MT-bench, and HumanEval.

5.1 Setup

Models. We use Qwen2.5-3B-Instruct (Qwen Team, 2024) as the target model and a single-layer EAGLE decoder as the draft model, following the EAGLE-2 architecture (Li et al., 2024a). We adopt HASS (Zhang et al., 2024) as our baseline training framework.

Training Data. We use the ShareGPT dataset¹ (~68k conversations). Following EAGLE (Li et al., 2024b), we extract the first question from each conversation and generate answers using the target model.

Evaluation Benchmarks. We evaluate on three diverse benchmarks: (1) GSM8K (Cobbe et al., 2021) for mathematical reasoning, (2) MT-bench (Zheng et al., 2023) for multi-turn conversation, and (3) HumanEval (Chen et al., 2021) for code generation.

Configurations. We compare four configurations: (1) **HASS** (Zhang et al., 2024) is the baseline using smooth L1 loss; (2) **HASS+MSE** replaces smooth L1 with MSE; (3) **HASS+MSE+Tree** adds our tree-based confidence calibration loss; and (4) **CATS-SD** is our full method combining MSE, BiLD, and tree-based losses.

Implementation Details. On ShareGPT, we pre-train with MSE and BiLD for 40 epochs, then fine-tune with the tree-based loss for 10 additional epochs; the HASS baseline trains for 40 epochs with smooth L1. For ablation on GSM8K (smaller datasets), the tree-based loss is applied from scratch (40 epochs). The tree-based loss uses $m=5$ roots, depth $D=16$, top- $k=8$, and decay $\gamma=4$. For the token-level loss weights (Eq. 1), we set $w_c=0.1$

and $w_t=w_m=w_b=1.0$. BiLD uses temperature $T_{\text{BiLD}}=3.0$ and top- $k=8$. We evaluate under two inference settings: **Fixed Depth**, which expands the full tree to depth $D \in \{6, 11, 16\}$; and **OPT-Tree** (Wang et al., 2025), which applies dynamic early stopping with threshold $\delta \in \{0.2, 0.5, 0.8\}$ and maximum depth 16. All experiments use top- $k=8$ expansion, 32 verification tokens, and greedy decoding ($T=0$); inference runs on a single H20 GPU. All results are deterministic under greedy decoding, so we report a single run without error bars.

5.2 Metrics

We report the following metrics:

- **Acceptance Length τ :** The average number of tokens accepted per drafting-verification cycle, including the root token.
- **Acceptance Rate:** The ratio of accepted tokens (excluding the root) to the average tree depth, i.e., $\text{Acceptance Rate} = (\tau - 1) / \text{Depth}$.
- **Speedup:** End-to-end wall-clock acceleration relative to autoregressive decoding.
- **Pearson ρ :** Correlation between path-level confidence and acceptance rate, measuring calibration quality via our APC metric.

5.3 Acceptance Length Results

Figure 4 summarizes fixed-depth results across all three benchmarks. On GSM8K, CATS-SD achieves $\tau = 5.68$ at depth 16, representing a 10.7% improvement over HASS (5.13). We observe similar trends on MT-bench, where CATS-SD achieves $\tau = 4.42$ at depth 16, an 11.3% improvement over HASS (3.97). On HumanEval, CATS-SD consistently outperforms HASS as well, with

