THE ILLUSION OF CERTAINTY: DECOUPLING CAPABILITY AND CALIBRATION IN ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) is an increasingly important paradigm for post-training language models. However, we identify a pervasive *Scaling Law of Miscalibration*: while OPD effectively improves task accuracy, it systematically traps models in severe overconfidence. We trace this failure to an information mismatch: teacher supervision is formed under privileged context available during training, whereas the deployed model must report confidence using only deployment-time information. We formalize this perspective theoretically, showing that teacher-conditioned success is generally not a valid target for deployment-time confidence and that helpful privileged context induces *entropy collapse* and a systematic *optimism bias*. To address this, we propose a calibration-aware OPD framework, CaOPD, that estimates empirical confidence from model rollouts, replaces self-reported confidence with this student-grounded target, and distills the revised response through the same self-distillation pipeline. Experiments across various models and domains show that CaOPD achieves Pareto-optimal calibration while maintaining competitive capability, generalizing robustly under out-of-distribution and continual learning. Our findings highlight that capability distillation does not imply calibrated confidence, and that confidence should be treated as an essential objective in post-training.

> *"... AI tools do not rate their own confidence accurately. And this lowers their usefulness. We would appreciate more honest AIs."*
>
> — Terence Tao, *The Atlantic*, "The Edge of Mathematics" (2026)

1 INTRODUCTION

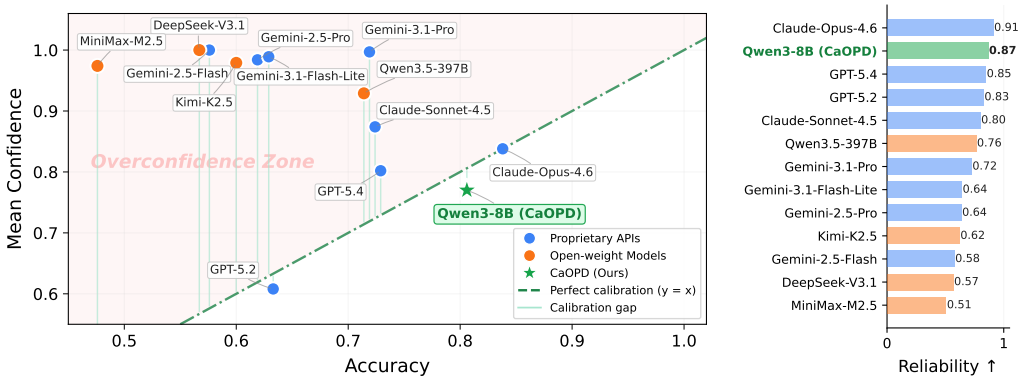

Figure 1: **The Scaling Law of Miscalibration.** (Left) Mean Confidence vs. Accuracy on Science Q&A. Almost all modern LLMs are trapped in the red *Overconfidence Zone*, exhibiting massive calibration gaps. Scaling up capability does not resolve this blind optimism. (Right) **CaOPD** structurally eliminates this bias by decoupling capability from calibration. It pulls the model back to the ideal calibration line, enabling a compact 8B model to achieve top-tier reliability (1-Brier Score) that rivals frontier LLMs (APIs & Open-weight Models).

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.

As LLMs achieve breakthroughs in complex reasoning and problem-solving, their role is shifting from text generators to autonomous agents and scientific co-pilots. For AI to be a trustworthy collaborator in rigorous domains, it must possess reliable *uncertainty signaling*, the ability to accurately self-assess and express doubt. Unfortunately, modern LLMs consistently fail this criterion. As shown in Figure 1, we observe a pervasive **Scaling Law of Miscalibration**: even the most advanced proprietary models (e.g., GPT (Achiam et al., 2023), Claude (Anthropic, 2024), Gemini (Team et al., 2023)) and scaled-up open-weight models (e.g., DeepSeek (Liu et al., 2024), Kimi (Team et al., 2026)) exhibit severe overconfidence. Scaling up capability does not resolve this blind optimism.

This widespread pathology is deeply rooted in the modern post-training paradigm. To achieve advanced capabilities, developers increasingly rely on **On-Policy Distillation (OPD)** (Agarwal et al., 2024; Lu & Lab, 2025; Ye et al., 2026) and self-distillation frameworks (Sang et al., 2026; Kim et al., 2026; Zhao et al., 2026). Methods such as SDPO (Hübotter et al., 2026) and SDFT (Shenfeld et al., 2026) generate high-quality training trajectories by conditioning a teacher policy on *privileged contexts*, such as environment verifier feedback, expert demonstrations, or ground-truth solutions. While OPD is phenomenally successful at transferring capability, it completely destabilizes confidence calibration. Indeed, as recently argued by Kalai et al. (2025), modern language models are effectively trapped in a "test-taking mode," incentivized to output plausible falsehoods rather than admitting uncertainty. We trace the algorithmic root of this epidemic directly to the OPD paradigm itself.

We demonstrate that this overconfidence is not a random artifact but a mathematical consequence of OPD, rooted in a fundamental *information asymmetry* between training and deployment (Pechyony & Vapnik, 2010). The teacher operates "open-book," conditioning on privileged context to produce low-entropy reasoning paths; the deployed model operates "closed-book" with only the user prompt. Minimizing the divergence against this privileged distribution forces the student to artificially sharpen its logits (*Entropy Collapse*; Müller et al., 2019) and to inherit the declarative style of success-filtered trajectories (*Optimism Bias*; Anthony et al., 2017). In short: capability transfers by imitation, but confidence cannot safely transfer across asymmetric information states. Existing efforts to tame overconfidence typically rely on Reinforcement Learning (RL) with explicit reward shaping, such as penalizing Brier scores via RLCR (Damani et al., 2025). While these methods can suppress absolute confidence, they suffer from a severe *capability tax* (Askell et al., 2021), the models become overly conservative and sacrifice fundamental reasoning accuracy to avoid RL penalties.

To break this capability-calibration trade-off, we propose **Calibration-Aware On-Policy Distillation (CaOPD)**. Rather than fighting the optimizer with complex reward shaping, CaOPD addresses the root cause via *target decoupling*. We separate *what* the model should answer from *how certain* it should be. We sample multiple rollouts from the student policy, which under SDPO are already generated at zero extra cost, to compute a student-grounded, unbiased empirical success rate. We then perform a non-invasive *target replacement*: we preserve the teacher's high-quality trajectory for capability cloning, but overwrite the implicit confidence token with the student's actual confidence. By distilling this revised joint target, CaOPD effectively amortizes the heavy computational expense of test-time confidence estimation directly into the parameters of a single-pass inference model.

Our main contributions are summarized as follows:

- **Identifying the Scaling Law of Miscalibration:** We empirically demonstrate that standard capability scaling and OPD inherently induce systematic overconfidence, a pervasive pathology that traps frontier LLMs in blind optimism.
- **Theoretical Formalization:** We mathematically formalize the information asymmetry in standard OPD, proving that distilling from a privileged teacher context inherently induces *entropy collapse* and *optimism bias*.
- **The CaOPD Framework:** We propose CaOPD, a compute-efficient post-training framework that decouples behavior imitation from confidence supervision. CaOPD amortizes expensive test-time empirical confidence into a single-pass model.
- **Pareto-Optimal Performance and Robustness:** Across OPD regimes (SDFT, SDPO) and domains (Science Q&A, Tool Use), CaOPD achieves Pareto-optimal calibration without the capability tax of RL-based methods, generalizing robustly under OOD shifts and preventing "calibration forgetting" during continual learning.

2 Problem Formulation and Preliminaries

In this section, we formally define the problem of confidence calibration within the context of On-Policy Distillation (OPD). We abstract recent self-distillation and post-training paradigms into a unified mathematical framework centered on information asymmetry, laying the groundwork for our theoretical analysis in Section 3.

2.1 On-Policy Distillation and Self-Distillation

Let $\mathcal{X}$ denote the space of input queries. To study confidence calibration explicitly, we augment the standard generation format with a *verbalized confidence instruction* (Kadavath et al., 2022b; Zhou et al., 2023; Tian et al., 2023), prompting the model to append an explicit confidence statement after its reasoning. Each complete generation $y = (y_1, \ldots, y_T)$ is partitioned into two contiguous segments: $y = (a, c)$, where $a = (y_1, \ldots, y_{T_a})$ is the reasoning trajectory (including the final answer), and $c = (y_{T_a+1}, \ldots, y_T)$ is the **confidence segment**, the token subsequence encoding a verbalized confidence statement such as "`Confidence:0.85`". We write $\mathrm{val}(c) \in [0, 1]$ for the scalar confidence score extracted by parsing the confidence segment c.

Modern OPD methods employ a *self-distillation* architecture in which a single model plays both roles: **Student policy** $\pi_\theta(\cdot \mid x)$: generates y conditioned *only* on the input query x, simulating deployment conditions; **Teacher policy** $\pi_\theta(\cdot \mid x, z)$: the *same* model, but additionally conditioned on a *privileged context* $z \sim \mathcal{Z}(x)$. Depending on the OPD paradigm, z may instantiate as an expert demonstration, environment feedback, or a ground-truth solution. The shared structural property is that the teacher is conditioned on strictly richer evidence than the deployed model will ever observe.

2.2 The Ideal Calibration Objective

Let $R : \mathcal{X} \times \mathcal{A} \to \{0, 1\}$ be an objective verification function, where $R(x, a) = 1$ indicates that the reasoning trajectory a yields a correct answer. For the deployed model, the true probability of success on input x is marginalized over its own generation distribution:

$$\mu(x) \;:=\; \mathbb{E}_{a \sim \pi_\theta(\cdot|x)}\big[R(x, a)\big] \;=\; \Pr_{\pi_\theta}(R = 1 \mid x). \tag{1}$$

A *calibrated* model should produce a confidence segment whose parsed value faithfully reflects this probability: $\mathbb{E}[\mathrm{val}(c_\theta(x))] = \mu(x)$. Therefore, $\mu(x)$ serves as the ground-truth target for confidence alignment.

2.3 The OPD Objective and Its Theoretical Blind Spot

Following SDFT and SDPO, on-policy self-distillation minimizes the *reverse* KL divergence between the student and the privileged teacher. The student first generates a trajectory $y \sim \pi_\theta(\cdot \mid x)$; both the student $\pi_\theta(\cdot \mid y_{<t}, x)$ and the teacher $\pi_\theta(\cdot \mid y_{<t}, x, z)$ then compute token-level distributions along this trajectory. The per-token reverse KL objective is:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) \;=\; \mathbb{E}_{x,\, z \sim \mathcal{Z}(x)} \;\; \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid y_{<t}, x) \,\big\|\, \pi_\theta(\cdot \mid y_{<t}, x, z)\big) \right]. \tag{2}$$

Because the generation $y = (a, c)$ includes confidence tokens, this loss operates over *all* token positions, both the reasoning segment and the trailing confidence segment. This is the theoretical blind spot: the teacher, armed with the privileged context z (e.g., the correct answer), produces a distribution at confidence positions that is concentrated near extreme values such as $c \approx 1.0$. Standard OPD therefore **entangles** two objectives: it transfers the teacher's reasoning capability *and simultaneously* forces the student to imitate the teacher's unjustified certainty.

3 Why Does Capability Distillation Exacerbate Overconfidence?

In this section, we demonstrate that this miscalibration is a strict mathematical consequence of OPD, providing both high-level intuition and formal propositions.

