LOCALITY GUIDED SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) trains a language model on its own responses using supervision from the same model given privileged, training-only context. However, this privileged context can improve reasoning while also altering response style, so matching the teacher's full predictive distribution may reinforce changes unrelated to the desired reasoning improvement. We introduce the *Target-Locality Hypothesis*, with two testable predictions. **P1 (Correction Retention):** distillation targets that remain local to the student's predictive distribution can retain most of the useful corrections in the privileged proposal; **P2 (Preference Advantage):** learning from such local targets yields responses that align better with human preferences than vanilla student or full distillation. Motivated by this hypothesis, we propose *Locality Guided Self-Distillation* (LGSD), which constructs a distillation target that is as close as possible to the privileged proposal while remaining within a response-level KL budget around the current student, and then trains the student on this local target. Empirical studies, including human preference evaluations on Arena, provide evidence for both P1 and P2. Across mathematical and logic reasoning benchmarks, LGSD improves over matched OPSD and other baselines, while additional analyses further demonstrate improved training stability and reduced style drift.

1 INTRODUCTION

Self-distillation provides a way to improve a model using stronger behavior already latent in the model itself, without requiring a separate teacher or additional information at deployment. In reasoning tasks, the model under its ordinary deployment prompt acts as the student, while the same model conditioned on a privileged, training-only reasoning instruction acts as the teacher. The student first generates a response, and the teacher then rescores the same response prefixes, providing dense token-level supervision that can transfer the benefit of the privileged reasoning context back to the deployment model (Agarwal et al., 2023; Gu et al., 2023). On-policy distillation applies this supervision to prefixes actually generated by the current student, reducing the mismatch between the states used for training and those encountered at deployment (Agarwal et al., 2023). Building on this setup, On-policy self-distillation (OPSD) conditions the teacher on privileged training-time context and clips large pointwise teacher–student divergence contributions to stabilize learning when the two predictive distributions differ substantially (Zhao et al., 2026).

However, the privileged proposal produced by the teacher need not be an ideal distillation target. Privileged conditioning can expose useful reasoning corrections, but it may also introduce incidental shifts in response behavior; indeed, OPSD reports disproportionately large teacher–student divergence on some stylistic tokens, motivating pointwise clipping (Zhao et al., 2026). More generally, because the privileged proposal q^P is induced by additional training-time context (e.g., a reference solution, reasoning trace, or reasoning instruction) rather than constructed as the ideal teacher distribution, directly fitting it may transfer both useful corrections and unwanted shifts. This makes target construction a fundamental question: *how much of the privileged proposal should be transferred to the student*?

We study this question through *target locality*: constraining the distillation target to remain within a response-level KL neighborhood of the deployment-context student p on the same response prefixes, inspired by the trust-region principle (Schulman et al., 2015). As illustrated in Figure 1, an ideal KL-regularized target takes the exponential form $q_t^\star(v) \propto p_t(v)\exp(R_t(v)/\beta)$ (Rafailov et al., 2023). A privileged proposal q^P need not coincide with this ideal target, so its shift from p may contain

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.

both useful reasoning corrections and incidental changes induced by the privileged context. Full distillation transfers this entire shift.

This motivates our **Target-Locality Hypothesis**: reducing the overall shift from the student need not proportionally reduce useful correction. The hypothesis yields two testable predictions. **P1 (Correction Retention):** targets that remain local to the student's predictive distribution retain most of the useful correction in the privileged proposal, despite moving less from the student, as supported Figure 2. **P2 (Preference Advantage):** training on such local targets produces responses that align better with human preferences than either the initial student or full distillation to the privileged proposal, as supported in Figure 5.

To test these predictions, we introduce **Locality-Guided Self-Distillation** (LGSD), which constructs the target closest to the privileged proposal q^P subject to a response-level KL budget ϵ around the current student p. The resulting KL projection has the geometric interpolation form $q^C_{t,\alpha} \propto p_t^{1-\alpha}(q_t^P)^\alpha$, where $\alpha \in [0,1]$ is shared across positions and determined per response. This projection preserves as much privileged guidance as possible while limiting the total shift from the student, and can be used with the existing forward-KL distillation objective without additional model forward passes.

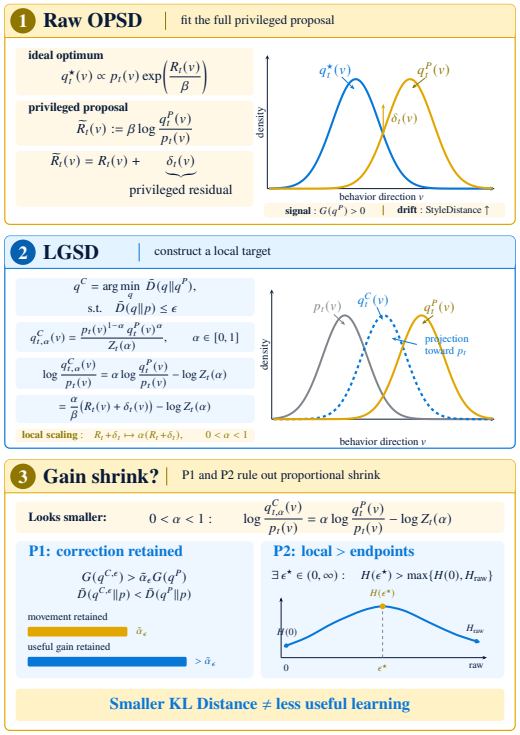

Figure 1: **LGSD Local Targets.** Bounding privileged shifts while preserving reasoning signals (P1 & P2).

Unlike approaches such as Veto (Jang et al., 2026), which primarily modify OPSD to improve optimization stability, LGSD addresses a different question: *how much of the privileged proposal should be transferred to the student?* We answer this through response-adaptive target construction under the Target-Locality Hypothesis.

Our contributions are threefold.

(I) A testable hypothesis for distillation target design. We formulate the *Target-Locality Hypothesis*, predicting that local targets can retain most useful privileged correction while yielding better human-preference alignment than either the initial student or full distillation.

(II) Locality-guided, response-adaptive target construction. We introduce LGSD, which constructs the target closest to the privileged proposal subject to a response-level KL budget around the student, without additional model forward passes or trainable modules.

(III) Empirical validation of reasoning, preference, and drift. Across mathematical and logical reasoning benchmarks, LGSD outperforms matched answer-free OPSD and other baselines. Human preference evaluation on Arena and additional analyses of training stability and style drift further support the two predictions of the *Target-Locality Hypothesis*.

We released in public our code[1], LGSD and OPSD checkpoints[2].

2 Background and Target-Locality Hypothesis

2.1 Privileged On-Policy Self-Distillation

On-policy self-distillation (OPSD) can condition the teacher on privileged information such as reference answers or reasoning traces, or on an answer-free reasoning instruction. We use the latter so

[1] https://anonymous.4open.science/r/clean_distill-31E4

[2] https://anonymous-hf.com/a/en52lzzj0jdx/, and https://anonymous-hf.com/a/rnfasn6kfn6u/

that the privileged view adds reasoning guidance without revealing task-specific answers, allowing us to isolate how the resulting proposal should be used for self-distillation.

Let $\mathcal{D}$ be the training set, V the vocabulary, and $\mathcal{B} = \{x_i\}_{i=1}^{B} \subset \mathcal{D}$ a minibatch. The current student snapshot $\pi_{\text{old}} = \pi_{\theta_{\text{old}}}$ samples one response per problem:

$$y_i = (y_{i,1}, \ldots, y_{i,T_i}) \sim \pi_{\text{old}}(\cdot \mid x_i), \qquad i = 1, \ldots, B. \tag{1}$$

We evaluate ordinary and privileged predictions on the same sampled prefix:

$$\begin{aligned} p_{i,t}(v) &= \pi_{\text{old}}(v \mid x_i, y_{i,<t}), \\ q^{P}_{i,t}(v) &= \pi_{\text{old}}(v \mid x_i, c, y_{i,<t}), \qquad v \in V. \end{aligned} \tag{2}$$

Both views share model parameters; only the privileged view receives c.

We use *target-to-student forward KL* to fit soft supervision across the vocabulary, rather than treating sampled tokens as ground truth. As per described in OPSD (Zhao et al., 2026), forward KL is comparatively better compared to reverse KL and Jensen-Shannon Diversity. For fixed targets, this is equivalent to soft-label cross-entropy up to a target-only constant. Writing $D_{\text{KL}}(a\|b) = \sum_{v\in V} a(v)\log[a(v)/b(v)]$, the unclipped objective for the trainable student π_θ is

$$\mathcal{L}_{\text{SD}}(\theta; q) = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{T_i}\sum_{t=1}^{T_i} D_{\text{KL}}(q_{i,t} \,\|\, \pi_\theta(\cdot \mid x_i, y_{i,<t}))\,. \tag{3}$$

Targets and sampled prefixes remain fixed during fitting. Our matched OPSD baseline uses $q_{i,t} = q^{P}_{i,t}$, whereas LGSD uses the projected target $q_{i,t} = q^{C}_{i,t}$. Both share the rollout protocol, optimizer, and pointwise loss clipping; only the supervision target differs. Below, single-response expressions omit i and use $T = T_i$.

2.2 Target-Locality Hypothesis and Empirical Study

We formalize the hypothesis through two predictions: P1 concerns correction retention before training, while P2 concerns preference advantage after training.

P1 (Correction Retention). For most locality budgets ε,

$$G(q^{C,\varepsilon}) > \tilde{\alpha}_\varepsilon G(q^{P}), \qquad \bar{D}(q^{C,\varepsilon}\|p) < \bar{D}(q^{P}\|p). \tag{4}$$

Here, $G(q)$ measures reference-token gain over the student, and $\tilde{\alpha}_\varepsilon$ is the fraction of the privileged proposal's TV movement retained by the local target. Thus, the fraction of useful gain retained by the local target, $G(q^{C,\varepsilon})/G(q^{P})$, exceeds the fraction of distributional movement retained, $\tilde{\alpha}_\varepsilon$, while remaining closer to the student, as measured by the mean target-to-student KL $\bar{D}(q\|p)$.

