Token-Level On-Policy Contrastive Learning for Black-box Distillation Without Full Rollouts

Jian Yang[1], Chern Hong Lim[1], Thuy-Trang Vu[2]
[1]School of Information Technology, Monash University Malaysia
[2]Faculty of Information Technology, Monash University, Australia
jyan0311@student.monash.edu
lim.chernhong@monash.edu
trang.vu1@monash.edu

Abstract

Black-box distillation trains student models using only teacher-generated outputs, making it practical when teacher models are accessible only through APIs. However, existing methods either rely mainly on positive imitation or require costly full-response rollouts to obtain corrective signals. We show that such corrective signals can be obtained locally from teacher trajectories alone. At each position, the teacher token provides a positive anchor, while a high-probability alternative sampled from the student provides an informative negative. Based on this insight, we propose Token-level On-policy contrastive Learning for black-box Distillation (TOLD), which samples a single negative token from the student's distribution and optimizes a Bradley-Terry objective anchored by negative log-likelihood. This enables the student to reinforce teacher tokens while suppressing its own likely mistakes, without teacher logits, reward models, or full-response rollouts. Experiments on reasoning and instruction-following tasks show that TOLD outperforms or matches strong black-box distillation baselines and achieves up to $1.88\times$ training speedup over rollout-based on-policy baselines. Code is available at https://github.com/jyan0311/TOLD.

1 Introduction

Knowledge distillation (KD) (Hinton et al., 2015; Kim and Rush, 2016) provides a practical way to transfer the capabilities of large language models (LLMs) to smaller, more efficient models. However, many frontier LLMs are proprietary and accessible only through APIs or pre-collected corpora (Achiam et al., 2023; Comanici et al., 2025; Guo et al., 2025), without access to their parameters or token probabilities. This motivates *black-box distillation*, where knowledge must be transferred using only teacher-generated responses (Kim and Rush, 2016; Guo et al., 2025). While substantially more accessible than white-box distillation, this setting provides much weaker supervision, making it challenging to effectively transfer the teacher's behavior to the student.

The dominant approach to black-box distillation is sequence-level knowledge distillation (SeqKD) (Kim and Rush, 2016; Guo et al., 2025), which maximizes the likelihood of teacher-generated responses. While simple and effective, SeqKD provides only positive supervision. It reinforces teacher-selected tokens without explicitly correcting the student's likely alternatives. On-policy distillation addresses this limitation by training on student-generated trajectories, exposing the model to states induced by its own predictions (Agarwal et al., 2024). However, existing approaches typically require teacher logits on these student-generated states and full autoregressive student rollouts. Sequence-level preference and adversarial approaches similarly rely on complete student responses to construct corrective signals (Rafailov et al., 2023; Ye et al., 2025). These requirements are particularly costly for long reasoning trajectories.

We argue that teacher trajectories contain richer supervision than maximum-likelihood training alone can exploit. At each decoding position, the teacher-selected token provides a positive anchor, while the student's high-probability alternatives reveal its likely errors under the same prefix. Although teacher probabilities over these alternatives are unavailable, their local contrast provides a useful corrective signal. Preference optimization can exploit this signal by reinforcing the teacher token while suppressing plausible student alternatives, enabling local error correction without teacher logits or reward models.

Building on this insight, we propose Token-level On-policy contrastive Learning for black-box Distillation (TOLD), which constructs token-level preferences by contrasting each teacher-selected

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 705–718
November 6-10, 2026 ©2026 Association for Computational Linguistics

token with a high-probability alternative sampled from the current student. These local preferences are optimized using a Bradley-Terry objective, with a frozen reference model providing implicit preference scores without requiring a discriminator or reward model. To prevent likelihood displacement, we combine preference optimization with an negative log likelihood (NLL) objective that anchors learning to the teacher trajectories (Razin et al., 2025). These two objectives reinforce teacher-selected tokens while adaptively suppressing likely student errors, focusing corrective updates where the student disagrees most with the teacher.

We evaluate TOLD on reasoning and instruction-following tasks across two model families and model scales. Experimental results show that TOLD consistently outperforms or matches strong black-box distillation baselines while reducing training cost by avoiding full student rollouts. These results demonstrate that teacher trajectories alone can provide effective local contrastive supervision for black-box distillation.

In summary, our contributions are:

- We show that teacher trajectories can provide local contrastive supervision by pairing each teacher token with an on-policy alternative sampled from the current student, requiring neither teacher logits nor full student rollouts.
- We propose TOLD, a token-level preference optimization framework that combines local contrastive learning with an NLL anchor and naturally focuses corrective updates on positions where the student disagrees with the teacher.
- We show that the gradient decomposes into an NLL term plus a focal-loss-style suppression term, naturally focusing learning on the student's hardest errors.
- Experiments across reasoning and instruction-following tasks show consistent gains in the main Qwen 2.5 setting, strong transfer at Llama 3.2-3B, and substantial training speed-up over rollout-based on-policy distillation.

2 Preliminaries

2.1 Blackbox Distillation

Let π_T denote a teacher model and π_θ a student model parametrised by θ. In black-box distillation, the teacher's parameters and token probabilities are inaccessible. Instead, we can query the teacher through an API or use pre-collected outputs to obtain $y^T = (y_1^T, \ldots, y_T^T) \sim \pi_T(\cdot \mid x)$, forming a dataset $\mathcal{D} = \{(x, y^T)\}$. The goal of black-box distillation is to transfer the behavior of π_T to π_θ using only these observed teacher trajectories. A standard approach is sequence-level knowledge distillation (SeqKD) (Kim and Rush, 2016), which performs supervised fine-tuning (SFT) on teacher trajectories:

$$\mathbb{E}_{(x,y^T)\sim\mathcal{D}} \sum_{t=1}^{T} \log \pi_\theta(y_t^T \mid x, y_{<t}^T). \tag{1}$$

SeqKD is simple, but provides only positive supervision as it reinforces teacher-selected tokens without explicitly correcting the student's likely mistakes.

2.2 On-Policy Distillation

On-Policy Distillation (OPD) instead trains on trajectories generated by the student, $y^S \sim \pi_\theta(\cdot \mid x)$, with the teacher providing token-level supervision along these trajectories. The training objective is as follow

$$\mathbb{E}_{x,\ y^S\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}\Big(\pi_\theta(\cdot \mid x, y_{<t}^S) \,\|\, \pi_T(\cdot \mid x, y_{<t}^S)\Big) \right]. \tag{2}$$

By training on student-generated contexts, OPD reduces the train-inference mismatch of offline distillation and provides supervision on the student's own mistakes (Agarwal et al., 2024). However, it requires access to teacher logits and full student rollouts, introducing costly autoregressive generation during training, particularly for long sequences.