¹https://huggingface.co/datasets/Aeala/ShareGPT_Vicuna_unfiltered

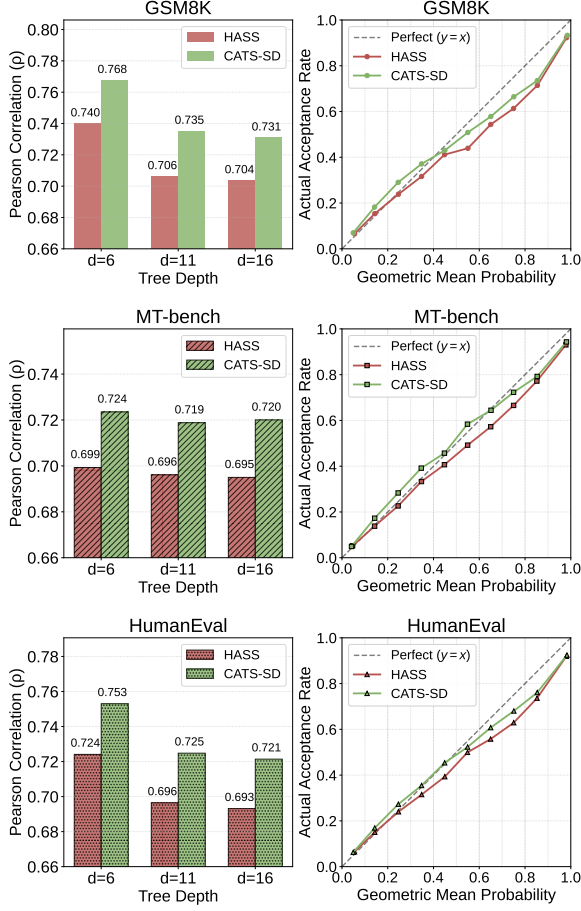


Figure 5: Confidence calibration results on GSM8K, MT-bench, and HumanEval. (a) Pearson correlation (ρ) at different fixed tree depths. (b) Calibration curves comparing HASS (red) and CATS-SD (green) against perfect calibration (dashed grey).

$\tau = 5.50$ at depth 16, a 10.4% improvement over HASS (4.98). These gains hold across all depths, indicating that our calibration approach is robust to different tree configurations.

5.4 Confidence Calibration Results

We report Pearson correlation coefficients between conditional probabilities and acceptance labels at different tree depths, using our proposed APC metric. Figure 5 summarizes both Pearson correlation and calibration curves on all three benchmarks.

On GSM8K, CATS-SD achieves $\rho = 0.731$ at depth 16, a 3.8% improvement over HASS (0.704). We observe consistent improvements on MT-bench ($\rho = 0.720$ vs 0.695, a 3.6% gain) and HumanEval ($\rho = 0.721$ vs 0.693, a 4.0% gain).

The calibration curves in Figure 5(b) further illustrate these improvements. Across all three benchmarks, HASS (red curve) exhibits systematic overconfidence, with the curve below the diagonal

(perfect calibration line) across nearly the entire probability range. For instance, on HumanEval, when HASS predicts a geometric mean probability of 0.45, the actual acceptance rate is only 0.39—a gap of 0.06. In contrast, CATS-SD (green curve) at the same probability achieves an acceptance rate of 0.45, matching the perfect calibration line and yielding more reliable estimates for early stopping strategies like OPT-Tree.

5.5 Early Stopping with OPT-Tree

Since OPT-Tree relies on draft model confidence for early stopping decisions (Section 2.1), improved calibration should directly benefit its performance. We compare HASS and CATS-SD when combined with OPT-Tree across different δ thresholds. A larger δ makes OPT-Tree more aggressive in early stopping, resulting in shallower average tree depth. Table 1 reports average tree depth, acceptance rate (AR), acceptance length (τ), and end-to-end speedup.

Under the same δ setting, CATS-SD outperforms HASS in acceptance rate, acceptance length, and end-to-end speedup across all three benchmarks. Note that the average tree depth is determined by OPT-Tree’s confidence-based stopping criterion, not optimized directly. At comparable depths, CATS-SD achieves substantially higher acceptance rate and acceptance length. For example, at $\delta = 0.2$ on GSM8K, both methods expand to similar depths (7.95 vs 7.98), yet CATS-SD improves τ from 5.11 to 5.59 (+9.4%) and speedup from $2.51\times$ to $2.71\times$. Since CATS-SD’s confidence estimates more faithfully reflect actual acceptance probability, OPT-Tree can make better early stopping decisions and utilize each draft model call more effectively. Similar improvements are observed on MT-bench (speedup $2.17\text{--}2.26\times$ vs $2.13\text{--}2.18\times$) and HumanEval (speedup $2.56\text{--}2.67\times$ vs $2.41\text{--}2.57\times$).

5.6 Generalization to Qwen3-1.5B with EAGLE-3

To evaluate whether draft overconfidence and CATS-SD generalize beyond Qwen2.5-3B-Instruct and EAGLE-2-style draft models, we further evaluate on Qwen3-1.5B, a reasoning-oriented target model that produces explicit thinking traces before final answers. We use EAGLE-3 as the baseline speculative decoding framework and apply CATS-SD to the same setting. For a controlled comparison, both EAGLE-3 and CATS-SD are trained from scratch for 40 epochs on the same ShareGPT train-

Table 1: Dynamic early stopping performance with OPT-Tree on GSM8K, MT-bench, and HumanEval.