P2 (Preference Advantage). A local training target can outperform both endpoints:

$$\exists\, \varepsilon^\star \in (0, \infty): \quad H(\varepsilon^\star) > \max\{H(0), H_{\text{raw}}\}. \tag{5}$$

Here, $H(\varepsilon)$ denotes the human-preference score of the policy trained with locality budget ε. For some intermediate budget $\varepsilon^\star$, $H(\varepsilon^\star)$ is predicted to exceed both the initial student, $H(0)$, and full distillation to the privileged proposal, H_{raw}.

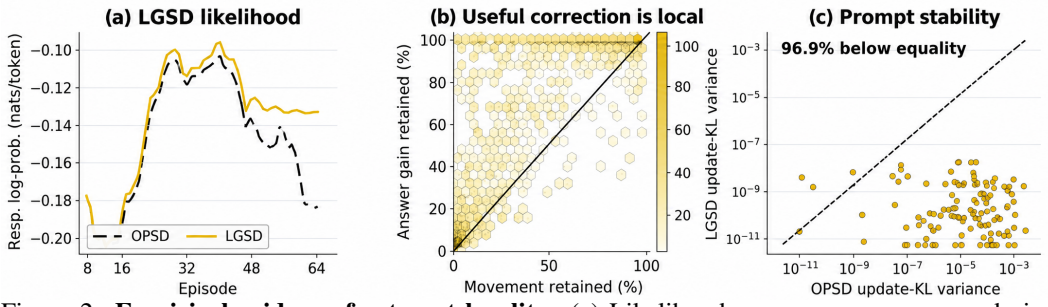

Figure 2: **Empirical evidence for target locality.** (a) Likelihood on common responses during training. (b) Retained correction versus target movement. (c) Variation in KL across various privileged context.

Figure 2 (b) supports P1 on the DeepMath audit set (He et al., 2025b). We freeze the student and privileged proposal, vary the locality budget ε, and plot the fraction of reference-token gain retained,

$G(q^{C,\varepsilon})/G(q^P)$, against the fraction of distributional movement retained, $\tilde{\alpha}_\varepsilon$, as defined in Eq. (4). The diagonal represents equal retention of gain and movement. Most points lie above it, indicating that local targets retain disproportionately more useful gain than distributional movement, directly supporting P1.

Figure 2 (a) and Figure 2 (c) provide complementary evidence for the mechanism behind P1, identically over DeepMath dataset. Figure 2 (a) tracks the token-averaged log-likelihood $\frac{1}{T}\sum_{t=1}^{T}\log \pi_\theta(y_t \mid x, y_{<t})$ on the same common responses throughout training; because both methods score identical response tokens, the comparison reflects how their predictions change over time. The trajectories are similar early on, but LGSD avoids the sharp late-stage decline seen with OPSD, indicating more stable learning. Figure 2 (c) tests sensitivity to the privileged context by varying multiple equivalent phrasings c, and measuring $\mathrm{Var}_c\big[\bar{D}(q(c)\|p)\big]$. The lower variance under LGSD shows that equivalent instructions produce more consistent supervision targets. Together, (a) and (c) support the mechanism behind P1: locality reduces unwanted target variation and leads to more stable training. **P2 is tested separately** using held-out human-preference pairs in Section 4.2.

3 The Proposed LGSD Method

LGSD follows two steps (Figure 3). First, each sampled response is scored under the ordinary prompt to obtain p_t and under the privileged instruction to obtain q_t^P. We then use bisection to find the largest interpolation coefficient that keeps the teacher target within the response-level KL budget ε around p_t, yielding q^C. Second, we keep q^C fixed and train the ordinary-prompt student to match it. The process is repeated for each new response using the updated student.

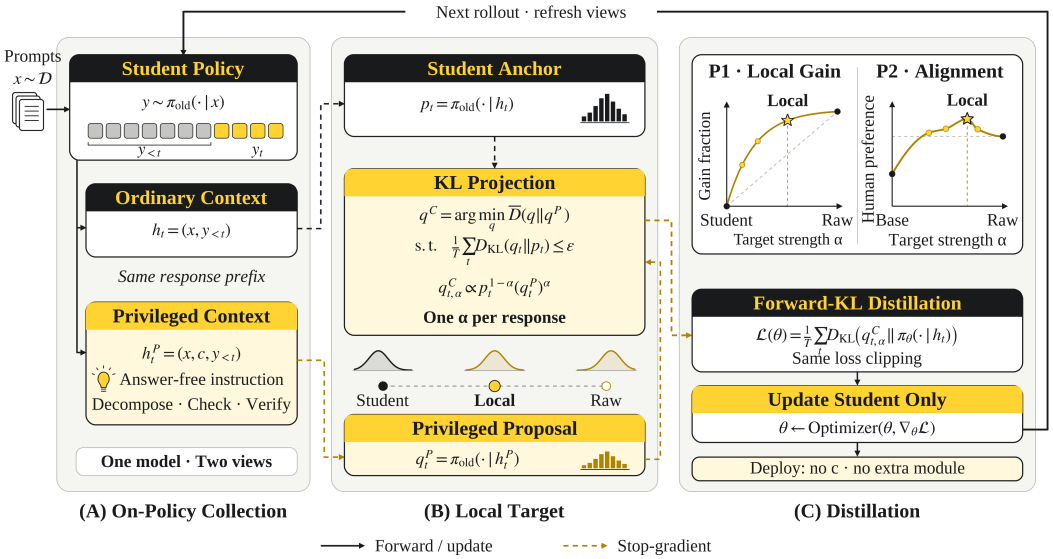

Figure 3: **LGSD training pipeline.** A trajectory-shared projection produces the local target q^C before distillation.

3.1 Constructing the Local Target

P1 suggests a simple target-construction principle: move toward the privileged proposal as much as possible, but stop before the supervision moves too far from the student. We therefore choose, for each response, the distribution closest to the privileged proposal subject to a fixed KL budget around the student:

$$q^C = \arg\min_q \frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}(q_t\|q_t^P) \quad \text{s.t.} \quad \frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}(q_t\|p_t) \le \varepsilon. \tag{6}$$

We tune a single locality radius ε on the development set and then fix the selected value across all backbones and training settings. For any chosen ε, Proposition 1 gives a unique projected target. The selection procedure is described in Appendix B.1.

Proposition 1 (Trajectory-shared geometric projection). *For full-support p_t, q_t^P and $\varepsilon \geq 0$, the unique solution to Equation 6 is*

$$q_{t,\alpha}^C(v) = \frac{p_t(v)^{1-\alpha} q_t^P(v)^\alpha}{\sum_u p_t(u)^{1-\alpha} q_t^P(u)^\alpha}, \qquad \alpha \in [0,1], \tag{7}$$

with the same α at every position. If q^P is feasible, $\alpha = 1$; otherwise the constraint is tight. At $\varepsilon = 0$, the target is p.

The coefficient α controls how far the target moves from the student toward the privileged proposal: $\alpha = 0$ recovers the student, while $\alpha = 1$ recovers the full proposal. If the proposal already satisfies the KL budget, we keep it unchanged. Otherwise, we use bisection on $[0, 1]$ to find the largest α that satisfies the constraint, thereby staying within the budget while remaining as close as possible to the proposal.

Projection in one line. q^P decides *where* supervision wants to move; p and ε decide *how far*; q^C is the farthest proposal-faithful point on the geometric path allowed by that radius.

3.2 Theory Guarantee and Style Drift

We measure style drift using the StyleDistance encoder (Patel et al., 2024). Given responses y_i^M and y_i^0 to the same question from a trained model and Base, respectively, we compute $\text{StyleDistance}(M, 0) = \frac{1}{N}\sum_{i=1}^{N}\left[1 - \cos\big(h(y_i^M), h(y_i^0)\big)\right]$, where h maps each response to its style embedding. Lower values indicate less stylistic change from Base.

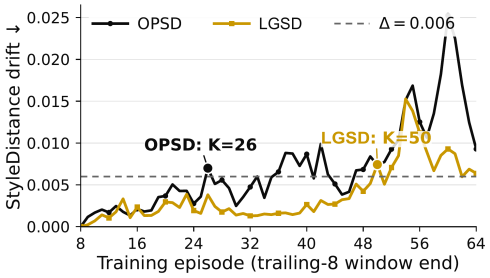

Figure 4: LGSD reduces and delays style drift.

Figure 4 reports the trailing-eight-episode windowed audit. StyleDistance first crosses .006 at episode 26 for OPSD and episode 50 for LGSD, an observed delay of $1.92\times$. Thus, local-target training reaches the same measured style-drift threshold later in this run.

Closeness to the global optimum. On fixed prefixes, let $q_t^\star(v) \propto p_t(v)\exp(R_t(v)/\beta)$ be the global optimum of the KL-regularized reward objective, as defined in Figure 1, and these rewards define the comparison and are not used in training.

Proposition 2 (LGSD targets are closer to the global optimum). *Assume p_t, q_t^P have full support on a finite vocabulary and $\varepsilon > 0$. If the global optimum $q^\star$ lies within the response-level KL budget around p but the raw proposal q^P does not, the exact* LGSD *target q^C is strictly closer to $q^\star$ than the raw OPSD target:*

$$\frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}(q_t^\star \| q_t^C) < \frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}(q_t^\star \| q_t^P). \tag{8}$$

Appendix E provides the proof and approximate-bisection correction.

3.3 Forward-KL Distillation of the Local Target

Forward KL uses the teacher's full next-token distribution as soft labels. For a fixed target, it is equivalent to cross-entropy, with unclipped logit gradient $p_t - q_t$ at the current student. Thus, errors in the teacher's probabilities directly become errors in the training signal. OPSD also reports its strongest gains with forward KL among the objectives tested (Zhao et al., 2026).

We retain this loss but replace the raw privileged target q^P with the local target q^C. This changes the supervision without changing how the student learns from it. When the ideal target lies within the KL budget but the raw proposal does not, our projection gives a tighter bound on squared logit-gradient error: the LGSD bound is at most α times the corresponding OPSD bound.

Proposition 3 (Improved supervision for forward-KL training). *Assume the conditions of Proposition 2. For fixed prefixes, let $g_t(q) = p_t - q_t$ denote the unclipped forward-KL logit gradient at the*

rollout student. Define its error relative to supervision from $q^\star$ as

$$\mathcal{E}(q) := \frac{1}{T}\sum_{t=1}^{T} \|g_t(q) - g_t(q^\star)\|_1^2.$$

Exact LGSD projection with coefficient $\alpha \in (0,1)$ satisfies

$$\mathcal{E}(q^C) \le 2\overline{D}(q^\star\|q^C) \le 2\alpha\,\overline{D}(q^\star\|q^P), \quad \mathcal{E}(q^P) \le 2\overline{D}(q^\star\|q^P). \tag{9}$$

Thus, the LGSD error bound is at most α times the corresponding raw-OPSD bound.