3 Token-level On-policy Contrastive Learning for Black-box Distillation

To address the lack of teacher logits and the high cost of full student rollouts, we propose TOLD, which obtains on-policy corrective signals without either requirement. Our key observation is that, at each teacher prefix, the teacher token serves as a positive anchor, while the current student provides a plausible negative alternative. TOLD exploits this local contrast to perform on-policy preference learning at the token level.

3.1 Local Token-Level Contrastive Signals

At each position t, given the teacher prefix $c_t^T = (x, y_{<t}^T)$, the teacher token y_t^T provides a positive

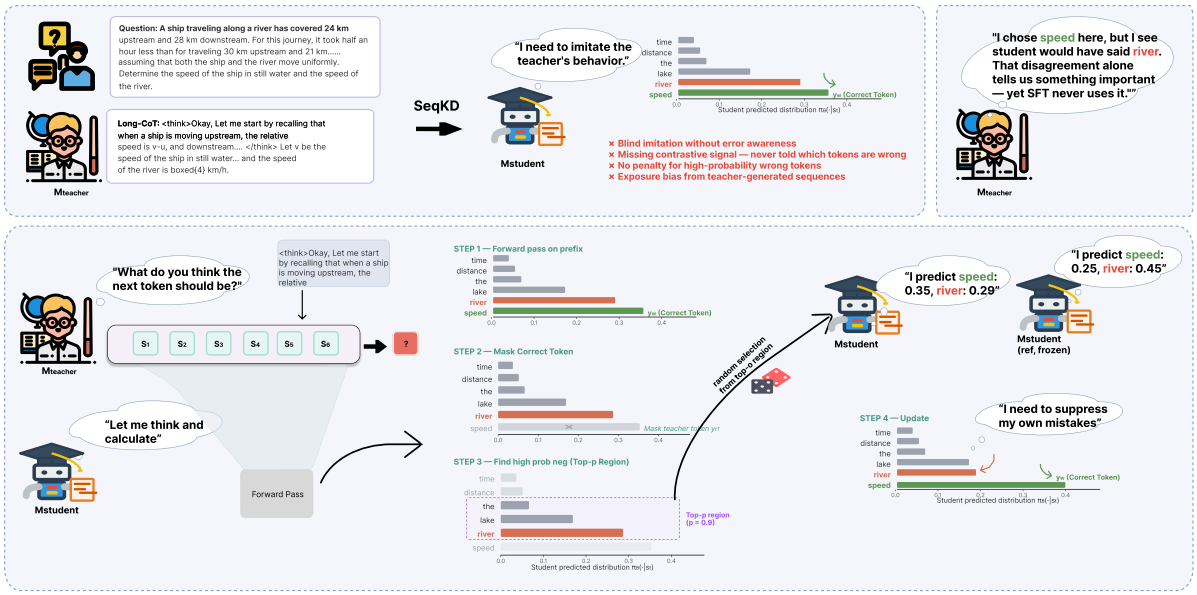


Figure 1: Overview of TOLD. At each token position t, we sample an on-policy negative $\tilde{a}_t$ from the student and impose a local preference constraint $y_t^T \succ \tilde{a}_t$, using the teacher token as the positive anchor for a local contrast against the student's own high-probability alternative. The combined objective (NLL + STN-DPO) exploits both the positive and contrastive signal in teacher trajectories without autoregressive rollout.

target, while the current student distribution $\pi_\theta(\cdot \mid c_t^T)$ provides plausible alternatives. We sample a high-probability alternative from the student[1] and form a local preference $(y_t^T, \tilde{a}_t)$, such that

$$\tilde{a}_t \sim \pi_\theta(\cdot \mid c_t^T,\ a \neq y_t^T), \qquad y_t^T \succ \tilde{a}_t. \tag{3}$$

Contrasting the teacher token with the student's own high-probability alternative provides a corrective signal that focuses learning on positions where the student disagrees with the teacher. Since the negative token $\tilde{a}_t$ is sampled from the current student, this signal is on-policy and naturally adapts as the student evolves. It requires neither teacher logits nor teacher-labeled negative examples. Moreover, all token distributions can be obtained in a single teacher-forced forward pass, avoiding costly autoregressive student rollouts.

3.2 Token-Level Preference Optimization

Given the local preference pair $y_t^T \succ \tilde{a}_t$, we optimize the student to increase the relative preference for the teacher token over the student-sampled negative. Following Rafailov et al. (2023) and Azar et al. (2024), we define the implicit token-level preference score

$$\delta_t(a) = \log \pi_\theta(a \mid c_t^T) - \log \pi_{\text{ref}}(a \mid c_t^T), \tag{4}$$

where π_{ref} is a frozen reference policy obtained by fine-tuning the student on teacher trajectories. The score measures how much the probability of token a has shifted from the reference policy.

We then optimize a token-level Bradley–Terry objective:

$$\mathcal{L}_{\text{pref}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log \sigma\Big(\beta\left[\delta_t^w - \delta_t^l\right]\Big), \tag{5}$$

where

$$\begin{aligned} \delta_t^w &= \text{clip}\big(\delta_t(y_t^T), -C, C\big), \\ \delta_t^l &= \text{clip}\big(\delta_t(\tilde{a}_t), -C, C\big). \end{aligned} \tag{6}$$

Here, β controls the strength of the preference contrast, while clipping with $C > 0$ prevents a small number of extreme token-level scores from dominating optimization.

The objective increases the preference margin between the teacher token and the student's own alternative. Its gradient is weighted by $1 - \sigma\big(\beta(\delta_t^w - \delta_t^l)\big)$, assigning larger updates to positions where the student does not yet strongly prefer the teacher token, while diminishing updates once the local preference is satisfied. This provides an implicit hard-example focusing effect (Lin et al., 2017), concentrating corrective supervision where it is most needed.

[1] In practice, we first mask y_t^T by setting its logit to $-\infty$, apply a top-k pre-truncation ($k = 256$) for memory efficiency, and then perform nucleus sampling with threshold $p = 0.9$.

3.3 Training Objective

The preference objective in Section 3.2 optimizes the relative ordering between the teacher token and the student-sampled alternative. However, preference optimization alone does not guarantee that the likelihood of the teacher token increases. Both probabilities can decrease while preserving their relative preference. This *likelihood displacement* phenomenon has been observed in prior preference optimization methods (Rafailov et al., 2023; Zeng et al., 2024; Ethayarajh et al., 2024).

