ON-POLICY SELF-TRAINING: DATA-FREE SELF-IMPROVEMENT IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation enables post-training foundation models without relying on stronger external teachers. However, existing approaches rely on ground-truth question-answer pairs, human-curated problem sets, or other supervised signals. In this work, we reformulate on-policy self-distillation as a self-training framework, driven by self-generated question-answer pairs and privileged reasoning demonstrations. We push supervision to the limit: guided solely by coarse, high-level topic descriptions, the model synthesizes its own curriculum of queries and privileged reasoning traces without access to any human-curated prompts or answers. We find that self-distillation on fully synthetic privileged information enables a model to improve across multiple domains while retaining its capabilities on domains it is not self-trained on. We demonstrate the generality of this framework across text and multimodal domains, achieving consistent self-improvement on standard language benchmarks and Visual Question Answering. Our findings demonstrate that foundation models possess rich latent capabilities that can be amplified without external task supervision or curated problem distributions.

1 INTRODUCTION

Post-training is a primary driver of capability gains in large language models (Ouyang et al., 2022), and its paradigms differ along two axes: whether training trajectories come from the model itself, and how dense the learning signal is. Supervised finetuning (SFT) imitates curated demonstrations, providing dense token-level targets on teacher-forced trajectories the model never visits at inference, which induces exposure bias (Gu et al., 2024; Agarwal et al., 2024) and erodes existing capabilities (Shenfeld et al., 2026b). Reinforcement learning with verifiable rewards (RLVR) trains on the model's own rollouts (Guo et al., 2025; Shao et al., 2024), avoiding this mismatch but compressing supervision into a single scalar per trajectory. On-policy distillation (OPD) combines the two: the student samples its own trajectories, and a teacher supplies dense next-token targets at the states it visits (Agarwal et al., 2024; Gu et al., 2024). On-policy self-distillation (OPSD) removes the need for a stronger teacher by constructing teacher and student from the same parameters under asymmetric conditioning (Zhao et al., 2026a; Shenfeld et al., 2026a). Across this progression, external supervision is relocated rather than removed: SFT requires reference solutions, RLVR gold answers, OPD a stronger teacher, and OPSD a privileged context, typically the ground-truth solution.

Curating this required supervision at scale is costly: process-level annotation of a single mathematics corpus required about 800,000 human step labels (Lightman et al., 2024), and open instruction datasets rest on crowdsourcing efforts spanning over 13,000 volunteers (Köpf et al., 2023). This cost has motivated a shift toward synthetic supervision, in which the model generates its own instructions (Wang et al., 2023b), rationales (Zelikman et al., 2022), or pseudo-solutions (Li et al., 2026). However, these methods still start from human-written problems, so human curation still decides which problems the model trains on and, in turn, what it can learn.

In language models, self-training is an established approach to self-improvement (Zelikman et al., 2022; Huang et al., 2023; Singh et al., 2024; Zuo et al., 2025). The model solves problems, filters its solutions by agreement with a reference answer or its own majority vote, and is finetuned on those it retains. These methods, however, still train on a curated problem set, which must be collected anew for every task (Singh et al., 2024). Yet a language model can pose problems, not only solve the ones it is given (Zhao et al., 2025; Chen et al., 2026). Therefore, we investigate whether a language

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.

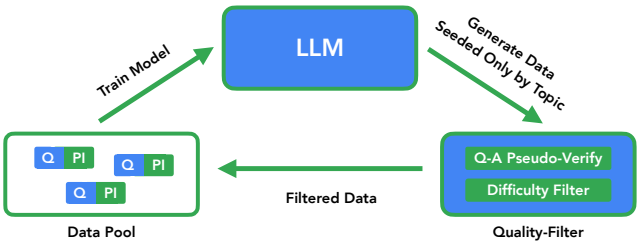

Figure 1: **On-Policy Self-Training loop.** Seeded only by topic descriptors, the model generates and filters its own question-answer pairs, then trains on them via on-policy self-distillation from a privileged EMA teacher, iterating for data-free self-improvement.

model can self-train without labels or a curated problem set, requiring only topic descriptors from which it generates its own training problems.

In this work, we propose **On-Policy Self-Training (OPST)**, an approach to self-train a language model with dense supervision derived from self-generated privileged information (Figure 1). We remove both ground-truth answers and the curated problem set: conditioned only on a coarse topic descriptor, the model synthesizes both a query and the privileged information (PI) used to train on it. Since the generated question-answer pairs are no longer human-curated, the quality of this data can be noisy. Hence, we filter the generated data along two axes: Pseudo-verification ensures that a problem is well-formed, requiring the model to reproduce its answer in a fresh context and under a paraphrase. Further, we apply a consensus gate to ensure that the problem is informative, discarding problems whose answer the student already consistently reproduces across on-policy rollouts. Because the gate is evaluated against the current student, the difficulty of the retained curriculum implicitly evolves as the student model improves. On the quality-filtered samples, we train the model with on-policy self-distillation (Figure 2), where the teacher provides high-quality token-level supervision to the student, limiting catastrophic forgetting of existing capabilities acquired from supervised data (Shenfeld et al., 2026b;a). These effects come from only the teacher being conditioned on the privileged information, while the student receives the question alone.

This asymmetry is crucial in a purely synthetic setting. SFT on the generated answer, or RL rewarding agreement with it, treats the generated answer as the optimization target, which bounds the supervision by the quality of the generation. We show that providing the synthetic answer as context to the teacher instead enables it to produce a reasoning rollout from its own parametric knowledge that serves as an enriched learning signal to the student. The student's supervision is therefore extended by what the teacher can infer from the privileged, generated context, exceeding what the generated answer alone provides. Iterating this loop improves the model over successive rounds, while an exponential moving average (EMA) teacher stabilizes against self-induced distribution drift.

In our experiments, we show that OPST improves language and multimodal reasoning without human-curated questions or answers, outperforming alternative training strategies. On Gemma 4 E2B (Gemma Team et al., 2026), self-training increases MMLU (Hendrycks et al., 2021a) accuracy by $+10.2$ points, with consistent gains on GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2024), and HellaSwag (Zellers et al., 2019). We also find that self-training in-domain transfers to out-of-distribution domains, with performance either remaining stable or also improving. Applied to vision-language models with unlabeled images alone, Visual OPST improves VQAv2 accuracy by $+6.9$ points.