When the optimal distribution lies within the budget but the raw proposal does not, LGSD provides supervision closer to that optimum and a tighter bound on training-signal error, using the same forward-KL loss. This compares error bounds, not actual errors after training. Appendix F gives the proof and the correction for numerical projection.

4 Experiments

4.1 Experimental Setup

Models, data, and hardware. We run the experiments on NVIDIA B200, H200, and H100 GPUs. We train Qwen3-1.7B, Qwen3-8B (Yang et al., 2025), and GPT-OSS-20B (Agarwal et al., 2025) on one ordered DeepMath-103K stream (He et al., 2025b). Matched target comparisons share data order, optimizer, generation budget, loss, and student and scorer prompts. Each reasoning-table row reports one deterministic-order training run; Appendix D gives the configurations and exact interfaces.

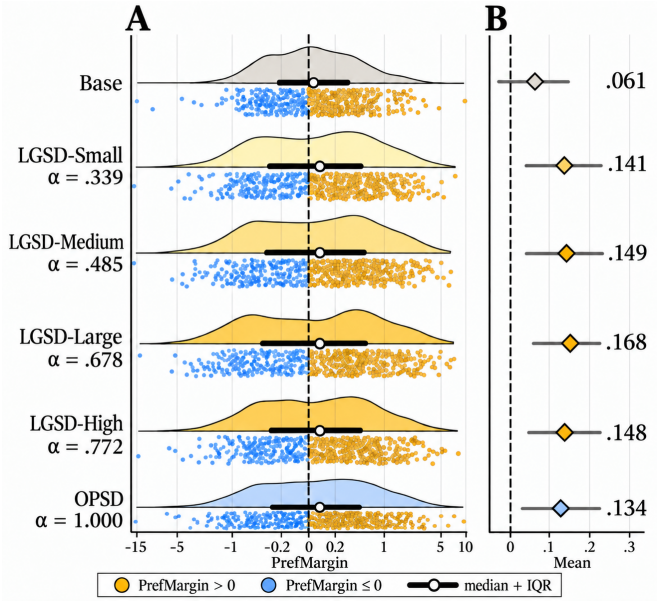

Figure 5: **Preference likelihood peaks at a local radius.** (A) Pair-level margins with median and IQR. (B) Mean ± SEM.

Baselines and matched target control. Under our matched answer-free protocol, OPSD fits the raw privileged proposal (Zhao et al., 2026), while LGSD projects it before fitting with the same pointwise-clipped loss. Qwen baselines additionally include Veto, which penalizes teacher log scores for tokens with low student confidence (Jang et al., 2026); SRPO, which routes correct samples to GRPO and failed samples to self-distillation, using entropy-aware weighting to downweight unreliable, high-entropy teacher targets (Li et al., 2026); and progress reward-based GRPO-PRM (Shao et al., 2024; He et al., 2025a). Base denotes the unadapted model.

Complementary evaluation. Arena human votes evaluate open-domain preference alignment (Chiang et al., 2024). Math-Verify scores AMC/AIME reasoning (Hugging Face, 2024); PySAT/Z3 verify held-out SATQuest and LogicSkills (Mathematical Association of America, 2025; Pan et al., 2026; Zhao et al., 2025; Rabern et al., 2026). These formal-logic tasks complement the mathematical benchmarks by testing whether reasoning learned from DeepMath transfers beyond mathematical problem solving. In particular, they evaluate constraint following and logical consistency on held-out tasks with formal verifiers. Math Acc@1 and Logic Pass@1 are verifier acceptance rates as default in all tables.

4.2 HUMAN-PREFERENCE ALIGNMENT

Human-preference alignment. To deliver experimental and empirical support of **P2**, we compare Qwen3-8B Base, four LGSD α (0.001, 0.004, 0.016, and 0.040), and matched OPSD on the same Arena human-voted response pairs, using checkpoints after 1,000 optimizer steps. Runs share the same training conditions and differ only in target construction. For each pair (x_i, y_i^+, y_i^-), where y_i^+ is the human-preferred response and y_i^- the rejected response, we define

$$\mathrm{PrefMargin}_i(M) = H(M) = \frac{1}{|y_i^+|}\sum_t \log \pi_M(y_{i,t}^+ \mid x_i, y_{i,<t}^+) - \frac{1}{|y_i^-|}\sum_t \log \pi_M(y_{i,t}^- \mid x_i, y_{i,<t}^-). \tag{10}$$

Figure 5 shows how human-preference likelihood changes as the target moves from Base toward the raw OPSD proposal. Higher PrefMargin means that the model assigns relatively more likelihood to the human-preferred response. The mean margin increases for local targets, reaches its maximum at an intermediate radius, and then decreases as the target moves further toward raw OPSD. This directly supports P2, $H(\varepsilon^\star) > \max\{H(0), H_{\mathrm{raw}}\}$, showing that an intermediate local target can align better with recorded human preferences than either endpoint.

		DeepMath-aligned math ID (%)				Logical OOD (%)			Style drift
Method	Ep.	AMC 2022/2023	AIME 2024	AIME 2025	ID All	SATQuest	LogicSkills	OOD All	Style Distance↓
Qwen3-1.7B									
Base	0	56.63	33.33	23.33	44.76	9.97	49.67	22.22	0.0000
GRPO-PRM	64	61.45	36.67	**33.33**	50.35	9.52	54.33	23.35	0.0104†
SRPO	16	59.04	36.67	26.67	47.55	10.12	53.67	23.56	0.0167†
SRPO	64	60.24	36.67	26.67	48.25	8.48	54.33	22.63	0.0169†
OPSD	16	66.27	30.00	23.33	49.65	11.90	53.33	24.69	0.0132
Veto	16	60.24	30.00	23.33	46.15	11.76	54.00	24.79	0.0150
LGSD	16	**67.47**	33.33	**33.33**	**53.15**	**13.10**	52.58	**25.29**	**0.0116**
OPSD	64	24.10	6.67	3.33	16.08	12.05	37.67	19.96	0.0423
Veto	64	28.92	3.33	3.33	18.18	12.42	**54.67**	25.46	0.0434
LGSD	**64**	**67.47**	**40.00**	**33.33**	**54.55**	**24.80**	38.99	**29.18**	**0.0412**
Qwen3-8B									
Base	0	65.06	50.00	26.67	53.85	12.50	81.67	33.85	0.0000
GRPO-PRM	64	72.29	56.67	36.67	61.54	12.35	81.00	33.54	0.0133†
SRPO	16	69.88	53.33	33.33	58.74	12.65	81.67	33.95	0.0241†
SRPO	64	73.49	56.67	36.67	62.24	12.20	80.33	33.23	0.0246†
OPSD	16	71.08	46.67	26.67	56.64	14.43	80.67	34.88	0.0166
Veto	16	69.88	**50.00**	30.00	57.34	13.10	81.67	34.26	0.0235
LGSD	16	**72.29**	46.67	**36.67**	**59.44**	**17.38**	**81.78**	**37.29**	**0.0158**
OPSD	64	77.11	56.67	30.00	62.94	15.03	56.33	27.78	0.0370
Veto	64	75.90	53.33	43.33	64.34	19.49	**81.67**	38.68	0.0312
LGSD	**64**	**83.13**	**70.00**	**50.00**	**73.43**	**30.06**	76.06	**44.28**	**0.0307**
GPT-OSS-20B									
Base	0	81.93	70.00	46.67	72.03	25.15	73.67	40.12	0.0000
OPSD	16	**85.54**	56.67	60.00	74.13	24.70	72.33	39.40	0.0211
LGSD	**16**	84.34	**70.00**	**66.67**	**77.62**	**25.45**	**74.33**	**40.53**	**0.0156**
OPSD	64	37.35	6.67	6.67	24.48	15.62	35.33	21.71	0.1431
LGSD	**64**	**75.90**	**53.33**	**33.33**	**62.24**	**24.70**	**73.33**	**39.71**	**0.0423**

Table 1: **Reasoning accuracy (%) and style drift.** †: reward-guided RL; lower drift is expected from external rewards and self-proposed, reverse-KL-like optimization.

4.3 MATHEMATICAL REASONING AND FORMAL-LOGIC TRANSFER

Mathematical reasoning and formal-logic transfer. Figure 7 shows that LGSD's advantage over OPSD widens with continued training on Qwen3-8B. Table 1 shows that this benefit extends beyond the matched comparison: both Qwen models achieve the highest final aggregate math accuracy among the tested methods, outperforming Veto and the reward-guided baselines. On GPT-OSS-20B, the main benefit is instead reduced late-stage degradation, although performance still falls

below Base. The improvements also extend to held-out formal logic, where final aggregate accuracy exceeds OPSD across all backbones and Base on both Qwen models. This transfer is uneven: the Qwen models improve on SATQuest but remain below Base on LogicSkills. Together, these results support local-target training as a way to improve mathematical reasoning and aggregate logic transfer, while showing that it mitigates rather than eliminates degradation during prolonged self-distillation.

Training dynamics. Figure 6 connects the endpoint gains to the full run. Late in training, LGSD maintains higher verifier reward as its responses shorten. Its lower late-stage entropy also shows that entropy alone does not order the methods by accuracy.

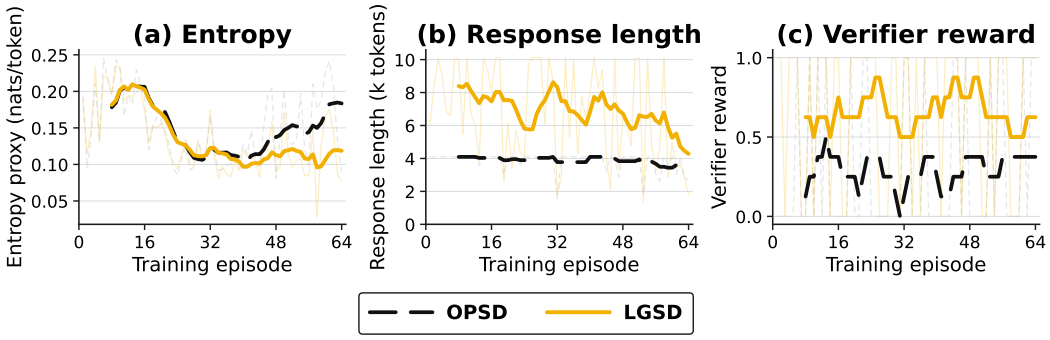

Figure 6: **Qwen3-8B training dynamics** over 64 episodes.

