FROM PRIVILEGE TO PARAMETERS: LEARNING DYNAMICS OF ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) has recently emerged as a promising post-training paradigm for improving language models. A teacher conditioned on privileged information, additional context available during training but unavailable at inference, supervises an unprivileged student's predictions at student-generated prefixes. However, how privileged information becomes internalized during training remains poorly understood. We revisit OPSD from the perspective of privilege transmission, combining a local learning-dynamics analysis with controlled random key–value recall experiments. We find that OPSD internalizes information through behavioral learning: privileged information is acquired only through the supervision it induces on the behavior the student is asked to produce. Internalization further exhibits a prefix-first progression aligned with the structure of autoregressive generation. These findings motivate **Recontextualizing Privileged Information**, a simple yet effective strategy that reuses privileged examples across task contexts while retaining task-local reference guidance, allowing information associated with one task to contribute to supervision on others. Experiments with Qwen3 models ranging from 1.7B to 8B parameters demonstrate improvements over vanilla OPSD, with gains of up to **17.61 percentage points** in best-checkpoint mean Avg@12 across eight mathematical reasoning benchmarks.

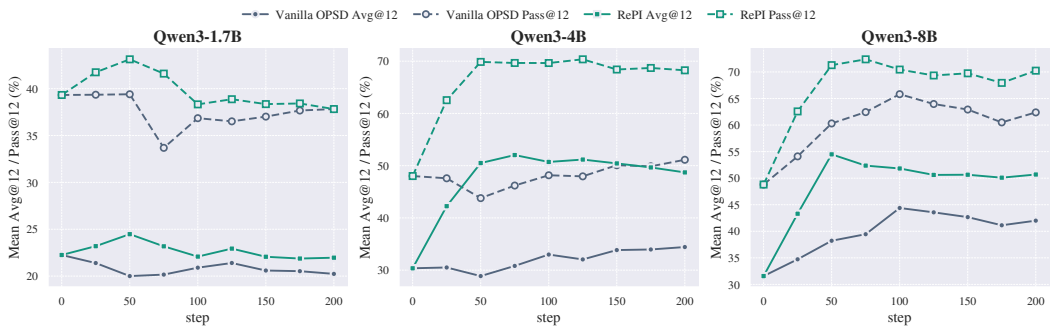


Figure 1: Training dynamics of Avg@12 and Pass@12 under vanilla OPSD and our Recontextualizing Privileged Information (RePI) strategy. Curves show the unweighted mean over all eight mathematics benchmarks for each model scale; step 0 is the untrained base model. RePI achieves higher best-checkpoint mean Avg@12 at all three scales.

1 INTRODUCTION

On-policy self-distillation (OPSD) provides dense post-training supervision without external reward models or task-specific verifiers during optimization (Zhao et al., 2026a; Shenfeld et al., 2026; Hübotter et al., 2026). A teacher conditioned on privileged reference solutions or feedback supervises an unprivileged student's next-token distributions at student-generated prefixes (Agarwal et al., 2024), translating training-time context into independent inference-time capabilities (Snell et al., 2022; Ye et al., 2026).

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.

However, how privileged information becomes internalized during OPSD remains poorly understood. Although the training objective specifies how to align teacher and student predictions, it does not by itself explain which aspects of the privileged information are acquired or how their acquisition unfolds over time (Shrestha & Tessier, 2026). A reference solution may convey task-specific facts, intermediate deductions, and general problem-solving strategies, yet the teacher's access to these does not ensure their internalization by the student. We therefore study OPSD from the perspective of **privilege transmission**, tracing how privileged information induces local supervision on student trajectories across task contexts, propagates through shared parameters, and becomes behavior the student can express independently.

To investigate this process, we combine a local analysis of learning dynamics (Ren & Sutherland, 2025) with controlled random key–value recall experiments. The experiments hold the teacher's privileged information fixed while varying its use in training tasks, allowing us to distinguish information availability from information acquisition. We find that OPSD internalizes information through **behavioral learning**: privileged information is acquired only through the supervision it induces on the behavior the student is asked to produce. Internalization further exhibits a **prefix-first progression** aligned with the structure of autoregressive generation, with earlier components typically acquired before later ones. These results indicate that internalization is governed by where privileged information induces supervision, suggesting that the task contexts of its use are themselves a design dimension of OPSD.

These findings motivate **Recontextualizing Privileged Information (RePI)**: supplementing the current reference solution with other tasks' worked examples in the privileged context, while students generate from the problem alone. Each worked example then induces supervision not only on its own query but also on other queries' student trajectories, broadening where a fixed reference corpus exerts its training value. Across eight mathematical reasoning benchmarks, Qwen3 models at 1.7B, 4B, and 8B improve over vanilla OPSD, gaining up to **17.61 percentage points** in best-checkpoint mean Avg@12.

We summarize our key contributions as follows: **(1) A privilege transmission framework.** We develop a local learning-dynamics analysis of OPSD that characterizes how privilege-induced supervision on student trajectories propagates through shared parameters to influence model predictions. **(2) Behavioral and progressive information internalization.** Through controlled random key–value recall experiments, we show that OPSD internalizes information through behavioral learning, with acquisition exhibiting a prefix-first progression aligned with autoregressive generation. **(3) Recontextualizing Privileged Information.** We propose a strategy that reuses privileged examples across task contexts while retaining task-local reference guidance, achieving consistent improvements over vanilla OPSD across all three model scales and mathematical reasoning benchmarks.

2 Related Work

On-Policy Self-Distillation. Same-architecture students can outperform their teachers through distillation (Furlanello et al., 2018); STaR and rationale distillation use generated reasoning as supervision (Zelikman et al., 2022; Hsieh et al., 2023). MiniLLM and GKD address distribution mismatch through student-generated trajectories (Gu et al., 2024; Agarwal et al., 2024), while context distillation internalizes training-time context (Snell et al., 2022; Ye et al., 2026). OPSD, SDFT, and SDPO condition self-teachers on reference solutions, demonstrations, and feedback, respectively (Zhao et al., 2026a; Shenfeld et al., 2026; Hübotter et al., 2026); these methods establish what privileged information to provide, but not how the student comes to acquire it.

Supervision Selection and Adaptation. Subsequent OPSD variants target supervision through position weighting, error-localized correction, and evidence-masked guided rollouts (Liu et al., 2026; Zhao et al., 2026b; Lazaridis et al., 2026). Teacher-side adaptation adjusts reference information or updates teachers as students evolve (Han et al., 2026; Chen et al., 2026). USD couples token selection and privilege provision under a capacity-sensitive learning-difficulty budget (Yang et al., 2026). Contrastive supervision and signal decomposition reduce undesirable conditioning effects (Pan et al., 2026; Shen et al., 2026). OP^{2}SD replaces the current reference with another problem's solution (Ichihara et al., 2026); RePI instead retains it and adds worked examples, extending their supervision across task contexts.

Information Internalization and Learning Dynamics. Generalized distillation unifies knowledge distillation and privileged-information learning (Lopez-Paz et al., 2015). Theory studies data geometry, optimization, and regularization in simplified settings (Phuong & Lampert, 2019; Mobahi et al., 2020); local learning dynamics connect updates to cross-example prediction changes (Ren & Sutherland, 2025). OPSD studies identify reasoning changes under privileged conditioning (Kaur et al., 2026), but gains and distributional alignment alone cannot establish reference information's learning contribution (Shrestha & Tessier, 2026). On-policy distillation studies examine teacher–student compatibility, state visitation, and progressive alignment (Li et al., 2026b; Fu et al., 2026). Li et al. (2026a) find that a handful of cross-domain prompts can recover most in-domain gains, attributing this to student states exposed to supervision. We distinguish query-elicited states from privilege-induced prediction differences driving local updates, combining local analysis and controlled random key–value recall to trace behavioral, prefix-first acquisition.

3 Learning Dynamics of OPSD

We introduce OPSD, decompose a single update into its privilege-transmission components, and study its training dynamics through controlled random key–value recall.

3.1 Preliminary

Autoregressive language models. Given an input x, an autoregressive language model π_θ parametrized by θ generates a response $y = (y_1, \ldots, y_T)$ according to $\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t})$. At each decoding step, the model maps the current prefix $(x, y_{<t})$ to a vector of logits $z_\theta(x, y_{<t}) \in \mathbb{R}^{|\mathcal{V}|}$, which induces the next-token distribution $\pi_\theta(\cdot \mid x, y_{<t}) = \mathrm{softmax}\big(z_\theta(x, y_{<t})\big)$. We use $s_t = (x, y_{<t})$ to denote a model state, when this causes no confusion.

