DECOUPLED ON-POLICY DISTILLATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

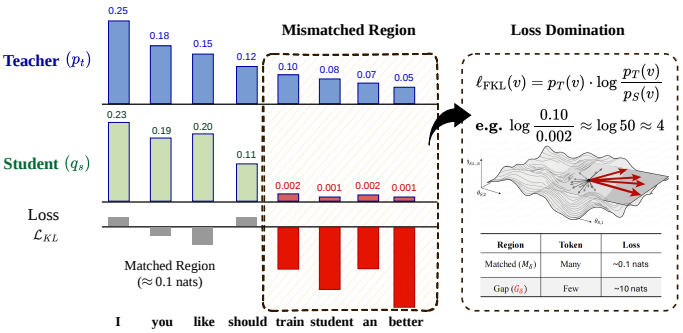
ABSTRACT

On-policy distillation (OPD) commonly relies on forward or reverse KL divergence (FKL/RKL) to align the student with the teacher. However, FKL has a particular limitation in OPD: its per-token loss and gradient concentrate on a small set of candidate tokens on which the teacher probability p_T far exceeds the student probability p_S. We call this high-disparity set of candidates the *Mismatched Region*, where p_T exceeds p_S by one to two orders of magnitude, and this concentration makes OPD prone to response-length inflation, unreasonable truncation, and accuracy collapse. RKL, in contrast, avoids this domination through its p_S-self-weighting but can under-cover teacher modes through its mode-seeking behaviour. To address these issues, we propose a simple but effective optimisation objective, **Decoupled-KL Distillation (DKLD)**, a loss that *decomposes* the KL into two contributors, the **Matched Region** and the **Mismatched Region**. In detail, we simply use forward KL in the Matched Region, and Reverse KL with p_S-self-weighting takes responsibility within the Mismatched Region. Furthermore, the student rollouts required by OPD simultaneously provide the group-relative advantages for GRPO, but their gradients can conflict. To reduce this interference, we propose an advantage-weighted distillation module that balances these two objectives and combines OPD with GRPO in a single framework. Through comprehensive experiments, across mathematical reasoning and code generation tasks, DKLD reaches state-of-the-art performance in all configurations we ran and drastically mitigates the length-inflation collapse.

Anonymous Code: `github.com/anon4rev/DKLD`

Dataset: `huggingface.co/datasets/anonymous2paper/CompMath-700`

Weight: `huggingface.co/anonymous2paper/DKLD_0.6B`



Method	Step 80 pass@1 ↑	Step 80 trunc. ↓
GRPO	76.0 ±1.5	19.8
FKL	13.7 ±0.6	91.2
RKL	74.8 ±1.3	19.7
AKL	76.6 ±1.2	21.1
ToDi	76.6 ±0.7	18.2
DKLD (ours)	**81.5** ±1.5	**14.6**

Figure 1: **Left:** FKL loss and bounded logit gradients concentrate on a few Mismatched-Region tokens ($p_T \gg p_S$, red; Section 3.1). **Right:** A table presents a clear collapse about FKL at step 80. FKL collapses to 13.7% pass@1 with 91.2% truncation, and DKLD achieves 81.5% with 14.6% truncation. Results are on MATH-500 (Qwen3-8B to Qwen3-1.7B); both columns are percentages, and pass@1 reports mean ± sample SD over seeds 10, 20, and 30 (Table 7). KD baselines use the uniform weight $w = 1$; DKLD uses the AWD gate (Table 8 separates the two).

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

1 INTRODUCTION

On-policy distillation (OPD) has become a gold standard in transferring large model reasoning ability to a small student (Agarwal et al., 2024; Hübotter et al., 2026; Ye et al., 2026). A key limitation of offline distillation (Gou et al., 2021; Kim et al., 2024; Li & Jin, 2022) is that the student conditions on fixed dataset prefixes during training but on its own prefixes at inference (Wei et al., 2022; Li et al., 2025). OPD overcomes this mismatch by having the student generate its own chains of thought while a frozen teacher provides token-level supervision along the generated trajectory. This token-level supervision is typically implemented by matching the teacher and student distributions at each generation step. Forward KL (FKL) is a common choice for this objective (Hinton et al., 2015; Zhao et al., 2022; Zhang et al., 2023). However, we find that its loss can become concentrated on a few candidate tokens with high teacher probability but low student probability. In our experiments, this concentration is associated with longer responses, more truncation, and a sharp drop in accuracy (Figure 1). Reverse KL (RKL) reduces this concentration by weighting each term with the student probability, but may instead under-cover teacher modes (Yao et al., 2025; Wu et al., 2025).

Several methods combine FKL and RKL to address these limitations. AKL (Wu et al., 2025) sets its weights based on teacher–student differences in the head and tail of the vocabulary. ToDi (Jung et al., 2025) uses a sigmoid weight based on the teacher–student probability ratio. Other work uses teacher entropy to choose the weights (Jin et al., 2026) or combines on-policy and off-policy training (Zhu et al., 2026). We take a different approach by separating two questions: how much total probability the student should assign to the mismatched candidates, and how it should distribute that probability among them. Decoupled KD (Zhao et al., 2022) and loss-level KL decomposition (Cui et al., 2024) use related decompositions for classification. We apply this idea to on-policy autoregressive distillation to reduce the concentration of loss on a few candidates.

We propose **Decoupled-KL Distillation (DKLD)**. At each generation step, we divide the vocabulary into two regions. Candidates with high teacher probabilities and low student probabilities form the *Mismatched Region* G; all other candidates form the *Matched Region* M (Definition 1). FKL splits exactly into three terms: a binary KL that compares the total probabilities of the two regions, a weighted KL within M, and a weighted KL within G (Proposition 2). DKLD keeps the first two terms and replaces the third with a reverse KL weighted by the student's probability mass in G. The binary term still guides how much total probability the student assigns to G. Within G, the new term gives less weight to candidates with very low student probabilities. This reduces their loss contributions while preserving supervision on the region's total probability.

We also combine DKLD with GRPO (Shao et al., 2024), using the same student rollouts for both objectives. The distillation and policy-gradient updates can push the probability of a sampled token in opposite directions. To reduce this conflict, we adjust the distillation weight for each rollout. We apply distillation only to correct rollouts and scale their distillation loss by $\sigma(-\hat{A}/\tau)$, where σ is the sigmoid function, $\hat{A}$ is the group-relative advantage, and $\tau > 0$ is a temperature. A correct rollout with a high advantage receives less distillation because it already receives a strong policy-gradient signal. A correct rollout with an advantage near zero receives more distillation.

Contributions.

- We show that a few mismatched candidates carry disproportionately large FKL loss terms and dominate the logit gradient in OPD. We connect this concentration to unstable training and show how changing the objective within G reduces it (Section 3.1).
- We propose DKLD, which keeps FKL supervision on region probabilities and within M, but uses RKL within G. We also introduce an advantage-based weight for combining distillation with GRPO (Sections 3.2.1–3.3).
- DKLD outperforms GRPO, FKL, RKL, AKL, and ToDi on five math benchmarks and LiveCodeBench v6. Our experiments distill Qwen3-8B into Qwen3-0.6B/1.7B and Qwen3-14B into Qwen3-4B (Section 4). We also release **CompMath-700** to reduce the risk of benchmark contamination. This held-out set contains 700 competition problems translated from several languages and checked by two authors (Appendix D).

2 PRELIMINARIES

2.1 ON-POLICY KNOWLEDGE DISTILLATION

Standard knowledge distillation (Hinton et al., 2015; Cho & Hariharan, 2019; Sun et al., 2024) trains a student model $p_S(\cdot;\theta)$ to mimic a frozen teacher p_T on a fixed dataset D. In autoregressive language models (Recchia, 2021; Schuurmans et al., 2024), the student observes the *dataset's* token prefixes during training, but must condition on its *own* previously generated tokens at inference time. This train-inference distribution mismatch causes compounding errors in long-horizon generation tasks such as chain-of-thought reasoning (Wei et al., 2022).

On-policy distillation (Agarwal et al., 2024; Song & Zheng, 2026; Fu et al., 2026) resolves this mismatch by generating training sequences from the student's current policy. At each training step, the student samples K responses per prompt $\{y^{(1)}, \ldots, y^{(K)}\} \sim p_S(\cdot \mid x;\theta)$, and a frozen teacher provides token-level supervision on these student-generated sequences. In our experiments, the teacher also receives an answer-aware system prompt $h_T(x)$ containing the correct answer; all teacher-based methods use the same prompt (Appendix C). The on-policy distillation objective is

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim D}\,\mathbb{E}_{y\sim p_S(\cdot|x;\theta)}\left[\sum_{t=1}^{|y|} \mathrm{KL}(p_T(\cdot \mid h_T(x), x, y_{<t}) \;\|\; p_S(\cdot \mid x, y_{<t};\theta))\right] \tag{1}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes a Kullback–Leibler divergence between the teacher and student distributions at each token position; we instantiate it as the forward or reverse KL in Section 2.2. Because the training distribution now matches the inference distribution, on-policy distillation substantially reduces compounding errors.