4.4 Hyperparameter Sensitivity

The locality radius ε controls how far the projected target may move from the current student: smaller values enforce more local supervision, while larger values approach the raw privileged proposal. We sweep ε on a random 5% DeepMath development split using answer-free diagnostics, including achieved KL, style movement, prompt sensitivity, and constraint activity. Based on this sweep, we select $\varepsilon = 0.004$ and fix it across all backbones and training episodes. The complete sensitivity analysis and post-selection target-space audit are reported in Appendix Table 6 and Figure 8.

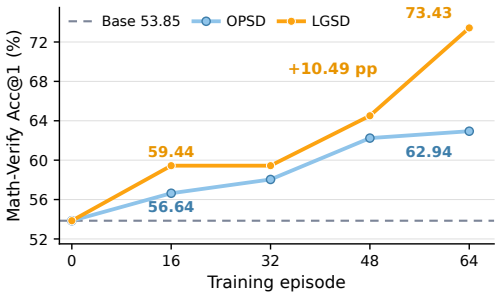

Figure 7: Qwen3-8B accuracy over training.

4.5 Ablation Study

Table 2: **Response-level target allocation.** Qwen3-8B, episode 64.

Construction	Mean α	Target KL	Response budget	Correct ↑	Cap hits ↓
Raw proposal (OPSD)	1.0000	0.014728	Unconstrained	90	43
Token-wise α_t	–	0.001220	Per-token budgets	90	45
Fixed shared α	0.5596	0.004637	51.56% violated	95	45
Trajectory-adaptive LGSD	**0.5596**	**0.003882**	**All feasible**	**105**	**25**

Why this locality target construction? Table 2 compares LGSD with three alternative target constructions. The raw proposal imposes no locality control, while token-wise budgets constrain each position independently and recover no accuracy gain over OPSD despite producing the smallest target KL. A fixed shared coefficient allows movement to be distributed across tokens, but cannot adapt to how far each response lies from its privileged proposal and therefore violates many response budgets. LGSD combines both properties: it shares one budget across positions while solving the interpolation coefficient separately for each response. This keeps every target within its prescribed locality region while improving correctness and reducing budget exhaustion.

Non-triviality of local target construction. Table 3 tests whether LGSD can be approximated by modifying optimization while keeping the raw OPSD target fixed. We vary learning rate, adapter rank, policy regularization, gradient clipping, and update scale. None reproduces the effect of local target construction. The strongest control solves 92 questions with 44 cap hits, while LGSD solves 105 with 25. Thus, simply shrinking or regularizing the OPSD update does not recover LGSD.

Table 3: **Raw-target attenuation does not match LGSD.** Qwen3-8B, 143 questions.

Method / control	Correct↑	Cap hits↓
OPSD	90	43
Half learning rate	82	61
LoRA rank 4, scale 8	88	55
Policy KL ($\beta = 1$)	80	61
Gradient clip 0.5	92	44
Update scale 0.5	77	64
LGSD	**105**	**25**

Performance and stability. Table 4 summarizes complementary audits. Across independent Qwen3-8B training repeats, LGSD reaches $70.86 \pm 2.46\%$ at episode 64, compared with $64.57 \pm 1.61\%$ for OPSD. Repeated decoding of fixed checkpoints yields $69.23 \pm 0.70\%$ for LGSD and $65.27 \pm 1.07\%$ for OPSD. These correspond to mean gains of 6.29 and 3.96 percentage points over OPSD under training and decoding variation, respectively. On GPT-OSS-20B, local targets reduce KL from initialization and retain more predictive entropy, consistent with the smaller late-stage accuracy loss in Table 1. The Qwen3-8B repeats show higher mean endpoint accuracy under both sources of variation; the GPT-OSS-20B audit provides separate evidence of reduced policy drift. Appendix B.3 expands the repeated-run evidence and Appendix A discusses the mechanism.

Table 4: **Performance and stability.** Accuracy: mean ± sample SD over three repeats; policy metrics: fixed-prefix audits.

Qwen3-8B, episode 64 Accuracy (%): mean ± SD, three repeats		
Audit	OPSD	LGSD
Training	64.57 ± 1.61	70.86 ± 2.46
Decoding	65.27 ± 1.07	69.23 ± 0.70
GPT-OSS-20B, episode 64		
Point estimates	OPSD	LGSD
KL to initialization	0.1834	0.0966
Entropy retention	0.730	0.891

Computational cost. Projection adds no model forward pass: measured active time is 4.4596 versus 4.6832 H100-hours, with identical checkpoint size and effectively unchanged peak memory. The complete systems measurements appear in Table 10.

5 Related Work

On-policy and self-distillation. Recent distillation methods increasingly train students on states induced by their own generations rather than only on fixed teacher data. GKD applies teacher supervision to student-generated sequences (Agarwal et al., 2023), while MiniLLM studies reverse-KL objectives for language-model distillation (Gu et al., 2023). Self-distillation further removes the need for a separate teacher by transferring supervision within the same model family (Furlanello et al., 2018), with theoretical work connecting repeated self-distillation to implicit regularization (Mobahi et al., 2020).

Learning from privileged context. Generalized distillation transfers training-only information via teacher supervision (Lopez-Paz et al., 2015; Choi et al., 2023; Shin et al., 2024). Concurrent work, β-OPSD (Xu et al., 2026), uses global reference–teacher interpolation to calibrate self-distillation. Our focus is complementary: guided by the Target-Locality Hypothesis, LGSD imposes a response-level KL budget and solves the interpolation coefficient per response by bisection. Thus, the distinction lies not in interpolation itself, but in the locality hypothesis and its response-adaptive constrained construction.

6 Conclusion

We introduced the *Target-Locality Hypothesis*: useful privileged correction can be retained without fully inheriting the privileged proposal's distributional shift. LGSD realizes this idea through response-adaptive local targets. Across frozen audits, reasoning benchmarks, and human-preference evaluation, local targets retain disproportionately more useful correction than distributional movement and can outperform both the initial student and full distillation. Our results highlight target construction as a key design choice in privileged self-distillation.

Reproducibility Statement

Anonymous links provide code, training and evaluation scripts, configurations, prompts, and example checkpoints. The appendices collect the projection derivations, experimental protocols, metric definitions, supplementary audits, systems measurements, and model interfaces accompanying the reported results.

AI Use Statement

Generative AI tools assisted with language editing, readability, and literature searches. The authors are responsible for reviewing these edits, checking sources, and verifying the final claims, results, and manuscript.

References

OpenAI Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Hai-Biao Bao, Boaz Barak, Ally Bennett, Tyler Bertao, N. Archer Brett, Eugene Brevdo, Greg Brockman, Sébastien Bubeck, Cheng Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, C. Dvorak, K. Fives, Vlad Fomenko, T. Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam P. Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominika Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran N. Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub W. Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, K. Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, Daniel Sculley, Harshit S. Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Phil Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy E. Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, K. Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Hu Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b&gpt-oss-20b model card. 2025. URL https://api.semanticscholar.org/CorpusID:280671456.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stańczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:263610088.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference. In *International Conference on Machine Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:268264163.

Eunbi Choi, Yongrae Jo, Joel Jang, Joonwon Jang, and Minjoon Seo. Fixed input parameterization for efficient prompting. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:259859012.

Imre Csiszár. i-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3:146–158, 1975. URL https://api.semanticscholar.org/CorpusID:18053591.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:4110009.

Yu-Xian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models. 2023. URL https://api.semanticscholar.org/CorpusID:259164722.

Jujie He, Jiacai Liu, Chris Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiao-Yu Zhang, Fu-Xiang Zhang, Jiacheng Xu, Wei Shen, Si-Yuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report. *ArXiv*, abs/2505.22312, 2025a. URL https://api.semanticscholar.org/CorpusID:278959882.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *ArXiv*, abs/2504.11456, 2025b. URL https://api.semanticscholar.org/CorpusID:277787455.

Hugging Face. Math-Verify: A robust mathematical expression evaluation system. https://github.com/huggingface/Math-Verify, 2024. Accessed 2026-08-13.

Ijun Jang, Je Won Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Annual Meeting of the Association for Computational Linguistics*, 2026. URL https://api.semanticscholar.org/CorpusID:284647873.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Ming-Yang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jin-Qiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *ArXiv*, abs/2604.02288, 2026. URL https://api.semanticscholar.org/CorpusID:287074130.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Naumovich Vapnik. Unifying distillation and privileged information. *CoRR*, abs/1511.03643, 2015. URL https://api.semanticscholar.org/CorpusID:8125776.

Mathematical Association of America. MAA mathematical competitions. https://maa.org/student-programs/amc/, 2025. Accessed 2026-08-03.

Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. Self-distillation amplifies regularization in hilbert space. *ArXiv*, abs/2002.05715, 2020. URL https://api.semanticscholar.org/CorpusID:211096976.

Le-Yi Pan, Shuchang Tao, Yun-Peng Zhai, Ling-Zhe Zhang, Zhaoyang Liu, Bolin Ding, Ai-Wei Liu, and Lijie Wen. Rlcsd: Reinforcement learning with contrastive on-policy self-distillation. *ArXiv*, abs/2606.11709, 2026. URL https://api.semanticscholar.org/CorpusID:289166712.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Christopher Callison-Burch. Styledistance: Stronger content-independent style embeddings with synthetic parallel examples. *ArXiv*, abs/2410.12757, 2024. URL https://api.semanticscholar.org/CorpusID:273375301.

B Rabern, Philipp Mondorf, and Barbara Plank. Logicskills: A structured benchmark for formal reasoning in large language models. *ArXiv*, abs/2602.06533, 2026. URL https://api.semanticscholar.org/CorpusID:285401876.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290, 2023. URL https://api.semanticscholar.org/CorpusID:258959321.