To prevent this failure mode, we therefore anchor the student to the teacher trajectory with an negative log likelihood (NLL) loss:

$$\mathcal{L}_{\mathrm{NLL}}(\theta) = -\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \log \pi_\theta(y_t^T \mid c_t^T). \quad (7)$$

The final TOLD objective combines positive imitation with local corrective supervision:

$$\mathcal{L}_{\mathrm{TOLD}}(\theta) = \mathcal{L}_{\mathrm{NLL}}(\theta) + \mathcal{L}_{\mathrm{pref}}(\theta). \quad (8)$$

The NLL term reinforces the teacher-selected tokens, while the preference term suppresses high-probability alternatives proposed by the student. Together, they encourage the student to increase the likelihood of teacher behavior while correcting its own local ranking errors. A formal gradient decomposition and analysis of likelihood displacement are provided in Appendix C.

Algorithm 1 summarizes the training procedure of TOLD. For each mini-batch of teacher trajectories, we perform teacher-forced forward passes through the student and frozen reference model (Line 5). At each position, we sample a negative token from the student's top-p distribution (Line 6) and compute the preference and NLL losses (Lines 8-9). Since all token-level signals are obtained under teacher forcing, TOLD avoids full student rollouts and requires no autoregressive student generation.

Algorithm 1 TOLD Training

Require: Teacher corpus $\mathcal{D} = \{(x, y^T)\}$; student π_θ initialized from π_{ref}; hyper-parameters β, C

1: **for** each mini-batch $(x, y^T) \sim \mathcal{D}$ **do**
2: **for** each token position $t \in \{1, \ldots, T\}$ **do**
3: $c_t^T \leftarrow (x, y_{<t}^T)$ ▷ teacher prefix
4: $\tilde{a}_t \sim \pi_\theta^{\text{top-}p}(\cdot \mid c_t^T)$ ▷ on-policy negative
5: $\delta_t^w \leftarrow \mathrm{clip}\Big(\log \frac{\pi_\theta(y_t^T|c_t^T)}{\pi_{\mathrm{ref}}(y_t^T|c_t^T)}, -C, C\Big)$
6: $\delta_t^l \leftarrow \mathrm{clip}\Big(\log \frac{\pi_\theta(\tilde{a}_t|s_t)}{\pi_{\mathrm{ref}}(\tilde{a}_t|s_t)}, -C, C\Big)$
7: **end for**
8: $\mathcal{L}_{\mathrm{NLL}} \leftarrow -\frac{1}{T}\sum_t \log \pi_\theta(y_t^T \mid s_t)$
9: $\mathcal{L}_{pref} \leftarrow -\frac{1}{T}\sum_t \log \sigma\big(\beta(\delta_t^w - \delta_t^l)\big)$
10: Update θ using

$$\nabla_\theta \Big[\mathcal{L}_{\mathrm{NLL}} + \mathcal{L}_{pref}\Big].$$

11: **end for**

4 Experiments

We empirically validate TOLD on two complementary task families: long-form mathematical reasoning (Section 4.1) and short-form instruction-following (Section 4.2). Subsequently, we analyse the training dynamics (Section 4.3) and design choices (Section 4.4).

Baselines. All methods train on the same pre-collected teacher trajectories without accessing teacher logits.

- *SeqKD* (Kim and Rush, 2016): pure NLL training on teacher trajectories ($\mathcal{L}_{\mathrm{NLL}}$ only).
- *DPO (w/o SFT)*: sequence-level DPO without any NLL/SFT component.
- *SFT + DPO*: sequence-level DPO jointly optimized with the NLL loss.
- *GAD* (Ye et al., 2025): an explicit discriminator jointly trained with the student via GRPO.

4.1 Mathematical Reasoning

Setup. We train 1.5B and 3B Qwen 2.5 student models on OpenR1-Math-8192 (Hugging Face, 2025), which provides pre-generated long chain-of-thought reasoning traces. All methods are trained on the same fixed supervision data without additional teacher rollouts. The reference policy π_{ref} is initialized from a SFT model trained for one epoch on the same data and then frozen. We evaluate pass@1 accuracy on AIME 2024/2025 (Shao et al., 2024), MATH-500 (Hendrycks et al., 2021), and GPQA (Rein et al., 2023) using LightEval (Habib et al., 2023). Additional implementation details are provided in Appendix A. We fix $(\beta, C) = (0.1, 3)$ across all experiments. Sensitivity analysis is provided in Section 4.4.

Results. Table 1 shows that TOLD achieves the best or tied-best result on 7 out of 8 column metrics across both model scales. At 3B, TOLD improves the average trend across benchmarks, with the largest numerical gains on AIME 2024 (+6.67

Method	Qwen2.5-1.5B				Qwen2.5-3B			
	AIME24	AIME25	MATH	GPQA	AIME24	AIME25	MATH	GPQA
Base	3.33	0.00	26.31	13.13	6.67	3.33	44.00	25.76
DPO (w/o SFT)	**6.67**	3.33	47.00	27.78	6.67	6.67	48.23	28.00
SFT+DPO	3.33	6.67	49.40	30.81	6.67	10.00	57.40	31.00
SeqKD	3.33	**10.00**	49.80	**31.32**	10.00	**13.33**	59.80	29.29
GAD	**6.67**	**10.00**	52.31	28.10	6.67	10.00	58.78	23.33
TOLD (Ours)	**6.67**	**10.00**	**53.40**	30.29	**16.67**	**13.33**	**64.40**	**35.80**

Table 1: Reasoning distillation results (pass@1) on OpenR1-Math-8192. *Base*: pre-trained checkpoint without fine-tuning. Best per column is **bold**.

over the next-best method) and GPQA (+4.80). A detailed cross-task and cross-scale analysis follows in Section 4.3. We assess statistical reliability via bootstrap resampling (1,000 replicates, 95% CI). Although TOLD improves over SeqKD on AIME 2024 at 3B scale (16.67% vs. 10.00%), a bootstrap test does not establish significance ($p = 0.18$; $n = 30$). In contrast, the gain on MATH-500 (64.40% vs. 59.80%) is significant ($p < 0.05$). We therefore view the consistent gains across benchmarks and scales as stronger evidence than any single AIME comparison.

4.2 General Instruction-Following

To assess the generality of TOLD beyond long-form reasoning, we evaluate on a complementary regime: short-form instruction-following with limited training data.