On-policy self-distillation. Let p denote privileged information available during training but unavailable at inference time. OPSD (Zhao et al., 2026a) evaluates the same model under two information branches. The unprivileged branch predicts $\pi_\theta(\cdot \mid x, y_{<t})$, while the privileged branch additionally conditions on p, $\pi_\theta(\cdot \mid x, p, y_{<t})$. The training trajectory itself is sampled on-policy from the unprivileged model, $y \sim \pi_\theta(\cdot \mid x)$. Let $P(x, p)$ denote the training distribution over query–privilege pairs. Along this student-generated trajectory, the unprivileged distribution is trained to match the privileged distribution, with gradients stopped through the privileged branch, and the objective is computed with forward-KL:

$$\mathcal{L}_{\mathrm{OPSD}}(\theta) = \mathbb{E}_{(x,p)\sim P}\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}\left(\mathrm{sg}\left[\pi_\theta(\cdot \mid x, p, y_{<t})\right] \middle\| \pi_\theta(\cdot \mid x, y_{<t})\right)\right]. \quad (1)$$

Figure 2 summarizes the transmission mechanism and the empirical acquisition pattern studied below.

3.2 Privilege Transmission in OPSD

We next characterize how privileged information is transmitted through a single OPSD update. We analyze an idealized forward-KL update to isolate the transmission structure, rather than reproduce optimizer-level dynamics. For a query–privilege pair (x, p) and a student-generated trajectory $y \sim \pi_\theta(\cdot \mid x)$, each visited state $s_t = (x, y_{<t})$ induces an unprivileged prediction $\pi_\theta(\cdot \mid s_t)$ and a privileged prediction $\pi_\theta(\cdot \mid s_t, p)$. Conditioning on the sampled trajectory and treating the privileged branch as fixed, the local forward-KL loss at s_t is

$$\ell(\theta; s_t, p) = D_{\mathrm{KL}}\big(\mathrm{sg}[\pi_\theta(\cdot \mid s_t, p)] \big\| \pi_\theta(\cdot \mid s_t)\big). \quad (2)$$

Its negative gradient with respect to the student logits is

$$-\nabla_{z_\theta(s_t)} \ell(\theta; s_t, p) = \pi_\theta(\cdot \mid s_t, p) - \pi_\theta(\cdot \mid s_t). \quad (3)$$

Thus, under forward-KL distillation, privileged information affects optimization only through the change it induces in the model prediction at student-visited states. To characterize where such

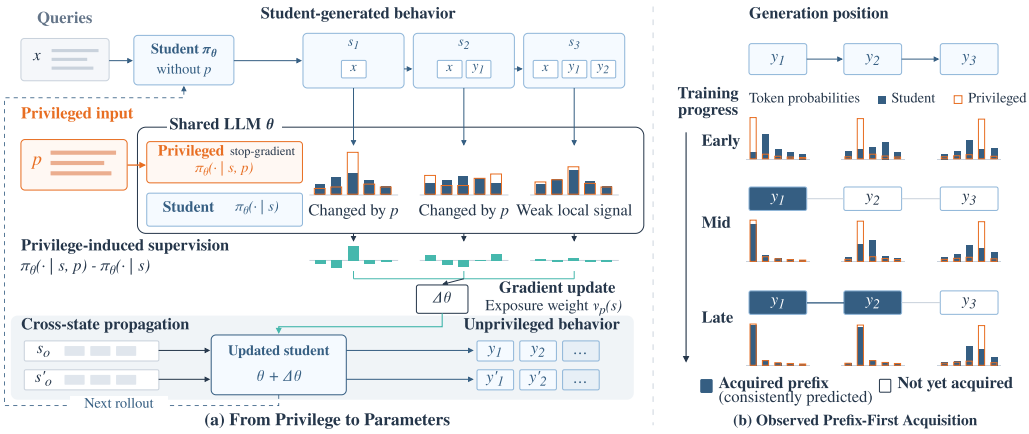


Figure 2: From privilege to parameters. **(a)** At student-generated states, privileged conditioning changes the shared model's next-token predictions, inducing local supervision of varying strength. Exposure-weighted updates propagate through shared parameters to unprivileged behavior, which supplies subsequent rollouts. **(b)** Schematic of the prefix-first acquisition observed in controlled recall.

supervision can arise, we condition the training distribution $P(x, p)$ on a fixed privilege p, giving $P(x \mid p)$. For each query x, define the token-level state occupancy induced by the current student policy as

$$d_{\pi_\theta}(s \mid x) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[\frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\{s = s_t\} \right]. \tag{4}$$

The resulting exposure of p over student states is $\nu_p(s) = \mathbb{E}_{x \sim P(\cdot|p)} \left[d_{\pi_\theta}(s \mid x) \right]$. Hence, privilege transmission is governed jointly by the states at which a privilege is exposed, captured by ν_p, and the prediction difference it induces at those states. Let $J_\theta(s) = \partial z_\theta(s)/\partial\theta \in \mathbb{R}^{|\mathcal{V}| \times \dim(\theta)}$ denote the logit Jacobian. Under a gradient-descent update, the expected one-step parameter contribution associated with p is then

$$\mathbb{E}[\Delta\theta \mid p] = \eta \mathbb{E}_{s \sim \nu_p} \left[J_\theta(s)^\top \left(\pi_\theta(\cdot \mid s, p) - \pi_\theta(\cdot \mid s) \right) \right]. \tag{5}$$

This expression characterizes how privilege-induced supervision is written into the shared parameters. Inspired by the local learning-dynamics analysis of Ren & Sutherland (2025), we next characterize how this update propagates to an arbitrary observation state s_o. Let $\mathcal{A}_\theta(s_o) = I - \mathbf{1}\pi_\theta(\cdot \mid s_o)^\top$ denote the log-softmax Jacobian, and let $\mathcal{K}_\theta(s_o, s) = J_\theta(s_o) J_\theta(s)^\top$ denote the cross-state empirical neural tangent kernel. Both act in vocabulary space; probability vectors are column vectors. These quantities give the following one-step transmission dynamics.

Proposition 1 (One-Step Privilege Transmission)**.** *For a fixed observation state s_o, consider one gradient-descent OPSD update with learning rate η, holding sampled trajectories and privileged predictions fixed during differentiation. Under the smoothness and boundedness assumptions in Appendix A.1,*

$$\mathbb{E}\left[\Delta \log \pi_\theta(\cdot \mid s_o) \mid p\right] = \eta \sum_s \nu_p(s) \mathcal{A}_\theta(s_o) \mathcal{K}_\theta(s_o, s) \left[\pi_\theta(\cdot \mid s, p) - \pi_\theta(\cdot \mid s)\right] + O(\eta^2). \tag{6}$$

The proof is given in Appendix A.1. Its transmission structure can be summarized schematically as

$$\underbrace{\nu_p(s)}_{\text{exposure}} \times \underbrace{\left[\pi_\theta(\cdot \mid s, p) - \pi_\theta(\cdot \mid s)\right]}_{\text{privilege-induced local supervision}} \times \underbrace{\mathcal{A}_\theta(s_o)\mathcal{K}_\theta(s_o, s)}_{\text{cross-state propagation}}. \tag{7}$$

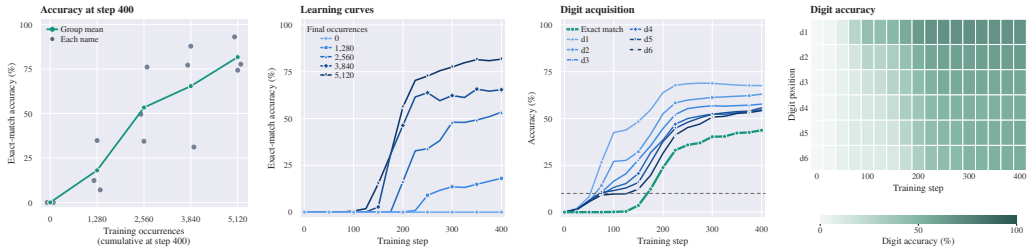


Figure 3: Random key–value recall under OPSD. Recall is measured from generated key–value assignment lines in unprivileged sum queries of the training task family. **(a)** Exact-match accuracy at step 400 versus cumulative training occurrences per key; small dots denote individual keys and large dots denote three-key group means. **(b)** Group-mean learning curves, labeled by cumulative occurrences at step 400. **(c)** Whole-value exact match and digit-wise accuracy, averaged over all 15 keys. **(d)** The same digit-wise accuracies across checkpoints; d_1 is the leftmost digit and d_6 the rightmost. All panels use one training run, evaluated every 25 steps without smoothing.

This decomposition separates where a privilege is exposed, what local supervision it induces, and how that supervision propagates. The query distribution and student policy determine exposure, privileged conditioning produces the local prediction differences, and shared parameters propagate their effects across states.

3.3 Learning Dynamics in Controlled Recall

We instantiate privilege transmission with arbitrary key–value mappings supplied in the privileged context. Queries specify which mappings the student must express, allowing us to study how task-side use affects acquisition while keeping teacher-side availability fixed.