Igal Sason and Sergio Verdú. f -divergence inequalities. *IEEE Transactions on Information Theory*, 62:5973–6006, 2015. URL https://api.semanticscholar.org/CorpusID:7855260.

John Schulman, Sergey Levine, P. Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015. URL https://api.semanticscholar.org/CorpusID:16046818.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024. URL https://api.semanticscholar.org/CorpusID:267412607.

Haebin Shin, Lei Ji, Yeyun Gong, Sungdong Kim, Eunbi Choi, and Minjoon Seo. Generative prompt internalization. In *North American Chapter of the Association for Computational Linguistics*, 2024. URL https://api.semanticscholar.org/CorpusID:274234121.

Jiawei Xu, Minghui Liu, Juzheng Zhang, Tom Goldstein, and Furong Huang. β-opsd: Deriving with policy optimization, training with self-distillation, 2026. URL https://arxiv.org/abs/2607.28582.

An Yang, An-Feng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bo-Wen 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, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Ming-Ze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tian-Yi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Ying-Er Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhen-Ru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

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*, abs/2601.18734, 2026. URL https://api.semanticscholar.org/CorpusID:285050713.

Yanxiao Zhao, Yaqian Li, Zihao Bo, Rinyoichi Takezoe, Haojia Hui, Mo Guang, Lei Ren, Xiaolin Qin, and Kaiwen Long. Satquest: A verifier for logical reasoning evaluation and reinforcement fine-tuning of llms. In *Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://api.semanticscholar.org/CorpusID:281079725.

APPENDIX CONTENTS

A Privileged Context and Degradation **13**

B Supplementary Experiments **14**

C Strawman Test: Is the Gain Just a Decoding-Budget Effect? **16**

D Experimental Settings and Prompts **16**

E Closeness to the Global Optimum **19**

F From target locality to distillation gradients **21**

A PRIVILEGED CONTEXT AND DEGRADATION

Privileged-context screening. We screen the original instruction and four alternatives on frozen GPT-OSS-20B, using the same audited DeepMath subset, paired decoding seeds, and fixed generation budget. All settings receive the same problems and answer-formatting request, without reference answers, solutions, or outcome feedback. Base uses medium reasoning effort without an additional system instruction; effort remains medium unless otherwise indicated.

Table 5: GPT-OSS-20B context screening. ✓: observed net accuracy gain over Base; ✗: no net gain. Marks summarize prompted generation, not trained-student performance or statistical significance.

Context	Guidance summary	Gain
`current`	Decompose into subgoals, track constraints and invariants, check boundary cases, and verify through an alternative route when possible.	✗
`verify`	Check calculations, original constraints, signs, integer restrictions, extraneous roots, and missing cases; correct errors without restarting a verified solution.	✓
`concise`	Use a short solution with exact arithmetic; avoid revisiting settled steps, change methods once if needed, and reserve space for checking and stating the answer.	✗
`high`	Increase reasoning effort to high without additional guidance.	✗
`high_verify`	Combine the verification instruction with high reasoning effort.	✗

Additional guidance is not uniformly beneficial. Only `verify` produces a small observed accuracy gain. The `current` and `concise` settings tie Base, while both high-effort settings reduce accuracy and increase budget exhaustion. Prompted accuracy does not directly measure token-level target quality, but these findings caution against treating the full privileged shift as uniformly useful. Repeated self-distillation may transfer incidental behavioral changes alongside useful corrections, with these changes accumulating as the updated student also generates subsequent proposals. This is a plausible explanation for degradation, not a causal conclusion from the screen alone.

Implications for the main comparison. Table 1 uses the original `current` instruction for both methods, isolating target construction under matched answer-free supervision. On GPT-OSS-20B, LGSD improves over Base at episode 16 but falls below it at episode 64, while substantially mitigating the late degradation observed with OPSD. Lower fixed-prefix drift and greater entropy retention are consistent with this stabilizing role (Table 7). Locality limits target divergence; it does not verify the guidance or prevent cumulative drift as the student anchor evolves. The endpoint result therefore supports reduced degradation relative to matched OPSD, rather than guaranteed improvement over Base.

B SUPPLEMENTARY EXPERIMENTS

B.1 TARGET-SPACE AUDITS

Table 6 reports the radius sweep; the allocation comparison in Table 2 tests how a response-level budget distributes movement across tokens.

Table 6: **Supplementary target-space evidence.** Radius sweep with answer-free selection diagnostics and a separate post-selection Gold/raw audit.

Answer-free radius selection

Radius	Mean α	Target KL	Gold/raw ↑	Style/raw ↓	Prompt/raw ↓	Active budgets
$\varepsilon = .001$	.3225	.001000	1.027	.316	.097	3/3
$\varepsilon = .002$	.4726	.002000	1.244	.464	.209	3/3
$\varepsilon = .004$	**.7022**	**.004000**	**1.319**	**.693**	**.467**	**3/3**
$\varepsilon = .008$	.9955	.007230	1.013	.995	.980	1/3
$\varepsilon \geq .016$	1.0000	.007290	1.000	1.000	1.000	0/3

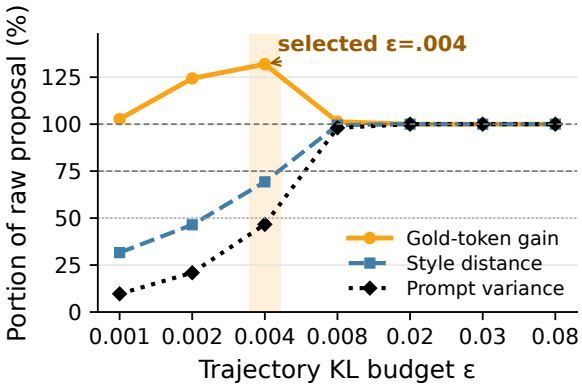

Figure 8: **Answer-free radius selection.**

Radius selection. We select $\varepsilon = .004$ as the largest tested radius active for all three equivalent wrappers. At .008, only one remains active; at .016 and above, projection is inactive. At the selected radius, style movement and wrapper sensitivity are $.693\times$ and $.467\times$ their raw-proposal values. The subsequent gold-token audit finds $1.319\times$ the raw gain, supporting P1. Gold-token outcomes do not enter radius selection.

B.2 LEARNED-POLICY AND COMPLETION AUDITS

The following audits use the episode-64 checkpoints from Table 1 to examine completion efficiency and sensitivity to decoding choices.

Table 7: **Supplementary learned-policy diagnosis.** Both backbones use the episode-64 checkpoints from Table 1. GPT-OSS-20B rows report supplementary reevaluations on the same 143 questions; the budget audit aggregates these repeated evaluations.

Audit	OPSD	LGSD
Qwen3-8B completion behavior		
Math Acc@1 (%)	62.94	**73.43**
Budget-cap rate (%)	30.07	**17.48**
Mean generated tokens	6,318	**5,986**
GPT-OSS-20B fixed-checkpoint collapse diagnosis		
Acc@1, seeds 09/10/11 (%)	19.6 / 0.7 / 14.7	**57.3 / 60.8 / 54.5**
Cap-hit rate, seeds 09/10/11 (%)	32.2 / 76.9 / 29.4	**0.0–0.7**
Acc@1, 10,240 → 20,480 tokens (%)	11.7 → 22.8	**57.6 → 61.1**
$D_{\mathrm{KL}}(\pi_{64} \Vert \pi_0)$	.1834	**.0966**
Entropy retention, $H(\pi_{64})/H(\pi_0)$	.730	**.891**

Table 8: Decoding-seed robustness. Fixed checkpoints; three repeats.

Episode	OPSD	LGSD	Δ (pp)
16	59.21 ± 1.07	59.21 ± 1.61	$+0.00$
32	57.81 ± 3.59	60.14 ± 2.10	$+2.33$
48	62.70 ± 2.46	63.87 ± 2.25	$+1.17$
64	65.27 ± 1.07	69.23 ± 0.70	$+3.96$

Table 9: Training-seed reproducibility. Fixed decoding; three repeats.

Episode	OPSD	LGSD	Δ (pp)
16	59.44 ± 1.21	58.28 ± 1.76	-1.17
32	58.51 ± 2.46	61.77 ± 2.83	$+3.26$
48	61.77 ± 1.46	65.27 ± 1.07	$+3.50$
64	64.57 ± 1.61	70.86 ± 2.46	$+6.29$

Qwen3-8B achieves higher accuracy with fewer cap hits and fewer generated tokens. For GPT-OSS-20B, repeated decodes and longer generation budgets preserve a substantial accuracy separation; fixed-prefix KL and entropy measurements support a learned-policy difference associated with raw-target degradation.

B.3 Repeated Evaluation and Training

Decoding seeds. We reevaluate each fixed Qwen3-8B checkpoint using three decoding seeds on the same 143 questions. The methods tie in mean accuracy at episode 16. LGSD leads by 2.33, 1.17, and 3.96 percentage points at episodes 32, 48, and 64, respectively.

Training seeds. We repeat training with three seeds while holding the pretrained checkpoint, training-example order, and other settings fixed. All runs share one fixed decoding seed. LGSD trails OPSD by 1.17 points at episode 16, then leads by 3.26, 3.50, and 6.29 points at episodes 32, 48, and 64. These results show higher mean accuracy at later checkpoints, with gains that vary across training progress.

Tables 8 and 9 report accuracy (%) as mean ± sample standard deviation across three repeats. Δ denotes LGSD minus OPSD in percentage points.

B.4 Systems Measurements

Table 10: **Training overhead.** Aggregate over two instrumented runs.

Measure	OPSD mean	LGSD mean	Ratio
Active time (H100-h)	4.6832	4.4596	$0.9523\times$
Checkpoint (MiB)	348.81365	348.81365	$1.0000\times$
Peak memory (GiB)	22.3429	22.3401	$0.9999\times$

Training cost. The projection solves one shared scalar per response and adds no model forward pass or trainable module. Across two instrumented runs, mean active time is 4.4596 versus 4.6832 H100-hours, checkpoint sizes are identical, and peak memory is effectively unchanged (Table 10). These measurements show that projection fits within the existing training cost.