3.1 Intuition: Information Asymmetry and Distributional Warping

The root cause of OPD-induced overconfidence is the information asymmetry between the teacher's privileged evidence z and the student's accessible input x. When the student minimizes the per-token reverse KL against $\pi_\theta(\cdot \mid y_{<t}, x, z)$ (Equation equation 2) without access to z, it is forced to artificially sharpen its logits to approximate the teacher's deterministic-like behavior. At deployment, this translates into extreme verbalized confidence — the model learns the teacher's certainty without possessing the teacher's evidence. We now formalize the three mechanisms through which this asymmetry induces miscalibration.

3.2 Theoretical Analysis: The Mechanics of Miscalibration

Proposition 1 (Information Gap and Non-Identifiability). *Suppose the privileged teacher context Z is informative about the correctness R beyond the input X, such that the conditional mutual information $I(R; Z \mid X) > 0$. Then, the teacher-conditioned success probability $\mu_T(X, Z)$ is not measurable with respect to X alone. Specifically, there exists no deterministic function $g(X)$ such that $g(X) = \mu_T(X, Z)$ almost surely. Furthermore, under squared error, the optimal X-measurable predictor of $\mu_T(X, Z)$ is exactly the student's deployment success probability $\mu(X)$, leaving a strictly positive irreducible projection error:*

$$\min_g \mathbb{E}\left[\big(\mu_T(X, Z) - g(X)\big)^2\right] = \mathbb{E}_X\left[\mathrm{Var}\big(\mu_T(X, Z) \mid X\big)\right] > 0. \tag{3}$$

Proposition 2 (Entropy Collapse via Privileged Conditioning). *Let $H(\cdot)$ denote the Shannon entropy. Assume the privileged context Z provides actionable information about the correct reasoning trajectory A that is not present in X, such that the conditional mutual information $I(A; Z \mid X) > 0$. Then, the expected entropy of the teacher's trajectory distribution is strictly lower than the conditional entropy of the trajectories given only the accessible input X:*

$$\mathbb{E}_{X,Z}\big[H(\pi_\theta(A \mid X, Z))\big] < \mathbb{E}_X\big[H(A \mid X)\big]. \tag{4}$$

Furthermore, by minimizing the per-token reverse KL against $\pi_\theta(\cdot \mid y_{<t}, X, Z)$, the model $\pi_\theta(\cdot \mid y_{<t}, X)$ is penalized for expressing the natural uncertainty $H(A \mid X)$, forcing its internal logits to become artificially sharpened.

Proposition 3 (Selection Bias Induces Optimism). *In OPD, the privileged context Z is not sampled randomly, but is typically conditioned on successful or high-quality outcomes (e.g., demonstrations, correct solutions). Assume this training distribution of contexts, denoted $\mathcal{D}_{\textit{helpful}}$, is strictly helpful, meaning it increases the expected correctness of the teacher over the student's marginal capability: $\mathbb{E}_{Z \sim \mathcal{D}_{\textit{helpful}}}[\mu_T(X, Z) \mid X] \geq \mu(X)$, with strict inequality on a set of positive measure. Then, the implicit target distilled into the student is an upward-biased surrogate for the true deployment-time success probability:*

$$\mathbb{E}_{X,Z \sim \mathcal{D}_{\text{helpful}}}\big[\mu_T(X, Z) - \mu(X)\big] > 0. \tag{5}$$

Full proofs and extended discussions for all three propositions are provided in Appendix B.

4 Calibration-Aware On-Policy Distillation (CaOPD)

Theoretical analysis establishes a grim reality: capability distillation and confidence calibration are fundamentally misaligned under asymmetric information. OPD forces the student to chase an unidentifiable, artificially sharpened, and systematically optimistic target. To overcome this mathematical limitation, the supervision target for confidence *must* be explicitly decoupled from the teacher's capability trajectory. It must be anchored to an unbiased estimator of $\mu(x)$ that relies solely on the student's deployment-time information state. This mandate directly derives our proposed framework: **CaOPD**.

4.1 Constructing the Student-Grounded Target

The first step of CaOPD is to obtain an unbiased estimate of the model's true deployment-time success probability $\mu(x)$, without relying on the privileged context z. Since $\mu(x)$ is defined as

the expectation over the model's own generation distribution $\pi_\theta(\cdot \mid x)$ (Equation equation 1), we approximate it via Monte Carlo estimation. For a given input x, we sample K independent trajectories $(a_k, c_k) \sim \pi_\theta(\cdot \mid x)$ for $k = 1, \ldots, K$, evaluate each with the task verifier $R(x, a_k) \in \{0, 1\}$, and compute the empirical success rate: $\hat{\mu}(x) = \frac{1}{K} \sum_{k=1}^{K} R(x, a_k)$.

A key practical advantage is that on-policy OPD methods already generate multiple rollouts per prompt. Under SDPO, CaOPD *reuses the rollouts already generated by the base training loop* to compute $\hat{\mu}(x)$, requiring only a lightweight verifier evaluation on these existing trajectories. The distillation trajectory itself is sampled separately and incurs no extra cost beyond what SDPO already requires. By definition, $\hat{\mu}(x)$ is a student-grounded, execution-based target. While we utilize objective task verifiers in our primary evaluations, this target is highly modular; for open-ended domains lacking ground-truth verifiers, $\hat{\mu}(x)$ can be approximated using *Teacher-Anchored Self-Consistency* (Appendix C.6).

4.2 Target Replacement and the Decoupled Objective

With the empirical target $\hat{\mu}(x)$ computed, we integrate it into the OPD pipeline via a *Target Replacement* operation that modifies both the *completion* and the *teacher context* before computing the loss, while leaving the reverse-KL machinery entirely unchanged. Concretely, given a student-generated trajectory $y = (a, c)$:

(i) **Revise the completion.** Replace the confidence segment c with $\hat{\mu}(x)$, yielding the revised trajectory $\tilde{y} = (a,\ \hat{\mu}(x))$.

(ii) **Revise the teacher context.** In the privileged context z, overwrite the confidence score (originally near 1.0) with $\hat{\mu}(x)$, producing a revised context $\tilde{z}$.

Both the student and teacher then score the revised completion $\tilde{y}$ under their respective conditioning (x and $(x, \tilde{z})$). Recall that the token positions of $\tilde{y}$ partition into reasoning $\mathcal{I}_a = \{1, \ldots, T_a\}$ and confidence $\mathcal{I}_c = \{T_a{+}1, \ldots, T\}$. The CaOPD objective is:

$$\mathcal{L}_{\mathrm{CaOPD}}(\theta) \;=\; \mathbb{E}_{x,\,\tilde{z}}\ \mathbb{E}_{\tilde{y}}\Bigg[\underbrace{\sum_{t \in \mathcal{I}_a} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid \tilde{y}_{<t}, x) \;\big\|\; \pi_\theta(\cdot \mid \tilde{y}_{<t}, x, \tilde{z})\big)}_{\text{Capability Cloning (preserved)}} + \underbrace{\sum_{t \in \mathcal{I}_c} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid \tilde{y}_{<t}, x) \;\big\|\; \pi_\theta(\cdot \mid \tilde{y}_{<t}, x, \tilde{z})\big)}_{\text{Confidence Calibration (student-grounded)}} \Bigg]. \tag{6}$$

The target replacement is a clean decoupling of capability from calibration:

- **Capability Cloning is preserved.** At reasoning positions $t \in \mathcal{I}_a$, the prefix $\tilde{y}_{<t}$ contains only reasoning tokens (the confidence replacement has not yet appeared), and the privileged reasoning content in $\tilde{z}$ is left unmodified. The per-token KL at these positions is therefore *effectively identical* to standard OPD (Equation equation 2).
- **Confidence Imitation → Calibration.** At confidence positions $t \in \mathcal{I}_c$, the tokens being scored encode $\hat{\mu}(x)$ rather than the student's original c, and the teacher's context reflects $\hat{\mu}(x)$ rather than 1.0. The student is thus trained to produce the *student-grounded empirical success rate*, directly resolving the entropy collapse (Proposition 2) and optimism bias (Proposition 3).

Unlike RL-based calibration methods that inject scoring-rule penalties into the reward function, often causing optimization instability and a severe capability tax (Damani et al., 2025; Askell et al., 2021), CaOPD operates entirely within the standard distillation pipeline via target replacement, requiring no reward modification or additional optimization stages. The complete procedure is summarized in Algorithm 1; a visual overview is provided in Appendix C.1 (Figure 5).

Table 1: Mean confidence and OCG metrics across models, domains and methods.

Model	Domain	Base (OCG)	OPD		CaOPD (Ours)	
			SDFT (OCG)	SDPO (OCG)	SDFT (OCG)	SDPO (OCG)
Qwen3-8B	Science Q&A	0.897 (+58.7%)	0.972 (+48.1%)	0.935 (+12.9%)	**0.382** (-11.8%)	**0.770** (-3.3%)
	Tool Use	0.971 (+32.4%)	0.996 (+32.0%)	0.973 (+31.1%)	**0.699** (-0.7%)	**0.652** (-5.7%)
Olmo-3-7B-Instruct	Science Q&A	0.831 (+49.3%)	0.882 (+43.0%)	0.949 (+31.9%)	**0.529** (+8.1%)	**0.867** (+22.7%)
	Tool Use	0.911 (+36.7%)	0.981 (+30.5%)	0.985 (+30.5%)	**0.891** (+17.0%)	**0.786** (+11.0%)

Table 2: Main results comparing CaOPD against SDFT. Accuracy is reported in percentages. Values in parentheses indicate the absolute change relative to the SDFT baseline.

Method	Science Q&A				Tool Use			
	Acc. ↑	ECE ↓	BS ↓	SPR ↑	Acc. ↑	ECE ↓	BS ↓	SPR ↑
Qwen3-8B								
Base	31.0	0.587	0.559	0.307	64.7	0.324	0.328	0.396
+ SDFT	49.1	0.486	0.477	0.387	67.6	0.321	0.320	0.085
+ CaOPD	**50.0**	**0.266** (-0.22)	**0.281** (-0.20)	**0.599** (+0.21)	**70.6**	**0.228** (-0.09)	**0.242** (-0.08)	**0.555** (+0.47)
Olmo-3-7B-Instruct								
Base	33.8	0.492	0.462	0.488	54.4	0.367	0.381	0.351
+ SDFT	**45.2**	0.429	0.423	0.397	67.6	0.304	0.303	0.503
+ CaOPD	44.8	**0.176** (-0.25)	**0.260** (-0.16)	**0.568** (+0.17)	**72.1**	**0.177** (-0.13)	**0.186** (-0.12)	**0.580** (+0.08)

5 Experiments

5.1 Experiment Setup

Datasets and Models. We conduct experiments across two distinct domains: Science Question Answering (Science Q&A (Feng et al., 2024)) and Tool Use (ToolAlpaca (Tang et al., 2023)). We select two representative open-weight LLMs as our primary testbeds: Qwen3-8B (Yang et al., 2025) and Olmo-3-7B-Instruct (Olmo et al., 2025). We expand our evaluation to the broader Qwen3 family to conduct scaling analysis from 0.6B to 32B (see Appendix D.1).

Evaluation Metrics. We evaluate models across two dimensions: capability and calibration. For **capability**, we report final-answer **Accuracy**. For **calibration**, we report **Expected Calibration Error (ECE)** and **Brier Score (BS)** (Brier, 1950; Guo et al., 2017a; Desai & Durrett, 2020). To explicitly quantify the *direction* of miscalibration and the systematic optimism bias induced by post-training, we introduce the **Overconfidence Gap (OCG)**, defined as the mean confidence minus the empirical accuracy. A large positive OCG indicates severe overconfidence. Finally, to measure confidence **discrimination**, we introduce **Strict Pairwise Ranking (SPR)**. SPR measures the probability that a correct response receives a *strictly* higher confidence score than an incorrect one ($\mathbb{P}(c_+ > c_-)$). Unlike AUROC, SPR assigns zero credit to ties, heavily penalizing the confidence saturation (e.g., uniformly predicting $c = 1.0$) prevalent in OPD. More details are provided in Appendix D.2.