2.2 FORWARD AND REVERSE KL DIVERGENCE

The choice of KL direction in Eq. (1) is critical. At each token position t, we define the teacher and student distributions over the vocabulary V as $p_T(\cdot) \triangleq p_T(\cdot \mid h_T(x), x, y_{<t})$ and $p_S(\cdot) \triangleq p_S(\cdot \mid x, y_{<t};\theta)$, respectively. We omit the conditioning for notational brevity.

Forward KL (FKL) (Jerfel et al., 2021) minimises the standard $\mathrm{KL}(p_T\|p_S)$ used in knowledge distillation,

$$\mathcal{L}_{\text{FKL}} = \mathrm{KL}(p_T\|p_S) = \sum_{v\in V} p_T(v)\log\frac{p_T(v)}{p_S(v)} \tag{2}$$

FKL is a *mode-covering* (or *mean-seeking*) objective, which heavily penalises any token v where the teacher assigns non-negligible probability $p_T(v) > 0$ but the student assigns low probability $p_S(v) \approx 0$. This forces the student to spread its probability mass to cover *all* modes of the teacher distribution.

Reverse KL (RKL) (Wu et al., 2025; Wang et al., 2024) minimises $\mathrm{KL}(p_S\|p_T)$,

$$\mathcal{L}_{\text{RKL}} = \mathrm{KL}(p_S\|p_T) = \sum_{v\in V} p_S(v)\log\frac{p_S(v)}{p_T(v)} \tag{3}$$

RKL is a *mode-seeking* (or *zero-forcing*) objective, which penalises tokens where the *student* assigns high probability but the teacher does not. This encourages the student to concentrate on a subset of teacher modes rather than covering all of them.

2.3 GROUP RELATIVE POLICY OPTIMIZATION (GRPO)

GRPO (Shao et al., 2024) is a reinforcement learning algorithm for language models that eliminates the need for a learned value function. For each prompt x, GRPO samples a group of K responses $\{y^{(1)}, \ldots, y^{(K)}\}$ from the current policy and computes per-response rewards $\{r^{(1)}, \ldots, r^{(K)}\}$. The advantage of each response is estimated by normalising rewards within the group, $\hat{A}^{(i)} = \big(r^{(i)} - \text{mean}(\{r^{(j)}\}_{j=1}^{K})\big)/\big(\text{std}(\{r^{(j)}\}_{j=1}^{K}) + \epsilon\big)$. The policy loss is a clipped surrogate objective with a KL

penalty against the reference policy p_{ref},

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\Big(\min\Big(\rho_t\hat{A}^{(i)},\ \text{clip}_\varepsilon(\rho_t)\hat{A}^{(i)}\Big) - \beta\,\text{KL}(p_S(\cdot \mid x, y_{<t}) \,\|\, p_{\text{ref}}(\cdot \mid x, y_{<t}))\Big)\right] \tag{4}$$

where $\rho_t = p_S(y_t \mid x, y_{<t}; \theta)/p_{\text{old}}(y_t \mid x, y_{<t})$ is the importance ratio against the behaviour policy p_{old} that generated the current rollouts, p_{ref} is the static reference policy used only in the KL penalty, and $\text{clip}_\varepsilon(x) := \text{clip}(x, 1-\varepsilon, 1+\varepsilon)$. GRPO provides a reward-based learning signal that complements the distributional signal from knowledge distillation.

3 METHOD

3.1 WHY FKL DESTABILISES ON-POLICY TRAINING

In on-policy distillation (Agarwal et al., 2024), FKL fails because a few candidates carry disproportionately large loss terms and pin the gradient at a token position. We make this precise by writing both quantities per candidate.

Proposition 1 (Per-token FKL loss and logit gradient)**.** *The forward KL has the per-token decomposition*

$$\mathcal{L}_{\text{FKL}} = \sum_{v\in V}\ell_{\text{FKL}}(v), \qquad \ell_{\text{FKL}}(v) \triangleq p_T(v)\,\log\frac{p_T(v)}{p_S(v)}, \tag{5}$$

and the logit gradient at vocabulary entry v is

$$\frac{\partial\mathcal{L}_{\text{FKL}}}{\partial z_v} = p_S(v) - p_T(v). \tag{6}$$

Each logit-gradient component is bounded by $|p_S(v) - p_T(v)| \le 1$ (proof in Appendix A.1).

Loss concentration. The contribution of a candidate, $\ell_{\text{FKL}}(v) = p_T(v)\log\big(p_T(v)/p_S(v)\big)$, is large exactly when $p_T(v)$ is high and $p_S(v)$ is low. For one such Mismatched-Region candidate in our measurement it is 2.3 nats; for a Matched-Region candidate it is 0.011 (Table 1). Summed over the top-128 teacher candidates used in this analysis (training transmits top-256; Table 6), FKL measures 0.616 nats per generated token at initialisation and reaches about 10 nats by step 40.

Gradient pinning. On the same candidates $p_S(v) \approx 0$, so Eq. (6) gives $\partial\mathcal{L}_{\text{FKL}}/\partial z_v \approx -p_T(v)$. Each component is bounded by one, but the update is pinned to the same few teacher-preferred candidates at every step, and the student is pushed towards tokens it almost never samples. The parameter gradient, $\nabla_\theta\mathcal{L}_{\text{FKL}} = \sum_v\big(p_S(v) - p_T(v)\big)\nabla_\theta z_v$ by the softmax chain rule, inherits this allocation.

What DKLD changes. Section 3.2.2 splits FKL into a binary term over the region masses r_G^T and r_G^S (Eq. (8)) and one term inside each region. The FKL gradient on G splits the same way: an aggregate part from the binary term, whose components over G sum to $r_G^S - r_G^T$, and a within-region part $r_G^T\big(\tilde{q}_G(v) - \tilde{p}_G(v)\big)$ that sums to zero over G and pulls the student's distribution inside G towards the teacher's. DKLD keeps the aggregate part and replaces only the within-region part by the gradient of the reverse KL, which scales with $p_S(v)$ on every candidate (Appendix A.2). The total push towards G is unchanged, but it is now spread by the student's own distribution $\tilde{q}_G(v)$ instead of $p_T(v)$, and the student is no longer asked to copy the teacher inside G. In the loss, the within-region contribution on G becomes $p_S(v)\log\big[\tilde{q}_G(v)/\tilde{p}_G(v)\big]$, which for the candidate above is 0.046 nats instead of 2.3, a $50\times$ reduction; the Matched-Region contribution stays at 0.011 (Table 1). Per generated token, DKLD moves from 0.032 nats at initialisation to about 0.1 by step 40, where FKL reaches about 10. Checkpoints trained with DKLD also show smaller logit-gradient norms than those trained with FKL (Figure 3).

These measurements link the candidate-level concentration to the training curves in Table 5, where FKL collapses and DKLD does not. They also settle the design: RKL's self-weighting is what we want inside G, and its mode-seeking behaviour is why we keep FKL on the Matched Region (Gu et al., 2024; Ko et al., 2024).

Table 1: **Loss concentration on the Mismatched Region.** Top: within-region loss contribution of one candidate (base Qwen3-1.7B student, Qwen3-8B teacher); ratios are FKL/DKLD. Bottom: share of the analysed top-128 candidate slots that fall in G and share of the forward KL they carry, aggregated over 12,288 on-policy positions of the base Qwen3-0.6B student with the Qwen3-8B teacher ($\epsilon_S = 0.01$). DKLD keeps forward KL on the Matched Region M and, through the binary term, on the aggregate masses of the two regions; only the within-region objective on G is switched to p_S-weighted reverse KL (Eq. (10)), so the binary term continues to supervise how much mass the student places on G.

Scope	Quantity	FKL	DKLD	Ratio
Absolute within-region contribution (nats)				
One candidate v	Mismatched Region G	2.3	**0.046**	50×
One candidate v	Matched Region M	0.011	0.011	1×
Share over all analysed positions (%)				
Region G	Candidate slots occupied	0.079	–	–
Region G	Loss carried	40.0	–	–

3.2 Decoupled-KL Distillation

The analysis above locates the problem inside G, so **Decoupled-KL Distillation (DKLD)** changes the objective only there (Figure 1). Which KL direction is better depends on the local teacher–student relationship at each token, as token-adaptive schemes such as AKL (Wu et al., 2025) and ToDi (Jung et al., 2025) also assume, and the decoupling principle of DKD (Zhao et al., 2022) shows how to isolate one part of a divergence. We first define G (Section 3.2.1) and then split FKL exactly along it (Section 3.2.2).

3.2.1 The Mismatched Region

When distilling from a large teacher to a much smaller student (Hsieh et al., 2023; Stanton et al., 2021), p_T typically has broader support than p_S at each token position.