B.5 Training Dynamics and Endpoint Audits

Supplementary collapse audit. Table 7 examines the GPT-OSS-20B endpoint under three decoding seeds, a doubled output budget, and fixed-prefix policy diagnostics. LGSD retains a large accuracy advantage across seeds and budgets. On the fixed prefixes, it also has lower KL from initialization and higher entropy retention. The combined evidence links the endpoint separation to a learned-policy difference that persists across evaluation conditions.

B.6 Multi-sample mathematical reasoning

We complement the single-response evaluation with eight sampled responses per question for Qwen3-8B and its adapted checkpoints. Pass@8 measures the fraction of questions for which at least one of the eight responses is correct. Table 11 reports results on AMC 2022/2023 (83 questions), AIME 2024 (30 questions), and AIME 2025 (30 questions). The aggregate score weights all 143 questions equally rather than averaging the three benchmark percentages.

Method	Ep.	AMC 22/23	AIME 2024	AIME 2025	Math All
Qwen3-8B					
Base	0	89.16	73.33	66.67	81.12
OPSD	16	92.77	73.33	56.67	81.12
Veto	16	84.34	73.33	60.00	76.92
LGSD	16	91.57	73.33	60.00	81.12
OPSD	64	91.57	80.00	66.67	83.92
Veto	64	92.77	76.67	73.33	85.31
LGSD	64	93.98	83.33	76.67	88.11

Table 11: **Mathematical reasoning with eight samples (pass@8, %).** Math All aggregates correctness over all 143 questions.

C Strawman Test: Is the Gain Just a Decoding-Budget Effect?

A natural strawman explanation is that LGSD performs better simply because raw OPSD generates longer responses and hits the decoding limit more often. Under the standard 10,240-token budget, OPSD reaches 62.94% accuracy with a 30.07% cap-hit rate, while LGSD reaches 73.43% with a 17.48% cap-hit rate. Since an unfinished response without a verifiable final answer is counted as incorrect, part of the gap could come from truncation rather than better reasoning.

We test this by reevaluating the same episode-64 checkpoints with a doubled 20,480-token budget. This substantially improves OPSD from 62.94% to 71.33% and reduces its cap-hit rate to **0%**. However, LGSD still reaches 79.02%, retaining a **7.69-point advantage**. Therefore, decoding truncation explains part of the original gap, but cannot explain the LGSD gain.

Table 12: **Decoding-budget strawman test on Qwen3-8B at episode 64.** Even when raw OPSD has no cap hits under a doubled budget, LGSD remains 7.69 points more accurate.

Method	**Budget**	**AMC22/23**	**AIME24**	**AIME25**	**Acc@1**	**Cap hit**
OPSD	10k	77.11	56.67	30.00	62.94	30.07
LGSD	10k	83.13	70.00	50.00	**73.43**	17.48
OPSD	20k	78.31	66.67	56.67	71.33	**0.00**
LGSD	20k	85.54	70.00	70.00	**79.02**	9.79

D Experimental Settings and Prompts

Experiments use NVIDIA H100, H200, and B200 GPUs. Matched OPSD–LGSD runs share initialization, ordered examples, generation settings, optimizer, adapters, and vocabulary support. The systems comparison uses the same H100 instrumentation for both methods.

D.1 Human-Preference Evaluation

The Qwen3-8B preference study compares Base, four LGSD radii (.001, .004, .016, .040), and matched raw OPSD after 1,000 optimizer steps. Runs share training conditions and the same Arena human-voted evaluation pairs. Each policy teacher-forces the preferred and rejected responses. With N evaluated pairs, define

$$s_M(x,y) = \frac{1}{|y|}\sum_t \log \pi_M(y_t \mid x, y_{<t}),$$

$$m_{M,i} = s_M(x_i, y_i^+) - s_M(x_i, y_i^-), \qquad P_M = \frac{1}{N}\sum_{i=1}^{N} m_{M,i}. \tag{11}$$

Figure 5 reports pair-level margins and $P_M \pm \mathrm{SEM}(P_M)$, where $\mathrm{SEM}(P_M) = \mathrm{SD}_i(m_{M,i})/\sqrt{N}$. PrefMargin measures likelihood assigned to recorded human choices. The token-weighted mean α_{tok} weights each realized response token equally. Preference outcomes evaluate the sweep and do not select the radius used for reasoning experiments.

D.2 Data and Training Progress

Training uses a frozen, conflict-filtered stream of DeepMath-103K level-7–10 problems. Duplicate normalized texts are grouped, conflicting answer–solution records are removed, and canonical records follow one deterministic order; shorter runs use prefixes of that order. The 64-episode Qwen3-8B run contains 3,413 optimizer steps, averaging 53.3 steps per episode.

Held-out mathematics covers AMC 2022/2023, AIME 2024, and AIME 2025. The internal split name "AMC23" contains both 2022 and 2023 competitions. Training, development, and held-out mathematical sets are disjoint under exact and normalized problem text. SATQuest and LogicSkills are evaluation-only: neither enters training or radius selection.

D.3 Matched Configuration

Table 13: **Core configuration for the matched comparison.**

Component	Setting
Backbones	Qwen3-1.7B, Qwen3-8B, and GPT-OSS-20B
Adaptation	LoRA rank 8, scale 16; matched module selection within each backbone
Optimizer	AdamW, learning rate 2×10^{-5}, gradient norm 1.0, BF16
Sampling	Temperature .6, top-p .95, top-k 20
LGSD radius	The reasoning study uses global $\varepsilon = .004$; the preference study uses .001, .004, .016, and .040 (Small–High), plus raw OPSD
Projection	Full-vocabulary geometric path with one trajectory-shared α and six bisection steps
Distillation loss	Full-vocabulary target-to-student KL with a detached target
Math decoding	One matched-seed response per query with a 10,240-token budget
Logic decoding	Greedy Pass@1 with benchmark-specific formal verification

The current model scores each sampled prefix under ordinary and answer-free privileged contexts. Matched OPSD fits the raw proposal; LGSD fits its projected, detached target through the same forward KL (Appendix B.1). Learning rate, gradient-norm clipping, adapter size, and update scale remain matched.

Veto. The student-confidence penalty coefficient starts at $\beta = .8$ and decays linearly to zero, following the mathematical-reasoning setting of Jang et al. (2026). Penalized teacher log scores are normalized and fitted through target-to-student KL under the common protocol.

D.4 Repeated Runs and Systems Measurement

The decoding audit reevaluates each fixed Qwen3-8B checkpoint three times on the same 143 questions. The training audit repeats the complete matched run three times, holding other settings and example order fixed. These tables report sample standard deviations across evaluations or training runs, respectively. Active H100 time, serialized checkpoint size, and peak allocated memory use identical instrumentation.

D.5 Verification and Audit Metrics

Verifier-based correctness. Math-Verify checks boxed AMC/AIME answers. SATQuest uses PySAT, and LogicSkills uses task-specific PySAT or Z3 verification. Malformed outputs or responses without a verifiable final answer are counted as incorrect. We report Math Acc@1 and Logic Pass@1 as query-level correctness.

Target-space audits. For a target q and student p, target locality is

$$\overline{D}(q\|p) = \frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}(q_t\|p_t). \tag{12}$$

Gold/raw is the reference-token gain of q over p, normalized by the gain of the raw proposal q^P. It is used only after radius selection. Prompt variance measures target sensitivity across equivalent privileged instructions on identical prefixes.

Forward versus reverse KL. Reverse KL, $D_{\mathrm{KL}}(s_\theta \| q)$, is student-weighted and more mode-seeking. This makes it natural for on-policy distillation methods that optimize from student-generated samples. Forward KL, $D_{\mathrm{KL}}(q \| s_\theta)$, is teacher-weighted and directly fits the teacher distribution when its full vocabulary probabilities are available.

OPSD uses forward KL because the privileged self-teacher provides a dense distribution on every realized prefix. LGSD keeps the same choice:

$$\mathcal{L}_{\mathrm{LGSD}} = \frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}}\left(q_t^C \,\|\, \pi_\theta(\cdot \mid x, y_{<t})\right). \tag{13}$$

Thus, matched OPSD and LGSD use the same distillation direction and training pipeline; they differ only in the supervision target, q^P versus q^C.

Endpoint policy audits. The GPT-OSS-20B audits use the same episode-64 checkpoints and 143 held-out questions. The decoding-budget test changes only the generation limit from 10,240 to 20,480 tokens. Policy movement is measured by $D_{\mathrm{KL}}(\pi_{64} \| \pi_0)$ on fixed prefixes, and entropy retention is the ratio of endpoint to initialization entropy. Table 7 reports the corresponding point estimates.

D.6 Exact prompts and model interfaces

D.6.1 Training interface

Ordinary student: user message

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

Privileged scorer: training-only system message

```
Private reasoning-method instruction for the teacher:  decompose the
problem into explicit subgoals, track constraints and invariants,
check boundary cases, and verify the chosen route against an
independent alternative when possible.  Use only the problem
statement.
```

The scorer receives the same user message and realized response prefix as the ordinary student.

D.6.2 Held-out mathematics interface

```
{problem}
Please reason step by step.  You have a strict response budget of at
most 10,240 generated tokens.  Finish the reasoning and put the final
answer within \boxed{} before reaching that limit; prioritize
completing the answer over extending the analysis.
```

D.6.3 Logical evaluation interfaces

SATQuest receives the benchmark's requested Boolean assignment or satisfiability decision in its canonical machine-readable format. LogicSkills receives the task-provided interface for validity, symbolization, or finite countermodels. Decoding is greedy, no mathematical privileged instruction is supplied, and the extracted object is checked by the benchmark-specific PySAT or Z3 verifier.

D.6.4 Robustness-audit instructions

The prompt-robustness audit replaces only the privileged system message. The terse control asks for the shortest direct solution with minimal exposition. The verbose control asks for explicit decomposition, checks, and alternative routes. The equivalent-wrapper family paraphrases the main reasoning-method instruction without adding answers, reference solutions, or task-specific hints. All variants score the same fixed prefixes, making changes in the resulting target attributable to the privileged interface rather than to different sampled trajectories.

D.6.5 EXAMPLE COMPARISON

A completed AMC example: retaining the remainder's $+x$

Qwen3-8B, episode 64; the same query and decoding settings. Both responses finish naturally within the 10,240-token budget.

Problem

Let P be the polynomial of least degree with rational coefficients satisfying

$$P \equiv x+2 \pmod{x^2+x+1}, \qquad P \equiv 2x+1 \pmod{x^2+1}.$$

What is the sum of the squared coefficients of P?