Setup. We train Qwen 2.5-Instruct students at 1.5B and 3B scales on 10K teacher-generated responses sampled from Ultrafeedback (Cui et al.), a multi-domain instruction-following dataset with substantially shorter responses than the chain-of-thought traces used in reasoning distillation. The reference policy π_{ref} is initialized identically to the reasoning setup: one epoch of supervised fine-tuning on teacher trajectories followed by freezing. All hyperparameters are reused without task-specific tuning. We evaluate on AlpacaEval 2 (Dubois et al., 2024) (length-controlled and raw win rates against the GPT-4 reference model, `gpt-4-1106-preview`) and MT-Bench (Zheng et al., 2023) (first- and second-turn scores). Additional evaluation details are provided in Appendix B.

Results. Table 2 shows that TOLD achieves the best result on all 8 column metrics. The gains are most pronounced on AlpacaEval: at 3B, AE-LC reaches 52.32 versus 38.78 for the next-best baseline (GAD), a margin of +13.54 points.

Cross-family instruction-following. To test whether the gains are specific to Qwen 2.5, we additionally evaluate Llama 3.2 students under the same UltraFeedback 10K teacher-response protocol. All methods are initialized from the same warmup SFT checkpoint.

Table 3 shows that TOLD is not tied to the Qwen 2.5 family. The 1B setting is mixed, while at 3B TOLD achieves the best result on all four metrics.

AlpacaEval judge robustness. Because our main AlpacaEval evaluation uses an open judge for reproducibility, we also re-judge Qwen 2.5-3B outputs with Azure `gpt-4o-mini` and compute length control using the official AlpacaEval procedure.

Table 4 indicates that the conclusion is not an artifact of the original judge: absolute calibration changes, but TOLD remains best under both raw and length-controlled AlpacaEval.

4.3 Analysis

We analyse the results from four perspectives: cross-task consistency, training dynamics, dataset-level statistical evidence, and training efficiency.

Consistency across tasks and scales. Comparing the 3B columns in Tables 1 and 2, TOLD achieves the best results on the competition-level AIME benchmarks, both AlpacaEval axes, and all MT-Bench turns for Qwen 2.5, and shows similar gains on Llama 3.2 at 3B (Table 3). In contrast, sequence-level DPO variants are less stable:

Method	Qwen2.5-1.5B				Qwen2.5-3B			
	AE-LC	AE-WR	MT-1T	MT-2T	AE-LC	AE-WR	MT-1T	MT-2T
Teacher (7B-Instruct)	75.47	92.73	8.90	7.76	75.47	92.73	8.90	7.76
Base	2.56	4.66	1.58	1.51	27.45	25.78	5.60	4.66
DPO (w/o SFT)	25.11	23.42	4.56	2.52	31.32	30.90	4.68	3.59
SFT+DPO	21.53	20.12	4.15	2.57	37.43	38.13	5.23	3.48
SeqKD	18.62	18.60	4.19	2.87	36.94	36.77	4.90	3.32
GAD	27.50	35.84	5.44	3.43	38.78	56.09	6.45	4.24
TOLD (Ours)	**38.14**	**49.17**	**5.95**	**3.50**	**52.32**	**63.04**	**6.59**	**5.20**

Table 2: Instruction-following distillation results (Qwen 2.5 7B-Instruct → 1.5B / 3B, Ultrafeedback). *Base*: pre-trained checkpoint without fine-tuning. Best per column is **bold**.

Model	Method	AE-LC	AE-WR	MT-1T	MT-2T
Llama 3.2-1B	DPO	16.50	30.29	3.28	2.19
	GAD	**18.88**	32.42	**3.62**	2.45
	TOLD	17.84	**34.81**	3.61	**2.57**
Llama 3.2-3B	DPO	45.14	60.12	5.62	3.75
	GAD	44.90	62.48	5.38	4.14
	TOLD	**53.87**	**65.96**	**5.71**	**4.83**

Table 3: Cross-family instruction-following results on Llama 3.2. TOLD is not uniformly dominant at 1B, but gives the strongest overall result at 3B.

Judge	Method	AE-LC	AE-WR
Qwen3 judge	SeqKD	36.94	36.77
	GAD	38.78	56.09
	TOLD	**52.32**	**63.04**
`gpt-4o-mini`	SeqKD	40.50	53.63
	GAD	49.98	70.40
	TOLD	**63.10**	**73.26**

Table 4: AlpacaEval judge robustness on Qwen 2.5-3B. Absolute scores vary across judges, but the relative ranking is stable.

SFT+DPO is competitive on reasoning but trails on chat, while DPO (w/o SFT) underperforms on both. This instability reflects the coarse-grained nature of sequence-level preference. A single reward for an entire trajectory averages over many token decisions, diluting the corrective signal. TOLD's token-level contrast avoids this averaging.

The advantage also grows with model capacity: on reasoning, the AIME 2024 margin over the next-best baseline widens from a tie at 1.5B to +6.67 at 3B; on chat, the AE-LC margin increases from +10.64 at 1.5B (38.14 vs. 27.50) to +13.54 at 3B (52.32 vs. 38.78). This suggests that larger students can better leverage the fine-grained token-level contrastive signal provided by TOLD.

Training dynamics under token-level contrast. To better understand the source of the observed gains, we define

$$\Delta_w = \log \frac{\pi_\theta(y_t^T \mid s_t)}{\pi_{\text{ref}}(y_t^T \mid s_t)}, \qquad \Delta_l = \log \frac{\pi_\theta(\tilde{a}_t \mid s_t)}{\pi_{\text{ref}}(\tilde{a}_t \mid s_t)},$$

where Δ_w and Δ_l denote the teacher-token and negative-token margins, respectively. Their difference, $\Delta_w - \Delta_l$, corresponds to the preference gap optimized by the Bradley–Terry objective.

Figure 2 shows two trends: SFT increases Δ_w while leaving $\Delta_l \approx 0$; TOLD drives Δ_l substantially negative while maintaining positive Δ_w, widening the preference gap. This confirms that TOLD improves distillation by actively suppressing competing alternatives.

Dataset-level suppression evidence. The aggregate training dynamics confirm suppression on average; we now verify that the effect is *position-adaptive* and *systematic* across the data.