Definition 1 (Mismatched Region)**.** *Given a teacher distribution $p_T(v)$ and a student distribution $p_S(v)$ over vocabulary V at one token position, and thresholds $\epsilon_T, \epsilon_S > 0$, we define the **Mismatched Region** as*

$$G = \{v \in V \mid p_T(v) > \epsilon_T,\ \ p_S(v) < \epsilon_S\}\,, \tag{7}$$

*where $\epsilon_T = \kappa \max_{u \in V} p_T(u)$ is relative to the teacher's peak probability at that position and $\epsilon_S(s) = \epsilon_0 + (\epsilon_1 - \epsilon_0)\, s/s_{\max}$ is annealed linearly with the training-step index s over the $s_{\max}$ steps of the full run; our default is $(\kappa, \epsilon_0, \epsilon_1) = (0.1, 0.01, 0.001)$. The region selects candidates whose teacher probability exceeds a fraction of the teacher peak and whose student probability falls below the scheduled threshold. We call the complementary set $M = V \setminus G$ the **Matched Region**.*

The rule implies both $p_T(v) - p_S(v) > \epsilon_T - \epsilon_S$ and $p_T(v)/p_S(v) > \epsilon_T/\epsilon_S$, so it combines a probability-gap criterion with a probability-ratio criterion.

3.2.2 Decoupled Decomposition of Forward KL and the DKLD Loss

We now split FKL into a binary KL between the total masses of M and G plus one KL inside each region.

For each region $A \in \{M, G\}$, define the aggregate masses and, when A is nonempty, the within-region distributions.

$$r_A^T = \sum_{v \in A} p_T(v), \quad r_A^S = \sum_{v \in A} p_S(v), \quad \tilde{p}_A(v) = \frac{p_T(v)}{r_A^T}, \quad \tilde{q}_A(v) = \frac{p_S(v)}{r_A^S}\ \ (v \in A), \tag{8}$$

so $r_M^T + r_G^T = r_M^S + r_G^S = 1$ and $p_T(v) = r_A^T \tilde{p}_A(v)$, $p_S(v) = r_A^S \tilde{q}_A(v)$ for $v \in A$. For an empty region, its weighted KL term is defined as zero and its within-region normalisation is skipped; the binary term uses the convention $0 \log(0/0) = 0$.

Proposition 2 (Decoupled decomposition of FKL)**.** *The forward KL at each token position satisfies the exact identity*

$$\mathrm{KL}(p_T\|p_S) = \underbrace{r_M^T \log \frac{r_M^T}{r_M^S} + r_G^T \log \frac{r_G^T}{r_G^S}}_{\mathcal{L}_{Binary}} + \underbrace{r_M^T \cdot \mathrm{KL}(\tilde{p}_M\|\tilde{q}_M)}_{\mathcal{L}_{Matched}} + \underbrace{r_G^T \cdot \mathrm{KL}(\tilde{p}_G\|\tilde{q}_G)}_{\mathcal{L}_{Mismatched}} \tag{9}$$

The proof splits the sum over $V = M \cup G$ and substitutes Eq. (8) in each part; it is deferred to Appendix A.3.

We *keep FKL on $\mathcal{L}_{Binary}$ and $\mathcal{L}_{Matched}$, and switch to RKL on $\mathcal{L}_{Mismatched}$*, giving the per-token DKLD loss

$$\mathcal{L}_{\text{DKLD}} = \mathcal{L}_{\text{Binary}} + r_M^T \cdot \mathrm{KL}(\tilde{p}_M\|\tilde{q}_M) + r_G^S \cdot \mathrm{KL}(\tilde{q}_G\|\tilde{p}_G), \tag{10}$$

whose third term carries RKL's p_S-self-weighting on G through the student mass r_G^S. Because of this swap, $\mathcal{L}_{\text{DKLD}}$ no longer equals $\mathrm{KL}(p_T\|p_S)$; it is a surrogate loss in which every term stays non-negative (Appendix A.4).

3.3 Combining OPD with GRPO

GRPO (Shao et al., 2024; Guo et al., 2025; Lambert et al., 2025) and OPD share the same student rollouts but disagree on which tokens to push. We reconcile them with two rollout-level gates.

KD only on correct rollouts. We compute KD only on correct rollouts, using the verifier mask $m^{(i)} = \mathbb{1}\big[r^{(i)} > 0\big]$.

Advantage-weighted distillation. Among correct rollouts, we scale KD by

$$w_{\text{awd}}(\hat{A}^{(i)}) = \sigma\Big(-\hat{A}^{(i)}/\tau\Big). \tag{11}$$

where $\tau > 0$ is a temperature (we use $\tau = 1$ in all experiments without tuning it; see Table 6). A correct rollout with a high group-relative advantage already receives a strong policy-gradient signal, so the gate lowers its KD weight. As the advantage approaches zero, the weight approaches 0.5 and teacher supervision takes over. The gate thus reduces KD exactly where it can oppose the policy update (Appendix A.5).

Combined objective.

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda_{\text{KD}} \cdot \frac{1}{K}\sum_{i=1}^{K} m^{(i)}\, w_{\text{awd}}(\hat{A}^{(i)}) \cdot \frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|} \mathcal{L}_{\text{DKLD}}^{(t)}\Big(y^{(i)}\Big) \tag{12}$$

where $m^{(i)}$ selects correct rollouts and w_{awd} adjusts their KD strength according to the group-relative advantage.

4 Experiments

4.1 Setup and Benchmarks

We evaluate three instruction-tuned students, Qwen3-0.6B/1.7B/4B (Yang et al., 2025), with Qwen3-8B as teacher for the 0.6B/1.7B and Qwen3-14B for the 4B student. The math training set consists of 2,531 OpenThoughts (Guha et al., 2025) problems (middle-school to competition, thinking mode), excluding Hard_mini150 and the validation set. Validation questions are selected using the same semantic-uncertainty procedure as Hard_mini150. For code, we train and evaluate on LiveCodeBench v6 using a split at the *test-case* level within each problem. A random subset of the official judge tests is reserved for evaluation, and the remaining tests provide the training reward. Training uses the combined DKLD objective (Eq. 12) on 4×H100 GPUs (Appendix G) with a binary $0/1$ correctness reward. We train each configuration with seeds 10, 20, and 30 and report the mean and standard deviation (SD). Full hyperparameters are in Appendix C. We compare DKLD against GRPO (Shao et al., 2024, no KD), FKL (Hinton et al., 2015, $\mathrm{KL}(p_T\|p_S)$ on correct rollouts), RKL (Gu et al., 2024), AKL (Wu et al., 2025, adaptive FKL/RKL weighting), and ToDi (Jung et al., 2025, per-token

Table 2: **Mathematical reasoning results (%; pass@1, with avg@8 for AIME 2025).** Teachers are Qwen3-8B for the Qwen3-0.6B and Qwen3-1.7B students, and Qwen3-14B for the Qwen3-4B student. Best student results are **bolded**. Checkpoints are selected on validation and scores for trained methods are mean ± SD over three independent training seeds (10, 20, 30).

Method	GSM8K	MATH-500	Hard_mini150	CompMath-700	AIME 2025 (avg@8)	Avg.
Teacher						
Qwen3-8B *(for 0.6B / 1.7B students)*	95.6	85.0	74.0	58.1	42.1	71.0
Student: Qwen3-0.6B						
GRPO (Shao et al., 2024)	74.5 ±0.5	63.5 ±0.8	32.9 ±1.4	40.0 ±2.1	13.5 ±1.0	44.9 ±0.8
FKL (Hinton et al., 2015)	73.0 ±0.8	63.3 ±0.6	32.7 ±2.0	42.5 ±1.1	14.4 ±1.0	45.2 ±1.0
RKL (Gu et al., 2024)	74.1 ±0.9	63.7 ±1.7	35.6 ±1.9	43.0 ±0.9	14.7 ±1.2	46.2 ±1.3
AKL (Wu et al., 2025)	75.0 ±0.7	66.1 ±1.6	36.2 ±2.5	43.2 ±1.4	14.4 ±1.3	47.0 ±1.5
ToDi (Jung et al., 2025)	75.8 ±0.7	67.6 ±1.7	39.1 ±2.1	44.0 ±0.8	14.4 ±1.3	48.2 ±0.9
DKLD (Ours)	**77.9** ±1.3	**70.5** ±1.1	**43.3** ±2.4	**44.9** ±0.6	**16.5** ±0.9	**50.6** ±0.5
Student: Qwen3-1.7B						
GRPO (Shao et al., 2024)	89.6 ±0.3	78.2 ±0.9	66.4 ±3.9	52.0 ±1.6	29.2 ±1.4	63.1 ±1.3
FKL (Hinton et al., 2015)	89.0 ±0.5	81.8 ±1.6	61.8 ±2.1	53.5 ±1.2	26.9 ±1.7	62.6 ±0.7
RKL (Gu et al., 2024)	89.2 ±0.5	82.5 ±0.7	64.0 ±4.1	54.9 ±0.6	29.7 ±0.9	64.1 ±1.2
AKL (Wu et al., 2025)	89.3 ±0.5	81.7 ±1.4	65.8 ±5.0	54.6 ±0.5	29.7 ±1.0	64.2 ±1.6
ToDi (Jung et al., 2025)	89.6 ±0.7	81.3 ±1.1	67.6 ±4.2	54.9 ±0.7	29.7 ±0.9	64.6 ±0.9
DKLD (Ours)	**90.3** ±0.5	**83.1** ±0.8	**73.1** ±3.7	**56.3** ±0.7	**32.5** ±0.8	**67.1** ±0.9
Teacher						
Qwen3-14B *(for 4B student)*	95.9	93.8	85.3	63.4	56.3	78.9
Student: Qwen3-4B						
GRPO (Shao et al., 2024)	91.3 ±0.5	88.1 ±1.9	79.8 ±2.8	61.2 ±0.7	47.2 ±2.8	73.5 ±1.4
FKL (Hinton et al., 2015)	89.6 ±0.5	84.1 ±1.8	74.9 ±1.9	58.3 ±1.1	46.0 ±2.7	70.6 ±1.1
RKL (Gu et al., 2024)	91.6 ±0.7	87.8 ±1.6	80.9 ±4.4	60.9 ±1.5	50.3 ±2.3	74.3 ±1.8
AKL (Wu et al., 2025)	91.9 ±0.2	87.1 ±1.1	77.3 ±4.2	61.0 ±1.2	49.3 ±3.0	73.3 ±1.2
ToDi (Jung et al., 2025)	92.3 ±0.7	88.3 ±1.9	80.7 ±3.7	61.7 ±0.9	50.0 ±1.5	74.6 ±0.5
DKLD (Ours)	**94.9** ±0.3	**91.3** ±0.8	**84.7** ±3.1	**63.3** ±1.9	**52.4** ±1.5	**77.3** ±0.7