Benchmark	Method	$\delta = 0.2$				$\delta = 0.5$				$\delta = 0.8$			
		Depth	AR $\uparrow$	τ $\uparrow$	Speedup $\uparrow$	Depth	AR $\uparrow$	τ $\uparrow$	Speedup $\uparrow$	Depth	AR $\uparrow$	τ $\uparrow$	Speedup $\uparrow$
GSM8K (Cobbe et al., 2021)	HASS	7.95	51.7%	5.11	2.51 $\times$	6.17	65.9%	5.06	2.71 $\times$	4.66	79.3%	4.70	2.72 $\times$
	CATS-SD	7.98	57.5%	5.59	2.71$\times$	6.18	71.0%	5.39	2.81$\times$	4.60	83.7%	4.86	2.76$\times$
MT-bench (Zheng et al., 2023)	HASS	5.98	50.4%	4.01	2.13 $\times$	4.66	62.8%	3.93	2.18 $\times$	3.51	75.1%	3.64	2.15 $\times$
	CATS-SD	5.85	57.9%	4.39	2.25$\times$	4.51	70.4%	4.18	2.26$\times$	3.37	80.1%	3.70	2.17$\times$
HumanEval (Chen et al., 2021)	HASS	7.71	51.3%	4.95	2.41 $\times$	5.99	64.0%	4.83	2.53 $\times$	4.51	77.6%	4.50	2.57 $\times$
	CATS-SD	7.95	55.2%	5.39	2.56$\times$	6.16	67.4%	5.16	2.67$\times$	4.58	79.5%	4.64	2.60$\times$

Table 2: Generalization to Qwen3-1.5B with EAGLE-3. We report acceptance length (τ) at different tree depths.

Benchmark	Method	d=6	d=11	d=16
GSM8K	EAGLE-3	4.823	5.255	5.309
	CATS-SD	5.084	5.699	5.795
HumanEval	EAGLE-3	4.546	4.740	4.751
	CATS-SD	4.806	5.065	5.061
MT-bench	EAGLE-3	4.252	4.469	4.487
	CATS-SD	4.497	4.811	4.857

Table 3: Ablation study: Acceptance length (τ) and Pearson correlation (ρ) at tree depth $d = 16$.

Method	GSM8K		MT-bench	
	τ $\uparrow$	ρ $\uparrow$	τ $\uparrow$	ρ $\uparrow$
HASS	5.13	0.704	3.97	0.695
HASS+MSE	5.38	0.710	4.13	0.703
HASS+MSE+Tree	5.60	0.728	4.32	0.712
HASS+MSE+BiLD	5.45	0.714	4.23	0.714
CATS-SD	5.68	0.731	4.42	0.720

ing data. Both methods use the same decoding configuration as the main experiments: top- $k = 8$ expansion, 32 verification tokens, greedy decoding ($T=0$), and fixed tree depths $d \in \{6, 11, 16\}$. Table 2 reports acceptance length under fixed-depth expansion.

CATS-SD consistently improves over EAGLE-3 across all three benchmarks and tree depths. The gains are especially clear at deeper trees: at $d = 16$, CATS-SD improves τ by 9.2% on GSM8K (5.795 vs. 5.309), 6.5% on HumanEval (5.061 vs. 4.751), and 8.2% on MT-bench (4.857 vs. 4.487). These results directly address the concern that CATS-SD may be specific to Qwen2.5-3B-Instruct or EAGLE-2/HASS; draft overconfidence persists in a stronger EAGLE-3 setting, and our calibration framework remains effective for a reasoning-oriented target model.

5.7 Ablation Studies

Loss Component Ablation. Table 3 shows the incremental benefit of each training loss component

at tree depth $d = 16$. HASS+MSE improves τ by 4.9% on GSM8K compared to HASS (5.38 vs 5.13) and ρ by 0.9% (0.710 vs 0.704), suggesting that MSE provides better hidden-state alignment than smooth L1 loss. Building on this improved baseline, we analyze each calibration component:

Adding BiLD loss (HASS+MSE+BiLD) improves τ by 1.3% over HASS+MSE on GSM8K (5.45 vs 5.38) and ρ by 0.6% (0.714 vs 0.710), confirming the value of token-level ranking alignment. Adding tree loss (HASS+MSE+Tree) yields larger gains: τ by 4.1% (5.60 vs 5.38) and ρ by 2.5% (0.728 vs 0.710), showing that path-level supervision provides stronger calibration. CATS-SD combines both levels and achieves the best results: τ of 5.68 on GSM8K (+10.7% over HASS) and 4.42 on MT-bench (+11.3%), with ρ of 0.731 (+3.8%) and 0.720 (+3.6%), respectively, indicating that the two calibration objectives are complementary.