Experimental setup. We associate 15 random keys with distinct six-digit values and train Qwen3-4B on queries that require listing the values of three keys before computing their sum. The student receives only the query and output format; the privileged branch additionally receives all 15 mappings, four worked examples, and the current query's reference solution. We vary how often each key participates in training queries while keeping the mapping fixed and available in every privileged context. Three keys are assigned to each of five groups with 0, 1,280, 2,560, 3,840, or 5,120 cumulative occurrences over 400 optimization steps. We periodically evaluate the unprivileged model on sum queries of the same task family, using target combinations excluded from training queries, and score the generated key–value assignments. Appendix B.1 provides the full prompts and experimental details.

Behavioral learning. Figure 3(a,b) shows that recall improves with a key's participation in training queries, despite all mappings being available to the privileged branch throughout training. At step 400, group-mean exact-match accuracy rises from 0% for keys never queried during training to 18.1%, 53.4%, 65.4%, and 81.7% for the four increasingly frequent groups. These results support behavioral learning: making information available in the privileged context alone does not ensure its acquisition; using it in the requested output behavior provides opportunities for privilege-induced supervision along student-generated trajectories.

Prefix-first progression. Figure 3(c,d) reveals a broadly left-to-right progression within the six-digit values. Earlier digits become accurate before whole values are reliably recovered, and partial acquisition persists even at the final checkpoint: mean accuracy is 67.6% for the first digit and 54.2% for the last, compared with 43.7% for whole-value exact match. This separation shows that internalization develops progressively along the autoregressive output. Learning earlier components can change the prefixes on which later supervision is evaluated.

Together, these observations suggest that privileged information becomes behavior the student can express independently through the supervision it induces in particular task contexts, rather than through availability alone. This perspective opens a design opportunity: beyond choosing what

information the teacher receives, we can also choose the task contexts in which that information is used to supervise the student.

4 RECONTEXTUALIZING PRIVILEGED INFORMATION

Section 3 shows that privileged information's training value depends on where it induces supervision. **Recontextualizing Privileged Information (RePI)** retains each query's reference solution while reusing other worked examples in its privileged context, broadening the use of an existing reference corpus without introducing a new supervision source.

4.1 METHOD

From task-local references to cross-task supervision. Consider a training set $\mathcal{D} = \{(x_i, y_i^*)\}_{i=1}^n$. Vanilla OPSD associates query x_i with its reference solution y_i^*: the student generates from x_i alone, and the privileged branch uses y_i^* to supervise that generation. This pairing provides task-specific guidance, but confines each reference's direct contribution to trajectories generated for its own query. RePI gives each worked example two roles: as the reference for its own query and as reasoning context for teacher supervision on other queries.

Retain and augment. For each training item, we randomly sample m distinct partner indices excluding itself and fix this assignment throughout training. Each partner contributes its question and solution; the current query's reference solution is appended last. Thus, p_i retains a correct solution to x_i while adding cross-task reasoning context that may influence teacher supervision, without a separate retrieval mechanism. Algorithm 1 gives the procedure.

The intervention is in the supervision. The additional examples are neither student inputs nor trajectories that the student is trained to reproduce. Their contribution is mediated by the privileged branch at the prefixes the student actually generates. For a fixed state $s = (x_i, y_{<t})$ and fixed parameters, the change in the idealized negative forward-KL logit gradient relative to vanilla OPSD is

$$\big[\pi_\theta(\cdot \mid s, p_i) - \pi_\theta(\cdot \mid s)\big] - \big[\pi_\theta(\cdot \mid s, y_i^*) - \pi_\theta(\cdot \mid s)\big] = \pi_\theta(\cdot \mid s, p_i) - \pi_\theta(\cdot \mid s, y_i^*).$$

Under this idealized update, the change in local supervision is exactly the teacher-prediction difference above. This signal propagates through the shared-parameter mechanism described in Section 3. Because the teacher processes the examples jointly, their contributions can interact.

A minimal intervention with evolving effects. RePI preserves the student prompt, rollout procedure, distillation objective, and branch parameterization. At fixed parameters, teacher-side conditioning does not change student-state visitation; over training, its updates can alter future prefixes and supervision. Inference uses the original unprivileged prompt.

4.2 EXPERIMENTS

4.2.1 SETUP

Models and data. We evaluate RePI with Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Team, 2025). Training uses the same OpenThoughts math corpus as OPSD (Zhao et al., 2026a), containing 29,434 problem–solution pairs. We evaluate on eight mathematics benchmarks: AIME 2024, AIME 2025, HMMT February 2025, HMMT November 2025, AMO-Bench (50-problem test split), MATH-500, Minerva, and OlympiadBench. These tasks test whether reusing privileged solutions improves reasoning when no reference solutions are available at inference time.

Comparison protocol. Vanilla OPSD uses the current reference; RePI adds nine worked examples ($m = 9$). Both start from the same base checkpoint and use full-parameter training for 200 steps, batch size 32, and identical optimizer, schedule, rollout, and clipped forward-KL settings. Comparisons match optimization steps; the added cost of RePI is confined to the teacher's forward pass over longer contexts, which adds on average 5.8% of wall-clock training time across the three scales.

Algorithm 1 Recontextualizing Privileged Information (RePI)

Require: Training dataset $\mathcal{D} = \{(x_i, y_i^*)\}_{i=1}^{n}$, LLM policy π_θ with parameters θ, additional-example count $m < n$, training steps R, learning rate η.

// Phase I: Privileged Information Generation

$\mathcal{D}' \leftarrow \emptyset$ ▷ Initialize the dataset with privileged information

for $i = 1, \ldots, n$ **do**

  Sample m distinct indices $i_1, \ldots, i_m$ from $\{1, \ldots, n\} \setminus \{i\}$

  $p_i \leftarrow \text{concat}\big(\underbrace{(x_{i_1}, y^*_{i_1}), \ldots, (x_{i_m}, y^*_{i_m})}_{\text{PI from other questions}},\ y_i^*\big)$ ▷ m worked examples plus the current reference solution

  $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{(x_i, y_i^*, p_i)\}$

end for

// Phase II: Standard OPSD Training

for $r = 1, \ldots, R$ **do**

  Sample a mini-batch $\mathcal{B} \subset \mathcal{D}'$

  for $(x_i, y_i^*, p_i) \in \mathcal{B}$ **do**

    Sample $y_i \sim \pi_\theta(\cdot \mid x_i)$ without privileged information

    Evaluate stop-gradient teacher predictions conditioned on x_i and p_i along y_i

  end for

  Update θ using the standard OPSD loss, mini-batch reduction, and optimizer with learning rate η

end for

We also include an OP^{2}SD-style baseline (Ichihara et al., 2026), denoted OP^{2}SD: our reminder variant supplies one unrelated worked example without the current reference solution and trains for 200 steps. Appendix B.2 specifies its prompt variant and different evaluation length limit.

GRPO provides a full-parameter reference baseline trained for 500 steps, evaluated every 50 steps, and selected by eight-benchmark mean Avg@12. Appendix C.1 compares the aggregate trajectories of GRPO, OPSD, and RePI.

Metrics. We sample 12 responses per evaluation problem. Avg@12 measures the fraction correct, averaged over problems; Pass@12 measures the fraction of problems solved at least once. We report individual benchmarks and their unweighted mean. For OPSD, OP^{2}SD, and RePI, we select the checkpoint with highest eight-benchmark mean Avg@12 among steps 25–200, evaluated every 25 steps, and report both metrics at that checkpoint. Figure 1 shows the full trajectories; Appendix B.2 provides full settings and evaluation details.

4.2.2 Main Results

Table 1 reports Base, GRPO, vanilla OPSD, OP^{2}SD, and RePI on every benchmark. RePI improves mean Avg@12 over vanilla OPSD by 3.07, 17.61, and 10.10 percentage points at 1.7B, 4B, and 8B, respectively, and exceeds the base model in both aggregate metrics at all three scales. The largest gain occurs at 4B, where mean Avg@12 increases from 34.42% to 52.03% and mean Pass@12 from 51.12% to 69.66%. At the selected checkpoints, RePI improves Avg@12 over vanilla OPSD on all eight benchmarks at 4B and 8B, and on seven of eight at 1.7B.

At 1.7B, vanilla OPSD falls below Base in selected mean Avg@12 (21.40% versus 22.25%), whereas RePI reaches 24.47% at step 50, matching GRPO's 24.49% selected at step 250. At 4B and 8B, RePI exceeds Base, vanilla OPSD, and GRPO on every benchmark in Avg@12 at their selected checkpoints. RePI also improves mean Pass@12 over vanilla OPSD at all three scales. Compared with OP^{2}SD, RePI has higher mean Avg@12 and Pass@12 at all three scales. All gains are measured with unprivileged evaluation prompts.