sigmoid mix). All KD baselines share λ_{KD} and top-k, use the same answer-aware teacher system prompt, and are combined with GRPO. All KD baselines apply the correct-rollout mask with a uniform weight $w = 1$; only DKLD uses the advantage-weighted gate of Section 3.3. Because correct rollouts have $\hat{A} \geq 0$ under a $0/1$ reward, the gate satisfies $w \leq 0.5$, so DKLD's effective KD coefficient is at most half of the baselines'.

We evaluate on five benchmarks of increasing difficulty: **GSM8K** (Cobbe et al., 2021) (1,319 test problems, grade-school math); **MATH-500** (Hendrycks et al., 2021) (500 competition-level problems); **Hard_mini150** (we use semantic entropy (Kuhn et al., 2023) to rank questions from the OpenThoughts pool by the uncertainty of five sampled responses per question and select the top 150, see Appendix C); **CompMath-700** (700 problems, a held-out evaluation set we curated from recent open math-competition problems in several languages to mitigate contamination from publicly released benchmarks); **AIME 2025** (30 problems from the American Invitational Mathematics Examination). For code generation, we evaluate on **LiveCodeBench v6** (Jain et al., 2025), using the held-out test-case split described in Section 4.1. Pass@4 is reported on the held-out tests only.

4.2 Main Results: Mathematical Reasoning

DKLD is best on every benchmark and every student size (Table 2). Over the strongest baseline, ToDi, it raises the average by $+2.4$ points on Qwen3-0.6B (50.6 vs. 48.2), $+2.5$ on Qwen3-1.7B (67.1 vs. 64.6), and $+2.7$ on Qwen3-4B (77.3 vs. 74.6). The gains are widest on Hard_mini150 ($+4.2$, $+5.5$, and $+4.0$ points at the three scales). With the uniform weight $w = 1$ of the baselines, DKLD still leads ToDi on Qwen3-1.7B: 68.9 vs. 67.6 on Hard_mini150 (Table 3) and 77.5 vs. 76.6 on MATH-500 at step 80 (Tables 7 and 8).

Training stability. Table 5 tracks MATH-500 during training for the 1.7B student. Between steps 35 and 50, FKL+GRPO falls from 44.6 to 20.2 pass@1 while its truncation rate rises from 40.2% to 80.6%; DKLD stays between 80.2 and 82.0 with truncation between 14.6% and 15.2% (evaluation details in Appendix E). This collapse is the training-time face of the loss concentration in Section 3.1.

For code, we distill Qwen3-8B into Qwen3-1.7B on LiveCodeBench v6 and plot pass@4 on the held-out test cases against the truncation rate (Figure 2). DKLD has the highest pass@4 and the lowest truncation rate of all plotted methods.

4.3 Ablation Studies

All ablations use Qwen3-1.7B: component variants on Hard_mini150 (Table 3) and the region thresholds on MATH-500 and AIME 2025 (Table 4).

Swapping the two KL directions, so that RKL acts on M and FKL on G, drops Hard_mini150 pass@1 from 73.1 to 48.9 under the same AWD gate, a loss of 24.2 points. FKL belongs on the Matched Region and RKL on the Mismatched Region.

Replacing AWD with a uniform weight $w = 1$ lowers pass@1 from 73.1 to 68.9, and removing the GRPO term as well, so that KD alone trains the student, gives 66.9. The gate is thus worth 4.2 points and the GRPO term a further 2.0. The gate also caps the effective KD weight at $w \leq 0.5$; Table 8 shows the same ordering at a fixed step. The KD-only variant lands within the seed spread of the GRPO-only run in Table 2 (66.9 ± 2.5 vs. 66.4 ± 3.9), so on this benchmark neither signal alone comes close to the combined objective.

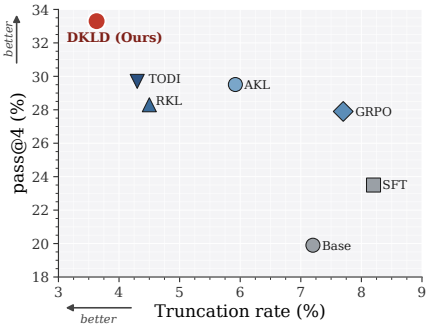


Figure 2: **Code generation on LiveCodeBench v6** (Qwen3-8B to Qwen3-1.7B). Pass@4 on held-out test cases versus truncation rate. DKLD has the highest pass@4 and the lowest truncation rate among the plotted methods.

The gate helps the baselines much less. It raises FKL from 61.8 to 62.7 and RKL from 64.0 to 65.1, about one point each, against 4.2 points for DKLD. Under the AWD gate, DKLD exceeds RKL by 8.0 points and FKL by 10.4; under the uniform gate the margins are 4.9 and 7.1. The gate is therefore not the source of the gap between DKLD and the fixed-direction losses.

The two within-region divergences act on the renormalised distributions $\tilde{p}_A$ and $\tilde{q}_A$ of Eq. (8), so they do not see the region masses; only $\mathcal{L}_{\text{Binary}}$ does. Dropping it lowers Hard_mini150 pass@1 from 73.1 to 60.7, below plain FKL (61.8). Without it, nothing in the objective asks the student to place mass on G, since the Mismatched term $r_G^S \, \mathrm{KL}(\tilde{q}_G \| \tilde{p}_G)$ itself vanishes as $r_G^S \to 0$. This term is what separates Eq. (10) from a per-token reweighting of FKL and RKL.

Table 3: Component ablation (Qwen3-1.7B, Hard_mini150 pass@1). All rows except "w/o GRPO" train with GRPO, and "uniform" sets $w(\hat{A}) = 1$ on every correct rollout. "w/o Binary term" drops $\mathcal{L}_{\text{Binary}}$ from Eq. (10), leaving a token-level FKL/RKL switch between the two regions. Scores are mean $\pm$ SD over seeds 10, 20, and 30.

Setting	**KD loss**	**AWD gate** $w(\hat{A})$	**Hard_mini150**
Full DKLD	DKLD	$\sigma(-\hat{A}/\tau)$	**73.1** $\pm$3.7
w/o AWD (uniform weight)	DKLD	uniform	68.9 $\pm$3.9
w/o GRPO (KD only)	DKLD	uniform	66.9 $\pm$2.5
w/o Binary term	DKLD $-\,\mathcal{L}_{\text{Binary}}$	$\sigma(-\hat{A}/\tau)$	60.7 $\pm$4.0
Plain FKL (uniform weight)	FKL	uniform	61.8 $\pm$2.1
Plain FKL (AWD gate)	FKL	$\sigma(-\hat{A}/\tau)$	62.7 $\pm$3.5
Plain RKL (uniform weight)	RKL	uniform	64.0 $\pm$4.1
Plain RKL (AWD gate)	RKL	$\sigma(-\hat{A}/\tau)$	65.1 $\pm$3.0
Swapped regions	swapped	$\sigma(-\hat{A}/\tau)$	48.9 $\pm$3.0

Sensitivity to the region thresholds. Table 4 probes the thresholds that define G in Eq. (7). Unlike the dynamic rule of Definition 1 used elsewhere, this diagnostic fixes both thresholds at absolute values, $\epsilon_S = 0.005$ and $\epsilon_T \in \{0.01, 0.05, 0.50\}$; the $\epsilon_T = 0.10$ row is the main-result configuration of Table 2, which uses the dynamic rule with $\kappa = 0.10$. Pure RKL and pure FKL (first and last rows) are the two endpoints of the sweep. The main configuration is best on both MATH-500 (83.1) and AIME 2025 (32.5). Over $\epsilon_T \in \{0.05, 0.10, 0.50\}$ the scores stay within 81.2–83.1 and 27.5–32.5, and only the lowest threshold $\epsilon_T = 0.01$ falls off, to 79.8 and 25.0.