2 Related Work

On-policy self-distillation. OPSD (Zhao et al., 2026a) constructs both policies from a single model, conditioning the teacher on a verified reference solution. Since on-policy methods train on the model's own samples, they forget less than SFT (Shenfeld et al., 2026b), and self-distillation has been shown to enable continual learning across multiple domains (Shenfeld et al., 2026a). Subsequent work varies the privileged context between expert demonstrations (Shenfeld et al., 2026a), privileged observations (Penaloza et al., 2026), and environmental feedback (Hübotter et al., 2026). Others address information leakage and instability (Yang et al., 2026) or extend the setting to agentic tasks (Liu et al., 2026). Closest to our work, Li et al. (2026) replace the gold solution with

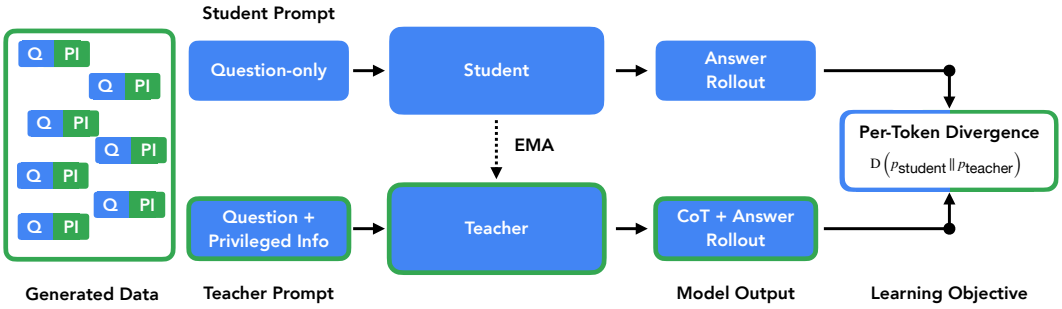

Figure 2: **Model training.** As in OPSD (Zhao et al., 2026a), the student sees only the question and matches the next-token distribution of a teacher that also sees privileged information. In OPST, this information is self-generated, and the teacher is an EMA of the student.

a majority-vote consensus over the model's own rollouts, eliminating answer supervision. All of these, however, operate on a given distribution of problems. OPST removes that assumption: the model synthesizes its own queries from a coarse topic descriptor, so neither the problems nor their solutions are externally supplied.

LLM Self-Training. Early self-training methods filter solutions by agreement with a reference answer (Zelikman et al., 2022; Singh et al., 2024) or by self-consistency across samples (Huang et al., 2023), and label-free reinforcement learning has since replaced the reference answer with majority votes (Zuo et al., 2025) or the model's own confidence (Zhao et al., 2026b). These methods remove answers but not problems: each still trains on a curated problem set, which must be collected anew for every task (Singh et al., 2024). Self-play methods let a proposer generate tasks, seeded by as little as a topic description and checked by a code executor or by majority voting over the solver's answers (Zhao et al., 2025; Huang et al., 2026; Chen et al., 2026). Self-generated problems, however, carry information that self-play reduces to a scalar reward: the answer and rationale the model produces while posing them. Used as privileged context for a self-distillation teacher, it instead yields dense, token-level supervision. On curated problems, Li et al. (2026) show that self-generated answers are considerably more effective as teacher context than as a reward, and Zhang et al. (2026) combine self-play with self-distillation for search agents. While their method requires no external training data, question construction is grounded in documents retrieved from an external corpus, whereas OPST generates problems from the model's parametric knowledge alone. Whether a model can improve from problems it poses itself, with dense token-level supervision and without retrieval or other external grounding, has to our knowledge not yet been studied for general reasoning.

3 Method

In this section, we present On-Policy Self-Training. We first define the data-free post-training setting (Section 3.1) and review on-policy self-distillation (Section 3.2). Following, we describe how the model generates problems and privileged information from topic descriptors (Section 3.3), how it filters them intrinsically (Section 3.4), and how it trains on the result (Section 3.5). Repeating these steps yields multiple rounds of self-training (Section 3.6). Finally, we extend OPST to vision-language models (Section 3.7).

3.1 Problem Formulation: Data-Free Post-Training

Post-training typically relies on a curated dataset of human-written queries and ground-truth answers $\mathcal{D} = \{(x_i, y_i^*)\}$, or at least on a curated prompt distribution $\mathcal{D}_{\text{prompts}} = \{x_i\}$. In *data-free post-training*, neither is available. Training is seeded only by a set of coarse topic descriptors $\mathcal{T} = \{\tau_1, \ldots, \tau_K\}$, where each τ_k names a broad subject area, such as *"Abstract Algebra"* or *"Commonsense Causal Reasoning"*. From $\mathcal{T}$, the model generates a synthetic training set $\mathcal{D}_{\text{syn}} = \{(x_i, z_i)\}_{i=1}^{N}$, where x_i is a generated query and z_i its privileged information, consisting of a rationale and a candidate answer (Section 3.3).

The setting has two objectives. First, the model must improve its reasoning in the seeded domains using only self-generated problems and privileged information. Second, because the training signal comes from its own generations without human oversight, the model must retain its general reason-

ing and previously acquired parametric knowledge on domains it is not self-trained on, a prerequisite for repeating the process over multiple rounds.

3.2 Preliminaries: On-Policy Self-Distillation

On-policy self-distillation (Zhao et al., 2026a) derives the student π_θ and the teacher π_ϕ from the same model and replaces the capability gap of standard distillation (Kim & Rush, 2016) with an information gap. Given a problem x, the student samples a response $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$, with next-token distribution

$$p_t = \pi_\theta(\cdot \mid x, y_{<t}) \tag{1}$$

at step t. The teacher evaluates the same prefix $y_{<t}$, but is additionally conditioned on privileged information z that the student does not see:

$$q_t = \pi_\phi(\cdot \mid x, z, y_{<t}). \tag{2}$$

The student is trained to match the teacher along its own trajectory,

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{(x,z)\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\frac{1}{\sum_t M_t} \sum_{t=1}^{T} M_t \, D(p_t \,\|\, q_t) \right], \tag{3}$$

where D is a divergence between distributions, $M_t \in \{0, 1\}$ masks padding tokens, and no gradient flows through the teacher. In standard OPSD, z is a reference solution (Zhao et al., 2026a).

Existing OPSD methods, however, assume a curated problem set and typically use a reference solution as z. The following sections describe how OPST generates both with the model itself.

3.3 Topic-Seeded Curriculum Generation

Instead of drawing problems from a curated distribution, OPST generates them from the topic descriptors $\mathcal{T}$. Given a topic $\tau \in \mathcal{T}$, a generator π_{gen} produces a query x together with privileged information z,

$$(x, z) \sim \pi_{\text{gen}}(\cdot \mid \tau), \quad z = (r, a), \tag{4}$$

where r is a rationale and a a candidate answer. Queries can be multiple-choice or open-ended, and we sample at temperature T_{gen} to obtain diverse problems. While the student receives the query x alone, the teacher also receives z as additional privileged information. We embed the topic name into a targeted data generation prompt, asking the model to pose problems on the given topic along with generated privileged information. In practice, the model generates partial or full answers to the posed problem. Where available, topics are data-domain category labels. The applied prompt templates are provided in Appendix A.7.

We instantiate π_{gen} either as the EMA teacher π_ϕ (Section 3.5), so that data quality can improve as training progresses, or as a frozen copy of the initial model π_{θ_0}. The student does not generate data, but its rollouts determine which samples are retained (Section 3.4).

3.4 Pseudo-Verification & Consensus Gate

Removing human supervision removes not only human data curation but also quality assurance. Generated samples can contain ill-posed questions or incorrect candidate answers, and training on them propagates these errors. At the same time, not every well-posed sample is worth training on: if the student already solves a problem reliably, the teacher conditioned on z produces nearly the same next-token distribution as the student, and the sample contributes little beyond its generation and rollout cost.

We therefore filter any generated sample for two criteria: *Pseudo-verification* first estimates whether a sample is well-formed, and the *consensus gate* then approximates whether it is informative for the current student.

Pseudo-Verification. We evaluate if a given generated sample possesses two properties of a well-posed problem: its answer should be reproducible without the generator's own intent as context, and it should not depend on the particular wording of the question.