LGSD: correct *3,585 generated tokens*

The expansion retains the contribution from the remainder:

$$\begin{aligned} P &= (x^2+x+1)(mx+n)+x+2 \\ &= mx^3+(m+n)x^2+(m+n\mathbf{+1})x+(n+2). \end{aligned}$$

Comparing with $P=(x^2+1)(px+q)+2x+1$ gives $m=n=1$ and $q=2$. Thus,

$$P = x^3+2x^2+3x+3, \qquad 1^2+2^2+3^2+3^2 = \boxed{23}.$$

OPSD: incorrect *5,156 generated tokens*

Starting from $P=(x^2+x+1)(mx+n)+x+2$, the response writes the coefficient of x as

$$\underbrace{m+n}_{\text{used by OPSD}} \qquad \text{instead of} \qquad \underbrace{m+n+1}_{\text{correct coefficient}},$$

omitting the $+x$ contributed by the remainder. The response subsequently reports

$$P = x^3+3x^2+3x+4, \qquad 1^2+3^2+3^2+4^2 = \boxed{35}.$$

Observation. Both responses complete, but OPSD makes a coefficient-expansion error that LGSD avoids. Truncation therefore does not explain the correctness difference in this example.

Presentation and decoding. Derivations are condensed summaries, not verbatim transcripts. One response per query; temperature 0.6, top-p 0.95, top-k 20, and seed $54{,}486$.

E CLOSENESS TO THE GLOBAL OPTIMUM

We compare the raw proposal and the LGSD target with the same global optimum. Fix T response prefixes, a finite vocabulary V, and distributions $p_t(v), q_t^P(v) > 0$. Let $\mathcal{P} = \prod_{t=1}^{T} \Delta(V)$ contain all token distributions on these prefixes, and write $D_{\mathrm{KL}}(a\|b) = \sum_{v\in V} a(v)\log[a(v)/b(v)]$. For fixed finite rewards $R_t(v)$ and $\beta > 0$, consider

$$\mathcal{J}(q) = \frac{1}{T}\sum_t \mathbb{E}_{q_t} R_t - \frac{\beta}{T}\sum_t D_{\mathrm{KL}}(q_t\|p_t), \qquad q \in \mathcal{P}. \tag{14}$$

The reward-tilted distribution (Rafailov et al., 2023) is

$$q_t^\star(v) = \frac{p_t(v)\exp(R_t(v)/\beta)}{Z_t^\star}, \qquad Z_t^\star = \sum_{u\in V} p_t(u)\exp(R_t(u)/\beta). \tag{15}$$

To verify global optimality, use $\log(q_t^\star/p_t) = R_t/\beta - \log Z_t^\star$. For every $q \in \mathcal{P}$,

$$\begin{aligned} \frac{\beta}{T}\sum_t D_{\mathrm{KL}}(q_t\|q_t^\star) &= \frac{\beta}{T}\sum_{t,v} q_t(v)\left[\log\frac{q_t(v)}{p_t(v)} - \frac{R_t(v)}{\beta} + \log Z_t^\star\right] \\ &= \frac{\beta}{T}\sum_t D_{\mathrm{KL}}(q_t\|p_t) - \frac{1}{T}\sum_t \mathbb{E}_{q_t} R_t + \frac{\beta}{T}\sum_t \log Z_t^\star \\ &= \frac{\beta}{T}\sum_t \log Z_t^\star - \mathcal{J}(q). \end{aligned} \tag{16}$$

Setting $q = q^\star$ gives $\mathcal{J}(q^\star) = \beta T^{-1} \sum_t \log Z_t^\star$. Therefore,

$$\begin{aligned} \mathcal{J}(q^\star) - \mathcal{J}(q) &= \frac{\beta}{T} \sum_t D_{\mathrm{KL}}(q_t \| q_t^\star) \geq 0, \\ q^\star &= \arg\max_{q \in \mathcal{P}} \mathcal{J}(q). \end{aligned} \tag{17}$$

Equality holds only at $q = q^\star$. Thus, $q^\star$ is the unique global maximizer over all distributions in $\mathcal{P}$, independently of the locality budget. Privileged conditioning need not produce this distribution: its implicit reward is $\widetilde{R}_t(v) = \beta \log[q_t^P(v)/p_t(v)]$, which induces $q_t^\star$ only when $\widetilde{R}_t - R_t$ is constant across tokens. These rewards define the comparison and are not required by LGSD. We now prove that projection brings the target closer to this global optimum under the condition

$$0 < \varepsilon, \qquad \frac{1}{T} \sum_t D_{\mathrm{KL}}(q_t^\star \| p_t) \leq \varepsilon < \frac{1}{T} \sum_t D_{\mathrm{KL}}(q_t^P \| p_t). \tag{18}$$

This condition does not redefine the optimum within a smaller search space. It states that the global optimum lies within the budget, whereas the raw proposal does not. Consider the geometric path

$$q_{t,\alpha}(v) = \frac{p_t(v)^{1-\alpha} q_t^P(v)^\alpha}{Z_t(\alpha)}, \qquad Z_t(\alpha) = \sum_{u \in V} p_t(u)^{1-\alpha} q_t^P(u)^\alpha, \qquad \alpha \in [0,1]. \tag{19}$$

Let $a_t(v) = \log[q_t^P(v)/p_t(v)]$ and $\psi(\alpha) = T^{-1} \sum_t \log Z_t(\alpha)$. Since $Z_t(\alpha) = \sum_v p_t(v) e^{\alpha a_t(v)}$, differentiation gives $\psi'(\alpha) = T^{-1} \sum_t \mathbb{E}_{q_{t,\alpha}} a_t$ and $\psi''(\alpha) = T^{-1} \sum_t \mathrm{Var}_{q_{t,\alpha}}(a_t)$. Consequently,

$$\begin{aligned} \frac{1}{T} \sum_t D_{\mathrm{KL}}(q_{t,\alpha} \| p_t) &= \alpha \psi'(\alpha) - \psi(\alpha), \\ \frac{d}{d\alpha} \left[\frac{1}{T} \sum_t D_{\mathrm{KL}}(q_{t,\alpha} \| p_t) \right] &= \alpha \psi''(\alpha). \end{aligned} \tag{20}$$

The proposal differs from p at some position, so full support implies $\psi''(\alpha) > 0$. The radius is therefore continuous and strictly increasing on $[0,1]$, from zero to $\frac{1}{T} \sum_t D_{\mathrm{KL}}(q_t^P \| p_t)$. Equation 18 ensures a unique $\alpha \in (0,1)$ satisfying $\frac{1}{T} \sum_t D_{\mathrm{KL}}(q_{t,\alpha} \| p_t) = \varepsilon$. The following identity both establishes the projection and compares it with the global optimum. For any $r \in \mathcal{P}$, expanding the logarithm of Equation 19 gives

$$\begin{aligned} \frac{1}{T} \sum_t D_{\mathrm{KL}}(r_t \| q_{t,\alpha}) &= \frac{1}{T} \sum_{t,v} r_t(v) \left[(1-\alpha) \log \frac{r_t(v)}{p_t(v)} + \alpha \log \frac{r_t(v)}{q_t^P(v)} + \log Z_t(\alpha) \right] \\ &= \frac{1-\alpha}{T} \sum_t D_{\mathrm{KL}}(r_t \| p_t) + \frac{\alpha}{T} \sum_t D_{\mathrm{KL}}(r_t \| q_t^P) + \psi(\alpha). \end{aligned} \tag{21}$$

Setting $r = q_\alpha$ makes the left-hand side zero. Hence $\psi(\alpha) = -T^{-1} \sum_t [(1-\alpha) D_{\mathrm{KL}}(q_{t,\alpha} \| p_t) + \alpha D_{\mathrm{KL}}(q_{t,\alpha} \| q_t^P)]$. Substituting this expression back yields

$$\begin{aligned} \frac{1}{T} \sum_t D_{\mathrm{KL}}(r_t \| q_{t,\alpha}) = \frac{\alpha}{T} \sum_t \big[D_{\mathrm{KL}}(r_t \| q_t^P) - D_{\mathrm{KL}}(q_{t,\alpha} \| q_t^P) \big] \\ + \frac{1-\alpha}{T} \sum_t \big[D_{\mathrm{KL}}(r_t \| p_t) - D_{\mathrm{KL}}(q_{t,\alpha} \| p_t) \big]. \end{aligned} \tag{22}$$

Since $\frac{1}{T} \sum_t D_{\mathrm{KL}}(q_{t,\alpha} \| p_t) = \varepsilon$, every feasible r satisfies

$$\begin{aligned} &\frac{1}{T} \sum_t \big[D_{\mathrm{KL}}(r_t \| q_t^P) - D_{\mathrm{KL}}(q_{t,\alpha} \| q_t^P) \big] \\ &\quad = \frac{1}{\alpha T} \sum_t D_{\mathrm{KL}}(r_t \| q_{t,\alpha}) + \frac{1-\alpha}{\alpha} \left[\varepsilon - \frac{1}{T} \sum_t D_{\mathrm{KL}}(r_t \| p_t) \right] \geq 0. \end{aligned} \tag{23}$$

Both terms are nonnegative, and equality requires $r = q_\alpha$. Thus, q_α is exactly the unique LGSD projection q^C. This is a direct instance of KL-projection geometry (Csiszár, 1975). Now set $r = q^\star$

in Equation 22. The global optimum satisfies $\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|p_t) \le \varepsilon$, so

$$\begin{aligned}\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^C) &= \frac{\alpha}{T}\sum_t \big[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(q_t^C\|q_t^P)\big] \\ &\quad + (1-\alpha)\left[\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|p_t) - \varepsilon\right] \\ &\le \frac{\alpha}{T}\sum_t \big[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(q_t^C\|q_t^P)\big] \\ &\le \frac{\alpha}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P) < \frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P).\end{aligned} \tag{24}$$

The final inequality is strict because $\alpha < 1$ and $q^\star \neq q^P$: one is feasible and the other is not. This proves Proposition 2. Equivalently,

$$\frac{1}{T}\sum_t \mathbb{E}_{q_t^\star} \log \frac{q_t^C}{q_t^P} = \frac{1}{T}\sum_t \big[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(q_t^\star\|q_t^C)\big] > 0. \tag{25}$$