Table 4: Threshold sensitivity (Qwen3-1.7B; fixed $\epsilon_S = 0.005$ except †). †: main-result configuration of Table 2 (dynamic rule, $\kappa = 0.10$); RKL and FKL rows from Table 2.

	MATH-500	AIME 2025
RKL (ref.)	82.5	29.7
$\epsilon_T = 0.01$	79.8	25.0
$\epsilon_T = 0.05$	81.2	30.4
$\epsilon_T = 0.10^{\dagger}$	**83.1**	**32.5**
$\epsilon_T = 0.50$	82.2	27.5
FKL (ref.)	81.8	26.9

Table 5: MATH-500 pass@1 and truncation rate during training (Qwen3-1.7B, Qwen3-8B teacher, shared 8,192-token generation limit), evaluated every 5 steps. FKL+GRPO collapses while DKLD stays stable.

	FKL + GRPO		DKLD (ours)	
Step	pass@1	trunc.	pass@1	trunc.
35	44.6	40.2%	80.2	14.8%
40	43.8	44.4%	81.4	15.2%
45	38.6	59.0%	82.0	14.6%
50	20.2	80.6%	81.6	14.8%

5 Related Work

Classical KD (Hinton et al., 2015; Wang & Yoon, 2022) matches student outputs to the teacher softmax and has been extended to autoregressive and encoder models by sequence-level KD (Kim & Rush, 2016; Cappellazzo et al., 2023; Mun'im et al., 2019; Sanh et al., 2019); these approaches all suffer from train-inference distribution mismatch. On-policy distillation (Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024; Xu et al., 2026) fixes this by training on student rollouts, but still applies one fixed divergence at every token; that is the limitation we address. An orthogonal line precomputes teacher log-probabilities offline (Wu et al., 2026), changing the recipe rather than the divergence.

Forward KL is mode-covering but spreads probability mass too broadly, while reverse KL is mode-seeking but ignores teacher modes outside student support. Token-adaptive variants interpolate the two using a head/tail split (AKL (Wu et al., 2025)), a per-token sigmoid on the probability ratio (ToDi (Jung et al., 2025)), teacher-entropy gating (Jin et al., 2026), hybrid on-/off-policy mixes (Zhu et al., 2026), or correctness-routed dual paths (Zheng et al., 2026); all rely on continuous mixing weights rather than a hard region partition. Decoupling-style losses (Zhao et al., 2022; Cui et al., 2024) restructure the same divergence but target classification, not on-policy autoregressive generation. We instead ask *why* FKL fails in the on-policy setting: a few Mismatched-Region candidates carry disproportionately large loss terms and pin the per-token gradient, even though each logit-gradient component stays bounded. This adds a gradient-level mechanism to token-level accounts of OPD success and failure (Li et al., 2026), and it motivates a hard partition with the KL *direction* switched per region rather than weighted.

GRPO (Shao et al., 2024) and critic-free variants such as RLOO (Ahmadian et al., 2024) replace the value function with group-relative advantages, but rely on sparse terminal rewards that give little guidance for intermediate tokens. Recent work combines OPD with reasoning-specific signals: answer-aware teacher prompting (Zhao et al., 2026) and length-compressing self-distillation (Sang et al., 2026) use dense teacher supervision to complement outcome rewards, while RL-aware KD (Zhang et al., 2026) and Generalised OPD (Yang et al., 2026) explicitly address interference between the RL and KD objectives by reweighting the global KD term. Our advantage-weighted gate $\sigma(-\hat{A}/\tau)$ instead targets this conflict per rollout, so that outcome verification shapes the global trajectory while DKLD adds dense per-token supervision exactly where standard OPD degrades.

6 Conclusion

We trace FKL's instability in on-policy KD to a concentration of loss and logit gradient on a few Mismatched-Region candidates, which drives length inflation and accuracy collapse. **DKLD** keeps FKL on the Matched Region and on the region masses, applies RKL inside the Mismatched Region, and uses an advantage-weighted gate to balance KD against GRPO. It gives the best student scores on all five math benchmarks (Table 2) and the highest pass@4 at the lowest truncation rate on LiveCodeBench (Figure 2). In the ablations, the binary term and the region assignment account for most of the gap to the fixed-direction losses (Section 4.3).

AI USE STATEMENT

In this work, we used generative AI tools for two purposes: to polish the wording of sentences in the manuscript; to translate the CompMath-700 problems from their source languages into English (Gemini 3.1 Pro), with every translation manually checked by two authors (Appendix D). We have not used generative AI tools for synthetic data generation, theoretical model development, mathematical formulation, proof assistance, hypothesis refinement, methodology design, implementation, qualitative analysis, or result interpretation. Open-weight models that serve as components of our pipeline (the Qwen3-4B semantic-equivalence judge in Appendix C) are part of the method and are described there. We have reviewed all AI-assisted edits and translations and verified that they preserve the intended technical meaning of the original text. We take responsibility for the final content of this work, including text, claims or artefacts produced with the aid of generative AI.

ETHICS STATEMENT

This work aims to make strong reasoning capabilities accessible in compact language models suitable for resource-constrained environments. By improving knowledge distillation efficiency, DKLD reduces the compute and energy cost of deploying capable reasoning models, which lowers the barrier for academic and small-organisation use of reasoning LLMs. The released distilled students inherit the capabilities of their public Qwen3 teachers and do not exceed them, so we do not introduce new dual-use risks beyond those already addressed by the Qwen3 release. CompMath-700 is collected from publicly available online math-competition sources and contains no personal data. We do not foresee specific negative societal impacts beyond those inherent to general language-model research.

REPRODUCIBILITY STATEMENT

All training configurations, evaluation scripts, and the distilled Qwen3-0.6B student checkpoint are released on GitHub and Hugging Face (links at the end of the abstract), along with the semantic-entropy annotations for Hard_mini150 and the full CompMath-700 evaluation set with its grading harness. The release includes a README documenting environment setup, training commands, and evaluation reproduction. The algorithm (Eqs. (9)–(12)), public datasets, and full hyperparameters (Table 6) are reported in the paper; the complete training loop is given in Algorithm 1, the distributed training architecture, teacher prompting, and evaluation protocol in Appendix C, the construction of CompMath-700 in Appendix D, and the per-seed training cost in Appendix G. Propositions 1 and 2 are stated with their assumptions in Definition 1 and proved, together with the auxiliary derivations, in Appendix A.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Vol. 1)*, pp. 12248–12267, 2024. URL https://aclanthology.org/2024.acl-long.662.pdf.

Umberto Cappellazzo, Muqiao Yang, Daniele Falavigna, and Alessio Brutti. Sequence-Level Knowledge Distillation for Class-Incremental End-to-End Spoken Language Understanding. In *Proc. INTERSPEECH 2023*, pp. 2953–2957, 2023. doi: 10.21437/Interspeech.2023-242. URL http://arxiv.org/abs/2305.13899.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4794–4802, 2019. URL http://openaccess.thecvf.com/content_ICCV_2019/html/Cho_On_the_Efficacy_of_Knowledge_Distillation_ICCV_2019_paper.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, November 2021. doi: 10.48550/arXiv.2110.14168. URL http://arxiv.org/abs/2110.14168.

Jiequan Cui, Zhuotao Tian, Zhisheng Zhong, Xiaojuan Qi, Bei Yu, and Hanwang Zhang. Decoupled kullback-leibler divergence loss. *Advances in Neural Information Processing Systems*, 37:74461–74486, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/87ee1bbac4635e7c948f3eea83c1f262-Abstract-Conference.html.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting On-Policy Distillation: Empirical Failure Modes and Simple Fixes. *arXiv preprint arXiv:2603.25562*, April 2026. doi: 10.48550/arXiv.2603.25562. URL http://arxiv.org/abs/2603.25562.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129(6):1789–1819, June 2021. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-021-01453-z. URL https://link.springer.com/10.1007/s11263-021-01453-z.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?noteId=5h0qf7IBZZ.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. OpenThoughts: Data Recipes for Reasoning Models. *arXiv preprint arXiv:2506.04178*, June 2025. doi: 10.48550/arXiv.2506.04178. URL http://arxiv.org/abs/2506.04178.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, August 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023. URL https://aclanthology.org/2023.findings-acl.507/.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement Learning via Self-Distillation. *arXiv preprint arXiv:2601.20802*, February 2026. doi: 10.48550/arXiv.2601.20802. URL http://arxiv.org/abs/2601.20802.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A. Heller, Yian Ma, and Michael I. Jordan. Variational refinement for importance sampling using the forward Kullback-Leibler divergence. In *Uncertainty in Artificial Intelligence*, volume 161, pp. 1819–1829. PMLR, 2021. URL https://proceedings.mlr.press/v161/jerfel21a.html.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. ToDi: Token-wise Distillation via Fine-Grained Divergence Control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8078–8091, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.409. URL https://aclanthology.org/2025.emnlp-main.409.