To pseudo-verify the generated answer, we require the generator to recover it when solving the question in a fresh context, without access to z and at a low temperature, which yields $\hat{a}(x) = a$. This separates authoring from solving: when query and answer are generated in one pass, a flawed or ambiguous question can still receive a confident answer. To pseudo-verify the generated problem statement, we prompt the generator to paraphrase x into $x' \sim \pi_{\text{gen}}(\cdot \mid x)$ with an instruction to preserve its meaning, solve x' under the same conditions, and require the answer to be unchanged, $\hat{a}(x') = a$. A well-posed question has the same answer regardless of its wording, so this rejects questions whose answer depends on incidental phrasing, which typically indicates that the question admits more than one reading. A sample is retained only if: $v(x, z) = \mathbb{I}\big[\hat{a}(x) = a\big] \cdot \mathbb{I}\big[\hat{a}(x') = a\big]$

Consensus Gate. We assume that the samples that carry the most signal are those the student cannot yet answer reliably: a question that is trivial for the student yields the same answer on every rollout, and the privileged teacher has little to add. We therefore measure answer agreement across sampled reasoning paths as in self-consistency (Wang et al., 2023a). For each verified sample, the student samples G rollouts $\{y_g\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid x)$, from which we extract answers $\{\hat{a}_g\}_{g=1}^{G}$. Samples with full agreement with the candidate answer, $c(x, z) = \frac{1}{G} \sum_{g=1}^{G} \mathbb{I}[\hat{a}_g = a] = 1$, provide little new signal for the model and are discarded.

The two filters act in opposite directions by design. Pseudo-verification keeps samples that are stable enough for the generated answer to serve as privileged information, while the consensus gate discards samples the student already solves. The retained samples therefore lie at the student's capability frontier, and since the gate depends on the current student, the difficulty of the curriculum evolves with the model.

3.5 Model Training

Given a pseudo-verified batch $\mathcal{B} = \{(x_i, z_i)\}_{i=1}^{B}$, we update the student with on-policy self-distillation (Section 3.2), where $p_{i,t}$ and $q_{i,t}$ denote the student and teacher distributions (Eqs. 1 and 2) at step t of the student's rollout on sample i.

Generalized Divergence. Following the generalized formulation of prior work (Agarwal et al., 2024; Zhao et al., 2026a), we instantiate the divergence D in Eq. (3) as a skewed Jensen–Shannon divergence. With the interpolated mixture distribution $\bar{q}_{i,t} = \alpha\, q_{i,t} + (1 - \alpha)\, p_{i,t}$ for $\alpha \in (0, 1)$, the per-token divergence is defined as

$$D_\alpha(p_{i,t} \,\|\, q_{i,t}) = \alpha\, D_{\mathrm{KL}}(q_{i,t} \,\|\, \bar{q}_{i,t}) + (1 - \alpha)\, D_{\mathrm{KL}}(p_{i,t} \,\|\, \bar{q}_{i,t}). \tag{5}$$

In the limits, $D_\alpha/\alpha \to D_{\mathrm{KL}}(q_{i,t} \,\|\, p_{i,t})$ as $\alpha \to 0$, recovering the mode-covering forward KL divergence, and $D_\alpha/(1 - \alpha) \to D_{\mathrm{KL}}(p_{i,t} \,\|\, q_{i,t})$ as $\alpha \to 1$, recovering the mode-seeking reverse KL divergence. Unlike unbounded KL objectives, D_α is upper-bounded by the binary entropy $H(\alpha)$, ensuring that tokens to which one distribution assigns near-zero probability, such as localized artifacts introduced by an imperfect synthetic hint, are prevented from dominating the gradient update. We set $\alpha = 0.8$, which favors the mode-seeking regime. The final objective is

$$\mathcal{L}_{\text{OPST}}(\theta; \mathcal{B}) = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{\sum_t M_{i,t}} \sum_{t=1}^{T} M_{i,t}\, D_\alpha(p_{i,t} \,\|\, q_{i,t}). \tag{6}$$

EMA Teacher. The teacher is initialized from the student, $\phi_0 = \theta_0$, and updated after every gradient step as $\phi_k = (1 - \beta)\, \phi_{k-1} + \beta\, \theta_k$. Since the teacher produces the dense supervision, it must remain stable. With $\beta = 1$, a single noisy batch immediately changes the targets for subsequent batches, and errors can reinforce themselves. The EMA instead averages out such fluctuations over roughly $1/\beta$ steps. Averaging the student's weights also yields a more accurate model than the student itself (Table 6d), consistent with momentum teachers in vision (Caron et al., 2021). For this reason, the teacher is the model we evaluate.

3.6 Iterative Self-Training

A single round of OPST is limited by the quality of the data and supervision that the initial model provides. We therefore repeat generation, verification, and training over N rounds. The student and

teacher weights carry over from one round to the next. For each additional round, π_{gen} generates a new training set, from which we remove problems that duplicate those of earlier rounds.

The iterative nature of our loop lets student and teacher co-evolve. Training on new data further improves the student, and through the EMA update, also the teacher. A stronger teacher, in turn, benefits the next round in three ways. It provides more accurate token-level targets on the student's rollouts. As the generator, it produces better problems and privileged information. And as the verifier (Section 3.4), it judges its own data more reliably, so fewer incorrect samples reach training. Because the teacher is an average over many training steps, only improvements that persist across batches enter this cycle.

3.7 VISUAL OPST

OPST is modality-agnostic and can be extended to the multimodal setting by adding an unlabeled modality. We therefore explore this for vision-language tasks by adding images to the process. Visual OPST uses a collection of unlabeled images $\mathcal{I}$ and visual topic descriptors $\mathcal{T}_{\text{vis}}$, such as *"Spatial Relations"*, *"Object Counting"*, or *"Attribute Recognition"*. Unlike the text setting, it requires unlabeled images as input, from which we generate the same data as detailed in Section 3.3.

Visual Curriculum Generation. Given an image $I \in \mathcal{I}$ and a topic $\tau \in \mathcal{T}_{\text{vis}}$, the generator produces a question about the image together with privileged information, $(x, z) \sim \pi_{\text{gen}}(\cdot \mid I, \tau)$. We consider two forms of privileged information. In the first, z is a detailed scene caption that the generator writes for I, describing its objects, attributes, and spatial arrangement. In the second, we prompt the model to detect the objects in the scene, from which we bootstrap bounding-box grounded spatial relations. The privileged caption informs the teacher rollout with explicit visual details that the student must extract from the image itself, while the detected spatial relations help ground the teacher with actual object positions. Both forms can be combined by conditioning the teacher on the caption and the spatial relations.

Visual Verification. Visual verification tests whether the candidate answer is supported by the image rather than by a hallucinated premise. The generator solves x on K augmented views $\mathcal{A}_k(I)$, obtained with color jitter, rescaling, and random cropping, and we measure agreement with the candidate answer:

$$v_{\text{aug}}(I, x, z) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\big[\hat{a}(\mathcal{A}_k(I), x) = a\big]. \tag{7}$$

Horizontal flips are excluded because they change the answer to questions about left and right.

Model Training. Training follows Eq. (6) with the image added to both contexts: the student samples $y_i \sim \pi_\theta(\cdot \mid I_i, x_i)$, and the teacher provides $q_{i,t} = \pi_\phi(\cdot \mid I_i, x_i, z_i, y_{i,<t})$.

4 EXPERIMENTS

Our experimental evaluation focuses on three aspects: in-domain self-improvement without human-curated data, knowledge retention on out-of-domain (OOD) benchmarks when mixing data generation domains, and transfer to multimodal inputs with Visual OPST.