We provide a matched-compute training comparison in Appendix B, including wall-clock time and continued-training baselines for HASS. We further evaluate inference-time temperature scaling and label smoothing on HASS in Appendix C, confirming that cheap post-hoc softening cannot reproduce CATS-SD’s acceptance-length gains.

Training Data Efficiency. To validate the data efficiency claim in Section 3.2, we train both HASS and CATS-SD on a small dataset of approximately 7.4K GSM8K training samples². As Table 4 shows, for such a limited amount of data, the improvement of CATS-SD over HASS becomes increasingly pronounced at greater depths (+2.7% at $d=6$, +6.6% at $d=11$, +9.1% at $d=16$). This is consistent with the intuition that deeper trees generate more counterfactual paths, amplify the supervision signal via synthetic negative examples, and thus yield larger gains.

Depth-Dependent vs. Uniform Weighting. Using the same 7.4K GSM8K training set, we eval-

²<https://huggingface.co/datasets/openai/gsm8k>

Table 4: Training data efficiency: Acceptance length (τ) on GSM8K with limited training data (7.4K samples).

Method	d=6	d=11	d=16
HASS	5.85	7.56	8.20
CATS-SD	6.01	8.06	8.95
Improvement	+2.7%	+6.6%	+9.1%

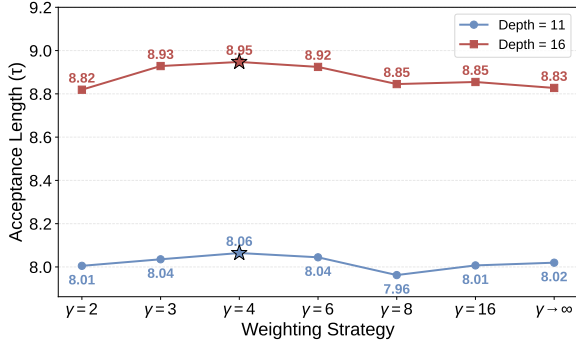


Figure 6: Effect of depth-dependent loss weighting (γ) on acceptance length (τ). Stars ($\star$) mark the best configuration ($\gamma = 4$), used as the CATS-SD default.

uate the effect of the depth-dependent weighting introduced in Eq. 3. We compare against uniform weighting (equivalent to $\gamma \rightarrow \infty$), where each layer contributes equally. Figure 6 reports acceptance length at different tree depths. Smaller γ values increasingly emphasize shallow layers, while uniform weighting does not distinguish between layers. Among all settings, $\gamma = 4$ achieves the best acceptance length at deeper trees ($d = 11$ and $d = 16$), suggesting that moderate emphasis on shallow-layer accuracy works best. At $d = 16$, $\gamma = 4$ improves τ by 1.4% over uniform weighting (8.95 vs 8.83).

6 Conclusion

We have presented CATS-SD, a training framework that addresses draft model overconfidence in tree-structured speculative decoding through a tree-based calibration loss. We also introduced the APC metric using Pearson correlation to evaluate calibration quality in this setting. Experiments on GSM8K, MT-bench, and HumanEval show up to 11.3% higher acceptance length and end-to-end speedup improving from $2.51\times$ to $2.71\times$. Future directions include using calibrated confidence to adaptively vary the branching factor per node instead of a fixed top- k , and studying the interaction between model quantization and confidence calibration.

Limitations

Single Target Model and Scale. Our experiments use Qwen2.5-3B-Instruct and Qwen3-1.5B as target models. We do not evaluate larger target models or other model families such as LLaMA and Mistral. However, draft overconfidence is primarily a property of the draft model’s confidence estimates during verification, rather than a target-model-specific phenomenon. Since our method only modifies the draft model’s training objective and does not depend on the target model architecture, we expect it to extend naturally to other target model families and larger scales.

Training Overhead. The tree-based loss requires additional forward passes for tree construction during training. However, this overhead can be mitigated by fine-tuning a pre-trained draft model with fewer epochs for faster convergence (Section 3.2). We report wall-clock training costs and matched-compute HASS baselines in Appendix B.

Draft Architecture Scope. Our experiments evaluate both a single-layer EAGLE-style draft model and EAGLE-3 (Li et al., 2025b), showing that CATS-SD can improve stronger speculative decoding baselines. Nevertheless, our evaluation remains limited to EAGLE-family draft models. We do not study other draft model architectures or decoding paradigms beyond tree-structured speculative decoding, such as chain-based speculative decoding. Extending confidence calibration to these settings is an important direction for future work.