Baselines. We evaluate CaOPD against two distinct categories of post-training methods. To illustrate the severe overconfidence degradation inherent to capability-focused post-training, we benchmark against state-of-the-art *OPD paradigms* (**SDFT** (Shenfeld et al., 2026), **SDPO** (Hübotter et al., 2026)) as well as a standard RL baseline (**GRPO** (Shao et al., 2024)). To demonstrate CaOPD's superiority over specialized calibration techniques, we compare with *calibration-aware RL methods*, specifically **RLCR** (Damani et al., 2025) and **CAR** (Xuan et al., 2026), which explicitly penalize miscalibration via reward shaping. Detailed descriptions are provided in Appendix D.3.

5.2 Does On-Policy Distillation Inherently Exacerbate Overconfidence?

OPD exacerbates baseline overconfidence. As shown in Table 1, base instruction-tuned models already exhibit severe optimism bias (e.g., a massive **+58.7%** OCG for Qwen3-8B on Science Q&A). Strikingly, applying standard OPD methods (SDFT and SDPO) systematically *worsens* this pathology. Across almost all settings, OPD drives mean confidence toward extreme saturation (e.g., **0.996** on

Table 3: Comparison of CaOPD against RL-based baselines. Values in parentheses indicate the absolute change relative to the SDPO baseline on the same backbone and domain.

Method	Science Q&A				Tool Use			
	Acc. ↑	ECE ↓	BS ↓	SPR ↑	Acc. ↑	ECE ↓	BS ↓	SPR ↑
Qwen3-8B								
+ RLCR	65.8	0.143	0.152	0.546	61.9	0.175	0.218	0.556
+ CAR	61.6	0.151	0.148	0.645	60.2	0.181	0.229	0.540
+ GRPO	74.5	0.169	0.184	0.522	65.7	0.304	0.315	0.528
+ SDPO	**80.6**	0.157	0.177	0.571	66.2	0.298	0.303	0.504
+ CaOPD	80.3	**0.141** (-0.02)	**0.134** (-0.04)	**0.669** (+0.10)	**70.9**	**0.133** (-0.17)	**0.164** (-0.14)	**0.573** (+0.07)
Olmo-3-7B-Instruct								
+ RLCR	53.4	0.229	0.266	0.523	60.6	0.214	0.227	0.541
+ CAR	50.8	0.243	0.261	0.567	58.4	0.230	0.241	0.539
+ GRPO	59.7	0.353	0.322	0.546	**68.4**	0.286	0.275	0.533
+ SDPO	63.0	0.286	0.281	0.505	68.0	0.314	0.308	0.494
+ CaOPD	**64.0**	**0.241** (-0.05)	**0.210** (-0.07)	**0.654** (+0.15)	67.6	**0.180** (-0.13)	**0.197** (-0.11)	**0.562** (+0.07)

Tool Use), expanding the OCG (highlighted in red). This strongly validates our theoretical analysis: optimizing student behavior under a privileged teacher context induces an optimism bias, forcing the model to blindly inherit absolute certainty without possessing the underlying evidence.

CaOPD structurally realigns confidence. By decoupling capability imitation from confidence supervision, CaOPD successfully mitigates this dangerous drift. Replacing the teacher's privileged confidence with the student's empirical success drastically reduces mean confidence to realistic levels. Notably, for Qwen3-8B on Tool Use, CaOPD collapses the massive OCG from **+32.0%** (SDFT) down to an exceptionally aligned **-0.7%**. This demonstrates that CaOPD structurally corrects the confidence distribution, proving that true calibration can be explicitly learned during distillation.

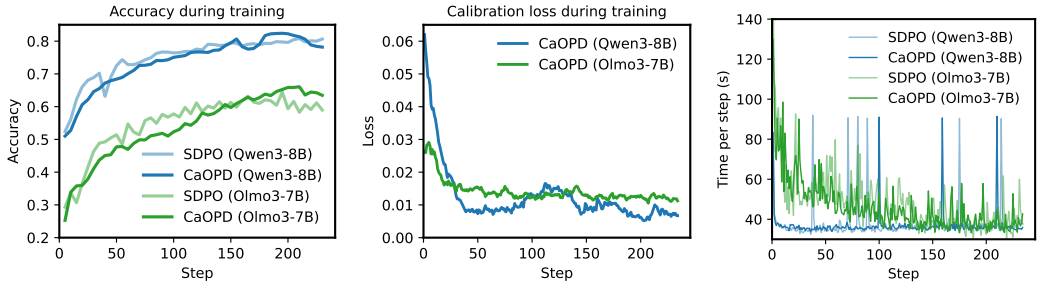

Figure 2: Optimization trajectories. CaOPD perfectly matches the accuracy improvements of standard SDPO (**left**) while simultaneously driving the calibration loss to convergence (**middle**), effectively bypassing the capability-calibration trade-off. The time per optimization step is nearly indistinguishable from SDPO (**right**), demonstrating that CaOPD introduces negligible latency during the gradient update phase.

5.3 Aligning Confidence Without Sacrificing Capability

The next critical question: *Can we explicitly calibrate the model's confidence without damaging the reasoning capabilities acquired through privileged distillation?*

Overcoming OPD Saturation with Strong Discrimination. Table 2 and 3 compare CaOPD against SDFT, SDPO, GRPO and other baselines across different domains. CaOPD drastically reduces ECE and BS while maintaining or strictly improving task accuracy. Crucially, true calibration requires confidence to accurately *rank* correctness, which is measured by SPR. For instance, on Qwen3-8B Tool Use, standard SDFT yields a highly low SPR of **0.085**, indicating a near-complete loss of discriminative power. CaOPD restores this metric to **0.555**, proving it actively learns to assign strictly higher confidence to correct answers over mistakes.

Avoiding the Capability Tax of RL-based Calibration. RL with explicit reward shaping is a common alternative for taming overconfidence. However, as shown in Table 3, pushing calibration

penalties into the RL reward fundamentally creates a capability-calibration trade-off, severely taxing the model's task performance. On Qwen3-8B, Science Q&A, RLCR and CAR drop accuracy to 65.8% and 61.6%, respectively, lagging far behind GRPO (74.5%) and SDPO (80.6%). In contrast, CaOPD operates via direct target replacement rather than fighting the RL optimizer. It matches or exceeds the high capability of SDPO (e.g., boosting Tool Use accuracy from 66.2% to 70.9%) while achieving superior calibration metrics.

Visualizing the Decoupling. Figure 2 explains *why* CaOPD bypasses this capability penalty. The left panel shows that CaOPD's accuracy trajectory perfectly shadows SDPO throughout training; the middle panel shows the calibration loss converging rapidly and stably. This confirms our core design principle: capability distillation and confidence calibration are algorithmically orthogonal. The model learns *what* to answer from the teacher's context, while learning *how certain* it should be from the empirical target.

Rollout Budget and Confidence Reshaping. We ablate the number of rollouts $K \in \{1, 2, 4, 8, 16, 32\}$ (Appendix E.3, Figure 6). Task accuracy remains flat across all K, confirming that capability is fully decoupled from confidence sampling variance. For calibration, $K \geq 8$ emerges as a compute-efficient sweet spot where the overconfidence gap collapses; smaller K yields overly quantized targets that trap the model in overconfidence. Qualitatively, standard OPD collapses the confidence distribution into a delta at 1.0; CaOPD restores a dispersed distribution that the single-pass model internalizes, placing uncertain predictions into lower bins and restoring its capacity to differentiate successes from failures.

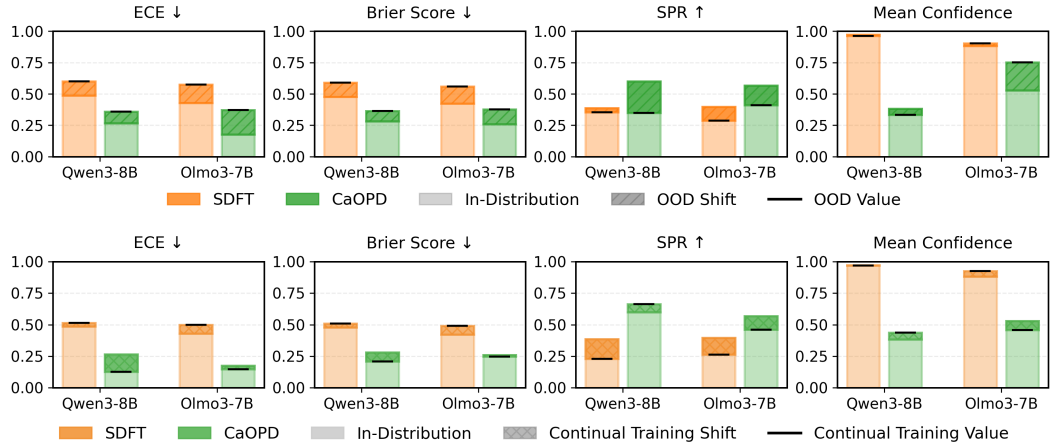

Figure 3: Generalization of Calibrated Confidence. Evaluation on the **Science Q&A (Chemistry)**(Feng et al., 2024) test domain under **(a)** Out-of-Distribution shift (top row; Train: Tool Use → Test: Chemistry) and **(b)** Continual Training (bottom row; Train: Tool Use → Chemistry → Test: Chemistry). Solid lighter bars represent In-Distribution (ID) baselines (trained on Chemistry), while hatched regions and black lines denote the absolute shift and final metric values after distribution changes.

5.4 Generalization and Continual Learning

A common vulnerability of post-training calibration is overfitting to training-domain verbal patterns, causing failures on unseen tasks. To verify whether CaOPD learns a robust uncertainty representation, we evaluate it under Out-of-Distribution (OOD) shifts and Continual Training (CT). We employ the SDFT backbone, which was explicitly designed to enable continual learning (Shenfeld et al., 2026). Figure 3 illustrates the performance shifts on Qwen3-8B. Full results across all models are deferred to Appendix E (Tables 7 and 8).

OOD Generalization: Learning a Meta-Skill. When transferring from Tool Use to Science Q&A(Feng et al., 2024) (Figure 3a), standard SDFT suffers a complete calibration collapse. Despite maintaining a reasonable OOD accuracy of 36.7%, its ECE spikes to a massive 0.599, and its SPR plummets to 0.353. In contrast, CaOPD generalizes exceptionally well, achieving an OOD ECE of 0.358 (a 40% relative reduction) and maintaining a healthy Overconfidence Gap. By grounding

supervision in empirical rollouts, CaOPD avoids memorizing the confidence distribution of the source training domain, instead learning a generalizable meta-skill: mapping its internal uncertainty to reliable verbalized confidence.

Preventing "Calibration Forgetting" in CT. In the CT setting, models are first trained on Tool Use, then continually trained on Science Q&A (Figure 3(b)). Prior work showed SDFT successfully prevents the catastrophic forgetting of *capabilities*. However, we reveal a critical new pathology: *capability preservation does not guarantee calibration preservation* (Figure 3b). SDFT's calibration is broken by sequential updates (ECE 0.513, SPR 0.230). CaOPD solves this "calibration forgetting." When trained continually, CaOPD matches SDFT's recovered accuracy but restores calibration, achieving a CT ECE of **0.126** and an SPR of **0.662**.