4.1 EXPERIMENTAL SETUP

Models and Datasets. As base models, we utilize the Gemma 4 E2B and E4B models (Gemma Team et al., 2026). Data Generation is seeded with topic descriptors covering the domains of four *in-domain* benchmarks: MMLU (Hendrycks et al., 2021a), GPQA (Rein et al., 2024), GSM8K (Cobbe et al., 2021), and HellaSwag (Zellers et al., 2019). To measure transfer and retention, we additionally evaluate on three *OOD* benchmarks that no topic descriptor targets: ARC-Challenge (Clark et al., 2018), MATH-500 (Lightman et al., 2024; Hendrycks et al., 2021b), and Winogrande (Sakaguchi et al., 2021). For the multi-domain experiments (Table 3), GPQA is treated as out-of-domain.

Table 1: **Synthetic-data results.** Performance comparison on language reasoning benchmarks for Gemma 4 models trained on self-generated data only.

	Method	Signal	Gen. Answer	MMLU	GPQA	GSM8K*	HellaSwag	Average
Gemma 4 E2B								
w/o Think.	Base			37.5	28.8	–	48.7	38.3
	+ SFT	Dense	Target	38.3$_{+0.8}$	30.8$_{+2.0}$	–	50.8$_{+2.1}$	40.0$_{+1.7}$
	+ GRPO	Sparse	Reward	37.3$_{-0.2}$	30.3$_{+1.5}$	–	50.3$_{+1.6}$	39.3$_{+1.0}$
	+ OPST (Ours)	Dense	Context	**47.7**$_{+10.2}$	**32.8**$_{+4.0}$	–	**56.2**$_{+7.5}$	**45.6**$_{+7.3}$
w/ Think.	Base			69.9	27.8	72.7	63.1	58.4
	+ SFT	Dense	Target	71.0$_{+1.1}$	30.3$_{+2.5}$	75.4$_{+2.7}$	64.5$_{+1.4}$	60.3$_{+1.9}$
	+ GRPO	Sparse	Reward	70.7$_{+0.8}$	32.3$_{+4.5}$	75.0$_{+2.3}$	62.5$_{-0.6}$	60.1$_{+1.7}$
	+ OPST (Ours)	Dense	Context	**71.1**$_{+1.2}$	**34.3**$_{+6.5}$	**80.9**$_{+8.2}$	**65.6**$_{+2.5}$	**63.0**$_{+4.6}$
Gemma 4 E4B								
w/o Think.	Base			54.3	30.3	–	53.7	46.1
	+ SFT	Dense	Target	58.1$_{+3.8}$	35.9$_{+5.6}$	–	**58.6**$_{+4.9}$	50.9$_{+4.8}$
	+ GRPO	Sparse	Reward	55.8$_{+1.5}$	31.8$_{+1.5}$	–	54.3$_{+0.6}$	47.3$_{+1.2}$
	+ OPST (Ours)	Dense	Context	**61.9**$_{+7.6}$	**36.4**$_{+6.1}$	–	58.2$_{+4.5}$	**52.2**$_{+6.1}$
w/ Think.	Base			76.0	43.9	84.4	80.9	71.3
	+ SFT	Dense	Target	76.2$_{+0.2}$	44.4$_{+0.5}$	87.1$_{+2.7}$	81.1$_{+0.2}$	72.2$_{+0.9}$
	+ GRPO	Sparse	Reward	75.4$_{-0.6}$	45.0$_{+1.1}$	83.2$_{-1.2}$	81.5$_{+0.6}$	71.3$_{\pm 0.00}$
	+ OPST (Ours)	Dense	Context	**80.1**$_{+4.1}$	**48.0**$_{+4.1}$	**93.0**$_{+8.6}$	**83.0**$_{+2.1}$	**76.0**$_{+4.7}$

*Evaluated with thinking only

Evaluation and Baselines. Table 1 compares training algorithms on identical self-generated data. *SFT* imitates the generated answer, *GRPO* (Shao et al., 2024) rewards agreement with it, and OPST supplies it as teacher context. Table 2 varies the level of supervision. *Supervised SFT* and *Supervised OPSD* train on curated questions with reference solutions, *u-OPSD* (Li et al., 2026) on curated questions with self-generated answers, and *R-Zero* (Huang et al., 2026) on self-generated problems with reinforcement learning. *Base* denotes the instruction-tuned model. In Table 2, methods above the light rule train on human-curated reference solutions, and **bold** marks the best result among the remaining methods. Elsewhere, **bold** and underline mark the best and second-best result. For OPST, we evaluate the EMA teacher (Section 3.5).

4.2 Results

Learning from Self-Generated Data. On Gemma 4 E2B without thinking, OPST improves MMLU by $+10.2$ points, GPQA by $+4.0$, and HellaSwag by $+7.5$ (Table 1). On average, OPST exceeds SFT on the same data by 5.6/1.3 points (2B/4B) without thinking and by 2.7/3.8 points with thinking. The two methods differ only in whether the generated answer is the optimization target or the teacher's context, so this gap isolates the effect of that choice. Only on 4B HellaSwag without thinking is SFT ahead (0.4 points). With thinking, the stronger base models leave less headroom on MMLU ($+1.2$ at 2B), whereas the gains on GPQA ($+6.5$) and GSM8K ($+8.2$) persist.

Comparison Across Supervision Levels. Table 2 compares OPST against methods that use human-curated data. Among methods without reference solutions, OPST attains the best average in all four settings, exceeding u-OPSD by 1.0 to 2.8 points and R-Zero by 1.4 to 5.2 points. As u-OPSD trains on curated questions and R-Zero operates at the same level of supervision as OPST, this suggests that a curated problem set is not necessary and that the improvements stem from our training approach rather than additional data. OPST also closes most of the gap to full supervision: at 2B without thinking, it recovers 88% of the supervised OPSD gain ($+7.3$ of $+8.3$), and with thinking, it matches supervised OPSD at both model sizes ($+0.2$ and $+0.7$ points).

Multi-Domain Curriculum & Domain Generalization. Table 3 evaluates whether OPST can self-train on several domains at once by combining the topic descriptors of two or all three in-domain benchmarks. Training on all three domains improves the in-domain average by 7.6 points, the largest gain among the mixtures, and comes within 1.0 point of single-domain training, indicating little interference between domains. Out-of-domain retention is mixed: the average improves by 0.8 points, but MATH-500 regresses by 4.9 points. Combining MMLU with GSM8K topics improves MMLU by 11.3 points, more than MMLU topics alone ($+10.2$), and yields the best out-of-domain average ($+3.9$), suggesting positive transfer from mathematical reasoning. Even without any MMLU topics, MMLU improves by 5.1 points, so part of the gain is not specific to the seeded domain.

Table 2: **Results across various levels of supervision.** Performance comparison on language reasoning benchmarks for Gemma 4 models across supervision levels. Methods above the light rule train on human-curated reference solutions. **Bold** marks the best result among the remaining methods.