Use of AI Assistants

We used Claude Opus 4.6 (Anthropic) to assist with English language polishing and editing of this manuscript. All technical content, experimental design, and scientific conclusions are solely the work of the authors.

References

- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. [Evaluating large language models trained on code](#). *arXiv preprint arXiv:2107.03374*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. [Training verifiers](#)

- to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koepl, Preslav Nakov, and Iryna Gurevych. 2024. [A survey of confidence estimation and calibration in large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. [On calibration of modern neural networks](#). In *International conference on machine learning*, pages 1321–1330. PMLR.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. [Fast inference from transformers via speculative decoding](#). In *International Conference on Machine Learning*, pages 19274–19286. PMLR.
- Minchong Li, Feng Zhou, and Xiaohui Song. 2025a. [BiLD: Bi-directional logits difference loss for large language model distillation](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1168–1182.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024a. [EAGLE-2: Faster inference of language models with dynamic draft trees](#). In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 7421–7432.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. [EAGLE: Speculative sampling requires rethinking feature uncertainty](#). *arXiv preprint arXiv:2401.15077*.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025b. [EAGLE-3: Scaling up inference acceleration of large language models via training-time test](#). *arXiv preprint arXiv:2503.01840*.
- Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, and 1 others. 2024. [SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification](#). In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949.
- Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. [Obtaining well calibrated probabilities using bayesian binning](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, pages 2901–2907.
- Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. 2018. [Analyzing uncertainty in neural machine translation](#). In *International Conference on Machine Learning*, pages 3956–3965. PMLR.
- Seo Yeon Park and Cornelia Caragea. 2022. [On the calibration of pre-trained language models using mixup guided by area under the margin and saliency](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5364–5374.
- Maja Pavlovic. 2025. [Understanding model calibration—a gentle introduction and visual exploration of calibration and the expected calibration error \(ece\)](#). *arXiv preprint arXiv:2501.19047*.
- Qwen Team. 2024. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Jikai Wang, Yi Su, Juntao Li, Qingrong Xia, Zi Ye, Xinyu Duan, Zhefeng Wang, and Min Zhang. 2025. [OPT-tree: Speculative decoding with adaptive draft tree structure](#). *Transactions of the Association for Computational Linguistics*, 13:188–199.
- Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2023. [Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3909–3925.
- Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. 2024. [Learning harmonized representations for speculative sampling](#). *arXiv preprint arXiv:2408.15766*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. [Judging LLM-as-a-judge with MT-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

Appendix

A Why ECE is Less Discriminative for Tree-Structured Decoding

We empirically demonstrate why Expected Calibration Error (ECE) is less suitable than Pearson correlation for evaluating calibration quality in tree-structured speculative decoding. Table 5 reports ECE and APC (Pearson ρ) metrics for HASS, HASS+MSE+BiLD, and CATS-SD across different tree depths on GSM8K and MT-bench.

ECE fails to capture calibration improvements. While CATS-SD shows expected ECE improvements over HASS (e.g., on GSM8K at depth 6: 0.0335 vs. 0.0362), examining HASS+MSE+BiLD reveals critical flaws in ECE as a metric (Table 5). On GSM8K, HASS+MSE+BiLD achieves higher APC than HASS (0.754 vs. 0.740 at depth 6; 0.714 vs. 0.704 at depth 16), yet its ECE is worse (0.0402 vs. 0.0362; 0.0519 vs. 0.0455). The

Table 5: Comparison of ECE and APC (Pearson ρ) on GSM8K and MT-bench across tree depths.

	ECE $\downarrow$		APC $\rho \uparrow$	
	d=6	d=16	d=6	d=16
GSM8K				
HASS	0.0362	0.0455	0.740	0.704
HASS+MSE+BiLD	0.0402	0.0519	0.754	0.714
CATS-SD	0.0335	0.0424	0.768	0.731
MT-bench	d=6	d=16	d=6	d=16
HASS	0.0199	0.0222	0.699	0.695
HASS+MSE+BiLD	0.0199	0.0214	0.717	0.714
CATS-SD	0.0187	0.0203	0.724	0.720

problem is even more pronounced on MT-bench: HASS+MSE+BiLD improves APC over HASS by 2.7% at depth 16 (0.714 vs. 0.695), but ECE remains nearly unchanged (0.0214 vs. 0.0222). In fact, at depth 6, the ECE is identical (0.0199) despite clear calibration improvements captured by APC. This demonstrates that ECE can fail to detect genuine calibration improvements or even misleadingly suggest degraded calibration when the model is actually better calibrated.