4.2.3 Scaling Recontextualization

We vary the number of additional worked examples on Qwen3-4B, keeping the current reference solution in every condition. Table 2 reports this sweep on AIME 2024, AIME 2025, and HMMT

Table 1: Results on eight mathematics benchmarks (%). Base denotes the untrained model. One checkpoint per trained method is selected by mean Avg@12 over all eight benchmarks and used in both panels. All entries use 12 samples per problem. Bold marks the best value within each model scale, including ties; shaded rows denote RePI. GRPO is trained for 500 steps, compared with 200 for OPSD, OP^{2}SD, and RePI.

A Avg@12 ↑

Method	Best ckpt. step	AIME 2024	AIME 2025	HMMT Feb. 25	HMMT Nov. 25	AMO-Bench	MATH-500	Minerva	Olympiad Bench	Mean
Qwen3-1.7B										
Base	—	13.89	10.00	6.94	5.00	1.17	72.68	27.85	40.48	22.25
GRPO	250	18.61	11.67	7.78	**5.83**	1.83	75.75	**30.30**	**44.16**	**24.49**
OPSD	125	15.28	11.67	2.78	3.06	**2.33**	71.77	25.80	38.51	21.40
OP^{2}SD	75	12.78	13.61	6.39	4.44	1.83	73.50	25.34	40.24	22.27
RePI	50	**20.28**	**14.17**	**8.89**	4.44	1.33	**76.02**	28.03	42.63	24.47
Qwen3-4B										
Base	—	22.22	18.61	13.61	10.00	3.17	83.62	39.12	52.52	30.36
GRPO	500	31.67	26.11	17.78	17.78	2.67	87.57	40.75	57.28	35.20
OPSD	200	35.00	23.06	16.67	16.94	2.33	86.42	39.19	55.79	34.42
OP^{2}SD	50	53.61	**56.11**	33.89	43.61	5.67	87.17	44.06	66.78	48.86
RePI	75	**63.61**	55.83	**35.28**	**46.11**	**6.83**	**93.87**	**44.73**	**69.97**	**52.03**
Qwen3-8B										
Base	—	27.78	21.67	13.33	8.61	2.33	84.65	39.74	54.43	31.57
GRPO	400	45.00	31.94	22.22	28.89	4.83	91.20	42.06	62.61	41.10
OPSD	100	56.11	45.28	25.28	30.56	5.17	90.20	39.58	62.92	44.39
OP^{2}SD	100	65.83	49.44	26.39	37.22	7.17	91.52	40.47	67.68	48.22
RePI	50	**71.94**	**58.06**	**39.17**	**48.61**	**8.50**	**94.25**	**43.81**	**71.55**	**54.49**

B Pass@12 ↑

Method	Best ckpt. step	AIME 2024	AIME 2025	HMMT Feb. 25	HMMT Nov. 25	AMO-Bench	MATH-500	Minerva	Olympiad Bench	Mean
Qwen3-1.7B										
Base	—	36.67	23.33	23.33	**23.33**	6.00	90.40	46.69	64.84	39.32
GRPO	250	**50.00**	30.00	20.00	20.00	10.00	**93.60**	**47.43**	**69.58**	42.58
OPSD	125	33.33	30.00	10.00	13.33	10.00	89.80	42.28	63.35	36.51
OP^{2}SD	75	30.00	**33.33**	23.33	16.67	**12.00**	91.20	40.44	64.54	38.94
RePI	50	46.67	**33.33**	**30.00**	**23.33**	8.00	93.40	42.65	67.66	**43.13**
Qwen3-4B										
Base	—	50.00	40.00	23.33	43.33	8.00	95.40	50.00	74.04	48.01
GRPO	500	66.67	53.33	33.33	40.00	10.00	96.80	53.31	78.49	53.99
OPSD	200	66.67	40.00	40.00	40.00	6.00	94.80	47.79	73.74	51.12
OP^{2}SD	50	**83.33**	76.67	**56.67**	70.00	**24.00**	96.60	**57.35**	82.64	68.41
RePI	75	**83.33**	**80.00**	**56.67**	**76.67**	22.00	**98.60**	56.62	**83.38**	**69.66**
Qwen3-8B										
Base	—	53.33	43.33	26.67	40.00	8.00	96.00	48.90	74.18	48.80
GRPO	400	76.67	56.67	43.33	53.33	20.00	97.80	52.94	80.12	60.11
OPSD	100	**83.33**	70.00	60.00	63.33	22.00	97.20	48.90	81.75	65.81
OP^{2}SD	100	80.00	73.33	60.00	63.33	18.00	96.80	49.63	82.94	65.50
RePI	50	**83.33**	**83.33**	**66.67**	**73.33**	**26.00**	**98.60**	**54.41**	**84.57**	**71.28**

February 2025, selecting checkpoints by mean Avg@12 over these three benchmarks. Figure 6 in Appendix C.2 complements these peak results with the full learning curves for each example count. Mean Avg@12 increases from 24.91% without additional examples to 51.76% with eight, remains near 51% with nine or sixteen, and decreases to 46.48% with thirty-two. This non-monotonic response shows that more privileged context is not uniformly better. In the transmission decomposition, the training value of additional information depends on the supervision it induces and its propagation through shared parameters.

Table 2: Scaling recontextualization on Qwen3-4B, averaged over the three competition benchmarks. One checkpoint per condition is selected by mean Avg@12 over these three benchmarks and used for both metrics. Bold marks the best value for each metric. All conditions retain the current reference solution. The 32-example condition uses a longer context limit to accommodate its input.

Additional examples m	0	1	2	4	8	9	16	32
Best ckpt. step	200	75	75	50	100	75	125	50
Avg@12 (%) ↑	24.91	29.54	37.87	46.11	**51.76**	51.57	51.20	46.48
Pass@12 (%) ↑	48.89	53.33	67.78	72.22	73.33	73.33	**75.56**	68.89

4.2.4 Behavioral Effects of Privileged Supervision

RePI also changes how the unprivileged student structures its solutions. We measure alignment between the structural change from base outputs to trained outputs and that from base outputs to reference solutions, using features such as headings, step markers, and mathematical notation. Appendix C.4 defines the metric and reports cross-scale trajectories.

Figure 4 shows that at step 200, alignment rises from -0.26 for vanilla OPSD to 0.58–0.67 with eight to sixteen additional examples. Teacher-side references thus shape the student's independently generated behavior beyond final-answer accuracy.



Figure 4: Reference-style alignment on Qwen3-4B versus additional example count (left) and training step (right).

Style and accuracy evolve differently: with thirty-two examples, alignment reaches 0.79 at step 75, while Avg@12 peaks at step 50 and falls from 46.48% to 31.67% by step 200. These trajectories connect the use of PI to multiple aspects of student behavior, whose changes need not track one another.

5 Conclusion

We studied how privileged information becomes internalized by an unprivileged student through on-policy self-distillation. Using a local learning-dynamics analysis, we connect student-state exposure, privilege-induced supervision, and cross-state propagation through shared parameters. We further conducted controlled random key–value recall experiments to distinguish teacher-side availability from behavioral use, revealing a prefix-first progression of acquisition. The findings motivate RePI: retain the current reference solution while reusing other privileged examples across task contexts. RePI improves best-checkpoint mean Avg@12 over vanilla OPSD at all three tested Qwen3 scales, with gains of up to 17.61 percentage points across eight mathematics benchmarks. These results recast the design question for verifier-free post-training: not only what privileged information to provide, but where it induces supervision.

Limitations

First, our analysis isolates an idealized one-step update; the experiments use the baseline optimizer and clipped objective described in the appendix. Second, the observed gains establish RePI as an effective training design, but not which reference-induced prediction changes produce them. Finally, our random assembly of privileged contexts prioritizes simplicity rather than example relevance; future work could explore retrieval strategies for selecting examples that induce useful supervision.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and editing, code development, and identifying potential bugs and vulnerabilities in the implementation. We did not use generative AI tools for research ideation, theoretical derivations, or the analysis and interpretation of experimental results. The authors reviewed the AI-assisted text for accuracy and consistency and verified the AI-assisted code through manual inspection and testing. The authors take full responsibility for the final content of this paper and its accompanying code, including all AI-assisted contributions.

REPRODUCIBILITY STATEMENT

Appendix A.1 provides the assumptions and proof of Proposition 1. Appendix B describes the experimental settings, prompt templates, and evaluation procedures for the controlled recall and mathematical reasoning experiments. Appendix C presents additional learning curves and analysis details. Our implementation, experimental configurations, and evaluation scripts are available in an anonymous repository at `https://anonymous.4open.science/r/repi-2B5C`.

ETHICS STATEMENT

This work investigates on-policy self-distillation using synthetic key–value tasks and existing mathematical reasoning datasets. It does not involve human-subject studies or the collection of personal data. Because privileged context shapes the supervision received by the student, it may also transmit biases or undesirable behavior contained in that context. Applications beyond the evaluated reasoning tasks therefore require appropriate assessment of the reference data and the resulting model behavior.