	Method	Human-Curated Question	Human-Curated Answer	MMLU	GPQA	GSM8K*	HellaSwag	Average
Gemma 4 E2B								
w/o Think.	Base			37.5	28.8	–	48.7	38.3
	+ Supervised SFT	✓	✓	47.3$_{+9.8}$	31.3$_{+2.5}$	–	52.8$_{+4.1}$	43.8$_{+5.5}$
	+ Supervised OPSD	✓	✓	49.6$_{+12.1}$	33.3$_{+4.5}$	–	56.8$_{+8.1}$	46.6$_{+8.3}$
	+ u-OPSD‡	✓	×	45.9$_{+8.4}$	**33.3**$_{+4.5}$	–	54.7$_{+6.0}$	44.6$_{+6.3}$
	+ R-Zero‡	×	×	43.5$_{+6.0}$	30.8$_{+2.0}$	–	51.2$_{+2.5}$	41.8$_{+3.5}$
	+ OPST (Ours)	×	×	**47.7**$_{+10.2}$	32.8$_{+4.0}$	–	**56.2**$_{+7.5}$	**45.6**$_{+7.3}$
w/ Think.	Base			69.9	27.8	72.7	63.1	58.4
	+ Supervised SFT	✓	✓	70.5$_{+0.6}$	30.8$_{+3.0}$	83.6$_{+10.9}$	63.7$_{+0.6}$	62.2$_{+3.8}$
	+ Supervised OPSD	✓	✓	70.1$_{+0.2}$	31.8$_{+4.0}$	85.3$_{+12.6}$	63.9$_{+0.8}$	62.8$_{+4.4}$
	+ u-OPSD‡	✓	×	70.8$_{+0.9}$	33.3$_{+5.5}$	76.6$_{+3.9}$	65.0$_{+1.9}$	61.4$_{+3.0}$
	+ R-Zero‡	×	×	70.3$_{+0.4}$	31.8$_{+4.0}$	80.3$_{+7.6}$	64.1$_{+1.0}$	61.6$_{+3.2}$
	+ OPST (Ours)	×	×	**71.1**$_{+1.2}$	**34.3**$_{+6.5}$	**80.9**$_{+8.2}$	**65.6**$_{+2.5}$	**63.0**$_{+4.6}$
Gemma 4 E4B								
w/o Think.	Base			54.3	30.3	–	53.7	46.1
	+ Supervised SFT	✓	✓	62.4$_{+8.1}$	34.3$_{+4.0}$	–	60.3$_{+6.6}$	52.3$_{+6.2}$
	+ Supervised OPSD	✓	✓	63.4$_{+9.1}$	36.9$_{+6.6}$	–	62.6$_{+8.9}$	54.3$_{+8.2}$
	+ u-OPSD‡	✓	×	60.0$_{+5.7}$	32.8$_{+2.5}$	–	56.5$_{+2.8}$	49.8$_{+3.7}$
	+ R-Zero‡	×	×	54.7$_{+0.4}$	31.3$_{+1.0}$	–	54.9$_{+1.2}$	47.0$_{+0.9}$
	+ OPST (Ours)	×	×	**61.9**$_{+7.6}$	**36.4**$_{+6.1}$	–	**58.2**$_{+4.5}$	**52.2**$_{+6.1}$
w/ Think.	Base			76.0	43.9	84.4	80.9	71.3
	+ Supervised SFT	✓	✓	78.5$_{+2.5}$	43.5$_{-0.4}$	93.2$_{+8.8}$	81.5$_{+0.6}$	74.2$_{+2.9}$
	+ Supervised OPSD	✓	✓	77.7$_{+1.7}$	47.5$_{+3.6}$	93.4$_{+9.0}$	82.4$_{+1.5}$	75.3$_{+4.0}$
	+ u-OPSD‡	✓	×	76.9$_{+0.9}$	44.3$_{+0.4}$	88.9$_{+4.5}$	82.8$_{+1.9}$	73.2$_{+1.9}$
	+ R-Zero‡	×	×	75.4$_{-0.6}$	44.0$_{+0.1}$	90.2$_{+5.8}$	82.1$_{+1.2}$	72.9$_{+1.6}$
	+ OPST (Ours)	×	×	**80.1**$_{+4.1}$	**48.0**$_{+4.1}$	**93.0**$_{+8.6}$	**83.0**$_{+2.1}$	**76.0**$_{+4.7}$

*Evaluated with thinking only. ‡Our re-implementation.

Table 3: **Multi-domain results.** Performance comparison on in-domain and out-of-domain benchmarks for Gemma 4 E2B self-trained on combinations of topic descriptors. *Single domain* reports separate models, each self-trained only on the topics of the respective benchmark.

	Training Topics	In-domain MMLU	In-domain HellaSwag	In-domain GSM8K*	In-domain Avg.	Out-of-domain GPQA	Out-of-domain ARC-C	Out-of-domain MATH-500*	Out-of-domain Winogrande	Out-of-domain Avg.
Gemma 4 E2B										
Reference	Base	37.5	48.7	72.7	53.0	28.8	39.1	56.5	50.0	43.6
	+ Single domain	*47.7*$_{+10.2}$	*56.2*$_{+7.5}$	*80.9*$_{+8.2}$	*61.6*$_{+8.6}$					
Mixtures	+ MMLU + HellaSwag	46.9$_{+9.4}$	**55.6**$_{+6.9}$	74.3$_{+1.6}$	58.9$_{+5.9}$	30.3$_{+1.5}$	44.9$_{+5.8}$	48.0$_{-8.5}$	51.8$_{+1.8}$	43.8$_{+0.2}$
	+ HellaSwag + GSM8K	42.6$_{+5.1}$	54.6$_{+5.9}$	77.0$_{+4.3}$	58.1$_{+5.1}$	**31.3**$_{+2.5}$	43.0$_{+3.9}$	55.8$_{-0.7}$	51.8$_{+1.8}$	45.5$_{+1.9}$
	+ MMLU + GSM8K	**48.8**$_{+11.3}$	50.8$_{+2.1}$	**78.5**$_{+5.8}$	59.4$_{+6.4}$	28.8$_{\pm 0.00}$	44.5$_{+5.4}$	**64.1**$_{+7.6}$	**52.7**$_{+2.7}$	**47.5**$_{+3.9}$
	+ All three	48.4$_{+10.9}$	55.4$_{+6.7}$	77.9$_{+5.2}$	**60.6**$_{+7.6}$	29.3$_{+0.5}$	**45.3**$_{+6.2}$	51.6$_{-4.9}$	51.2$_{+1.2}$	44.4$_{+0.8}$

*Evaluated with thinking. All other benchmarks are evaluated without thinking.

4.3 Multimodal Self-Training: Visual OPST

To evaluate whether our self-training framework extends beyond text-only reasoning, we apply Visual OPST to Gemma 4 E2B and E4B vision-language models (Gemma Team et al., 2026) using unlabeled images sampled from OpenImages V4 (Kuznetsova et al., 2020). Table 4 reports performance on visual question answering (VQAv2 (Goyal et al., 2017)), object hallucination detection (POPE (Li et al., 2023)), and object counting (CountBench (Paiss et al., 2023)).

As shown in Table 4, Visual OPST transfers to vision-language models using unlabeled images alone. Both forms of privileged information improve VQAv2 accuracy at 2B, by 6.6 (spatial QA) and 4.9 (scene caption) points. At 2B, combining them yields the largest VQAv2 gain (+6.9), while its POPE F1 gain (+2.4) matches spatial QA alone (+2.5). At 4B, spatial QA alone performs best on VQAv2 and POPE, so the two forms of privileged information are not consistently complementary. On POPE, the improvement comes mainly from recall (+4.5 points at 2B and +6.2 at 4B).

Table 4: **Multimodal results.** Performance comparison on vision-language benchmarks for Gemma 4 models self-trained on unlabeled images only.