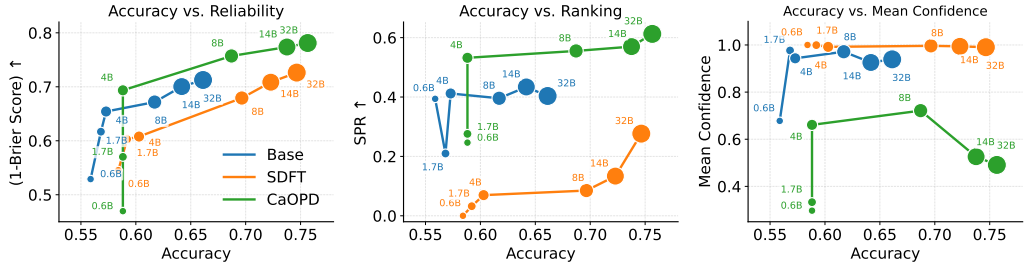

Figure 4: Decoupling Capability from Calibration across Model Scales. As Qwen3 scales from 0.6B to 32B (bubble size), standard SDFT (orange) fails to escape extreme overconfidence, remaining a flat horizontal line near 1.0 (**right**). In contrast, CaOPD (green) dynamically aligns confidence with actual capability, completely dominating the Pareto frontier for Reliability (1-BS) (**left**) and Discrimination SPR (**middle**), proving successful decoupling.

5.5 The Scaling Law of Miscalibration: Decoupling Capability from Calibration

A natural question is whether OPD-induced overconfidence is merely an artifact of smaller models. *Does scaling up parameters automatically cure miscalibration?*

Scaling rigidly entrenches blind optimism. As demonstrated in our macro-level analysis of leading LLMs (Figure 1, left), we observe a pervasive *Scaling Law of Miscalibration*: whether utilizing massive proprietary APIs or scaling up open-weight models, increasing capability (moving right along the X-axis) does not resolve the optimism bias. Models remain trapped in the overconfidence zone. To dissect this dynamically, Figure 4 tracks the Qwen3 family from 0.6B to 32B parameters. Under standard SDFT, mean confidence forms a flat horizontal line saturated near 1.0. Whether the model has 0.6B or 32B parameters, OPD forces it into absolute certainty, rigidly cementing the information asymmetry.

CaOPD achieves Pareto-optimal decoupling. CaOPD shatters this scaling curse. As models scale, CaOPD systematically pushes the Pareto frontier toward the top-right corner (Figure 4, left and middle), achieving strictly better Reliability and Discrimination (SPR). Crucially, CaOPD's mean confidence dynamically scales with its true accuracy, proving that capability distillation and calibration alignment can be safely decoupled. Ultimately, this decoupling enables a compact 8B model to achieve calibration quality rivaling frontier LLMs (Figure 1).

6 Conclusion

We identify a pervasive pathology in modern LLM post-training: the *Scaling Law of Miscalibration*. We demonstrate, both theoretically and empirically, that standard OPD traps models in an "illusion of certainty." To break this capability-calibration trade-off, we propose Calibration-Aware OPD (CaOPD), which decouples capability distillation from confidence supervision via a simple target replacement mechanism. Extensive evaluations confirm that CaOPD collapses the Overconfidence Gap across various scales, domains, and distributional shifts, without incurring the capability tax typical of RL-based reward shaping. By enabling models to produce calibrated confidence in a single forward pass, CaOPD provides a practical foundation for deploying trustworthy, uncertainty-aware agents.

AI USE STATEMENT

We disclose the use of LLMs in both our research and the preparation of this manuscript. For manuscript preparation, LLMs were utilized to assist in language editing and drafting content, and were employed to create specific graphical icons and conceptual illustrations featured in our diagrams (e.g., Figure 5). Within our research pipeline, LLMs were employed for automated evaluation (e.g., serving as an LLM-as-a-Judge for semantic equivalence in the Tool Use domain).

REFERENCES

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

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic AI, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Claude_3_Model_Family.pdf.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *The Fourteenth International Conference on Learning Representations*, 2025.

Glenn Wilson Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions. *arXiv preprint arXiv:2603.12273*, 2026.

Eunbi Choi, Yongrae Jo, Joel Jang, and Minjoon Seo. Prompt injection: Parameterization of fixed inputs. *arXiv preprint arXiv:2206.11349*, 2022.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*, 2025.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 295–302, 2020.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6577–6595, 2024.

Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017a.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017b.

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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022a.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022b.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023a.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023b.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. *arXiv preprint arXiv:2410.09724*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022a.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022b.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. *Advances in neural information processing systems*, 23, 2010.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. On-policy self-distillation for reasoning compression. *arXiv preprint arXiv:2603.05433*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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*, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

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

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. Openclaw-rl: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*, 2026.

Weihao Xuan, Qingcheng Zeng, Heli Qi, Yunze Xiao, Junjue Wang, and Naoto Yokoya. The confidence dichotomy: Analyzing and mitigating miscalibration in tool-use agents. *arXiv preprint arXiv:2601.07264*, 2026.

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.

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

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. Sac3: reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15445–15458, 2023.

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*, 2026.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori B Hashimoto. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5506–5524, 2023.

Contents

CONTENTS

1 Introduction **1**

2 Problem Formulation and Preliminaries **3**

2.1 On-Policy Distillation and Self-Distillation 3

2.2 The Ideal Calibration Objective 3

2.3 The OPD Objective and Its Theoretical Blind Spot 3

3 Why Does Capability Distillation Exacerbate Overconfidence? **3**

3.1 Intuition: Information Asymmetry and Distributional Warping 4

3.2 Theoretical Analysis: The Mechanics of Miscalibration 4

4 Calibration-Aware On-Policy Distillation (CaOPD) **4**

4.1 Constructing the Student-Grounded Target 4

4.2 Target Replacement and the Decoupled Objective 5

5 Experiments **6**

5.1 Experiment Setup 6

5.2 Does On-Policy Distillation Inherently Exacerbate Overconfidence? 6

5.3 Aligning Confidence Without Sacrificing Capability 7

5.4 Generalization and Continual Learning 8

5.5 The Scaling Law of Miscalibration: Decoupling Capability from Calibration 9

6 Conclusion **9**

A Related Work **16**

B Full Theoretical Statements and Proofs **16**

B.1 Proof of Proposition 1 16

B.2 Proof of Proposition 2 17

B.3 Proof of Proposition 3 19

C Algorithm and Implementation Details **20**

C.1 The CaOPD Framework Pipeline 20

C.2 Verbalized Confidence Prompt Templates 22

C.2.1 Domain: Science Q&A (Chemistry) 22

C.2.2 Domain: Tool Use (ToolAlpaca) 23

C.3 Constructing the Privileged Teacher Context (z) and Target Replacement 24

C.3.1 SDFT: Offline Demonstration Injection. 24

C.3.2 SDPO: Online Retrospection from Successful Rollouts. 26
C.4 Handling Target Non-Stationarity: Online Estimation of $\hat{\mu}(x)$ 27
C.5 Extended Discussion: Justification of the Empirical Target 28
C.6 Extending CaOPD Without Objective Verifiers: Teacher-Anchored Self-Consistency 28
C.7 Logit-Based vs. Verbalized Calibration: The Decoupling of Internal Entropy 29

D Additional Experimental Setup **30**
D.1 Datasets and Models 30
D.2 Extended Evaluation Metrics 31
D.3 Detailed Baseline Descriptions 31
D.4 Technical Setup and Hyperparameters 32

E Extended Results and Analysis **32**
E.1 Generalization and Continual Learning 32
E.2 The True Cost of Calibration: Test-Time Compute vs. Amortization 34
E.3 Detailed Analysis: Effect of Rollout Sample Size and Confidence Reshaping 35
E.4 Training Dynamics and Format Adherence Across Model Scales 35
E.5 Full API Evaluation Results and The Scaling Law of Miscalibration 36

F Qualitative Case Studies: The Illusion of Certainty **37**
F.1 Case Study 1: Restoring Capability and Calibrated Confidence 37
F.2 Case Study 2: Honest Failure and the "Illusion of Certainty" 38

G Broader Impact: Calibrated Small Models as Cascade Routers **39**

H Limitations and Future Work **40**

A Related Work

On-Policy Distillation for Post-Training. One recent paradigm shift in the field of LLM post-training is trending toward "On-Policy Distillation" (OPD). (Agarwal et al., 2024; Lu & Lab, 2025; Ye et al., 2026) and self-distillation (Sang et al., 2026; Kim et al., 2026; Zhao et al., 2026; Askell et al., 2021; Snell et al., 2022; Choi et al., 2022). Methods such as Self-Distillation Policy Optimization (SDPO) (Hübotter et al., 2026), Self-Distillation Fine-Tuning (SDFT) (Shenfeld et al., 2026), and Self-Distilled Reasoner (On-Policy Self-Distillation (OPSD)) (Zhao et al., 2026) have demonstrated that models can bootstrap their own reasoning capabilities by acting as their own teachers. A shared characteristic of these state-of-the-art frameworks is the use of *privileged context* such as environment verifier feedback, expert demonstrations, ground-truth solutions, or user interactions (Wang et al., 2026; Buening et al., 2026), to generate high-quality teacher trajectories. However, these works are overwhelmingly capability-centric. They focus on transferring the teacher's reasoning accuracy while mathematically ignoring the information asymmetry between the privileged teacher and the deployed student. Our work is the first to formalize how this asymmetry inherently induces entropy collapse and systematic overconfidence, and we propose a framework to fix this blind spot.

Confidence Calibration and Reward Shaping. The epidemic of overconfidence in modern LLMs has been widely documented, prompting significant interest in calibration methods (Guo et al., 2017b; Kadavath et al., 2022a; Lin et al., 2022a; 2023; Tian et al., 2023; Groot & Valdenegro-Toro, 2024). The prevailing approach involves Reinforcement Learning (RL) augmented with explicit reward shaping. Works such as RLCR (Damani et al., 2025), Rewarding Doubt (Bani-Harouni et al., 2025), CAR (Xuan et al., 2026), and Taming Overconfidence (Leng et al., 2024) integrate proper scoring rules (e.g., Brier score penalties) directly into the PPO (Schulman et al., 2017) or RL objective to penalize overconfident mistakes. While effective at suppressing absolute confidence, these RL-based interventions often suffer from optimization instability and a severe *capability tax* (Askell et al., 2021), the models become overly conservative, and their fundamental reasoning accuracy degrades to avoid penalties. In contrast, CaOPD avoids fighting the RL optimizer. By employing a non-invasive *target replacement* strategy on the confidence token, CaOPD achieves Pareto-optimal alignment, decoupling capability distillation from confidence calibration without sacrificing capability.

Test-Time Amortization and Hallucination. As recently highlighted by Kalai et al. (2025), modern language models are heavily incentivized to behave as aggressive "test-takers," opting to bluff or hallucinate rather than accurately signaling uncertainty (Kuhn et al., 2023a; Geng et al., 2024; Kirchhof et al., 2025). To detect and mitigate such hallucinations, substantial research has leveraged test-time multi-sampling. Frameworks like SelfCheckGPT (Manakul et al., 2023) and SAC^3 (Zhang et al., 2023) estimate uncertainty by measuring the consistency across multiple independent generations. While highly reliable, these sampling-based approaches increase inference latency and compute costs by a factor of $O(K)$, making them prohibitive for real-time or agentic deployment. CaOPD acknowledges that multi-sampling yields the reliable uncertainty signal, but fundamentally shifts *when* this computation occurs. By calculating the empirical success rate during the data generation phase and distilling it into the model, CaOPD amortizes the expensive test-time search cost into a single-pass inference model, resolving the hallucination-bluffing pathology with zero deployment (inference) overhead.