Figure 3 (left) traces token-level log-probabilities across 12 positions of a held-out sequence. The suppression gap is widest at positions where the student is initially most confused, directly instantiating the hard-example focusing property from Section 3.2. SeqKD leaves negative-token probabilities essentially untouched. Figure 3 (right) aggregates over 465 held-out sequences (100,963 token positions): under TOLD, 99% of sequences exhibit negative

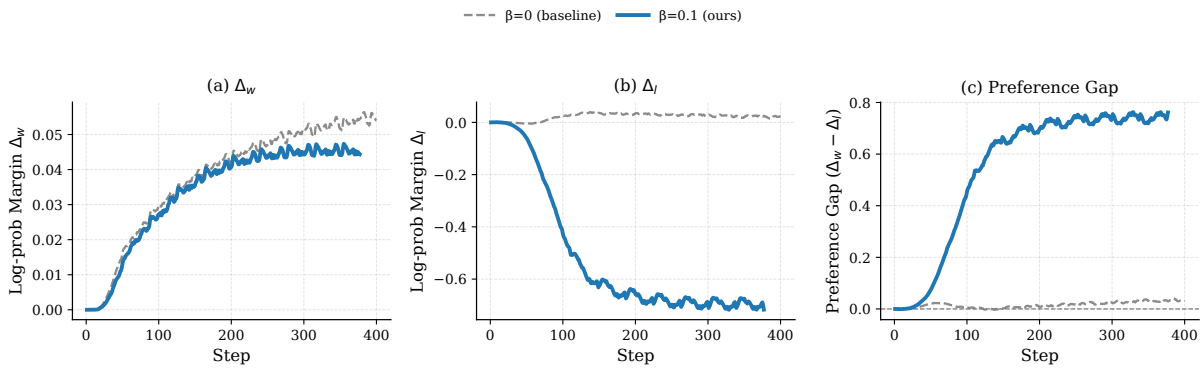


Figure 2: **Training dynamics under TOLD ($\beta = 0.1$) vs. SFT ($\beta = 0$). (a)** Δ_w: both methods raise teacher-token probability. **(b)** Δ_l: SFT leaves negatives unchanged (≈ 0); TOLD drives them negative ($\rightarrow -0.7$). **(c)** Preference gap: TOLD opens a gap an order of magnitude larger than SFT.

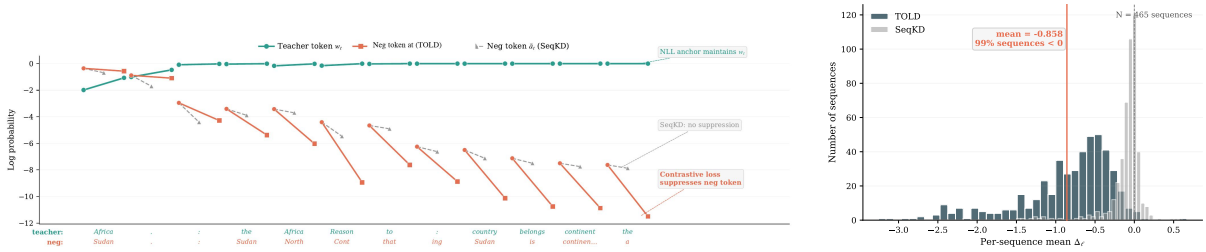


Figure 3: **Dataset-level evidence for active negative suppression. Left:** Log-probability trajectories on a held-out sequence. The teacher token (green, ●) stays near its reference level; the negative (red, ■) is strongly suppressed by TOLD; SeqKD (gray, ▲) leaves negatives unchanged. (For visualization, the negative is $\arg\max_{a \neq y_t^T} \pi_{\text{ref}}(a \mid s_t)$; during training, negatives are sampled from $\pi_\theta^{\text{top-}p}$.) **Right:** Per-sequence mean Δ_ℓ over 465 sequences; 99% show net suppression (mean $= -0.858$).

		Qwen2.5-1.5B		Qwen2.5-3B	
Method	**Rollout**	**Train (1 ep.)**	**Speedup**	**Train (1 ep.)**	**Speedup**
SeqKD	No	6.1	2.10×	9.2	2.55×
DPO (w/o SFT)	Yes	4.8	2.67×	11.9	1.97×
SFT+DPO	Yes	5.1	2.51×	12.2	1.93×
GAD	Yes	12.8	1.00×	23.5	1.00×
TOLD (Ours)	No	6.8	1.88×	12.7	1.85×

Table 5: Training-stage cost on OpenR1-Math-8192 (GPU-hours for one training epoch on a single A100-80G). *Train* reports the optimization cost for one epoch, excluding any one-time offline response generation. *Speedup* is the corresponding training-stage speedup relative to GAD.

mean Δ_ℓ (overall mean $= -0.858$), confirming that active suppression is a systematic property of the method.

Training efficiency. A core claim of TOLD is that token-level contrast avoids the autoregressive rollouts required by sequence-level methods. Table 5 reports training cost under identical hardware (A100-80G, reasoning setup).

We report training-stage acceleration using a one-epoch comparison, separating it from any one-time response generation required by response-level baselines. TOLD is not the fastest method per epoch: its cost is higher than SeqKD and slightly higher than DPO at 3B because it evaluates the frozen reference policy and samples token-level negatives. Nevertheless, it is 1.85×–1.88× faster than GAD in the training stage, while requiring no full-response rollout. SeqKD and TOLD train directly on the pre-collected teacher trajectories. DPO variants require one-time offline rejected-response generation outside the train-stage timing, whereas GAD uses rollout-related computation during training. The table reports the optimization-stage comparison only. This comparison focuses on optimization-stage acceleration and does not claim a fixed ranking for total end-to-end cost under different generation or epoch budgets.

Offline versus online DPO. To address whether fixed rejected responses make DPO artificially weak, we additionally compare with an online-refreshed DPO variant that regenerates rejected re-

Model	Method	Local LC	Local WR	GPT-4o LC/WR	MT-2T
Llama 3.2-1B	Offline DPO	16.50	30.29	16.08 / 28.73	2.19
	Online DPO	**19.33**	31.30	17.88 / 28.91	2.54
	TOLD	17.84	**34.81**	**18.91 / 32.36**	**2.57**
Llama 3.2-3B	Offline DPO	45.14	60.12	43.37 / 58.60	3.75
	Online DPO	49.66	62.11	45.95 / 59.60	3.84
	TOLD	**53.87**	**65.96**	**46.51 / 64.73**	**4.83**

Table 6: Offline versus online DPO on Llama 3.2. Online refresh strengthens DPO but requires additional full-response generation; TOLD avoids this cost.

β	AE-LC	AE-WR	MT-1T	MT-2T
0.1 (def.)	**38.14**	**49.17**	**5.95**	3.50
0.5	37.10	47.76	5.55	3.18
1	37.68	46.10	5.31	3.18
3	35.71	45.28	5.54	3.25
5	35.98	45.76	5.05	**3.66**

Table 7: Effect of β on instruction-following (Qwen 2.5 1.5B). Highlighted row is the default.

sponses from the current policy before subsequent DPO training rounds.

Table 6 shows that online refresh improves DPO, but TOLD remains stronger on most metrics, especially at 3B. The online variant is a stronger comparison in accuracy, but its additional full-response refreshes change the compute regime relative to the fixed-data setting used by TOLD.