		Privileged Info		VQAv2	POPE			CountBench
	Method	**Caption**	**Spatial**	**Acc. (↑)**	**F1 (↑)**	**Acc. (↑)**	**Rec. (↑)**	**Acc. (↑)**
Gemma 4 E2B								
	Base			49.1	82.2	83.2	72.9	31.7
	+ Visual OPST	×	✓	55.7$_{+6.6}$	**84.7**$_{+2.5}$	**85.1**$_{+1.9}$	**77.4**$_{+4.5}$	34.8$_{+3.1}$
Ours	+ Visual OPST	✓	×	54.0$_{+4.9}$	83.0$_{+0.8}$	83.9$_{+0.7}$	74.0$_{+1.1}$	33.9$_{+2.2}$
	+ Visual OPST	✓	✓	**56.0**$_{+6.9}$	84.6$_{+2.4}$	85.0$_{+1.8}$	**77.4**$_{+4.5}$	**35.7**$_{+4.1}$
Gemma 4 E4B								
	Base			55.9	84.8	85.2	78.0	49.4
	+ Visual OPST	×	✓	**59.6**$_{+3.7}$	**87.9**$_{+3.1}$	**87.4**$_{+2.2}$	**86.4**$_{+8.5}$	50.2$_{+0.8}$
Ours	+ Visual OPST	✓	×	58.0$_{+2.2}$	87.8$_{+2.9}$	87.2$_{+2.0}$	**86.4**$_{+8.5}$	**50.6**$_{+1.1}$
	+ Visual OPST	✓	✓	59.2$_{+3.3}$	87.0$_{+2.1}$	86.6$_{+1.4}$	84.2$_{+6.2}$	50.0$_{+0.6}$

Table 5: **Ablations on Gemma 4 E2B on MMLU.** Default settings are highlighted in gray.

Sample selection	MMLU
Base	37.5
+ OPST w/o verification	42.7
+ Verification	45.4
+ Consensus gate	**47.7**

(a) Sample selection.

Privileged info	MMLU
None	40.6
Placebo	40.2
Answer only	44.6
Rationale only	45.8
Full	47.7

(b) Privileged information.

p	SFT	OPST
0%	38.8	42.7
25%	38.1	42.1
50%	35.4	41.4
75%	33.1	40.8
Drop	−5.7	−1.9

(c) Wrong candidate answers.

5 Ablations

Quality Filtering. We ablate our quality filters by adding them incrementally, starting from OPST on all generated samples. Without filtering, OPST improves over the base model by $+5.2$ points (Table 5a). Pseudo-verification adds $+2.7$ points and the consensus gate a further $+2.3$.

Privileged Information. In Table 5b, removing privileged information shrinks the gain over the base model from 10.2 to 3.1 points, and a placebo from a different problem on the same topic yields only 2.7. The rationale alone recovers 8.3 points and the answer alone 7.1. The gain therefore stems from sample-specific privileged information, with the rationale carrying more signal than the answer; the residual gain without it likely reflects self-distillation towards the EMA teacher.

Robustness to Wrong Answers. In Table 5c, we disable both filters and replace a fraction p of the candidate answers with wrong ones. At $p = 75\%$, OPST loses only 1.9 points and remains 3.3 points above the base model, whereas SFT loses 5.7 points and falls 4.4 points below it. Conditioning the teacher on the generated answer, rather than imitating it, therefore limits the influence of incorrect generations. Further ablations are provided in Appendix A.2.

6 Conclusion

We introduced On-Policy Self-Training (OPST), which post-trains a model on self-generated problems and privileged information, seeded only by coarse topic descriptors. We showed that conditioning the teacher on the generated answer, rather than training on it, lets the teacher fill in omitted reasoning steps and limits the influence of potentially noisy data. We found that OPST outperforms SFT and GRPO on the same generated data and methods that rely on curated questions, and extends to multiple domains and vision-language models, demonstrating that models can self-improve from self-generated data alone.

Limitations. OPST can only amplify existing knowledge, and since pseudo-verification measures self-consistency rather than correctness, consistent errors still enter training. Our topic descriptors are hand-chosen, and our experiments are limited to small models. Promising directions include label-free stopping criteria and tool- or retrieval-grounded verification.

AI USE STATEMENT

In this work, we used generative AI tools for implementing training infrastructure and experiment scripts and for feedback on the experimental design. We have not used generative AI tools for generating core research ideas, formulating hypotheses, or modifying evaluation benchmark data, and tasks involving human-subject data are not applicable to this work. Additionally, we used generative AI tools for drafting and editing parts of the manuscript, improving readability, checking and formatting tables, plotting utilities, and debugging LaTeX compilation. Open-weight generative models (Gemma 4) serve as the primary object of study and generate the synthetic training data within our proposed self-training pipeline. We have reviewed all AI-assisted work: all generated code was manually audited and tested for correctness by the authors, all mathematical formulations and experimental numbers were verified against primary training logs, all AI-suggested interpretations were checked against the results, and all references were checked against original publications. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Lili Chen, Mihir Prabhudesai, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Self-questioning language models. In *ICLR 2026 Workshop on Multi-Agent Learning and Its Opportunities in the Era of Generative AI*, 2026. URL https://openreview.net/forum?id=8OOlhDmpXL.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 2025. doi: 10.1038/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021b.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. In *International Conference on Learning Representations*, volume 2026, pp. 130770–130790, 2026.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 1051–1068, 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.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36:47669–47681, 2023.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision. *arXiv preprint arXiv:2608.06296*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Haoran Liu, Yuwei Zhang, Xiyao Li, Bohan Lyu, and Jingbo Shang. Hero: Hindsight-enhanced reflection from environment observations for agentic self-distillation. *arXiv preprint arXiv:2606.11559*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3147–3157. IEEE, 2023.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

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

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026b.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Yaocheng Zhang, Yuanheng Zhu, Wenyue Chong, Songjun Tu, Qichao Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. π-Play: Multi-agent self-play via privileged self-distillation without external data. *arXiv preprint arXiv:2604.14054*, 2026.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *Advances in Neural Information Processing Systems*, 38:105816–105879, 2025.

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. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026a.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, volume 2026, pp. 2548–2581, 2026b.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2025.

A Appendix

A.1 Self-Distillation Objectives on Generated Data.

Unlike a ground-truth reference solution, generated privileged information $z = (r, a)$ can be ambiguous: the rationale r may miss important aspects, and the candidate answer a may be partially or fully incorrect. Therefore, training a model on z directly inherits these defects. SFT on the generated pairs maximizes

$$\log \pi_\theta(z \mid x) = \sum_{j=1}^{|z|} \log \pi_\theta(z_j \mid x, z_{<j}), \tag{8}$$

which supervises the model only on the tokens of z. Critically, it treats every token as correct, including any potential omissions and errors. OPSD instead uses z as context: at every state $y_{<t}$ the student visits, the teacher supplies a full next-token distribution, so supervision covers the entire student rollout rather than the $|z|$ tokens of the generation. Crucially, the teacher produces this rollout from its own knowledge under the exposure to the generated privileged information. This leads to a more nuanced, higher-quality supervision, since the teacher reweights the model's own next-token distribution by how consistent each continuation is with the privileged information. This has three consequences for generated data. First, z need not spell out a step for the teacher to supervise it: tokens that move the rollout toward a are upweighted, so the teacher completes reasoning that r omits. Second, the target is expressed in the model's own distribution rather than in the surface form of z, so the student does not imitate the phrasing or formatting of the generation. Third, the prior limits the influence of a faulty z: an error shifts the target only to the extent that the model finds it plausible, and where the prior favors a different step, the teacher retains that step instead of copying the error. The student therefore learns from what the model can infer given z as a hint, not from z itself. Consequently, conditioning the teacher on z yields dense, prior-grounded token-level supervision that filters generation noise, making on-policy self-distillation well suited for learning from purely synthetic data.