Hojin Kim, Seunghun Lee, Hyeon Kang, and Sunghoon Im. Offline-to-online knowledge distillation for video instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 159–168, 2024. URL https://openaccess.thecvf.com/content/WACV2024/html/Kim_Offline-to-Online_Knowledge_Distillation_for_Video_Instance_Segmentation_WACV_2024_paper.html.

Yoon Kim and Alexander M. Rush. Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DISTILLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Jia Li, Ge Li, Yongmin Li, and Zhi Jin. Structured Chain-of-Thought Prompting for Code Generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–23, February 2025. ISSN 1049-331X, 1557-7392. doi: 10.1145/3690635. URL https://dl.acm.org/doi/10.1145/3690635.

Lujun Li and Zhe Jin. Shadow knowledge distillation: Bridging offline and online knowledge transfer. *Advances in Neural Information Processing Systems*, 35:635–649, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/040d3b6af368bf71f952c18da5713b48-Abstract-Conference.html.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. *arXiv preprint arXiv:2604.13016*, April 2026. doi: 10.48550/arXiv.2604.13016. URL http://arxiv.org/abs/2604.13016.

Raden Mu'az Mun'im, Nakamasa Inoue, and Koichi Shinoda. Sequence-level knowledge distillation for model compression of attention-based sequence-to-sequence speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6151–6155. IEEE, 2019. URL https://ieeexplore.ieee.org/abstract/document/8683171/?casa_token=D1cysG-eUvIAAAAA:NqQMJzaPhPGjw9-PHLRmJNs_X2Rz5JwlZuWRa5tfkF7tNOJ1JMf7b3R9-s4TH4ETuwUwQoT6xw.

Gabriel Recchia. Teaching Autoregressive Language Models Complex Tasks By Demonstration. *arXiv preprint arXiv:2109.02102*, December 2021. doi: 10.48550/arXiv.2109.02102. URL http://arxiv.org/abs/2109.02102.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. CRISP: Compressed Reasoning via Iterative Self-Policy Distillation. *arXiv preprint arXiv:2603.05433*, March 2026. doi: 10.48550/arXiv.2603.05433. URL http://arxiv.org/abs/2603.05433.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS 2019 EMC^2 Workshop*, 2019.

Dale Schuurmans, Hanjun Dai, and Francesco Zanini. Autoregressive Large Language Models are Computationally Universal. *arXiv preprint arXiv:2410.03170*, October 2024. doi: 10.48550/arXiv.2410.03170. URL http://arxiv.org/abs/2410.03170.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, April 2024. doi: 10.48550/arXiv.2402.03300. URL http://arxiv.org/abs/2402.03300.

Mingyang Song and Mao Zheng. A Survey of On-Policy Distillation for Large Language Models. *arXiv preprint arXiv:2604.00626*, April 2026. doi: 10.48550/arXiv.2604.00626. URL http://arxiv.org/abs/2604.00626.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew G. Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34: 6906–6919, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/376c6b9ff3bedbbea56751a84fffc10c-Abstract.html.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15731–15740, 2024. URL http://openaccess.thecvf.com/content/CVPR2024/html/Sun_Logit_Standardization_in_Knowledge_Distillation_CVPR_2024_paper.html.

Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao, and Bing Qin. RKLD: Reverse KL-Divergence-based Knowledge Distillation for Unlearning Personal Information in Large Language Models. *arXiv preprint arXiv:2406.01983*, June 2024. doi: 10.48550/arXiv.2406.01983. URL http://arxiv.org/abs/2406.01983.

Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3048–3068, 2022. URL https://ieeexplore.ieee.org/abstract/document/9340578/?casa_token=Bpv3hqVlatoAAAAA:NvTs8IIRfBGwJPGfYBEnSbi5l4i0wy_fczo8a8UNrBAS5m9JigVAfOTcXANkU6r2Ui-sqzXtQg.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.383/.

Yecheng Wu, Song Han, and Hai Cai. Lightning OPD: Efficient Post-Training for Large Reasoning Models with Offline On-Policy Distillation. *arXiv preprint arXiv:2604.13010*, April 2026. doi: 10.48550/arXiv.2604.13010. URL http://arxiv.org/abs/2604.13010.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084*, April 2026. doi: 10.48550/arXiv.2604.14084. URL http://arxiv.org/abs/2604.14084.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, May 2025. doi: 10.48550/arXiv.2505.09388. URL http://arxiv.org/abs/2505.09388.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond Teacher: Generalized On-Policy Distillation with Reward Extrapolation. *arXiv preprint arXiv:2602.12125*, February 2026. doi: 10.48550/arXiv.2602.12125. URL http://arxiv.org/abs/2602.12125.

Wei Yao, Wenkai Yang, Ziqiao Wang, Yankai Lin, and Yong Liu. Revisiting weak-to-strong generalization in theory and practice: Reverse KL vs. forward KL. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2860–2888, 2025. URL https://aclanthology.org/2025.findings-acl.148/.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-Policy Context Distillation for Language Models. *arXiv preprint arXiv:2602.12275*, March 2026. doi: 10.48550/arXiv.2602.12275. URL http://arxiv.org/abs/2602.12275.

Quanshi Zhang, Xu Cheng, Yilan Chen, and Zhefan Rao. Quantifying the knowledge in a DNN to explain knowledge distillation for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5099–5113, 2023. URL https://ieeexplore.ieee.org/abstract/document/9864081/.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware Knowledge Distillation for LLM Reasoning. *arXiv preprint arXiv:2602.22495*, April 2026. doi: 10.48550/arXiv.2602.22495. URL http://arxiv.org/abs/2602.22495.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled Knowledge Distillation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11943–11952, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.01165. URL https://ieeexplore.ieee.org/document/9879819/.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734*, January 2026. doi: 10.48550/arXiv.2601.18734. URL http://arxiv.org/abs/2601.18734.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. SCOPE: Signal-Calibrated On-Policy Distillation Enhancement with Dual-Path Adaptive Weighting. *arXiv preprint arXiv:2604.10688*, April 2026. doi: 10.48550/arXiv.2604.10688. URL http://arxiv.org/abs/2604.10688.

Wenhong Zhu, Ruobing Xie, Rui Wang, and Pengfei Liu. Hybrid Policy Distillation for LLMs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL http://arxiv.org/abs/2604.20244.

A Detailed Derivations

This appendix collects the auxiliary derivations omitted from Section 3.

A.1 Proof of Proposition 1 (Per-Token FKL Loss and Logit Gradient)

Proof. Eq. (5) restates $\mathrm{KL}(p_T\|p_S)$. For the logit gradient, applying $\nabla_z \log p_S(v) = e_v - p_S$ to $\mathcal{L}_{\mathrm{FKL}} = -\sum_v p_T(v)\log p_S(v) + \text{const}$ gives $\partial\mathcal{L}_{\mathrm{FKL}}/\partial z_v = p_S(v) - p_T(v)$. □

A.2 Per-Term Parameter Gradient of RKL

Complementing Proposition 1, we derive the per-term gradient of $\mathcal{L}_{\mathrm{RKL}} = \sum_v p_S(v)\log\big(p_S(v)/p_T(v)\big)$. Differentiating with respect to θ and using $\nabla_\theta p_S(v) = p_S(v)\nabla_\theta \log p_S(v)$,

$$\nabla_\theta \mathcal{L}_{\mathrm{RKL}} = \sum_v \nabla_\theta p_S(v)\, \log\frac{p_S(v)}{p_T(v)} + \sum_v p_S(v)\,\nabla_\theta \log p_S(v) \tag{13}$$

$$= \sum_v \nabla_\theta p_S(v)\left[\log\frac{p_S(v)}{p_T(v)} + 1\right] \tag{14}$$

The implicit per-term coefficient $\log\big(p_S(v)/p_T(v)\big) + 1$ diverges to $-\infty$ as $p_S(v) \to 0$. However, the basis vector $\nabla_\theta p_S(v) = p_S(v)\big(\nabla_\theta z_v - \mathbb{E}_{u\sim p_S}\nabla_\theta z_u\big)$ contains a factor of $p_S(v)$ from the softmax derivative, so the entire per-term contribution scales as $p_S(v)\big[\log\big(p_S(v)/p_T(v)\big) + 1\big]\big(\nabla_\theta z_v - \mathbb{E}_{p_S}\nabla_\theta z_u\big)$, which *vanishes* as $p_S(v)\to 0$. Writing the prefactor as $p_S(v)\log p_S(v) - p_S(v)\log p_T(v) + p_S(v)$, the first piece is the $x\log x \to 0$ limit and the remaining two vanish trivially whenever $p_T(v) > 0$ (which holds under softmax). The vanishing of the per-term gradient contribution (not of the coefficient itself), contrasted with the bounded but G-concentrated FKL contribution from Proposition 1, is the structural reason RKL is stable on the Mismatched Region but cannot *discover* teacher-only modes.