4.4 Ablation Study

We ablate the BT temperature β and the single-token negative structure itself, and discuss the negative-sampling threshold.

BT temperature β. Table 7 varies β on instruction-following (Qwen 2.5 7B-Instruct $\rightarrow$ 1.5B, Ultrafeedback; $C=3$ fixed).

The default $\beta = 0.1$ achieves the best results on 3/4 metrics. Performance degrades monotonically with increasing β on AlpacaEval (AE-LC: $38.14 \rightarrow 35.98$). Intuitively, because the BT loss operates on single-token logit differences, a large β over-sharpens the contrastive signal and destabilizes optimization, especially for short responses where the NLL anchor already provides strong supervision. We therefore fix $\beta = 0.1$ across all experiments.

Negative-sampling nucleus threshold. Our default uses top-$p = 0.9$ when sampling the student negative. This choice is intended to draw plausible high-probability alternatives near the student's decision boundary while avoiding very low-quality tail tokens.

Config	Rollout	Neg. K	EM (%)	Time (min)
Default (ours)	1	1	**30.2**	44 (1.0×)
Rollout = 3	3	1	27.4	53 (1.2×)
Rollout = 5	5	1	28.1	62 (1.4×)
Neg $K = 3$	1	3	25.9	44 (1.0×)
Neg $K = 5$	1	5	29.6	44 (1.0×)

Table 8: Sensitivity to rollout steps and number of negatives on GSM8K (Qwen 2.5 1.5B). Highlighted row is the default.

Rollout steps and number of negatives. Table 8 tests two natural extensions on GSM8K (Qwen 2.5 1.5B, 2k subset, 3 epochs): longer autoregressive rollouts and multiple negatives per position.

The single-step, single-negative default achieves the best accuracy *and* the lowest cost. Longer rollouts (3 or 5 steps) hurt accuracy ($30.2 \rightarrow 27.4$ / 28.1) while adding up to $1.4\times$ wall-time. We attribute this to *prefix mismatch*: later steps evaluate teacher tokens under student-generated prefixes that diverge from the teacher trajectory, weakening the contrastive signal. Multiple negatives ($K = 3, 5$) add negligible compute but also reduce accuracy, suggesting that averaging over negatives dilutes the optimization pressure on the single hardest competitor. The simplest configuration thus provides the best tradeoff.

5 Related Work

Knowledge distillation for language models. Classical knowledge distillation (KD) transfers teacher behavior to a smaller student by matching token-level distributions during generation (Hinton et al., 2015; Kim and Rush, 2016). Most existing methods assume white-box access to teacher logits and optimize divergence-based objectives such as forward or reverse KL (Agarwal et al., 2024; Ko et al., 2024; Gu et al., 2024). Recent work explores balanced divergences and adaptive weighting strategies under settings where teacher logits or reward signals are available (Wen et al.). In contrast, we study the strict black-box setting where such information is unavailable.

Black-box distillation. When only teacher-generated outputs are available, supervised fine-

tuning (SFT) or sequence-level KD (SeqKD) becomes the dominant approach (Kim and Rush, 2016; Guo et al., 2025). However, training solely on teacher prefixes leaves the student with little direct correction for its own high-probability local alternatives, and can still suffer from train–test distribution mismatch during autoregressive generation (Ross et al., 2011). Recent approaches incorporate on-policy student rollouts to alleviate this mismatch (Agarwal et al., 2024; Li et al., 2026), but they require either teacher likelihood evaluation or teacher feedback on student-generated prefixes. GKD (Agarwal et al., 2024), for example, trains on student-generated sequences by leveraging feedback from the teacher on those sequences. This assumption differs from our strict black-box setting, where only pre-collected teacher trajectories are available.

GAD (Ye et al., 2025) trains an auxiliary discriminator to distinguish teacher and student outputs and updates the student via policy-gradient optimization, but adversarial training introduces discriminator balancing, high-variance rollout updates, and substantial sequence-level generation cost. Jiang et al. (2023) distil from API outputs via adversarial imitation, and Singh et al. (2025) cast distillation as preference-based optimization. These methods all operate at the sequence level and require full student rollouts. TOLD takes a different approach: it extracts contrastive supervision directly at the token level from teacher trajectories, requiring neither a learned discriminator nor autoregressive rollouts.

Preference optimization and implicit rewards. Recent preference-optimization methods replace explicit reward modeling with implicit objectives defined over policy log-ratios (Rafailov et al., 2023; Azar et al., 2024; Ethayarajh et al., 2024; Meng et al., 2024; Hong et al., 2024; Tang et al., 2024). Among them, DPO formulates preference learning through Bradley–Terry comparisons between preferred and rejected responses. TDPO extends DPO to token-level preference alignment by decomposing sequence-level preferred/rejected response pairs and adding token-level KL control for alignment and diversity (Zeng et al., 2024). This is closely related in objective form, but differs in supervision and access assumptions: TDPO starts from complete preference pairs (x, y_w, y_l), whereas TOLD studies black-box distillation from teacher trajectories alone and constructs a single-token contrast at each teacher-forced state using a student-sampled negative.

However, preference objectives are often shift-invariant and may induce likelihood displacement or reward squeezing (Razin et al., 2025; Park et al., 2024; Rafailov et al.). Several methods mitigate this issue by combining preference learning with SFT-style anchors (Chen et al., 2024; Pang et al., 2024; Yuan et al., 2024), though these approaches typically rely on sequence-level rollouts.

TOLD applies token-level preference optimization to strict black-box distillation, replacing pre-collected preference pairs with on-policy negatives from teacher trajectories and an NLL anchor.

6 Conclusion

We presented TOLD, a token-level contrastive framework for black-box knowledge distillation. By contrasting each teacher token with one student-sampled negative, TOLD improves local ranking without teacher logits, preference pairs, or full-response rollouts. Across Qwen 2.5 reasoning and instruction-following benchmarks, it matches or outperforms the baselines; cross-family evaluation shows strongest gains at Llama 3.2-3B while results at 1B are mixed.

The results support using teacher trajectories as local contrastive supervision and show that a single-negative update offers a useful training-stage cost–performance tradeoff without sequence-level rollouts.

Limitations

TOLD targets the pure black-box setting and uses only sampled teacher trajectories; it does not exploit teacher logits or intermediate hidden states, and we therefore do not compare against white-box distillation methods.

Our experiments cover the Qwen 2.5 and Llama 3.2 model families at small-to-medium scales and two task families (long-form mathematical reasoning and short-form instruction-following); generalisation to multilingual, multimodal, or larger-scale (>10B) regimes remains to be validated.