A.2 Further Ablations

To isolate the contribution of each design choice in OPST, we conduct systematic component ablations on Gemma 4 E2B across four core axes: the EMA update rate β, the synthetic dataset size generated per round, the number of outer self-training iterations N, and the evaluation target (π_ϕ vs. π_θ).

Table 6: **OPST component ablations on Gemma 4 E2B evaluated on MMLU.** Default settings are highlighted in gray.

rate β	MMLU
1.00 (no EMA)	38.1
0.20	44.2
0.05	**47.7**
0.01	45.2

(a) EMA update rate.

samples / round	MMLU
57	43.8
171	47.2
285	**47.7**
570	46.9

(b) Dataset size per round.

rounds N	MMLU
0 (base IT)	37.5
3	43.1
7	**47.7**
12	46.2
20	46.2

(c) Self-training rounds.

Eval checkpoint	MMLU
Student (π_θ)	45.9
EMA teacher (π_ϕ)	**47.7**

(d) Evaluation target.

EMA Update. The EMA teacher is meant to provide stable targets despite noisy updates on generated data (Section 3.5). We vary its rate β (Table 6a). Without an EMA ($\beta = 1$), the teacher follows

the student's current weights, and OPST barely improves over the base model (38.1 vs. 37.5). Too much smoothing also slows progress: with $\beta = 0.01$, the teacher incorporates the student's improvements too slowly and reaches 45.2, whereas $\beta = 0.05$ performs best (47.7).

Dataset Size per Round. The number of samples per round trades off the amount of data against how often the improving teacher regenerates it. We vary it from 57 to 570 samples (Table 6b). Accuracy increases up to 285 samples and then saturates, decreasing slightly to 46.9 at 570 samples. Larger rounds therefore add generation cost without improving results.

Number of Rounds. Iterating lets student and teacher improve each other, but errors in the generated data can also accumulate over rounds (Section 3.6). We evaluate the model after different numbers of rounds (Table 6c). Significant improvements emerge after the first three rounds (+5.6), and accuracy is highest after seven rounds (47.7). Further rounds bring no additional improvement, but performance remains stable at around 46, well above the base model.

Evaluation Target. Because the EMA teacher averages over the student's recent updates, it may also be the better model to deploy. We compare the student and the EMA teacher at the end of training (Table 6d). The EMA teacher is 1.8 points more accurate, consistent with the student's weights being affected by noise from recent batches that averaging removes.

A.3 Variance Across Seeds

To assess the stability of OPST, we repeat self-training on Gemma 4 E2B with three random seeds and report the mean and standard deviation in Table 7. The means correspond to the values reported in Table 1. Across both regimes, the standard deviation stays at or below 1.2 points and is therefore well below the gains of OPST over the base model. Without thinking, variance is low on all benchmarks. With thinking, it is highest on GPQA and GSM8K. On GPQA Diamond, which contains only 198 questions, a single question already corresponds to about 0.5 points. MMLU with thinking is the most stable benchmark (0.1), consistent with its large test set.

Table 7: **Variance across seeds.** Mean and standard deviation of OPST on Gemma 4 E2B over three seeds. Means correspond to Table 1.

Setting	MMLU	GPQA	GSM8K	HellaSwag	Average
w/o Think.	47.7 ± 0.8	32.8 ± 1.0	–	56.2 ± 0.8	45.6 ± 0.3
w/ Think.	71.1 ± 0.1	34.3 ± 1.0	80.9 ± 1.2	65.6 ± 0.5	63.0 ± 0.4

A.4 Generalization to Qwen

To test whether OPST is specific to the Gemma 4 model family, we additionally apply it to Qwen3-4B-Base (Yang et al., 2025). Unlike the instruction-tuned Gemma 4 models used in the main experiments, Qwen3-4B-Base is a pretrained model without instruction tuning. We use the same topic descriptors, prompt templates, and hyperparameters as for Gemma 4 E2B and evaluate with the protocol described in Appendix A.6. Table 8 reports the results. OPST improves all four benchmarks and raises the average accuracy by 6.0 points (66.7 → 72.7). The largest gain occurs on GSM8K (+13.3), followed by GPQA (+5.1), MMLU (+3.1), and HellaSwag (+2.5). Since the multiple-choice benchmarks are scored by log-likelihood over the answer options, the gains on MMLU, GPQA, and HellaSwag do not stem from improved answer formatting. OPST therefore transfers both to a different model family and to a model that has not undergone instruction tuning.

A.5 OPST Algorithm

Algorithm 1 summarizes the complete OPST training procedure. Both the student π_θ and the privileged teacher π_ϕ are initialized from the same weights θ_0, so training requires no external teacher model. Each of the N rounds consists of two stages: (1) a fresh curriculum of problems x with privileged information z is synthesized from the topic set $\mathcal{T}$ and filtered via pseudo-verification and the consensus gate (Sections 3.3 and 3.4), and (2) the student is trained on its own on-policy rollouts $y \sim \pi_\theta(\cdot \mid x)$ by minimizing $\mathcal{L}_{\text{OPST}}$ (Eq. 6) against the teacher conditioned on (x, z). After

Table 8: **Results on Qwen3-4B-Base.** Performance of OPST applied to a pretrained Qwen3 base model, evaluated with the protocol described in Appendix A.6.

Method	MMLU	GPQA	GSM8K	HellaSwag	Average
Qwen3-4B-Base					
Base	72.0	32.3	79.7	82.8	66.7
+ OPST (Ours)	**75.1**$_{+3.1}$	**37.4**$_{+5.1}$	**93.0**$_{+13.3}$	**85.4**$_{+2.5}$	**72.7**$_{+6.0}$

every update, the teacher tracks the student through an exponential moving average with rate β. The teacher therefore improves alongside the student while remaining a stable distillation target. After the final round, the EMA teacher π_ϕ is returned as the final model.

Algorithm 1 On-Policy Self-Training (OPST)

Require: topics $\mathcal{T}$, initial weights θ_0, rounds N, EMA rate β
$\theta \leftarrow \theta_0,\ \phi \leftarrow \theta_0$
for $n = 1, \ldots, N$ **do**
  Generate $(x_i, z_i) \sim \pi_{\text{gen}}(\cdot \mid \tau)$ for $\tau \in \mathcal{T}$ ▷ §3.3
  Pseudo-verification and consensus gate; discard noisy, trivial, and duplicate samples ▷ Section 3.4
  for each mini-batch $\mathcal{B}$ **do**
    Sample rollouts $y_i \sim \pi_\theta(\cdot \mid x_i)$ for all samples in $\mathcal{B}$
    Update θ with $\nabla_\theta \mathcal{L}_{\text{OPST}}(\theta; \mathcal{B})$ ▷ Eq. 6
    $\phi \leftarrow (1-\beta)\,\phi + \beta\,\theta$
  end for
end for
return π_ϕ

A.6 Evaluation Protocol

We follow the task formulations of the EleutherAI `lm-evaluation-harness` (Gao et al., 2023) and adapt them to the Gemma 4 chat template. Without thinking, we score all multiple-choice benchmarks, i.e., MMLU, GPQA Diamond, HellaSwag, ARC-Challenge, and Winogrande, by log-likelihood over the closed set of answer options. No answer is therefore parsed from generated text. We use the harness defaults, i.e., 5-shot prompting for MMLU, zero-shot prompting for the remaining multiple-choice benchmarks, and length-normalized accuracy where the harness specifies it. GSM8K is evaluated with 8-shot chain-of-thought prompting (Wei et al., 2022).