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 *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Yutong Chen, Guangfu Guo, Zhichao Xu, and Kunpeng Liu. Dualopsd: Adaptive privileged teachers for on-policy self-distillation. *arXiv preprint arXiv:2608.26019*, 2026.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, et al. Rethinking on-policy distillation of large language models ii: One training example. *arXiv preprint arXiv:2609.04172*, 2026.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pp. 1607–1616. PMLR, 2018.

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

Zihao Han, Tiangang Zhang, Huaibin Wang, and Yilun Sun. Adaptive teacher exposure for self-distillation in llm reasoning. *arXiv preprint arXiv:2605.11458*, 2026.

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.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Yuki Ichihara, Naoto Iwase, Mohammad Atif Quamar, and Junpei Komiyama. Privileged solutions or context-induced teacher behavior? dissecting on-policy self-distillation. *arXiv preprint arXiv:2608.09228*, 2026.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205, 2022.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026.

Aristotelis Lazaridis, Dylan Bates, Aman Sharma, Brian King, Vincent Lu, and Jack FitzGerald. Edge-opd: Internalizing privileged context with evidence guided on-policy distillation. *arXiv preprint arXiv:2605.23493*, 2026.

Gengsheng Li, Mao Zheng, Mingyang Song, Jie Sun, Zeyuan Liu, Ruiqi Liu, Qiyong Zhong, Haiyun Guo, Junfeng Fang, and Jinqiao Wang. Data-free on-policy distillation. *arXiv preprint arXiv:2609.14193*, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Xiaogeng Liu, Xinyan Wang, Yingzi Ma, Yechao Zhang, and Chaowei Xiao. When are teacher tokens reliable? position-weighted on-policy self-distillation for reasoning. *arXiv preprint arXiv:2605.21606*, 2026.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. Evaluating style transfer for text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 495–504, 2019.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. Rlcsd: Reinforcement learning with contrastive on-policy self-distillation. *arXiv preprint arXiv:2606.11709*, 2026.

Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International conference on machine learning*, pp. 5142–5151. PMLR, 2019.

Yi Ren and Danica Sutherland. Learning dynamics of llm finetuning. In *International Conference on Learning Representations*, volume 2025, pp. 70523–70563, 2025.

Zhanming Shen, Jintao Tong, Shaotian Yan, Chen Shen, Hao Chen, Wentao Ye, Xiaomeng Hu, Rui Miao, Haobo Wang, Junbo Zhao, et al. Purified opsd: On-policy self-distillation without losing how to think. *arXiv preprint arXiv:2607.02234*, 2026.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Samyak Shrestha and Alexander Tessier. Rethinking privileged information in on-policy self-distillation. *arXiv preprint arXiv:2608.18271*, 2026.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Yongkang Yang, Zhezheng Hao, Hong Zhang, Yi Liu, Xiankun Lin, Wence Ji, Fanjunduo Wei, Jiarui Yu, Qiang Lin, Xiaoyun Liang, et al. Matching supervision to the student's learning capacity: A unified framework for on-policy self-distillation. *arXiv preprint arXiv:2608.08176*, 2026.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

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*, 2026a.

Ziqi Zhao, Xinyu Ma, Liu Yang, Yujie Feng, Daiting Shi, Jingzhou He, Xin Xin, Zhaochun Ren, and Xiao-Ming Wu. Rosd: Reflective on-policy self-distillation for language model reasoning across domains. *arXiv preprint arXiv:2605.28014*, 2026b.

A Theoretical Analysis

A.1 Proof of One-Step Privilege Transmission

Setting and assumptions. We analyze a single update at fixed parameters θ, conditioning on a fixed privileged input p for which $P(x \mid p)$ is defined. Draw $x \sim P(\cdot \mid p)$ and $y \sim \pi_\theta(\cdot \mid x)$, with states $s_t = (x, y_{<t})$. We make the following assumptions:

1. The vocabulary and the conditional query support are finite, and each trajectory has length $1 \le T \le H$ for a finite horizon H.
2. Logits at every reachable state and at the fixed observation state s_o are twice continuously differentiable in an open neighborhood of θ.
3. The update is ordinary gradient descent on the trajectory-normalized, full-vocabulary forward-KL loss at temperature one. The sampled trajectory, its length, and the privileged predictions are held fixed during differentiation; the update uses neither clipping nor optimizer preconditioning.

These assumptions describe the idealized update in Proposition 1. Finite support and horizon ensure that the expectations and sums below are finite. Smoothness then gives uniform derivative bounds on a sufficiently small closed neighborhood of θ. All expectations in this proof condition on the pre-update parameters, which we leave implicit.

The fixed-trajectory objective. To distinguish the optimization variable from the parameters used to generate the trajectory, write the former as ϑ. For a sampled pair (x, y), define

$$\widehat{\mathcal{L}}(\vartheta; x, y, p) = \frac{1}{T} \sum_{t=1}^{T} \sum_{v \in \mathcal{V}} \pi_\theta(v \mid s_t, p) \log \frac{\pi_\theta(v \mid s_t, p)}{\pi_\vartheta(v \mid s_t)}, \tag{8}$$

$$\Delta\theta = -\eta \, \nabla_\vartheta \widehat{\mathcal{L}}(\vartheta; x, y, p)\Big|_{\vartheta=\theta}. \tag{9}$$

The teacher distribution in Equation 8 is evaluated before the update and detached. Although the two branches share parameters in the model, no derivative passes through this target. Likewise, differentiating the sampled objective does not differentiate the sampling distribution or the discrete trajectory. This is the detached-gradient convention used in on-policy distillation, rather than the total derivative of an expectation whose sampling law also varies with ϑ.

Proof of Proposition 1. **Step 1: local supervision.** For vocabulary coordinates u, v, the log-softmax derivative is

$$\frac{\partial \log \pi_\vartheta(v \mid s)}{\partial z_\vartheta(s)_u} = \mathbb{I}\{u = v\} - \pi_\vartheta(u \mid s). \tag{10}$$

Using the normalization of the detached privileged distribution gives

$$\frac{\partial \ell(\vartheta; s, p)}{\partial z_\vartheta(s)_u}\bigg|_{\vartheta=\theta} = -\sum_v \pi_\theta(v \mid s, p)\big[\mathbb{I}\{u = v\} - \pi_\theta(u \mid s)\big] \tag{11}$$

$$= \pi_\theta(u \mid s) - \pi_\theta(u \mid s, p). \tag{12}$$

Thus the negative logit gradient is exactly the privilege-induced prediction difference. With $J_\theta(s) \in \mathbb{R}^{|\mathcal{V}|\times\dim(\theta)}$, the chain rule yields

$$\Delta\theta = \frac{\eta}{T}\sum_{t=1}^{T} J_\theta(s_t)^\top \big[\pi_\theta(\cdot \mid s_t, p) - \pi_\theta(\cdot \mid s_t)\big]. \tag{13}$$

This identity is exact for the specified sampled update.

Step 2: averaging over student states. By the definition of trajectory-normalized occupancy, any vector-valued function f on the reachable states satisfies

$$\mathbb{E}_{x,y|p}\left[\frac{1}{T}\sum_{t=1}^{T} f(s_t)\right] = \sum_s \mathbb{E}_{x,y|p}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\{s = s_t\}\right] f(s) \tag{14}$$

$$= \sum_s \nu_p(s) f(s). \tag{15}$$

In particular, $\sum_s \nu_p(s) = 1$. Applying Equation 15 to Equation 13 gives

$$\mathbb{E}[\Delta\theta \mid p] = \eta \sum_s \nu_p(s) J_\theta(s)^\top \big[\pi_\theta(\cdot \mid s, p) - \pi_\theta(\cdot \mid s)\big]. \tag{16}$$

The factor $1/T$ remains inside the expectation when trajectory lengths vary. No gradient of ν_p enters this one-step calculation: it describes state occupancy at the fixed pre-update parameters.

Step 3: propagation to an observation state. For a fixed s_o, the log-softmax Jacobian is

$$\mathcal{A}_\theta(s_o) = I - \mathbf{1}\pi_\theta(\cdot \mid s_o)^\top, \tag{17}$$

$$\frac{\partial \log \pi_\theta(\cdot \mid s_o)}{\partial\theta} = \mathcal{A}_\theta(s_o) J_\theta(s_o). \tag{18}$$

Taylor expansion at the pre-update parameters therefore gives

$$\log \pi_{\theta+\Delta\theta}(\cdot \mid s_o) - \log \pi_\theta(\cdot \mid s_o) = \mathcal{A}_\theta(s_o) J_\theta(s_o)\Delta\theta + R(\Delta\theta). \tag{19}$$

The observation state is held fixed in this expansion; it is not regenerated after the update. Taking conditional expectations and substituting Equation 16, we obtain

$$\mathbb{E}[\Delta \log \pi_\theta(\cdot \mid s_o) \mid p] = \eta \sum_s \nu_p(s) \mathcal{A}_\theta(s_o) \underbrace{J_\theta(s_o) J_\theta(s)^\top}_{\mathcal{K}_\theta(s_o, s)} \big[\pi_\theta(\cdot \mid s, p) - \pi_\theta(\cdot \mid s)\big]$$
$$+ \mathbb{E}[R(\Delta\theta) \mid p]. \tag{20}$$

The kernel $\mathcal{K}_\theta(s_o, s)$ has dimensions $|\mathcal{V}| \times |\mathcal{V}|$, as required to map training-state vocabulary supervision to observation-state logits.