A.3 Proof of Proposition 2 (Decoupled Decomposition of FKL)

Proof. Split the sum over V along the partition $V = M \cup G$,

$$\mathrm{KL}(p_T\|p_S) = \sum_{v\in M} p_T(v)\log\frac{p_T(v)}{p_S(v)} + \sum_{v\in G} p_T(v)\log\frac{p_T(v)}{p_S(v)} \tag{15}$$

Consider the Matched-Region term. Substituting $p_T(v) = r_M^T \tilde{p}_M(v)$ and $p_S(v) = r_M^S \tilde{q}_M(v)$ from Eq. (8),

$$\sum_{v \in M} p_T(v) \log \frac{p_T(v)}{p_S(v)} = \sum_{v \in M} r_M^T \tilde{p}_M(v) \log \frac{r_M^T \tilde{p}_M(v)}{r_M^S \tilde{q}_M(v)} \tag{16}$$

$$= r_M^T \sum_{v \in M} \tilde{p}_M(v) \left[\log \frac{\tilde{p}_M(v)}{\tilde{q}_M(v)} + \log \frac{r_M^T}{r_M^S} \right] \tag{17}$$

$$= r_M^T \cdot \mathrm{KL}(\tilde{p}_M \| \tilde{q}_M) + r_M^T \log \frac{r_M^T}{r_M^S} \cdot \underbrace{\sum_{v \in M} \tilde{p}_M(v)}_{=1} \tag{18}$$

$$= r_M^T \cdot \mathrm{KL}(\tilde{p}_M \| \tilde{q}_M) + r_M^T \log \frac{r_M^T}{r_M^S} \tag{19}$$

The same algebra on G has

$$\sum_{v \in G} p_T(v) \log \frac{p_T(v)}{p_S(v)} = r_G^T \cdot \mathrm{KL}(\tilde{p}_G \| \tilde{q}_G) + r_G^T \log \frac{r_G^T}{r_G^S} \tag{20}$$

Summing Eqs. (19) and (20) and grouping the two $r^T \log(r^T / r^S)$ terms into $\mathcal{L}_{\text{Binary}}$ gives Eq. (9). □

A.4 Non-Negativity of Each Term in the Decoupled Decomposition

The decomposition in Proposition 2 writes $\mathrm{KL}(p_T \| p_S)$ as a sum of three terms; each is individually non-negative.

- $\mathcal{L}_{\text{Matched}} = r_M^T \cdot \mathrm{KL}(\tilde{p}_M \| \tilde{q}_M) \geq 0$ and $\mathcal{L}_{\text{Mismatched}} = r_G^T \cdot \mathrm{KL}(\tilde{p}_G \| \tilde{q}_G) \geq 0$ since each is a non-negative scalar times a KL divergence.
- $\mathcal{L}_{\text{Binary}} = r_M^T \log(r_M^T / r_M^S) + r_G^T \log(r_G^T / r_G^S) = \mathrm{KL}\big((r_M^T, r_G^T) \| (r_M^S, r_G^S)\big) \geq 0$, since it is the KL between two-point distributions.

This justifies treating the three terms as independent loss contributions, so replacing $\mathcal{L}_{\text{Mismatched}}$ with its reverse-KL counterpart preserves non-negativity of the total objective.

A.5 Gradient Conflict Between OPD and GRPO

For a positive-advantage rollout, consider the local policy update at the behaviour policy, where the importance ratio is one and clipping is inactive. Writing y_t for the sampled token and omitting the reference-policy regulariser, the gradient-descent directions at its logit are

$$-\frac{\partial \mathcal{L}_{\mathrm{PG},t}}{\partial z_{y_t}} = \hat{A}\big(1 - p_S(y_t)\big), \qquad -\frac{\partial \mathcal{L}_{\mathrm{FKL},t}}{\partial z_{y_t}} = p_T(y_t) - p_S(y_t). \tag{21}$$

When $p_T(y_t) < p_S(y_t)$, the policy update increases the sampled token's logit while FKL decreases it. This local opposition motivates reducing the KD weight on high-advantage rollouts. The gate $\sigma(-\hat{A}/\tau)$ decreases with $\hat{A}$, reducing KD's influence where the policy-gradient signal is strongest.

B DKLD Algorithm

Algorithm 1 summarises the full training loop.

C Implementation Details

C.1 Distributed Training Architecture

Our implementation uses Ray for distributed actor management and vLLM for efficient on-policy rollout generation.

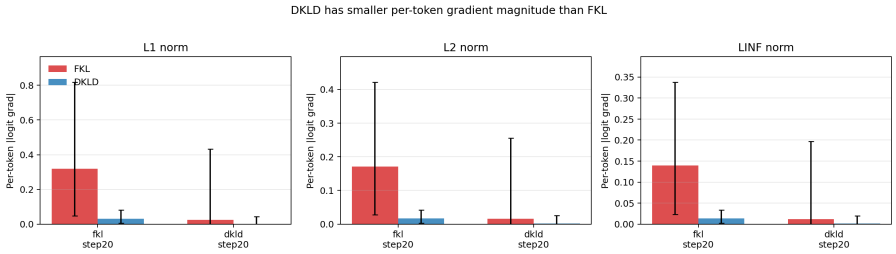


Figure 3: **Per-token logit-gradient norms under FKL and DKLD.** The panels report L_1, L_2, and L_∞ norms at step 20 for checkpoints trained with FKL or DKLD. DKLD yields smaller reported norms under both checkpoint settings.

Algorithm 1 DKLD: Decoupled-KL Distillation with GRPO

Require: Student $p_S(v)$, frozen teacher p_T, prompts $\mathcal{D}$, region thresholds $(\kappa, \epsilon_0, \epsilon_1)$, KD weight λ_{KD}, temperature τ

for each training step s **do**
  Sample prompt $x \sim \mathcal{D}$ and K rollouts $\{y^{(i)}\} \sim p_S(\cdot \mid x)$
  Compute verifier rewards $\{r^{(i)}\}$ and group-relative advantages $\{\hat{A}^{(i)}\}$ (Section 2.3)
  for each correct rollout i with $r^{(i)} > 0$ **do**
    for each token position t in $y^{(i)}$ **do**
      Query teacher top-k logits conditioned on $(h_T(x), x, y^{(i)}_{<t})$ and renormalise
      Partition the candidates into Matched M and Mismatched G with $\epsilon_T = \kappa \max_u p_T(u)$ and $\epsilon_S(s)$ (Def. 1)
      $\mathcal{L}^{(t)}_{\text{Binary}} \leftarrow \mathrm{KL}\big((r^T_M, r^T_G) \,\|\, (r^S_M, r^S_G)\big)$
      Initialise $\mathcal{L}^{(t)}_{\text{Matched}} \leftarrow 0$ and $\mathcal{L}^{(t)}_{\text{Mismatched}} \leftarrow 0$
      if $M \neq \varnothing$ **then**
        $\mathcal{L}^{(t)}_{\text{Matched}} \leftarrow r^T_M \,\mathrm{KL}(\tilde{p}_M \| \tilde{q}_M)$
      end if
      if $G \neq \varnothing$ **then**
        $\mathcal{L}^{(t)}_{\text{Mismatched}} \leftarrow r^S_G \,\mathrm{KL}(\tilde{q}_G \| \tilde{p}_G)$
      end if
      $\mathcal{L}^{(t)}_{\text{DKLD}} \leftarrow \mathcal{L}^{(t)}_{\text{Binary}} + \mathcal{L}^{(t)}_{\text{Matched}} + \mathcal{L}^{(t)}_{\text{Mismatched}}$
    end for
  end for
  Apply rollout-level mask $m^{(i)} = \mathbb{1}[r^{(i)} > 0]$ and AWD gate $w_{\text{awd}} = \sigma(-\hat{A}^{(i)}/\tau)$
  Update θ on combined objective $\mathcal{L} = \mathcal{L}_{\text{GRPO}} + \lambda_{\text{KD}} \cdot \frac{1}{K} \sum_{i=1}^{K} m^{(i)} \, w^{(i)}_{\text{awd}} \cdot \frac{1}{|y^{(i)}|} \sum_{t=1}^{|y^{(i)}|} \mathcal{L}^{(t)}_{\text{DKLD}}$ (Eq. 12)
  Synchronise updated weights to vLLM rollout engines
end for

Colocated model placement. Student, reference, teacher, and vLLM engines are colocated on the same GPUs using fractional GPU allocation. The student actor and reference model share GPU resources using DeepSpeed ZeRO Stage 2. Two vLLM generation engines serve the student's current policy for on-policy rollout generation, each with `gpu_memory_utilization=0.3`. After each gradient step, updated student weights are synchronised to both vLLM engines via the NCCL backend, with `vllm_enable_sleep` for memory efficiency.