With thinking, multiple-choice answers are parsed from the final `Answer: X` line. GSM8K is graded by exact match of the final numerical answer, and MATH-500 by symbolic equivalence of the final `\boxed{}` expression using SymPy. For all models and benchmarks, both before and after self-training, the rate of parsing failures and invalid answers is 0%. Performance differences in the thinking regime therefore do not stem from changes in answer formatting. All remaining settings follow the harness defaults.

A.7 OPST Prompt Templates

We list the prompts used for problem synthesis, student rollouts, and teacher conditioning. Synthesis prompts are shown in gray, student prompts in blue, and teacher prompts in orange, with placeholders filled at runtime shown in **`{color}`**. With thinking, the system turn `<|turn>system\n<|think|><turn|>\n` is prepended to both student and teacher prompts.

Problem Synthesis. The generator receives a topic descriptor, e.g., `"college_mathematics"` for MMLU or `"rate, time, and distance word problems"` for GSM8K. It returns a question x together with a rationale r and a candidate answer a. We use a multiple-choice template for MMLU, GPQA, and HellaSwag, and an open-ended template for GSM8K.

Synthesis Prompt (Multiple-Choice: MMLU, GPQA, HellaSwag)

```
You are an expert exam writer creating challenging multiple-choice
questions on
the topic of "{topic}".

Write ONE novel, self-contained, graduate-level multiple-choice
question testing
deep conceptual understanding or multi-step reasoning in "{topic}".
Provide
exactly four plausible answer choices labeled (A), (B), (C), and (D),
followed
by a rigorous step-by-step solution and the final correct option
letter.

Use the following exact format:
Question: <question text>
(A) <choice A>
(B) <choice B>
(C) <choice C>
(D) <choice D>
Solution: <step-by-step derivation explaining why the correct choice
holds>
Answer: <single letter A, B, C, or D>
```

Synthesis Prompt (Open-Ended: GSM8K)

```
You are an expert mathematics curriculum designer creating multi-step
word
problems on the topic of "{topic}".

Write ONE novel, self-contained arithmetic or algebraic word problem
about
"{topic}" that requires 3 to 5 reasoning steps to solve. Ensure all
numerical
quantities are consistent and yield an exact numerical answer. After
the
question, provide a complete step-by-step worked solution and the
final number.

Use the following exact format:
Question: <word problem text>
Solution: <step-by-step mathematical derivation>
Answer: <final numerical value>
```

Student Prompts. The student receives only the question x.

Student Prompt (Multiple-Choice)

```
Solve the following multiple-choice question step by step. After your
reasoning,
state your final choice on a new line in the exact format "Answer: X"
where X is
A, B, C, or D.

Question: {question}
(A) {choice_A}
(B) {choice_B}
(C) {choice_C}
(D) {choice_D}
```

Student Prompt (Open-Ended)

```
Solve the following math word problem step by step. Show all
intermediate
calculations and conclude with your final numerical result in the
exact format
"Answer: <number>".

Question: {question}
```

Teacher Prompts. The teacher receives the question together with the privileged information $z = (r, a)$ and is instructed to solve the problem without referring to it. If the generated rationale is incomplete, the teacher first completes it, conditioned on the question and the candidate answer.

Rationale Completion Prompt

```
Question: {question}
{choices_block}

The potentially correct answer to this problem is ({gen_answer}).
Write a clear, rigorous step-by-step derivation reasoning about
({gen_answer}).
Write the explanation as a natural, self-contained solution from first
principles. Do NOT mention that the correct answer was given to you
in advance.
```

Teacher Prompt

```
You are an expert tutor solving the problem below. You are provided
with a
potential reference solution and final answer as privileged
background context.
Use this privileged information to guide a clear, rigorous,
step-by-step
derivation of the problem from first principles. Do NOT mention the
reference
solution or state that the answer was provided to you; present the
reasoning as
a direct solution to the question.

### Privileged Reference Context:
Reference Rationale: {privileged_rationale}
Potential Answer: {gen_answer}

### Problem to Solve:
Question: {question}
{choices_block}

Provide your step-by-step reasoning and conclude with "Answer: ".
```

A.8 Visual OPST Prompt Templates

We list the prompts used in Visual OPST for visual question synthesis, privileged information extraction, visual verification, and student–teacher distillation. For all multimodal turns, the image token `<|image|>\n` is prepended to the user turn `<|turn>user\n...\n<turn|>\n<|turn>model\n`.

Visual Problem Synthesis. Given an unlabeled image $I \in \mathcal{I}$ and a visual topic descriptor $\tau \in \mathcal{T}_{\text{vis}}$, the generator synthesizes `{n}` question–answer pairs. We use a dedicated prompt for object counting and a shared template for the remaining visual topics.

Visual Synthesis Prompt (Object Counting)

```
Look at this image carefully. Generate {n} counting questions about
objects in
the image. Include some objects that appear multiple times and some
that do not
appear at all (count 0).
Format EXACTLY as:
Q: <question>
A: <number>

Generate {n} different questions.
```

Visual Synthesis Prompt (Attributes and Spatial Relations)

```
Look at this image carefully. Generate {n} questions about
{topic_description}.
Format EXACTLY as:
Q: <question>
A: <short answer>

Generate {n} different questions.
```

where `{topic_description}` is instantiated per topic $\tau \in \mathcal{T}_{\text{vis}}$ as:

- `color_attributes`: *"colors, sizes, shapes, or visual attributes of objects"*
- `spatial_relations`: *"spatial relationships between objects (left/right, above/below, in front/behind, near/far)"*

Privileged Information. To construct the privileged information z, the generator either writes a detailed scene caption for I or detects the objects in the scene with bounding boxes. From pairs of detected boxes, we derive grounded spatial relations between objects from their box centers (`"left of"`, `"right of"`, `"above"`, `"below"`).

Privileged Scene Caption Prompt

```
Describe this image in detail, including the main objects, their
attributes, and
their spatial arrangement.
```

Object Detection Prompt

```
Detect all objects in this image. For each object, provide the name
and bounding
box coordinates.
Format each detection as: name: [x1, y1, x2, y2]
Coordinates are normalized 0-1000.
List up to 10 objects.
```

Visual Verification. To compute $v_{\text{aug}}(I, x, z)$, the generator answers x on each of the K augmented views $\mathcal{A}_k(I)$ using a concise VQA prompt.

Visual Verification Prompt

```
{question}
Answer concisely in 1-5 words.
```

Student and Teacher Prompts. During distillation, the student receives the image I and the question x, while the teacher additionally receives the privileged information z prepended to x.

Visual Student Prompt

```
<|image|>
{question}
```

Visual Teacher Prompt

```
<|image|>
Image description: {privileged_caption}  Objects present:
{detected_objects}
{question}
```