Step 4: the remainder. The finite state set implies a bound G on the norm of the negative trajectory gradient in Equation 13, so $\|\Delta\theta\| \le \eta G$. Choose η small enough that every update segment stays in the smooth neighborhood. On a closed ball containing these segments, the Hessian of each observation-state log-probability is bounded in operator norm by a common constant M. Taylor's theorem then gives, for every vocabulary coordinate v,

$$|R_v(\Delta\theta)| \le \frac{M}{2}\|\Delta\theta\|^2 \le \frac{MG^2}{2}\eta^2. \tag{21}$$

Consequently,

$$\|\mathbb{E}[R(\Delta\theta) \mid p]\|_2 \le \frac{\sqrt{|\mathcal{V}|}\, MG^2}{2}\eta^2 = O(\eta^2), \tag{22}$$

which proves the stated expansion. □

Interpretation and scope. The proposition describes the conditional contribution of queries paired with a fixed p. It concerns log-probability changes, hence the log-softmax Jacobian. As a consistency check, $\pi_\theta(\cdot \mid s_o)^\top \mathcal{A}_\theta(s_o) = 0$, so the first-order log-probability change has zero mean under the pre-update distribution, consistent with probability normalization. The kernel term can propagate supervision to states that were not visited in the sampled trajectory; the result does not imply that unvisited states remain unchanged, or that every transferred update improves recall.

For an arbitrary sufficiently small optimizer displacement, Equation 19 still applies. The particular kernel-weighted expression follows from substituting the ordinary, unclipped forward-KL gradient-descent update. Optimizer preconditioning and clipping modify the parameter update; temperature, sampling truncation, and normalization modify the local supervision or occupancy. The proposition isolates the underlying transmission structure rather than asserting an exact optimizer-level identity for the experimental settings.

B Experiment Details

B.1 Random Key–Value Recall

Data and training tasks. We generate 15 unique random keys and associate them with distinct six-digit integers. The same mapping is used throughout training and evaluation.

Each training query asks the student to list the values of three keys and compute their sum. The student sees only the query and output format. The privileged branch additionally receives four worked examples, all 15 mappings, and the current query's reference solution. Both branches share model parameters; distillation is evaluated along the student's sampled continuation, with gradients stopped through the privileged branch.

Controlling training occurrences. An occurrence means that a key is included in a training query, regardless of the student's response. Three keys are assigned to each of five frequency groups. The groups have 0, $1{,}280$, $2{,}560$, $3{,}840$, or $5{,}120$ cumulative occurrences per key at step 400. All groups are trained together. Every privileged input contains all 15 mappings, including the zero-occurrence keys; the controlled variable is their use in student queries, not their availability to the privileged branch. Query wording and arithmetic format vary, while preserving the three assignment lines before the sum.

Training configuration. Table 3 lists the main training settings.

The forward-KL loss is averaged over student-generated completion tokens, with the privileged predictions detached. We apply an upper cap of 4.0 to each signed vocabulary contribution before summing over the vocabulary.

Recall probing and metrics. We probe the unprivileged model with sum queries from the training task family, using combinations of keys absent from training queries. Recall is scored from the generated assignment lines, without providing privileged information. Exact match measures whether each key's complete value is correct; digit accuracy measures correctness at each decimal position from left to right. Missing or malformed values count as incorrect. Figure 3(a,b) reports means within each three-key frequency group, while panels (c,d) average over all 15 keys. All panels follow the same training run.

Prompt templates. The templates below summarize the input structure. Braced fields are replaced by the corresponding keys, values, or text. Angle-bracket placeholders such as `<value>` are shown literally to the student. Ellipses indicate content omitted here for brevity, not text supplied to the model. All inputs use the model's chat template with thinking disabled.

Table 3: Random key–value recall training configuration. Learning-rate entries describe the configured optimizer and scheduler.

Parameter	Setting
Model and optimization	
Base model / trainable parameters	Qwen3-4B / full parameters
Hardware / precision	$8\times$ NVIDIA H200 / bfloat16
Optimizer	AdamW
Weight decay / gradient-norm cap	0 / 0.5
Initial learning rate	10^{-6}
Configured schedule / warmup	Linear / none
Effective batch size	32
Optimizer steps / checkpoint interval	400 / 25 steps
Student rollouts	
Temperature / top-p / top-k	1.1 / 0.95 / 20
Maximum rollout length	1,024 tokens
Thinking (student / privileged)	Disabled / disabled
Distillation	
Direction / teacher gradient	Forward KL / stopped
Loss temperature / vocabulary	1.1 / full vocabulary
Signed vocabulary-contribution cap	4.0 (upper cap only)

Student input

```
Compute {key_1} + {key_2} + {key_3}. The assignment lines below are the
    required step-by-step reasoning. Do not skip, reorder, or add any other
    text. Use exactly this structure:
{key_1} = <value>
{key_2} = <value>
{key_3} = <value>
<value> + <value> + <value> = <total>
The answer is \boxed{<total>}.
```

Privileged input

```
Problem: {student query and output structure}

Here are {k} privileged reference examples:
=== PI Examples Begin ===
=== Privileged Worked Example 1 Begin ===
Problem: Compute {example_key_1} + {example_key_2} + {example_key_3}.
Solution: {example_key_1} = {example_value_1}
{example_key_2} = {example_value_2}
{example_key_3} = {example_value_3}
{example_value_1} + {example_value_2} + {example_value_3} = {example_total}
The answer is \boxed{{example_total}}.
=== Privileged Worked Example 1 End ===
...
=== Additional Privileged Mappings ===
{mapping_key_1} = {mapping_value_1}
...
{mapping_key_15} = {mapping_value_15}
=== PI Examples End ===

Here is a reference solution to this problem:
=== Reference Solution Begin ===
```

```
{key_1} = {value_1}
{key_2} = {value_2}
{key_3} = {value_3}
{value_1} + {value_2} + {value_3} = {total}
The answer is \boxed{{total}}.
=== Reference Solution End ===

...
Return only the exact requested structure.
```

Evaluation input. Evaluation uses the student-input template above with evaluation key triples, without any privileged context. Alternative query phrasings and arithmetic formats preserve the same three key–value assignment lines.

B.2 RePI Mathematical Reasoning Experiments

Training data and comparison. We use the OpenThoughts mathematics training corpus distributed with OPSD (Zhao et al., 2026a), comprising 29,434 problem–solution pairs. Vanilla OPSD provides the current reference solution to the privileged branch. RePI additionally provides nine question–solution examples sampled without replacement from other training items, with the assignment fixed across training. Both use the same student prompt, model initialization, and optimization settings in Table 4. Their training example orders differ. Base denotes the corresponding model before training.

Table 4: Mathematical reasoning settings, shared by vanilla OPSD and RePI unless specified.

Parameter	Setting
Models	Qwen3-1.7B, Qwen3-4B, Qwen3-8B
Trainable parameters	Full model
Optimizer / initial learning rate	AdamW / 10^{-6}
Configured schedule / warmup	Linear / none
Weight decay / gradient-norm cap	0 / 0.5
Effective batch size	32
Training steps / checkpoint interval	200 / 25 steps
Training runs per condition	1; Qwen3-4B OPSD/RePI: 3 seeds (Appendix C.3)
Precision	bfloat16
Rollout temperature / top-p / top-k	1.1 / 0.95 / 20
Maximum rollout length	1,024 tokens
Training sequence limit	20,000 tokens
Thinking (both branches)	Disabled
Distillation direction / temperature	Forward KL / 1.1
Teacher target / vocabulary	Stop-gradient / full vocabulary
Upper cap per signed KL contribution	10^{-6}
Additional worked examples	OPSD: 0; RePI: 9

Both methods use the same baseline distillation implementation: signed vocabulary contributions to forward KL are capped before vocabulary summation, and the loss is averaged over valid student-completion tokens. The recontextualization sweep varies the number of additional examples over $\{0, 1, 2, 4, 8, 9, 16, 32\}$ on Qwen3-4B. The 32-example condition uses a sequence limit of 32,768 tokens to accommodate the longer teacher input; the other conditions use 20,000.

Evaluation benchmarks. Table 5 summarizes the evaluated collections. The suite combines short competition exams with broader mathematical and quantitative reasoning tasks. AMO-Bench uses the complete 50-problem test split of the evaluated release. For OlympiadBench, we evaluate the English, text-only, open-ended mathematics subset.