The single-token negative design replaces sequence-level rollouts with a local, position-wise contrast. While this yields substantial computational savings, it captures only token-level preferences and forgoes the long-range coherence signal that full rollouts could provide; the NLL anchor

partially compensates for this limitation but does not fully eliminate it.

Finally, TOLD relies on a frozen reference policy obtained from one epoch of SFT on teacher trajectories. The sensitivity of the framework to the quality and training duration of this reference initialisation has not been exhaustively studied; a suboptimal reference could compress the effective dynamic range of the implicit preference score and weaken the contrastive signal.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful comments. This research/work was supported by Monash eResearch capabilities, including M3.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, and 1 others. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. *URL https://arxiv. org/abs/2310.01377*, 13.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. LightEval: A lightweight framework for LLM evaluation.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3134–3154, Singapore. Association for Computational Linguistics.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and 1 others. 2026. Rethinking on-policy distillation of large language

models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Richard Y Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4998–5017.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q*: Your language model is secretly a q-function, 2024. *URL https://arxiv.org/abs/2404.12358*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. 2025. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, volume 2025, pages 24791–24834.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aasheesh Singh, Vishal Vaddina, and Dagnachew Birru. 2025. ORPO-distill: Mixed-policy preference optimization for cross-architecture LLM distillation. In *NeurIPS 2025 Workshop on Efficient Reasoning*.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. 2024. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation, 2023. *URL https://arxiv.org/abs/2307.15190*.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. 2025. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*.

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. *Advances in neural information processing systems*, 36:46595–46623.

A Implementation Details

This appendix provides the complete training recipe for all TOLD and baseline numbers in Tables 1 and 2. All methods use the same codebase, teacher trajectories, and nominal optimization budget, while their objectives and rollout mechanisms differ.

A.1 Models and Reference Policy

We use the Qwen 2.5 family (Qwen et al., 2025) at 1.5B and 3B scales. For reasoning, teacher trajectories are drawn from OpenR1-Math-8192 (Hugging Face, 2025), a public corpus of long chain-of-thought traces; for instruction-following, the teacher is Qwen 2.5-7B-Instruct. The reference policy π_{ref} is obtained by one epoch of SFT on the same teacher trajectories starting from the pretrained checkpoint, and is then frozen. The student π_θ is initialized from π_{ref}.

A.2 Dataset Statistics

Table 9 summarizes dataset characteristics.

Dataset	#Prompts	Avg Len	Max Len
OpenR1-Math-8192	∼45K	∼4,200	8,192
Ultrafeedback (subset)	10K	∼480	4,096

Table 9: Training data statistics.

A.3 Optimization Hyperparameters

We optimize $\mathcal{L}_{\text{TOLD}}$ (Equation (8)) with hyperparameters fixed across all tasks. Table 10 lists the full configuration.

Table 10: Shared hyperparameters for TOLD.

Hyperparameter	Value
BT temperature β	0.1
Symmetric clip C	3
Negative top-k pre-truncation	256
Negative top-p (nucleus)	0.9
Optimizer	AdamW
Learning rate	5×10^{-6}
LR schedule	cosine, 100-step warmup
Weight decay	0.0
Batch size (sequences)	64 (reason.) / 256 (chat)
Max sequence length	8192 (reason.) / 4096 (chat)
Epochs	6 (reason.) / 3 (chat)
Precision	bf16

A.4 Hardware

All runs use NVIDIA A100 80 GB GPUs with PyTorch FSDP.

A.5 Evaluation Details (Reasoning)

We evaluate pass@1 on AIME 2024/2025, MATH-500, and GPQA-Diamond using LightEval (Habib et al., 2023): sampling temperature 0.7, top-p 0.95. Maximum generation length: 10,240 tokens. The system prompt is listed in Appendix B.

A.6 Baseline-Specific Notes

- *SeqKD* (Kim and Rush, 2016): identical NLL term as TOLD, no contrastive component.
- *DPO (w/o SFT)* (Rafailov et al., 2023): sequence-level DPO with the same π_{ref} and β=0.1. Rejected responses are generated offline from π_{ref} (temperature 0.7, top-p 0.95), paired with teacher trajectories as chosen responses.
- *SFT + DPO*: SeqKD + sequence-level DPO at equal weight; rejected responses follow the same offline generation procedure.
- *GAD* (Ye et al., 2025): initialized from the same one-epoch SFT reference as other methods. At each update, student responses are sampled with temperature 0.7 and top-p 0.95 and optimized with GRPO-style rollout updates. We use β=5, λ_P=0.3, the same learning rate and batch-size budget as TOLD, and evaluate checkpoints with the same primary decoding and judge protocol used for all methods; the separate judge robustness check is described in Section 4.2.

B Prompts and Chat Evaluation Details

We list the exact prompts and evaluation configurations for full reproducibility. All chat tasks (training, AlpacaEval 2, MT-Bench) share the same system prompt:

```
You are a helpful assistant.
```

B.1 Reasoning Training & Evaluation

The following system prompt is prepended to every reasoning input during both training and evaluation:

```
Please reason step by step, and
put your final answer within
\boxed{}.
```

The user message is the problem statement verbatim from the dataset (e.g. AIME, MATH-500, GPQA). No few-shot examples are used.

B.2 Instruction-Following Training

During distillation on Ultrafeedback, we use the Qwen 2.5 chat template with the shared system

prompt above. The user message is the instruction from Ultrafeedback. The teacher response is generated by Qwen 2.5-7B-Instruct with temperature 0.7, top-p 0.95, max 4,096 tokens.

B.3 AlpacaEval 2 Evaluation

We follow Dubois et al. (2024): length-controlled (LC) and raw (WR) win rates against the GPT-4 reference model (`gpt-4-1106-preview`). The official AlpacaEval 2 prompt template (`alpaca_eval_gpt4_fn`) is used without modification; the system prompt is identical to training (see above).

- Judge model: `Qwen3-30B-A3B-Instruct`.
- Comparison reference: `gpt-4-1106-preview`.
- Student decoding: temperature 0.7, top-p 0.95, max 4,096 tokens.

Judge model justification. For the primary AlpacaEval 2 and MT-Bench comparisons, we use `Qwen3-30B-A3B-Instruct` rather than a closed GPT-4 judge for two reasons:

- reproducibility—closed-API judges are subject to silent model updates, making exact replication impossible;
- cost—evaluating all baselines across two model scales requires thousands of judge calls, which is prohibitive with commercial APIs.

All primary comparisons within each benchmark use this same judge, so relative differences are not confounded by judge changes. We also report a separate `gpt-4o-mini` re-judging experiment in Table 4; that table intentionally uses a second judge to test robustness across judge models.