B Full Theoretical Statements and Proofs

B.1 Proof of Proposition 1

Statement: Let $\mu(X) := \mathbb{P}(R = 1 \mid X)$ and $\mu_T(X, Z) := \mathbb{P}(R = 1 \mid X, Z)$. If $I(R; Z \mid X) > 0$, then $\mu_T(X, Z)$ is not X-measurable. The unique square-integrable minimizer $g^*(X)$ of $\mathbb{E}[(\mu_T(X, Z) - g(X))^2]$ is $\mu(X)$, with a minimum achievable error of $\mathbb{E}[\mathrm{Var}(\mu_T(X, Z) \mid X)]$.

Proof: We prove the proposition in three steps.

Step 1: $\mu_T(X, Z)$ is not X-measurable. Recall that $\mu_T(X, Z) = \mathbb{E}[R \mid X, Z]$. Suppose, for the sake of contradiction, that $\mu_T(X, Z)$ is X-measurable. Then there exists some measurable function $g(X)$ such that $\mathbb{E}[R \mid X, Z] = g(X)$ almost surely. Taking the conditional expectation with respect to X

on both sides yields:

$$\mathbb{E}\big[\mathbb{E}[R \mid X, Z] \mid X\big] = \mathbb{E}[g(X) \mid X] \implies \mathbb{E}[R \mid X] = g(X) \text{ a.s.} \tag{7}$$

This implies that $\mathbb{E}[R \mid X, Z] = \mathbb{E}[R \mid X]$ almost surely. Consequently, the conditional distribution of R given (X, Z) depends only on X, which means R and Z are conditionally independent given X. This is equivalent to $I(R; Z \mid X) = 0$, which directly contradicts our assumption that $I(R; Z \mid X) > 0$. Therefore, $\mu_T(X, Z)$ cannot be X-measurable.

Step 2: The optimal X-measurable predictor is the conditional expectation. We seek to minimize $\mathbb{E}[(\mu_T(X, Z) - g(X))^2]$ over all X-measurable functions g. By the standard property of minimum Mean Squared Error (MSE) estimation, the unique minimizer is the conditional expectation of the target given the observable variables:

$$g^*(X) = \mathbb{E}_{Z|X}\big[\mu_T(X, Z) \mid X\big]. \tag{8}$$

Step 3: The optimal projection equals deployment-time success. We substitute the definition of $\mu_T(X, Z)$ into the optimal predictor $g^*(X)$ and apply the Law of Total Expectation (the tower property):

$$g^*(X) = \mathbb{E}_{Z|X}\big[\mathbb{E}[R \mid X, Z] \mid X\big] = \mathbb{E}[R \mid X] = \mu(X). \tag{9}$$

Finally, substituting $g^*(X) = \mu(X)$ into the MSE objective yields the irreducible error:

$$\begin{aligned}\mathbb{E}\left[\big(\mu_T(X, Z) - \mu(X)\big)^2\right] &= \mathbb{E}_X\left[\mathbb{E}_{Z|X}\left[\big(\mu_T(X, Z) - \mathbb{E}[\mu_T(X, Z) \mid X]\big)^2 \mid X\right]\right] \\ &= \mathbb{E}_X\left[\mathrm{Var}\big(\mu_T(X, Z) \mid X\big)\right].\end{aligned}$$

Since $\mu_T(X, Z)$ is not X-measurable (from Step 1), the variance $\mathrm{Var}(\mu_T(X, Z) \mid X)$ is strictly positive on a set of non-zero measure, making the irreducible projection error strictly greater than zero. This completes the proof. ■

What this means for OPD: Proposition 1 formally explains the *Information Gap* introduced in Section 3.1. Because the teacher uses extra evidence z (e.g., a correct demonstration) to form its internal confidence, the student simply cannot "guess" this exact confidence value at test time. The best the student could theoretically do is predict its own marginal success rate $\mu(x)$. However, standard OPD minimizes the KL divergence point-wise on each trajectory, forcing the student to memorize the inaccessible $\mu_T(x, z)$ rather than learning the optimal projection $\mu(x)$. This structural mismatch guarantees that confidence distillation will fail to produce calibrated predictions.

Takeaway 1: The Information Gap in Practice

The mathematical impossibility of recovering the teacher's exact confidence $\mu_T(X, Z)$ from X alone means that standard OPD forces the student to chase a "ghost" target. The proof demonstrates that the theoretically optimal strategy for the student is to predict its own marginal success rate $\mu(X)$. However, the standard OPD loss function prevents this, coercing the student to memorize inaccessible context cues. This structural mismatch mathematically guarantees that confidence distillation, if left coupled with capability cloning, will fail to produce calibrated predictions at deployment.

B.2 Proof of Proposition 2

Statement: Let A represent the generated reasoning trajectory, X the user input, and Z the privileged context. If the context is informative about the trajectory such that $I(A; Z \mid X) > 0$, then the expected entropy of the model's distribution under privileged conditioning is strictly lower than the conditional entropy given only X: $\mathbb{E}_{X,Z}[H(\pi_\theta(A \mid X, Z))] < H(A \mid X)$.

Proof: The proof relies on the fundamental properties of Shannon entropy and conditional mutual information.

Step 1: Expanding the Conditional Entropy. By definition, the conditional entropy $H(A \mid X)$ measures the average uncertainty remaining about the reasoning trajectory A when only the input X

is observed:

$$H(A \mid X) = -\sum_{x \in \mathcal{X}} P(x) \sum_{a \in \mathcal{A}} P(a \mid x) \log P(a \mid x). \tag{10}$$

Similarly, the expected entropy of the model under privileged conditioning (i.e., conditioned on both X and Z) is:

$$H(A \mid X, Z) = -\sum_{x \in \mathcal{X}, z \in \mathcal{Z}} P(x, z) \sum_{a \in \mathcal{A}} \pi_\theta(a \mid x, z) \log \pi_\theta(a \mid x, z) = \mathbb{E}_{X,Z}\big[H(\pi_\theta(A \mid X, Z))\big]. \tag{11}$$

Step 2: Applying the Chain Rule of Information. Using the chain rule for Shannon entropy, the relationship between these two quantities can be expressed through the conditional mutual information $I(A; Z \mid X)$, which quantifies the reduction in uncertainty about A provided by knowing Z, given that X is already known:

$$H(A \mid X) = H(A \mid X, Z) + I(A; Z \mid X). \tag{12}$$

Step 3: Utilizing the Non-Negativity of Mutual Information. By the assumption of the proposition, the privileged context Z is informative, meaning $I(A; Z \mid X) > 0$. Rearranging the equation from Step 2, we obtain:

$$H(A \mid X, Z) = H(A \mid X) - I(A; Z \mid X). \tag{13}$$

Since $I(A; Z \mid X) > 0$, it strictly follows that:

$$H(A \mid X, Z) < H(A \mid X). \tag{14}$$

Substituting the expectation notation back into the inequality yields:

$$\mathbb{E}_{X,Z}\big[H(\pi_\theta(A \mid X, Z))\big] < \mathbb{E}_X\big[H(A \mid X)\big]. \tag{15}$$

Step 4: Implication on the Reverse KL Objective. In standard OPD, the model minimizes the per-token reverse KL divergence (Equation equation 2). For a given prefix $y_{<t}$ generated by the model in student mode, the per-token loss decomposes as:

$$D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid y_{<t}, x) \,\big\|\, \pi_\theta(\cdot \mid y_{<t}, x, z)\big) = \sum_{v \in \mathcal{V}} \pi_\theta(v \mid y_{<t}, x) \log \frac{\pi_\theta(v \mid y_{<t}, x)}{\pi_\theta(v \mid y_{<t}, x, z)}, \tag{16}$$

where $\mathcal{V}$ is the token vocabulary. The reverse KL is *mode-seeking*: it heavily penalizes the model (in student mode) for placing probability mass on tokens where the privileged-conditioned distribution $\pi_\theta(\cdot \mid y_{<t}, x, z)$ assigns low probability. Because the privileged distribution is sharply concentrated (as proven in Steps 1–3), the student-mode distribution $\pi_\theta(\cdot \mid y_{<t}, x)$ is driven to collapse its mass onto the same narrow set of high-probability tokens. This forces the model's internal logits to become artificially sharpened in order to match the teacher's certainty, even though the model in student mode lacks access to the mutual information $I(A; Z \mid X)$ that justifies this concentration. This completes the proof. ■

What this means for OPD: Proposition 2 formally explains the *Entropy Collapse* mechanism introduced in Section 3.1. Because the model under privileged conditioning possesses the "cheat sheet" z (e.g., correct demonstrations or verifier feedback), its next-token distribution is highly concentrated (low entropy). The reverse KL objective forces the model in student mode to match this sharp distribution. Being mode-seeking, the reverse KL drives the model to collapse its probability mass onto the teacher's high-confidence modes, drastically scaling up its internal logit magnitudes even though it lacks the evidence z to justify such certainty. When the model is subsequently asked to verbalize its confidence, it translates this artificially low-entropy internal state into an unjustified, near-absolute confidence score (e.g., $\mathrm{val}(c) = 0.99$).

Takeaway 2: Artificial Sharpness and Logit Collapse

This proof bridges information theory with model optimization dynamics. By demonstrating that $H(A \mid X, Z) < H(A \mid X)$, we show that the model under privileged conditioning

inherently operates with a lower-entropy (sharper) distribution. Because the reverse KL is mode-seeking, it forces the model in student mode to concentrate its probability mass on the teacher's high-confidence modes, artificially magnifying its internal logits during training. At test time, this manifests as extreme verbalized certainty. In essence, the model learns the *magnitude* of privileged-conditioned confidence without possessing the underlying *mutual information* $I(A; Z \mid X)$ that justifies it.

B.3 Proof of Proposition 3

Statement: Let $\mu(X) := \mathbb{P}(R = 1 \mid X)$ and $\mu_T(X, Z) := \mathbb{P}(R = 1 \mid X, Z)$. Let $\mathcal{D}_{\text{helpful}}$ be the distribution of the privileged context Z used during OPD training. Assume that conditioning on $Z \sim \mathcal{D}_{\text{helpful}}$ strictly improves expected correctness on a set of inputs $\mathcal{S} \subseteq \mathcal{X}$ with $\mathbb{P}(X \in \mathcal{S}) > 0$, such that $\mathbb{E}_{Z\sim\mathcal{D}_{\text{helpful}}}[\mu_T(X, Z) \mid X = x] > \mu(x)$ for all $x \in \mathcal{S}$, and $\mathbb{E}_{Z\sim\mathcal{D}_{\text{helpful}}}[\mu_T(X, Z) \mid X = x] \geq \mu(x)$ otherwise. Then, $\mathbb{E}_{X,Z\sim\mathcal{D}_{\text{helpful}}}[\mu_T(X, Z)] > \mathbb{E}_X[\mu(X)]$.

Proof: We evaluate the expectation of the difference between the teacher-conditioned success target and the true deployment-time success probability.