Table 5: Mathematics evaluation suite. Counts denote evaluated problems, with 12 responses sampled per problem.

Benchmark	Problems	Task coverage
AIME 2024	30	The 2024 AIME I and II competition problems.
AIME 2025	30	The 2025 AIME I and II competition problems.
HMMT February 2025	30	February HMMT competition mathematics, from the MathArena collection.
HMMT November 2025	30	November HMMT competition mathematics, from the MathArena collection.
AMO-Bench	50	Advanced mathematical reasoning with final-answer evaluation.
MATH-500	500	A MATH test subset spanning seven subjects and five difficulty levels.
Minerva	272	Quantitative STEM problems involving mathematical reasoning.
OlympiadBench	674	English, text-only competition mathematics covering algebra, combinatorics, geometry, and number theory.

Decoding and answer evaluation. All methods sample 12 responses per problem at temperature 1.0 and top-$p = 0.8$, without top-k filtering or thinking mode. Base, OPSD, and RePI use the same unprivileged evaluation prompt with a total context limit of 32,768 tokens, including the prompt. OP^{2}SD uses a maximum generation length of 38,912 tokens, a longer budget than OPSD and RePI. We extract the last boxed answer and check mathematical equivalence with the reference; missing or unparseable answers count as incorrect. The same answer-equivalence evaluator is used across methods and benchmarks, rather than the benchmark-specific official graders. In particular, AMO-Bench is not evaluated with its answer-type-dependent judging pipeline, and OlympiadBench's unit and tolerance metadata are not used.

Baselines. We compare RePI with vanilla OPSD, GRPO, and OP^{2}SD at all three Qwen3 model scales using full-parameter training. Vanilla OPSD (Zhao et al., 2026a) conditions the teacher on the current reference solution without additional examples and shares RePI's 200-step training settings in Table 4. GRPO provides a reinforcement-learning baseline trained for 500 steps. Our OP^{2}SD variant (Ichihara et al., 2026) replaces the current reference with one unrelated worked example sampled from the full training corpus. A reminder asks the teacher to use the example's style rather than its content, except when the target independently calls for the same technique; this adapts the published prompt and example-pool restriction. OP^{2}SD uses 200 training steps, batch size 32, learning rate 10^{-6}, and 1,024-token rollouts.

Aggregation and checkpoint selection. Avg@12 averages correctness over the 12 responses and then over problems; Pass@12 counts problems with at least one correct response. The main table and Figure 1 give an unweighted mean over all eight benchmarks. For each trained method and model scale, we select a single checkpoint by the highest eight-benchmark mean Avg@12 and use it for every benchmark and both metrics. OPSD, OP^{2}SD, and RePI are evaluated every 25 steps from step 25 to 200; GRPO is evaluated every 50 steps from step 50 to 500. The selected checkpoints are listed in Table 1. Base uses the original untrained 12-sample evaluations, not the separately sampled GRPO base evaluations. Figure 1 shows only the matched-horizon comparison of OPSD and RePI. The recontextualization-count sweep instead uses the three competition benchmarks AIME 2024, AIME 2025, and HMMT February 2025, selecting by mean Avg@12.

Prompt templates. The student prompt below is shared by both methods and used at evaluation. The teacher template shows RePI; vanilla OPSD omits the other-example blocks while retaining the current reference solution. Braced fields are substituted with task content, and ellipses abbreviate repeated examples and the shared instruction following the reference block.

Mathematics: student and evaluation input

```
Problem: {question}

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

Mathematics: RePI teacher input

```
Problem: {question}

Here is a reference solution to this problem:
=== Reference Solution Begin ===
=== Other Reference Example 1 Begin ===
Problem: {other_question_1}

Solution: {other_solution_1}
=== Other Reference Example 1 End ===
...
=== Other Reference Example {m} Begin ===
Problem: {other_question_m}