B.4 MT-Bench Evaluation

Following Zheng et al. (2023), we use the official MT-Bench prompt set (80 multi-turn questions across 8 categories) with the shared system prompt above.

- Judge model: `Qwen3-30B-A3B-Instruct` in single-answer grading mode (score 1–10).
- Decoding parameters match AlpacaEval.
- Both first-turn and second-turn are evaluated.

C Gradient Decomposition and Anti-Squeezing

We formalize the claim from Section 3.2: the per-position gradient of $\mathcal{L}_{\text{TOLD}}$ decomposes into an NLL term and a focal-loss-style suppression term, jointly preventing likelihood displacement (Razin et al., 2025).

Setup. Fix a prefix $s_t = (x, y^T_{<t})$. For a self-contained proof, write the unclipped counterparts of Equation (6) as

$$\delta^w_t = \log \frac{\pi_\theta(y^T_t \mid s_t)}{\pi_{\text{ref}}(y^T_t \mid s_t)}, \tag{9}$$

$$\delta^l_t = \log \frac{\pi_\theta(\tilde{a}_t \mid s_t)}{\pi_{\text{ref}}(\tilde{a}_t \mid s_t)}, \tag{10}$$

and define the BT confidence $g_t = \sigma\big(\beta(\delta^w_t - \delta^l_t)\big)$. Assume both scores lie inside the clip interval $(-C, C)$.

Proposition 1 (Gradient decomposition)**.** *The per-position gradient satisfies*

$$\begin{aligned}
&\nabla_\theta \mathcal{L}^{(t)}_{\text{TOLD}} \\
&= \underbrace{-\nabla_\theta \log \pi_\theta(y^T_t \mid s_t)}_{\textit{NLL: raise teacher token}} \\
&\quad + \underbrace{\beta(1-g_t)\, \nabla_\theta \log \pi_\theta(\tilde{a}_t \mid s_t)}_{\textit{focal suppression on } \tilde{a}_t} \\
&\quad - \underbrace{\beta(1-g_t)\, \nabla_\theta \log \pi_\theta(y^T_t \mid s_t)}_{\textit{focal reinforcement on } y^T_t}.
\end{aligned} \tag{11}$$

Proof. By definition,

$$\mathcal{L}^{(t)}_{\text{TOLD}} = -\log \pi_\theta(y^T_t \mid s_t) - \log \sigma\big(\beta(\delta^w_t - \delta^l_t)\big).$$

The NLL term contributes $-\nabla_\theta \log \pi_\theta(y^T_t \mid s_t)$. For the BT term, note $\frac{d}{du}\big[-\log \sigma(u)\big] = -(1-\sigma(u))$. Applying the chain rule:

$$\begin{aligned}
&-\nabla_\theta \log \sigma\big(\beta(\delta^w_t - \delta^l_t)\big) \\
&\quad = (1-g_t)\, \beta\, \big(\nabla_\theta \delta^l_t - \nabla_\theta \delta^w_t\big).
\end{aligned}$$

Since π_{ref} is frozen, $\nabla_\theta \delta^w_t = \nabla_\theta \log \pi_\theta(y^T_t \mid s_t)$ and $\nabla_\theta \delta^l_t = \nabla_\theta \log \pi_\theta(\tilde{a}_t \mid s_t)$. Rearranging yields (11). □

Interpretation. Three forces act at every position:

1. The NLL term unconditionally raises $\log \pi_\theta(y^T_t \mid s_t)$.
2. The focal suppression lowers $\log \pi_\theta(\tilde{a}_t \mid s_t)$ with weight $\beta(1-g_t)$, which *vanishes* when the teacher token is already strongly preferred ($g_t \to 1$) and is *maximal* at hard positions ($g_t \approx 1/2$)—focal-loss-style focusing (Lin et al., 2017).

3. The focal reinforcement *additionally* raises $\log \pi_\theta(y_t^T \mid s_t)$ with the same adaptive weight.

Corollary 2 (Anti-squeezing)**.** *Under the regularity condition that score functions of the teacher token and the sampled negative are non-positively correlated,*

$$\left\langle \nabla_\theta \log \pi_\theta(y_t^T \,|\, s_t),\ \nabla_\theta \log \pi_\theta(\tilde{a}_t \,|\, s_t) \right\rangle \leq 0, \quad (12)$$

the change in teacher-token log-probability per gradient step η satisfies

$$\begin{aligned} &\Delta \log \pi_\theta(y_t^T \mid s_t) \\ &\geq\ \eta\big(1+\beta(1-g_t)\big)\|\nabla_\theta \log \pi_\theta(y_t^T \,|\, s_t)\|^2 \\ &\quad - O(\eta^2), \end{aligned} \quad (13)$$

which is strictly positive for any non-degenerate gradient.

Proof. Consider a gradient step $\theta' = \theta - \eta\, \nabla_\theta \mathcal{L}_{\text{TOLD}}^{(t)}$. By first-order Taylor expansion:

$$\begin{aligned} &\Delta \log \pi_\theta(y_t^T \mid s_t) \\ &= -\eta \left\langle \nabla_\theta \log \pi_\theta(y_t^T \mid s_t),\ \nabla_\theta \mathcal{L}_{\text{TOLD}}^{(t)} \right\rangle + O(\eta^2). \end{aligned}$$

Substituting (11) and using $\left\langle \nabla_\theta \log \pi_\theta(y_t^T \mid s_t),\ -\nabla_\theta \log \pi_\theta(y_t^T \mid s_t) \right\rangle = -\|\nabla_\theta \log \pi_\theta(y_t^T \mid s_t)\|^2$ and the regularity condition (12):

$$\begin{aligned} &\Delta \log \pi_\theta(y_t^T \mid s_t) \\ &\geq \eta\big(1 + \beta(1-g_t)\big)\|\nabla_\theta \log \pi_\theta(y_t^T \mid s_t)\|^2 \\ &\quad - O(\eta^2). \end{aligned}$$

Since $0 < g_t < 1$ and $\beta > 0$, the leading coefficient exceeds unity, guaranteeing $\Delta > 0$ for small η. □

Discussion. Condition (12) holds exactly when y_t^T and $\tilde{a}_t$ partition the softmax mass (increasing one token's probability must decrease another's), and approximately in over-parameterized regimes where per-token score vectors are near-orthogonal. Corollary 2 formally confirms the empirical finding in Figure 2(a): $\log \pi_\theta(y_t^T \mid s_t)$ increases monotonically under TOLD, ruling out the likelihood-displacement regime where both δ_t^w and δ_t^l jointly drift downward.