Thus, LGSD assigns higher expected log-probability than the raw proposal to tokens drawn from the globally optimal distribution. The improvement concerns agreement with $q^\star$, not merely closeness to the student or fidelity to the proposal. It is measured in $\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|\cdot)$; the objective gap in Equation 17 uses the opposite KL direction. For the numerical implementation, let $\widehat{q}^C = q_{\widehat{\alpha}}$ be the feasible bisection approximation, with $0 \le \widehat{\alpha} \le \alpha < 1$, and define $\delta = \varepsilon - \frac{1}{T}\sum_t D_{\mathrm{KL}}(\widehat{q}_t^C\|p_t) \ge 0$. Equation 22 holds for every interpolation coefficient. Substituting $\widehat{\alpha}$ and using $\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|p_t) - \frac{1}{T}\sum_t D_{\mathrm{KL}}(\widehat{q}_t^C\|p_t) \le \delta$ therefore gives

$$\begin{aligned}\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|\widehat{q}_t^C) &\le \frac{\widehat{\alpha}}{T}\sum_t \big[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(\widehat{q}_t^C\|q_t^P)\big] + (1-\widehat{\alpha})\delta \\ &\le \frac{\widehat{\alpha}}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P) + (1-\widehat{\alpha})\delta.\end{aligned} \tag{26}$$

Hence $\delta < \frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P)$ is sufficient for the numerical target to remain strictly closer to the global optimum. If $q^\star$ lies within the achieved radius $\frac{1}{T}\sum_t D_{\mathrm{KL}}(\widehat{q}_t^C\|p_t)$, the slack term can instead be omitted, recovering the exact contraction bound.

F From target locality to distillation gradients

Fix T realized response prefixes and a finite vocabulary V. Assume $p_t(v), q_t^P(v) > 0$, and use natural logarithms throughout. As in the main text, write

$$D_{\mathrm{KL}}(a\|b) = \sum_{v\in V} a(v)\log\frac{a(v)}{b(v)}.$$

The reference $q^\star$ is the reward-optimal distribution defined above; it is used only for analysis and is not required to construct the target. All gradients below concern the unclipped forward-KL loss with fixed targets and prefixes.

Lemma 1 (Pinsker's inequality (Sason & Verdú, 2015))**.** *For probability distributions a, b on V,*

$$\|a-b\|_1^2 \le 2D_{\mathrm{KL}}(a\|b). \tag{27}$$

Consequently, for distributions on the fixed response prefixes,

$$\frac{1}{T}\sum_{t=1}^{T}\|a_t-b_t\|_1^2 \le \frac{2}{T}\sum_t D_{\mathrm{KL}}(a_t\|b_t). \tag{28}$$

Forward-KL gradients. Let $s_t = \mathrm{softmax}(z_t)$ be the trainable student's distribution at position t. For a fixed target q_t,

$$\ell_t(q; z_t) = D_{\mathrm{KL}}(q_t\|s_t) = \sum_v q_t(v)\log q_t(v) - \sum_v q_t(v)z_t(v) + \log\sum_v e^{z_t(v)}.$$

Differentiation therefore gives

$$\nabla_{z_t} \ell_t(q; z_t) = s_t - q_t.$$

At the rollout student $s_t = p_t$, define

$$g_t(q) := p_t - q_t.$$

For the same student and prefixes, the difference from the gradient under reference supervision is exactly

$$g_t(q) - g_t(q^\star) = q_t^\star - q_t. \tag{29}$$

Thus, target error transfers directly into logit-gradient error. Define its response-averaged squared magnitude as

$$\mathcal{E}(q) := \frac{1}{T} \sum_{t=1}^{T} \|g_t(q) - g_t(q^\star)\|_1^2.$$

Proposition 4 (Bounds on the distillation signal)**.** *Any target q^C satisfying $\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^C \| p_t) \le \varepsilon$ obeys*

$$\frac{1}{T} \sum_{t=1}^{T} \|g_t(q^C)\|_1^2 \le 2\varepsilon. \tag{30}$$

Moreover, suppose

$$0 < \varepsilon, \qquad \frac{1}{T} \sum_t D_{\mathrm{KL}}(q_t^\star \| p_t) \le \varepsilon < \frac{1}{T} \sum_t D_{\mathrm{KL}}(q_t^P \| p_t). \tag{31}$$

For the exact LGSD projection $q^C = q_\alpha$, with $\alpha \in (0, 1)$,

$$\begin{aligned} \mathcal{E}(q^C) &\le \frac{2}{T} \sum_t D_{\mathrm{KL}}(q_t^\star \| q_t^C) \\ &\le \frac{2\alpha}{T} \sum_t \left[D_{\mathrm{KL}}(q_t^\star \| q_t^P) - D_{\mathrm{KL}}(q_t^C \| q_t^P) \right] \\ &\le \frac{2\alpha}{T} \sum_t D_{\mathrm{KL}}(q_t^\star \| q_t^P). \end{aligned} \tag{32}$$

Proof. For the magnitude bound, apply Lemma 1 with $a_t = q_t^C$ and $b_t = p_t$:

$$\frac{1}{T} \sum_t \|g_t(q^C)\|_1^2 = \frac{1}{T} \sum_t \|p_t - q_t^C\|_1^2 \le \frac{2}{T} \sum_t D_{\mathrm{KL}}(q_t^C \| p_t) \le 2\varepsilon.$$

For the error bound, Equation 29 and Pinsker give

$$\mathcal{E}(q^C) = \frac{1}{T} \sum_t \|q_t^\star - q_t^C\|_1^2 \le \frac{2}{T} \sum_t D_{\mathrm{KL}}(q_t^\star \| q_t^C). \tag{33}$$

It remains to compare the projected and raw targets in this KL direction.

Consider the geometric path

$$q_{t,s}(v) = \frac{p_t(v)^{1-s} q_t^P(v)^s}{Z_t(s)}, \qquad Z_t(s) = \sum_{u \in V} p_t(u)^{1-s} q_t^P(u)^s,$$

and define $\psi(s) = T^{-1} \sum_t \log Z_t(s)$. Expanding the logarithm yields, for any reference distribution r,

$$\frac{1}{T} \sum_t D_{\mathrm{KL}}(r_t \| q_{t,s}) = \frac{1-s}{T} \sum_t D_{\mathrm{KL}}(r_t \| p_t) + \frac{s}{T} \sum_t D_{\mathrm{KL}}(r_t \| q_t^P) + \psi(s). \tag{34}$$

Setting $r = q_s$ makes the left-hand side zero, so

$$\psi(s) = -\frac{1-s}{T} \sum_t D_{\mathrm{KL}}(q_{t,s} \| p_t) - \frac{s}{T} \sum_t D_{\mathrm{KL}}(q_{t,s} \| q_t^P).$$

Substituting this expression into Equation 34 gives

$$\begin{aligned}\frac{1}{T}\sum_t D_{\mathrm{KL}}(r_t\|q_{t,s}) = \frac{s}{T}\sum_t \left[D_{\mathrm{KL}}(r_t\|q_t^P) - D_{\mathrm{KL}}(q_{t,s}\|q_t^P)\right] \\ + \frac{1-s}{T}\sum_t \left[D_{\mathrm{KL}}(r_t\|p_t) - D_{\mathrm{KL}}(q_{t,s}\|p_t)\right].\end{aligned} \tag{35}$$

Under Equation 31, the exact projection has an active constraint: $\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_{t,\alpha}\|p_t) = \varepsilon$. Taking $r = q^\star$ and $s = \alpha$ in Equation 35 therefore gives

$$\begin{aligned}\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^C) &= \frac{\alpha}{T}\sum_t \left[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(q_t^C\|q_t^P)\right] \\ &\quad + (1-\alpha)\left[\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|p_t) - \varepsilon\right] \\ &\le \frac{\alpha}{T}\sum_t \left[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(q_t^C\|q_t^P)\right] \\ &\le \frac{\alpha}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P).\end{aligned}$$

The first inequality uses reference feasibility; the second uses nonnegativity of KL. Combining this result with Equation 33 completes the proof. □

Comparison with the raw-target bound. The same Pinsker argument gives

$$\mathcal{E}(q^P) \le \frac{2}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P).$$

Equation 32 therefore bounds the local target's error by α times this raw-target upper bound. This compares upper bounds: it does not establish $\mathcal{E}(q^C) \le \alpha\mathcal{E}(q^P)$. The conclusion is conditional on Equation 31, not on an assumed ability to distinguish useful corrections from nuisances.

Finite-bisection correction. Let $\widehat{q}^C = q_{\widehat{\alpha}}$ be the feasible numerical target, where $0 \le \widehat{\alpha} \le \alpha < 1$, and define the unused budget

$$\delta := \varepsilon - \frac{1}{T}\sum_t D_{\mathrm{KL}}(\widehat{q}_t^C\|p_t) \ge 0.$$

The magnitude bound remains valid and can be sharpened to

$$\frac{1}{T}\sum_t \|g_t(\widehat{q}^C)\|_1^2 \le 2(\varepsilon - \delta).$$

For the reference error, apply Equation 35 with $r = q^\star$ and $s = \widehat{\alpha}$. Since

$$\frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|p_t) - \frac{1}{T}\sum_t D_{\mathrm{KL}}(\widehat{q}_t^C\|p_t) \le \delta,$$

Pinsker gives

$$\begin{aligned}\mathcal{E}(\widehat{q}^C) &\le \frac{2}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|\widehat{q}_t^C) \\ &\le \frac{2\widehat{\alpha}}{T}\sum_t \left[D_{\mathrm{KL}}(q_t^\star\|q_t^P) - D_{\mathrm{KL}}(\widehat{q}_t^C\|q_t^P)\right] + 2(1-\widehat{\alpha})\delta \\ &\le \frac{2\widehat{\alpha}}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P) + 2(1-\widehat{\alpha})\delta.\end{aligned} \tag{36}$$

Thus, $\delta < \frac{1}{T}\sum_t D_{\mathrm{KL}}(q_t^\star\|q_t^P)$ is sufficient for this upper bound to remain below the corresponding raw-target Pinsker bound. If $q^\star$ lies within the achieved radius $\frac{1}{T}\sum_t D_{\mathrm{KL}}(\widehat{q}_t^C\|p_t)$, the slack term can be omitted.