APC provides a consistent and reliable ranking.

In contrast, APC (Pearson ρ) yields a coherent ordering that aligns with actual calibration quality: CATS-SD > HASS+MSE+BiLD > HASS across all depths and benchmarks. This consistent ranking matches the acceptance length improvements reported in the main experiments, confirming that APC accurately captures the benefit of each training component.

These observations support our design choice of using Pearson correlation as the primary calibration metric. ECE’s binning-based formulation fails to capture the fine-grained relationship between confidence and acceptance in tree-structured decoding, where path-level dependencies and discrete acceptance decisions complicate the probability-accuracy relationship.

B Matched-Compute Training Comparison

To address whether CATS-SD’s gains simply come from additional training compute, we compare it against HASS variants under both standard and extended training budgets. Table 6 reports wall-clock time per epoch on 8 GPUs, total epochs, and GSM8K acceptance length at $d = 16$. The standard HASS and HASS+MSE+BiLD runs require 28 and 30 minutes per epoch, respectively, while

CATS-SD tree-loss fine-tuning takes 120 minutes per epoch due to per-step tree construction. Importantly, the continued-training settings use different initializations: CATS-SD starts from the pre-trained HASS+MSE+BiLD checkpoint and applies tree-loss fine-tuning for 10 epochs, whereas the HASS continued-training baselines start from the pretrained HASS checkpoint and continue optimizing with the original HASS objective. Despite the higher training cost, CATS-SD improves τ from 5.45 to 5.68 over its HASS+MSE+BiLD initialization. We include the HASS continued-training baselines to test whether a comparable extra training budget alone can close the gap.

Table 6: Matched-compute training comparison on GSM8K. Wall-clock time is measured per epoch using 8 GPUs. CATS-SD continues from the HASS+MSE+BiLD checkpoint; HASS continued-training baselines continue from the HASS checkpoint.

Method	Min./Epoch	Epochs	$\tau@d = 16$
Pretraining			
HASS	28	40	5.129
HASS+MSE+BiLD	30	40	5.452
Continued training			
HASS (continued)	28	10	5.147
HASS (matched wall-clock)	28	40	5.110
CATS-SD (tree FT)	120	10	5.676

The continued-training results show that HASS largely saturates after the standard 40-epoch training schedule. Adding 10 more epochs only marginally changes τ from 5.129 to 5.147, and extending HASS for a matched wall-clock budget does not improve performance (5.110). In contrast, CATS-SD improves substantially to 5.676 from the stronger HASS+MSE+BiLD initialization, suggesting that the gain is not simply due to additional optimization steps or wall-clock compute. Instead, the tree-based loss provides an additional supervision signal that helps the draft model escape the plateau reached by both the original HASS objective and the HASS+MSE+BiLD stage.

C Post-hoc Softening Baselines on HASS

A natural concern is that CATS-SD’s gains might be explained by generic distribution softening rather than by the path-level training objective. To test this, we apply post-hoc baselines on a frozen HASS draft model *without retraining*: (i) global temperature scaling T via softmax(logits/ T) during tree construction; (ii) depth-dependent temperature $T_d = T \cdot (1 + \alpha \cdot d)$; and (iii) label smoothing

$p' = (1 - \varepsilon)p + \varepsilon/|V|$, where $|V|$ is the vocabulary size. We sweep $T \in \{1.0, 1.2, 1.5, 2.0, 3.0\}$, $\varepsilon \in \{0.05, 0.1, 0.2\}$, and $\alpha \in \{0.1, 0.2, 0.3\}$ on GSM8K at tree depth $d = 16$.

Table 7: Post-hoc baselines on GSM8K ($d = 16$). Best settings per family are shown; stronger softening ($T \geq 2$ or $\alpha \geq 0.2$) further hurts τ .

Method	$\tau \uparrow$
HASS	5.13
HASS + temp. ($T = 1.2$)	5.18
HASS + label smooth. ($\varepsilon = 0.1$)	5.16
HASS + depth-temp. ($\alpha = 0.1$)	5.17
HASS + temp. ($T = 3.0$)	3.52
CATS-SD	5.68

Table 7 summarizes the best configuration in each family. On HASS, the strongest post-hoc variant reaches only $\tau = 5.18$, still far below CATS-SD (5.68). Aggressive softening substantially degrades acceptance length (e.g., $T = 3.0$ drops HASS to 3.52). These results indicate that CATS-SD’s improvements are not attributable to generic distribution softening, but to path-level training-time supervision.