Solution: {other_solution_m}
=== Other Reference Example {m} End ===

{current_reference_solution}
=== Reference Solution End ===
...
Please reason step by step, and put your final answer within \boxed{}.
```

C Further Experimental Results

C.1 Learning Dynamics with GRPO

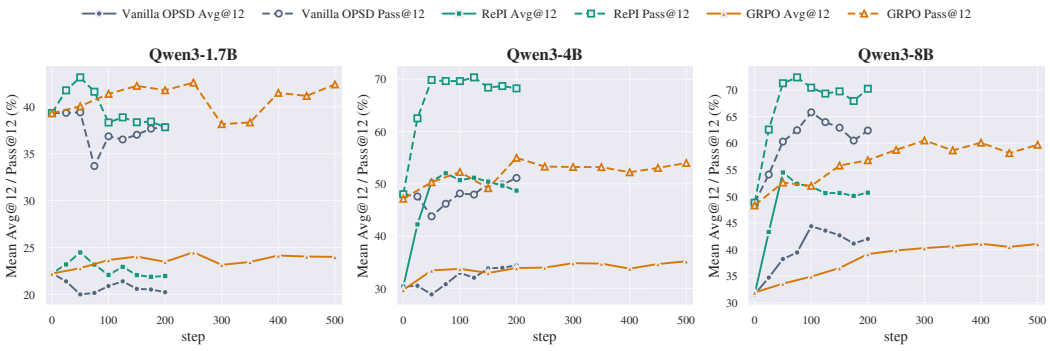


Figure 5: Learning dynamics of vanilla OPSD, RePI, and full-parameter GRPO across three Qwen3 model scales. Solid and dashed curves denote mean Avg@12 and Pass@12 over eight mathematics benchmarks, respectively. OPSD and RePI train for 200 steps, while GRPO trains for 500; the shared axis displays each method's full evaluated trajectory. GRPO uses its separately sampled base evaluation at step 0.

C.2 Recontextualization Learning Curves

C.3 Training-Seed Stability

Table 1 uses one training run per condition. We repeat the Qwen3-4B OPSD and RePI runs with two additional seeds, moving the training, data-order, and rollout seeds together; all other settings follow

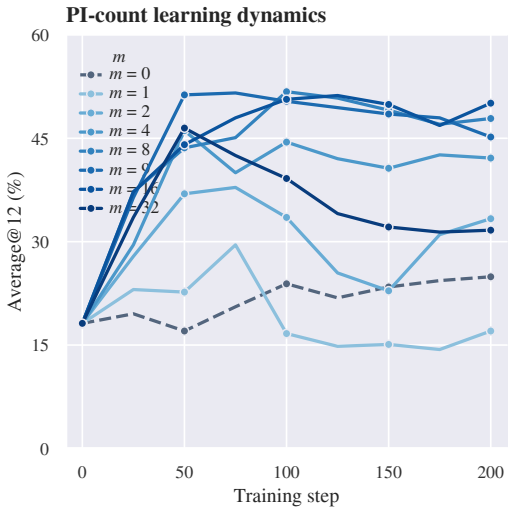


Figure 6: Learning dynamics with different numbers of additional worked examples on Qwen3-4B. Each curve reports mean Avg@12 over AIME 2024, AIME 2025, and HMMT February 2025; step 0 is the untrained base model. All conditions retain the current reference solution.

Table 4. Each run is evaluated on the three competition benchmarks, with checkpoints selected as in Table 2. Figure 7 and Table 6 report the results. The RePI–OPSD gap at the selected checkpoints is stable across seeds (+26.66 to +29.63 points). RePI trajectories are consistent across seeds, peaking at 44.91–51.57% between steps 75 and 125; OPSD trajectories diverge, with the two replications declining to 10.56% and 2.31% by step 200. Seed variation thus affects OPSD far more than RePI.

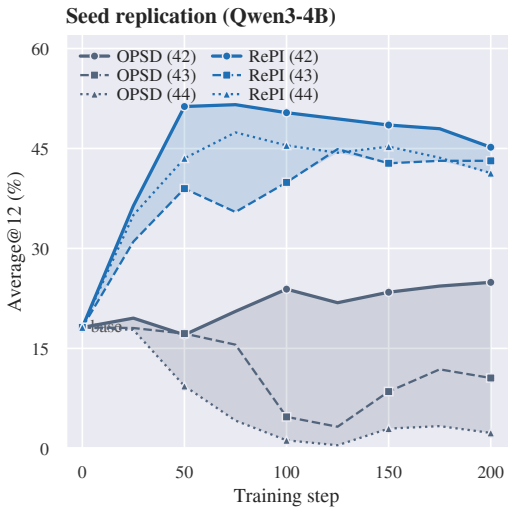


Figure 7: Training-seed stability on Qwen3-4B: mean Avg@12 over AIME 2024, AIME 2025, and HMMT February 2025 for seeds 42 (original) and 43, 44 (replications). Shaded regions span the min–max envelope across seeds.

C.4 Reference-Style Alignment

Text-style evaluation distinguishes stylistic change from content preservation and naturalness (Mir et al., 2019; Jin et al., 2022). We define a structural-alignment diagnostic using the features in Table 7.

Table 6: Seed replication of the Qwen3-4B OPSD and RePI runs, averaged over the three competition benchmarks. "Best step" and "Peak" follow the selection of Table 2; Δ is the RePI–OPSD gap at the selected checkpoints. Seed 42 is the original run of Table 1.

	OPSD			RePI			
Seed	Best step	Peak	Step 200	Best step	Peak	Step 200	Δ
42 (original)	200	24.91	24.91	75	51.57	45.19	+26.66
43	25	18.06	10.56	125	44.91	43.15	+26.85
44	25	17.78	2.31	75	47.41	41.30	+29.63
Mean	—	20.25	12.59	—	47.96	43.21	+27.71

Table 7: Structural features used for reference-style alignment.

Group	Feature	Statistic per response
Format	Headings	Count of Markdown heading lines
	Step markers	Count of numbered step lines
	Numbered items	Count of numbered list items
	Bullet items	Count of bullet list items
Mathematics	Inline math	Count of inline math spans
	Display math	Count of double-dollar math spans
	Equality lines	Fraction of nonempty lines containing =
	LaTeX presence	Binary indicator of LaTeX markers
Repetition	Repeated-line run	Longest run of identical nonempty lines
	Duplicate lines	Fraction of repeated nonempty lines
	20-word repetition	Maximum frequency of a 20-word sequence

Let μ, b, and q denote feature means over trained outputs, base outputs, and 500 reference solutions sampled from the training corpus, respectively. For Figure 4, the alignment score is

$$\cos\left(\log\frac{\mu+\epsilon}{b+\epsilon}, \log\frac{q+\epsilon}{b+\epsilon}\right), \qquad \epsilon = 10^{-3},$$

with elementwise logarithms and division. Positive values indicate that the structural change from the base model aligns with the reference direction; the score measures direction rather than distance to the reference. At initialization the displacement is zero, so the plotted score is set to zero by convention. The Qwen3-4B example-count comparison pools 1,080 responses per checkpoint from AIME 2024, AIME 2025, and HMMT February 2025.

Figure 8 shows the corresponding training trajectories at all three model scales on the five additional benchmarks: HMMT November 2025, AMO-Bench, MATH-500, Minerva, and OlympiadBench. Each checkpoint pools 18,312 responses. Each model's base feature means are computed on this same five-benchmark pool; the Qwen3-4B base is also used in the example-count comparison in Figure 4. The cross-scale plots use the original feature extraction, which counts dollar-delimited inline mathematics and floors feature means at ϵ before taking log ratios. The example-count plots additionally count parenthesis-delimited LaTeX inline mathematics and use the additive smoothing above. Both compare structural-change directions within their respective evaluation pools. The trajectories reveal scale-dependent dynamics: Qwen3-4B RePI increases from 0.51 at step 25 to 0.86 at step 200, while Qwen3-8B reaches an early peak of 0.40 at step 50 before returning to 0.14 at step 200. Qwen3-1.7B follows a recovery trajectory, moving from -0.42 at step 25 to -0.03 at step 200.

C.5 Per-Key Digit Acquisition

Figure 9 expands the averages in Figure 3(c,d) into per-key trajectories. Each panel tracks six digit positions; rows group keys by training frequency.

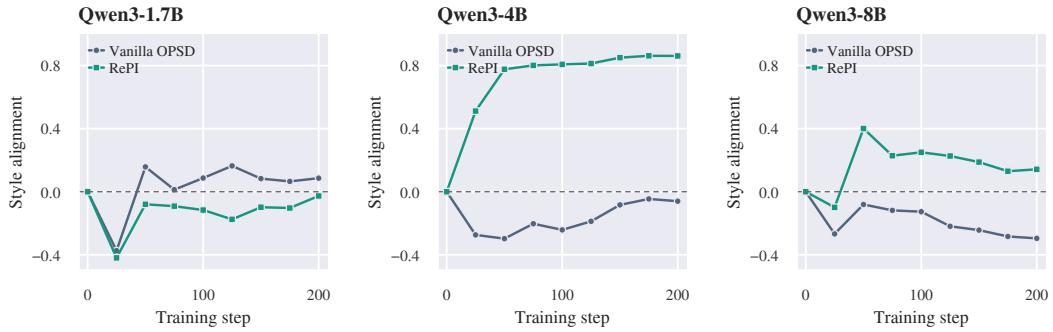


Figure 8: Reference-style alignment across training for Qwen3-1.7B, Qwen3-4B, and Qwen3-8B, evaluated on the five additional mathematics benchmarks. Curves trace changes in the structural-feature profile of unprivileged outputs relative to each model's base profile.

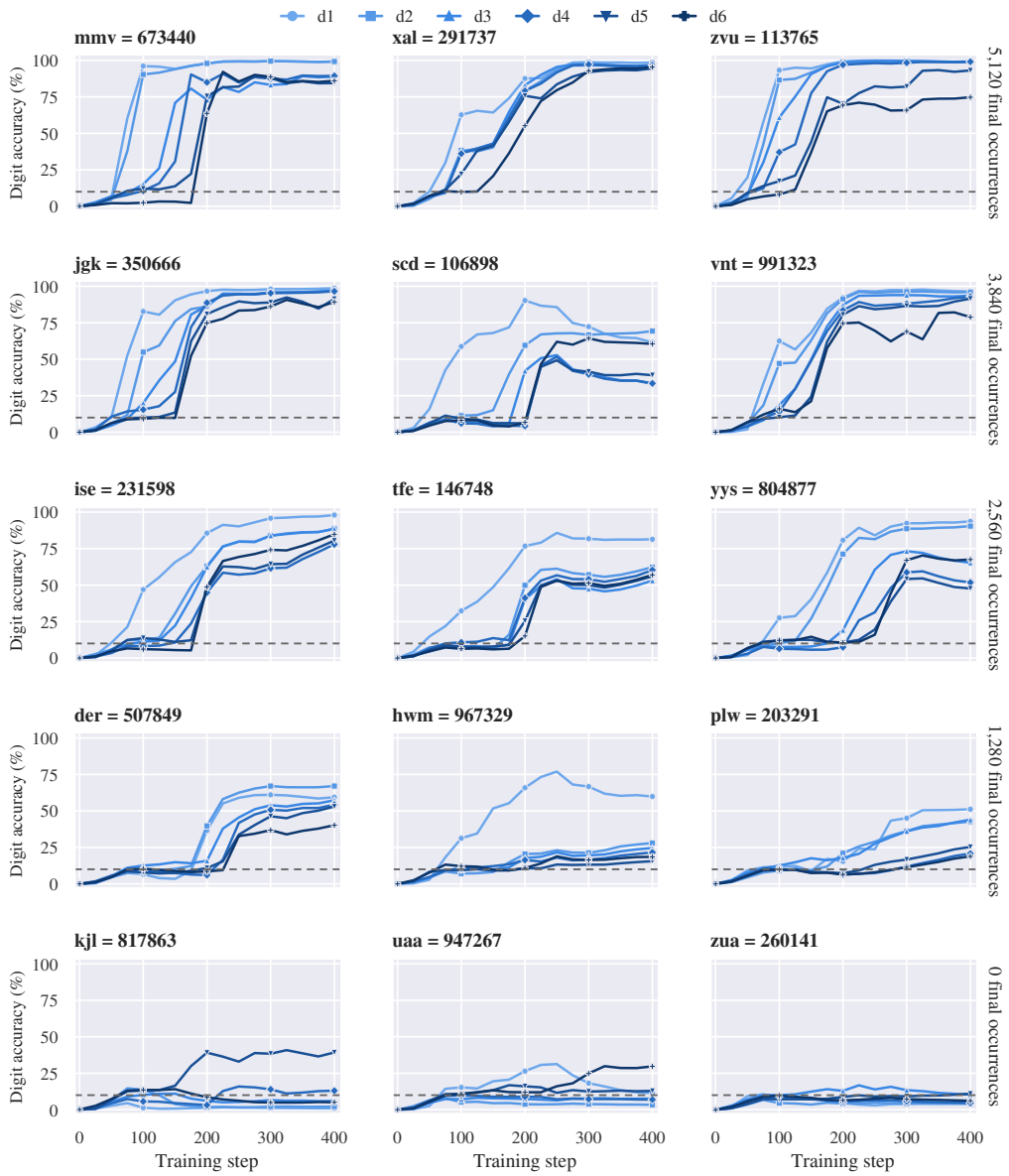


Figure 9: Digit-wise value recall for each of the 15 keys, evaluated on the same unprivileged sum queries as Figure 3. Curves show accuracy at decimal positions d_1 through d_6, from left to right. Rows group keys by cumulative training occurrences at step 400, in descending order. The dashed horizontal line marks 10% accuracy.