Step 1: Applying the Law of Total Expectation. We take the expectation of the difference over the joint training distribution of inputs X and privileged contexts $Z \sim \mathcal{D}_{\text{helpful}}$. By the linearity of expectation and the Law of Total Expectation (the tower property), we can condition on X:

$$\begin{aligned}\mathbb{E}_{X,Z\sim\mathcal{D}_{\text{helpful}}}\big[\mu_T(X, Z) - \mu(X)\big] &= \mathbb{E}_X\Big[\mathbb{E}_{Z\sim\mathcal{D}_{\text{helpful}}}\big[\mu_T(X, Z) - \mu(X) \mid X\big]\Big] \\ &= \mathbb{E}_X\Big[\mathbb{E}_{Z\sim\mathcal{D}_{\text{helpful}}}\big[\mu_T(X, Z) \mid X\big] - \mu(X)\Big]. \end{aligned} \tag{17}$$

Note that $\mu(X)$ acts as a constant with respect to the inner expectation over Z.

Step 2: Incorporating the Helpful Context Assumption. By the premise of the proposition, the training context Z is selected to be helpful (e.g., successful rollouts or gold demonstrations). Therefore, for every input X, the expected success probability of the teacher under this filtered context distribution is greater than or equal to the student's marginal success probability without the context:

$$\mathbb{E}_{Z\sim\mathcal{D}_{\text{helpful}}}\big[\mu_T(X, Z) \mid X\big] - \mu(X) \geq 0 \quad \text{almost surely.} \tag{18}$$

Step 3: Establishing the Strict Inequality. Furthermore, we assumed this inequality is strict on a subset of inputs $\mathcal{S}$ with non-zero probability measure ($\mathbb{P}(X \in \mathcal{S}) > 0$). Let the difference be denoted by $\Delta(X) = \mathbb{E}_{Z\sim\mathcal{D}_{\text{helpful}}}[\mu_T(X, Z) \mid X] - \mu(X)$. We can split the outer expectation over $\mathcal{X}$ into $\mathcal{S}$ and its complement $\mathcal{S}^c$:

$$\mathbb{E}_X[\Delta(X)] = \int_{\mathcal{S}} \Delta(x) d\mathbb{P}(x) + \int_{\mathcal{S}^c} \Delta(x) d\mathbb{P}(x). \tag{19}$$

Since $\Delta(x) > 0$ on $\mathcal{S}$ and $\mathbb{P}(\mathcal{S}) > 0$, the first integral is strictly positive. Since $\Delta(x) \geq 0$ on $\mathcal{S}^c$, the second integral is non-negative. Therefore:

$$\mathbb{E}_X[\Delta(X)] > 0. \tag{20}$$

Substituting back the definition of $\Delta(X)$, we conclude:

$$\mathbb{E}_{X,Z\sim\mathcal{D}_{\text{helpful}}}\big[\mu_T(X, Z) - \mu(X)\big] > 0. \tag{21}$$

This proves that mimicking the teacher under success-conditioned privileged context mathematically injects a strictly positive optimism bias into the student's expected confidence. ■

What this means for OPD: Proposition 3 is the mathematical formalization of the *Optimism Bias* introduced in Section 3.1. Methods like SDFT and SDPO do not train on random garbage; they specifically construct z to ensure that teacher has high-quality contexts. Consequently, the student is exclusively trained to imitate the highly certain, declarative linguistic style of a *successful* derivation. The model perfectly learns the "voice of a winner." At test time, when the student generates an incorrect answer due to its limited capability, it lacks the mechanism to express doubt. Instead, it blindly applies this inherited optimistic tone, resulting in the massive Overconfidence Gaps (OCG) we empirically observed in Table 1.

Takeaway 3: The Origin of the Optimism Bias

By evaluating the expectation over the successful subset of contexts ($\mathcal{D}_{\text{helpful}}$), the proof reveals a systemic flaw in distilling only high-quality trajectories. Because the student is mathematically shielded from failure cases during supervision, it severely underestimates the complexity of the prompt distribution. Consequently, the student exclusively learns the declarative, confident tone of a successful derivation. It blindly applies this inherited optimistic tone even on out-of-distribution or overly difficult test prompts, fully explaining the massive Overconfidence Gaps (OCG) we observe empirically.

C Algorithm and Implementation Details

C.1 The CaOPD Framework Pipeline

To provide an intuitive understanding of the Calibration-Aware On-Policy Distillation (CaOPD) framework, Figure 5 illustrates the complete end-to-end pipeline used during the post-training data generation and optimization phases. The framework explicitly decouples capability learning from confidence alignment through a five-step procedure:

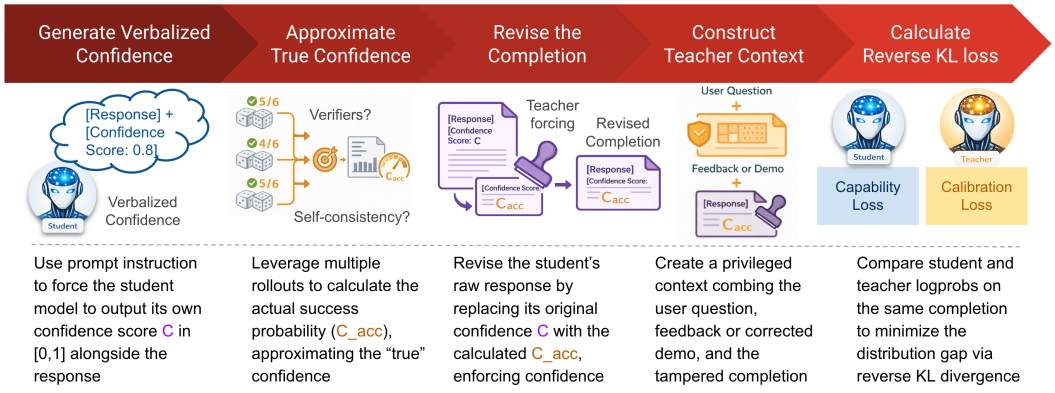

Figure 5: **Overview of the CaOPD Framework.** The pipeline consists of five stages: (1) querying the student for a base response with verbalized confidence, (2) approximating the true empirical confidence via student rollouts and an objective verifier, (3) revising the completion by replacing the original confidence tokens with the empirical success rate, (4) constructing the privileged teacher context with the same revised confidence, and (5) computing the per-token reverse KL between the student and teacher distributions on the revised completion.

1. Generate Verbalized Confidence: Given an input prompt x, the model (in student mode, i.e., conditioned only on x) generates a complete trajectory $y = (a, c)$, where a is the reasoning trajectory and c is the confidence segment. In standard OPD, $\text{val}(c)$ would typically be near 1.0, reflecting the teacher's optimism inherited during training.

2. Approximate True Confidence: Instead of trusting the single-pass verbalized confidence, we estimate the student's actual competence via empirical execution. We sample K independent rollouts $(a_k, c_k) \sim \pi_\theta(\cdot \mid x)$ and evaluate each with an objective task verifier to compute the empirical success rate $\hat{\mu}(x) = \frac{1}{K}\sum_{k=1}^{K} R(x, a_k)$. This serves as a statistically rigorous, student-grounded approximation of the true deployment-time confidence.

3. Revise the Completion (Target Replacement): This is the core decoupling step of CaOPD. We take the student's reasoning trajectory a from Step 1, but explicitly discard its original confidence segment c. We construct the revised completion $\tilde{y} = (a,\ \hat{\mu}(x))$ by overwriting the confidence tokens with the string representation of the empirical success rate. This guarantees that the supervision target for confidence reflects the student's actual competence rather than the teacher's privileged certainty.

4. Construct Revised Teacher Context: To ensure the model learns advanced reasoning capabilities, we construct a privileged context z by prepending the user question with environmental feedback, ground-truth demonstrations, or correct solutions, creating an "open-book" setting for the teacher. Crucially, the confidence score within z — originally near 1.0 — is also overwritten with $\hat{\mu}(x)$, producing a revised context $\tilde{z}$. This ensures that both the completion and the teacher's conditioning reflect the same student-grounded confidence target.

5. Compute Reverse KL Loss: Both the student $\pi_\theta(\cdot \mid \tilde{y}_{<t}, x)$ and the teacher $\pi_\theta(\cdot \mid \tilde{y}_{<t}, x, \tilde{z})$ perform a forward pass on the revised completion $\tilde{y}$. The per-token reverse KL divergence (Equation equation 6) is computed between these two distributions. As discussed in Section 4.2, this naturally decomposes into a *Capability Cloning* component at reasoning positions and a *Confidence Calibration* component at confidence positions.

The detailed pseudocode for this entire process is formally summarized in Algorithm 1.

Key Properties of CaOPD The CaOPD framework fundamentally restructures how post-trained LLMs learn to express uncertainty, exhibiting three desirable properties:

- **True Decoupling of Behavior and Self-Assessment.** Unlike RL methods (e.g., RLCR) that inject calibration penalties into a global reward function — often degrading task accuracy — CaOPD operates via direct target replacement within the standard distillation pipeline. The loss (Equation equation 6) preserves the teacher's capability signal at reasoning positions while redirecting the confidence signal to the student-grounded target.
- **Meta-Skill Generalization.** Because CaOPD supervises confidence using the student's actual execution outcomes rather than arbitrary heuristic penalties, the model learns a generalizable meta-skill: mapping its internal uncertainty to an accurate verbalized scale. As demonstrated in Section 5.4, this allows calibrated confidence to generalize out-of-distribution and persist through continual training.
- **Test-Time Compute Amortization.** Obtaining reliable confidence via self-consistency or multi-sampling (i.e., computing $\hat{\mu}(x)$ online) is computationally prohibitive at deployment. CaOPD computes $\hat{\mu}(x)$ **during the training phase** via iterative on-policy rollouts. By distilling this signal into the model's parametric weights, CaOPD amortizes the cost of test-time search into a single-pass inference model.

Algorithm 1 Calibration-Aware On-Policy Distillation (CaOPD)

Require: Dataset $\mathcal{D}$, Model π_θ, Verifier R, Rollout budget K

for each input $x \in \mathcal{D}$ **do**
  // 1. Student-Grounded Confidence Estimation
  for $k = 1$ **to** K **do**
    Sample student rollout $(a_k, c_k) \sim \pi_\theta(\cdot \mid x)$
  end for
  Compute empirical confidence $\hat{\mu}(x) = \frac{1}{K} \sum_{k=1}^{K} R(x, a_k)$

  // 2. Target Replacement
  Sample student trajectory $y = (a, c) \sim \pi_\theta(\cdot \mid x)$
  Revise completion: $\tilde{y} \leftarrow (a,\ \hat{\mu}(x))$ ▷ overwrite confidence tokens
  Construct privileged context $z \sim \mathcal{Z}(x)$ ▷ e.g., demo, feedback
  Revise teacher context: $\tilde{z} \leftarrow z$ with confidence overwritten by $\hat{\mu}(x)$

  // 3. Distillation
  Compute $\mathcal{L}_{\mathrm{CaOPD}}(\theta)$: per-token reverse KL between $\pi_\theta(\cdot \mid \tilde{y}_{<t}, x)$ and $\pi_\theta(\cdot \mid \tilde{y}_{<t}, x, \tilde{z})$ (Eq. equation 6)
  Update θ with optimizer (e.g., AdamW)
end for

C.2 Verbalized Confidence Prompt Templates

To elicit explicitly verbalized confidence scores from the language models, we adopt a simple, plug-and-play prompting strategy. Rather than designing entirely new prompts, we seamlessly integrate a standardized confidence instruction into the existing task-specific formatting templates. The core instruction appended to the formatting rules is:

```
Confidence: your confidence level (0.0-1.0) regarding
your answer, must be in numerical format, no other words or
explanation.
```

Below, we provide the full prompt templates and example model outputs for both the Science Q&A and Tool Use domains.