Teacher model. The frozen teacher (Qwen3-8B for 0.6B/1.7B students; Qwen3-14B for the 4B student) is loaded in inference mode and colocated with the student. For each student-generated rollout, the teacher receives the full sequence and returns top-k=256 raw logits and the corresponding token indices at every response position.

C.2 Semantic Entropy Computation

We compute semantic entropy (SE) offline on the OpenThoughts candidate pool. For each prompt, we sample N=5 teacher responses at temperature 0.7. Final answers are extracted using regex matching for `\boxed{}` patterns. Pairwise semantic equivalence is determined by exact string matching first, then a Qwen3-4B judge for non-matching pairs. Clusters are formed using Union-Find, and semantic entropy is computed over cluster probabilities. Hard_mini150 is the top-150 highest-SE subset of the OpenThoughts pool. We select validation questions using the same procedure and exclude both Hard_mini150 and the validation set from training.

C.3 Answer-Aware Teacher Prompting

Following Zhao et al. (2026), all teacher-based methods use the same system prompt containing the correct answer during distillation:

> *"You are an expert math teacher. You have access to the correct answer for this problem. Use this knowledge to provide the most helpful guidance. Correct answer: \boxed{[label]}"*

C.4 Inference and Evaluation Protocol

For evaluation, we decode at temperature 0.6 with top-p 0.95 and a maximum of 8,192 new tokens per response. Reported means and SDs are computed across three independent training seeds (10, 20, 30). Math evaluation uses pass@1, with avg@8 for the 30-problem AIME 2025 set. Code generation reports pass@4 on the held-out test cases using the LiveCodeBench v6 test harness. Truncation rate is the fraction of generated responses that hit the 8,192-token cap before emitting an EOS.

C.5 Training Hyperparameters

Table 6 lists the full training configuration. Every method and student size uses the same optimisation, rollout, RL-regularisation, and system settings, and all KD methods share the teacher top-k and λ_{KD}. The DKLD-specific values (τ, κ, ϵ_0, ϵ_1) are fixed across all experiments and were not tuned per model.

Table 6: Full hyperparameter and training configuration for DKLD.

Hyperparameter	**Value**
Optimisation	
Optimiser	AdamW
Learning rate	5×10^{-7}
Schedule	Cosine
Episodes	2
Batch size (train)	32
Batch size (rollout)	32
Micro batch size (train)	1
Micro batch size (rollout)	1
System	
Precision	bfloat16
DeepSpeed stage	ZeRO-2
Attention kernel	FlashAttention-2
Gradient checkpointing	✓
Hardware	4×H100 80GB

Hyperparameter	**Value**
Distillation	
Top-k (teacher logits)	256
KD coefficient (λ_{KD})	0.1
AWD temperature (τ)	1.0
Teacher threshold ratio (κ)	0.1
Student threshold start (ϵ_0)	0.01
Student threshold end (ϵ_1)	0.001
Anneal horizon ($s_{\max}$)	all training steps
Rollout and sequence	
Rollouts per prompt (K)	8
Rollout temperature	1.0
Max sequence length	8,192
Max new tokens	7,168
RL regularisation	
KL penalty (β)	0.01
PPO clip (ε)	0.2

D CompMath-700 Evaluation Set

CompMath-700 is a held-out evaluation set of 700 problems we collected from publicly available online math-competition sources in several countries and languages, to mitigate contamination from

publicly released English benchmarks. We translated each problem into English with Gemini 3.1 Pro, and two authors manually checked every translated problem and its answer against the source. Each problem is paired with a unique numeric or short-form symbolic answer to allow exact-match grading. We deduplicate against GSM8K, MATH-500, and AIME 2025 by both surface-string and semantic-equivalence judging, removing any problem with a positive match. Grading uses the same `\boxed{}` extractor as the public benchmarks.

E Collapse Dynamics of FKL

MATH-500 training trajectories. Figure 2 reports LiveCodeBench pass@4 against truncation rate. To show the collapse directly on math, we evaluated the full MATH-500 every 5 steps under one shared configuration (Qwen3-8B to Qwen3-1.7B, 8,192-token generation limit); Table 5 in the main text reports the step-35-to-50 window. From step 35 to step 50, the truncation rate of FKL+GRPO doubles from 40.2% to 80.6% while MATH-500 pass@1 falls from 44.6 to 20.2; under the same limit, DKLD stays between 14.6% and 15.2% truncation and 80.2–82.0 pass@1. Under the shared generation limit, FKL+GRPO loses accuracy as truncation increases, while DKLD remains stable over this window. The FKL entry of Table 2 for Qwen3-1.7B is the validation-selected checkpoint at step 10, before this degradation sets in; FKL degrades from step 15 onward.

All methods at a common training step. Table 7 compares all methods at step 80 under the shared configuration above. FKL has 13.7% pass@1 and 91.2% truncation, whereas DKLD achieves the highest pass@1 (81.5%) and the lowest truncation rate (14.6%). Figure 1 summarises these results.

Table 7: MATH-500 pass@1 and truncation rate of every method at training step 80 (Qwen3-1.7B student, Qwen3-8B teacher, shared 8,192-token generation limit). Pass@1 is the mean ± sample SD over training seeds 10, 20, and 30. KD baselines use the uniform weight $w = 1$; DKLD uses the AWD gate (Table 8 separates the two).

Method	pass@1	trunc.
GRPO	76.0 $\pm$1.5	19.8%
FKL	13.7 $\pm$0.6	91.2%
RKL	74.8 $\pm$1.3	19.7%
AKL	76.6 $\pm$1.2	21.1%
ToDi	76.6 $\pm$0.7	18.2%
DKLD (ours)	**81.5** $\pm$1.5	**14.6%**

Separating the KD objective from AWD at step 80. Table 8 complements the Hard_mini150 component ablation in Table 3 with a fixed-step MATH-500 comparison. Comparing FKL and DKLD under the same gate tests the effect of the KD objective; comparing gates within each row tests the effect of AWD, including its change to the effective KD weight. The KD objective determines stability. FKL collapses with or without the gate (13.7 and 15.3 pass@1, with 91.2% and 90.7% truncation), whereas DKLD reaches 77.5 without AWD and 81.5 with it, 63.8 and 66.2 points above FKL under the same gate. AWD adds 4.0 points to DKLD but only 1.6 points to FKL, so the gate improves an already stable objective but does not prevent the FKL collapse.

Table 8: **KD objective × AWD at training step 80.** Qwen3-8B teacher, Qwen3-1.7B student, and MATH-500 with an 8,192-token evaluation limit. The comparison holds GRPO, the correct-rollout mask, the base KD coefficient, and training seeds (10, 20, 30) fixed. Pass@1 is mean ± sample SD; both metrics are percentages. The FKL (without AWD) and DKLD (with AWD) entries are reproduced from Table 7. The truncation rate was not recorded for DKLD without AWD.

	Without AWD ($w = 1$)		**With AWD** ($w = \sigma(-\hat{A}/\tau)$)	
KD objective	pass@1 ↑	trunc. ↓	pass@1 ↑	trunc. ↓
FKL	13.7 $\pm$0.6	91.2	15.3 $\pm$0.9	90.7
DKLD	77.5 $\pm$0.6	–	81.5 $\pm$1.5	14.6

F LIMITATIONS

We highlight three limitations. First, our downstream task evaluation uses the Qwen3 family with 8B/14B teachers and 0.6B/1.7B/4B students; whether the Mismatched-Region phenomenon and the FKL/RKL split scale identically to substantially larger teachers (e.g., 70B+) is not verified. Second, the region thresholds $(\kappa, \epsilon_0, \epsilon_1)$ follow a hand-set rule and schedule rather than being learned. Third, the advantage-weighted gate $\sigma(-\hat{A}/\tau)$ relies on the group-relative advantage; very small groups provide fewer samples for estimating this signal.

G COMPUTE REPORTING

All experiments use 4×NVIDIA H100 80GB GPUs in bfloat16 with DeepSpeed ZeRO-2 and FlashAttention-2. Table 9 reports the wall-clock time and GPU-hours for one training seed. Each configuration is trained with seeds 10, 20, and 30.

Table 9: Compute cost per single full training run.

Configuration	**GPUs**	**Wall-clock**	**GPU-hours**
Qwen3-0.6B student (teacher Qwen3-8B)	4×H100	∼10 h	∼40
Qwen3-1.7B student (teacher Qwen3-8B)	4×H100	∼10 h	∼40
Qwen3-4B student (teacher Qwen3-14B)	4×H100	∼12 h	∼48

The main math results comprise six training methods at three student sizes, each trained with three seeds: $6 \times 3 \times 3 = 54$ training runs. Using the per-seed costs in Table 9, these runs account for approximately $36 \times 40 + 18 \times 48 = 2{,}304$ H100 GPU-hours.