C.2.1 Domain: Science Q&A (Chemistry)

For the Science Q&A domain, the model is tasked with generating a reasoning trajectory, selecting the correct multiple-choice option, and appending its confidence.

```
Given a question and four options, please select the right answer.
    Respond in the following format:
<reasoning>
...
</reasoning>
<answer>
...
</answer>
Confidence: your confidence level (0.0-1.0) regarding your answer, must
    be in numerical format, no other words or explanation.

For the answer, only output the letter corresponding to the correct
    option (A, B, C, or D), and nothing else. Do not restate the answer
    text. For example, if the answer is "A", just output:
<reasoning>
...
</reasoning>
<answer>
A
</answer>
Confidence: your confidence level (0.0-1.0) regarding your answer, must
    be in numerical format, no other words or explanation.

Begin!
Question: Which of the following molecules is the only correct reactant
    used in the synthesis of the product
    "Nc1ccn(Cc2ccc(Cl)cc2C(F)(F)F)" ?
A: O=C1c2ccccc2C(=O)N1c1ccn(Cc2ccc(Cl)cc2C(F)(F)F)n1
B: ...
C: ...
D: ...
Please reason step by step.
```

Listing 1: User prompt: Science Q&A (Chemistry)

```
<reasoning>
The target product is a benzimidazolinone-like scaffold ...
</reasoning>
<answer>
A
</answer>
Confidence: 0.8
```

Listing 2: Example model generation: Science Q&A

C.2.2 DOMAIN: TOOL USE (TOOLALPACA)

For the Tool Use domain, we build upon the ReAct (Reasoning and Acting) format. The model needs to read API documentation and output its thought, the selected tool action, the JSON-formatted action input, and finally, its confidence in the success of the action.

```
Your task is to answer the user's question using available tools.
You have access to the following tools:
Name: Axolotl
Description: Collection of axolotl pictures and facts
Documentation:
getRandomAxolotlImage: Retrieve a random axolotl image with information on the
image source.
Parameters: {}
Output: Successful response.
- Format: application/json
- Structure: Object{url, source, description}
searchAxolotlImages: Search for axolotl images based on specific criteria such as
color, gender, and size.
Parameters: {"color": "string. One of: [wild, leucistic, albino]. The color of the
axolotl (e.g., 'wild', 'leucistic', 'albino', etc.).", "gender": "string. One
of: [male, female]. The gender of the axolotl ('male', 'female').", "size": "
string. One of: [small, medium, large]. The size of the axolotl ('small', '
medium', 'large').", "page": "integer. The page number for pagination purposes
."}
Output: Successful response.
- Format: application/json
- Structure: Object{results: Array[Object{url, source, description}], pagination:
Object{current_page, total_pages, total_results}}
getAxolotlFacts: Retrieve interesting facts about axolotls such as their habits,
habitats, and physical characteristics.
Parameters: {"category": "string. One of:[habits, habitat, physical
characteristics]. The category of facts to retrieve (e.g., 'habits', 'habitat',
'physical characteristics').", "limit": "integer. The maximum number of facts
to return."}
Output: Successful response.
- Format: application/json
- Structure: Array[Object{fact, source}]

Use the following format:
Thought: you should always think about what to do
Action: the action to take, should be one of the tool names.
Action Input: the input to the action, must be in JSON format. All of the action input must be realistic and from the user.
Confidence: your confidence level (0.0-1.0) regarding the success of this action, must be in numerical format, no other words or explanation.

Begin!
Question: Hey, can you show me a random picture of an axolotl?
```

Listing 3: User prompt: Tool Use (incorporating API documentation)

```
Thought: Retrieve a random axolotl image using the available tool.
Action: Axolotl
Action Input: {}
Confidence: 0.95
```

Listing 4: Example model generation: Tool Use

C.3 Constructing the Privileged Teacher Context (z) and Target Replacement

In Section 3, we theorized that On-Policy Distillation (OPD) suffers from information asymmetry because the teacher, the same model conditioned on a privileged context z, produces artificially certain distributions. To concretize this theoretical framework, we detail exactly how this privileged context is constructed in practice for both the SDFT and SDPO baselines, and crucially, **where CaOPD intervenes to replace the confidence target**.

Target Replacement during CaOPD: By structuring the output with an explicit `Confidence: [value]` suffix, it becomes trivial to extract the scalar confidence score using rule-based parsing. During the CaOPD training phase (Algorithm 1), we perform two replacements: (1) in the *student's completion*, we parse the confidence segment, drop the original numeric value, and overwrite it with the string representation of $\hat{\mu}(x)$; (2) in the *teacher's privileged context* z, we likewise overwrite any confidence value (typically 1.0) with $\hat{\mu}(x)$, producing the revised context $\tilde{z}$. The per-token reverse KL loss is then computed on this revised completion under both the student conditioning (x) and the teacher conditioning $(x, \tilde{z})$.

While the student is prompted strictly with the user query (Appendix C.2), the teacher receives a heavily augmented prompt, the same query prepended with the privileged context $\tilde{z}$. Below, we provide the **exact, full text** of the teacher prompts from our Tool Use domain experiments. We use colored annotations to highlight the exact token where CaOPD overwrites the default overconfidence with the student's empirical target $\hat{\mu}(x)$.

C.3.1 SDFT: Offline Demonstration Injection.

In standard SDFT, the privileged context z is constructed offline using the ground-truth golden responses provided in the dataset. The ground truth is injected into the prompt as a flawless in-context demonstration, effectively transforming the generation into an "open-book" copying task.

```
Your task is to answer the user's question using available tools.
You have access to the following tools:
Name: Domainsdb.info
Description: Registered Domain Names Search
Documentation:
searchDomains: Search for registered domain names based on keywords
Parameters: {"keywords": "Required. string. Keywords to search for in
    domain names.", "page": "integer. Page number for pagination.",
    "perPage": "integer. Number of results per page."}
Output: List of domain names matching the provided keywords.
 - Format: application/json
 - Structure: Array[Object{domainName, creationDate, expirationDate,
    registrarName}]
getDomainDetails: Retrieve domain name registration details for a
    specific domain
Parameters: {"domain": "Required. string. The domain name for which
    registration details are to be retrieved."}
Output: Registration details for the specified domain.
 - Format: application/json
 - Structure: Object{domainName, creationDate, expirationDate,
    registrarName}
listAllDomains: Retrieve a list of all registered domain names
Parameters: {"page": "integer. Page number for pagination.", "perPage":
    "integer. Number of results per page."}
Output: List of all registered domain names.
 - Format: application/json
```

```
 - Structure: Array[Object{domainName, creationDate, expirationDate, registrarName}]
getRegistrarDetails: Retrieve details for a specific domain registrar
Parameters: {"registrar": "Required. string. The name of the domain registrar for which details are to be retrieved."}
Output: Details for the specified domain registrar.
 - Format: application/json
 - Structure: Object{registrarName, contactInformation, website}
listRegistrars: Retrieve a list of all domain registrars
Parameters: {"page": "integer. Page number for pagination.", "perPage": "integer. Number of results per page."}
Output: List of all domain registrars.
 - Format: application/json
 - Structure: Array[Object{registrarName, contactInformation, website}]

Use the following format:
Thought: you should always think about what to do
Action: the action to take, should be one of the tool names.
Action Input: the input to the action, must be in JSON format. All of the action input must be realistic and from the user.
Confidence: your confidence level (0.0-1.0) regarding the success of this action, must be in numerical format, no other words or explanation.

Begin!
Question: I'm evaluating multiple domain registrars for my new business website. Can you provide me with a list of the top 10 domain registrars and their contact information, website, and other relevant details?

This is an example for a response to the question:
 I need to get a list of all domain registrars and their details.
Action: listRegistrars
Action Input: {"page": 1, "perPage": 10}
 The listRegistrars function did not return any results, so I will use the getRegistrarDetails function to get the details of the top 10 domain registrars.
Action: getRegistrarDetails
Action Input: {"registrar": "GoDaddy"}
 I have retrieved the details of the first domain registrar, GoDaddy. I will now use the listRegistrars function to get the details of the other 9 domain registrars.
Action: listRegistrars
Action Input: {"page": 1, "perPage": 10}
Confidence: 1.0 ←[CaOPD overwrites with empirical μ̂(x), e.g., 0.8]

Now answer with a response of your own, including the thinking process.
```

Listing 5: Full SDFT Teacher Prompt (Tool Use). The ground truth is injected as a golden example. CaOPD intercepts and modifies the final confidence score.

```
Thought: I need to retrieve a list of domain registrars and their details. The listRegistrars tool can be used to get the list of domain registrars, and then I can extract the relevant information from the response.
Action: listRegistrars
Action Input: {"page": 1, "perPage": 10}
Confidence: 1.0 ←[CaOPD overwrites with empirical μ̂(x), e.g., 0.8]
```

Listing 6: Model Output (SDFT Target). CaOPD overwrites this token before distillation.

C.3.2 SDPO: ONLINE RETROSPECTION FROM SUCCESSFUL ROLLOUTS.

In SDPO, the privileged context z is constructed dynamically. The system samples multiple trajectories, uses the verifier to identify a successful rollout from the current batch, and then prepends this successful trajectory as the `"Correct solution"`. The teacher policy is asked to "Correctly solve the original question" while explicitly observing this verified correct path, inevitably leading to a highly saturated confidence generation.

```
Your task is to answer the user's question using available tools.
You have access to the following tools:
Name: Axolotl
Description: Collection of axolotl pictures and facts
Documentation:
getRandomAxolotlImage: Retrieve a random axolotl image with information on the image source.
Parameters: {}
Output: Successful response.
 - Format: application/json
 - Structure: Object{url, source, description}
searchAxolotlImages: Search for axolotl images based on specific criteria such as color, gender, and size.
Parameters: {"color": "string. One of:[wild, leucistic, albino]. The color of the axolotl.", "gender": "string. One of: [male, female]. The gender of the axolotl.", "size": "string. One of:[small, medium, large]. The size of the axolotl.", "page": "integer. The page number for pagination purposes."}
Output: Successful response.
 - Format: application/json
 - Structure: Object{results: Array[Object{url, source, description}], pagination: Object{current_page, total_pages, total_results}}
getAxolotlFacts: Retrieve interesting facts about axolotls such as their habits, habitats, and physical characteristics.
Parameters: {"category": "string. One of: [habits, habitat, physical characteristics]. The category of facts to retrieve.", "limit": "integer. The maximum number of facts to return."}
Output: Successful response.
 - Format: application/json
 - Structure: Array[Object{fact, source}]

Use the following format:
Thought: you should always think about what to do
Action: the action to take, should be one of the tool names.
Action Input: the input to the action, must be in JSON format. All of the action input must be realistic and from the user.

Begin!
Question: Hey, can you show me a random picture of an axolotl?
Correct solution:

<think>
Okay, the user is asking for a random axolotl picture. Let me check the available tools. There's the getRandomAxolotlImage function which requires no parameters. That's perfect because the user didn't specify any details, just a random request. The other tools like searchAxolotlImages need specific criteria, which aren't provided here. So using getRandomAxolotlImage is the right choice. No need for any input parameters. I'm pretty confident this will work.
</think>

Thought: The user is requesting a random axolotl image, and the `getRandomAxolotlImage` tool is specifically designed to fulfill this need without requiring any additional parameters. This ensures a straightforward and efficient solution.

```

```
Action: getRandomAxolotlImage
Action Input: {}
Confidence: 0.8 ←[CaOPD overwrites with empirical $\hat{\mu}(x)$]


Correctly solve the original question.
```

Listing 7: Full SDPO Teacher Prompt (Tool Use). A verified successful rollout from the current batch is dynamically injected.

```
<think>
Okay, the user is asking for a random axolotl picture. Let me check the
    available tools. There's the getRandomAxolotlImage function which
    requires no parameters. That's perfect because the user didn't
    specify any details, just a random request. The other tools like
    searchAxolotlImages need specific criteria, which aren't provided
    here. So using getRandomAxolotlImage is the right choice. No need
    for any input parameters. I'm pretty confident this will work.
</think>

Thought: The user is requesting a random axolotl image, and the
    `getRandomAxolotlImage` tool is specifically designed to fulfill
    this need without requiring any additional parameters. This ensures
    a straightforward and efficient solution.

Action: getRandomAxolotlImage
Action Input: {}
Confidence: 0.8 ←[CaOPD overwrites with empirical $\hat{\mu}(x)$]
```

Listing 8: Model Output (SDPO Target). CaOPD replaces the saturated confidence token before computing the reverse-KL loss.

C.4 Handling Target Non-Stationarity: Online Estimation of $\hat{\mu}(x)$

A critical consideration in the CaOPD framework is the potential non-stationarity of the empirical confidence target. As the model π_θ undergoes gradient updates and improves its reasoning capabilities, an empirical target $\hat{\mu}(x)$ computed from a stale snapshot would rapidly become misaligned with the student's true competence.

To strictly prevent target staleness, our implementation of CaOPD does *not* rely on a static, pre-computed offline dataset. Instead, $\hat{\mu}(x)$ is dynamically refreshed within an iterative, on-policy training loop. Specifically, the training dynamics proceed as follows:

1. **EMA Teacher Tracking:** In self-distillation, the teacher is the same model with different conditioning (Section 2.1). To stabilize the teacher's distribution during training, we maintain an Exponential Moving Average (EMA) (Tarvainen & Valpola, 2017) copy of the model weights θ_{ema}, updated at every gradient step: $\theta_{\text{ema}} \leftarrow (1-\alpha)\,\theta_{\text{ema}} + \alpha\,\theta$. The teacher distribution is then $\pi_{\theta_{\text{ema}}}(\cdot \mid \tilde{y}_{<t}, x, \tilde{z})$, providing a smoothed target for the reverse KL.
2. **Dynamic Rollout Generation:** Periodically during training, the latest model weights θ are synchronized to a high-throughput inference engine (vLLM (Kwon et al., 2023)). The engine generates a fresh batch of K *student* rollouts $(a_k, c_k) \sim \pi_\theta(\cdot \mid x)$ for each prompt on-the-fly.
3. **Online Target Computation:** These newly generated student rollouts are immediately evaluated by the objective task verifier $R(x, a_k)$. The empirical success rate $\hat{\mu}(x) = \frac{1}{K}\sum_{k=1}^{K} R(x, a_k)$ is computed strictly from this *current* batch of rollouts.
4. **Target Replacement:** The freshly computed $\hat{\mu}(x)$ is used to construct the revised completion $\tilde{y}$ and revised teacher context $\tilde{z}$ for the subsequent gradient optimization steps.

This online, on-policy generation mechanism ensures that the supervision signal for confidence dynamically adapts to the model's learning trajectory. As the model becomes more capable, the

fresh rollouts yield a higher success rate, and the target $\hat{\mu}(x)$ naturally scales upward. Consequently, CaOPD completely avoids the optimization instability associated with stale targets in standard off-policy algorithms.

Note on Terminology: When we refer to the computational cost of rollouts being "amortized offline" in the main text, we specifically mean that this cost is absorbed *prior to deployment* (i.e., during the training phase), rather than requiring $O(K)$ test-time compute from the end-user.

C.5 Extended Discussion: Justification of the Empirical Target

A potential concern regarding the CaOPD framework is whether the sampling-based empirical frequency, $\hat{\mu}(x)$, serves as a rigorous representation of the model's true internal confidence, rather than merely a heuristic proxy. We argue that $\hat{\mu}(x)$ is, in fact, the most statistically rigorous and unbiased target for confidence alignment, supported by both the mathematical definition of calibration and extensive literature on epistemic uncertainty.

Statistical Unbiasedness. By the strict definition of calibration, a model's predicted confidence should perfectly equal its expected accuracy on a given prompt. For the deployed model $\pi_\theta(\cdot \mid x)$, its true expected success probability is exactly the marginal expectation over its own generation distribution: $\mathbb{E}_{a\sim\pi_\theta}[R(x,a)]$. Therefore, the Monte Carlo estimator $\hat{\mu}(x) = \frac{1}{K}\sum_{k=1}^{K} R(x,a_k)$ is a statistically unbiased estimate of the model's objective capability on query x. Training the model to produce $\mathrm{val}(c) = \hat{\mu}(x)$ by replacing the confidence tokens in the distillation target (Section 4.2), ensures that the verbalized confidence honestly reflects the model's own generation distribution.

Connection to Behavioral Consistency and Hallucination Detection. Our formulation strongly aligns with established findings in the hallucination and uncertainty estimation literature. Extensive prior work has demonstrated that sampling-based behavioral consistency is the "gold standard" for measuring an LLM's epistemic uncertainty. For instance, **Self-Consistency** (Wang et al., 2022) leverages multiple rollouts to find the most reliable reasoning paths. More directly, hallucination detection frameworks such as **SelfCheckGPT** (Manakul et al., 2023) and **SAC3** (Semantic-aware Cross-check) (Zhang et al., 2023) operate on the principle that when an LLM hallucinates (i.e., operates outside its knowledge boundary), its sampled responses exhibit high entropy and self-contradiction. Conversely, confident and factual knowledge yields highly consistent samples.

CaOPD explicitly adopts this gold-standard consistency signal as the training target. However, while methods like SelfCheckGPT are applied at *test time*, incurring an expensive $O(K)$ inference cost, CaOPD computes this signal during the training-phase data generation (Appendix C.4). By substituting it as the confidence target in the revised completion $\tilde{y}$, CaOPD *amortizes* the expensive test-time search cost into the model's parametric weights, achieving self-consistency-level calibration in a single $O(1)$ forward pass.

C.6 Extending CaOPD Without Objective Verifiers: Teacher-Anchored Self-Consistency

A potential critique of CaOPD is its reliance on an objective task verifier $R(x,a)$ to compute the empirical target $\hat{\mu}(x)$. To address this, we first clarify the dependency structure of On-Policy Distillation (OPD), and then demonstrate how CaOPD can seamlessly operate in open-ended domains using Teacher-Anchored Self-Consistency.

The "Free Lunch" of Verifier Dependency in OPD. It is crucial to recognize that the requirement for an objective verifier or ground-truth label is not uniquely introduced by CaOPD; it is a fundamental prerequisite of the underlying OPD paradigms themselves. SDPO explicitly requires environment verifiers (e.g., compilers or reward models) to generate the privileged feedback z. SDFT requires ground-truth demonstrations to construct the privileged teacher context z. Because the base OPD algorithms already assume access to verification signals to construct the privileged context, CaOPD simply reuses this *existing* infrastructure to score the student rollouts, introducing **zero additional assumptions or dependencies**.

Teacher-Anchored Self-Consistency. In scenarios where an objective verifier is strictly unavailable (e.g., open-ended alignment tasks), CaOPD can substitute $R(x,a)$ with an unsupervised agreement metric. We propose *Teacher-Anchored Self-Consistency*. Instead of computing the majority vote

Table 4: Ablation study on the confidence supervision target using Qwen3-8B. Teacher-Anchored Self-Consistency (TA-SC) serves as an unsupervised empirical target when an objective verifier is unavailable. While TA-SC experiences minor degradations due to teacher fallibility and spurious disagreements, it still drastically outperforms the standard SDFT baseline, effectively collapsing the overconfidence gap.

Method (Qwen3-8B)	Science Q&A				Tool Use			
	Acc. ↑	ECE ↓	BS ↓	SPR ↑	Acc. ↑	ECE ↓	BS ↓	SPR ↑
SDFT (Baseline)	49.1	0.486	0.477	0.387	67.6	0.321	0.320	0.085
CaOPD (Objective Verifier)	**50.0**	**0.266**	**0.281**	**0.599**	**70.6**	**0.228**	**0.242**	**0.555**
CaOPD (TA-SC)	49.5	0.271	0.285	0.577	68.3	0.240	0.251	0.547

among the student's K rollouts — which is vulnerable to "consistently wrong" snowballing hallucinations — we generate a reference trajectory from the model under privileged conditioning, $a_{\text{ref}} \sim \pi_\theta(\cdot \mid x, z)$, and compute the agreement rate between the student rollouts and this reference:

$$\hat{\mu}_{\text{SC}}(x) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\Big(\text{Equiv}(a_k, a_{\text{ref}})\Big) \tag{22}$$

where $\text{Equiv}(\cdot, \cdot)$ is a semantic equivalence function (e.g., LLM-as-a-Judge).

This formulation solves the consensus hallucination problem: if the student consistently converges on a false premise (e.g., 8 identical wrong answers), but the model under privileged conditioning provides the correct answer, the equivalence score remains exactly $\hat{\mu}_{\text{SC}}(x) = 0/8$. The confidence target is properly set to 0.0, accurately reflecting the student's inability to reach the correct state without privileged information.

Empirical Validation of TA-SC. To empirically validate the effectiveness of Teacher-Anchored Self-Consistency (TA-SC), we conducted an ablation study using Qwen3-8B under the SDFT backbone. We replaced the exact objective verifier with the unsupervised TA-SC target. Specifically, we utilized a rule-based exact-match for Science Q&A, and a lightweight LLM judge to evaluate semantic equivalence for Tool Use.

The results, summarized in Table 4, align with our theoretical expectations. While CaOPD with TA-SC exhibits a slight degradation compared to the objective verifier (e.g., ECE increases marginally from 0.266 to 0.271 on Science Q&A), it substantially outperforms the standard SDFT baseline.

The slight performance drop observed in the Tool Use domain (Accuracy decreases from 70.6% to 68.3%, SPR from 0.555 to 0.547) directly corroborates the *Spurious Disagreements* failure mode: open-ended JSON generations are inherently more susceptible to false-negative equivalence checks than rigid multiple-choice formats. Nevertheless, TA-SC drastically reduces the ECE (from 0.321 to 0.240) and restores discriminative ranking capability (SPR from 0.085 to 0.547) compared to standard distillation. This confirms that CaOPD is highly modular and maintains its decoupling even when relying on generative consensus in the absence of ground-truth verifiers.

C.7 Logit-Based vs. Verbalized Calibration: The Decoupling of Internal Entropy

A fundamental question regarding the CaOPD framework is its choice of targeting explicitly *verbalized* confidence rather than internal token-level probability distributions (logits), and whether altering this verbalized output inadvertently corrupts the model's internal reasoning confidence.

Why Verbalized Confidence? As summarized in Table 5, estimating LLM uncertainty typically falls into three paradigms: logit-based (Kadavath et al., 2022b), sampling-based (Kuhn et al., 2023b; Wang et al., 2022), and verbalized confidence (Lin et al., 2022b). While logit-based methods (e.g., perplexity) are mathematically straightforward, they suffer from critical deployment limitations: they are entirely inaccessible for black-box proprietary APIs and require complex length-normalization for long-horizon reasoning trajectories. Sampling-based methods are robust but computationally prohibitive ($O(K)$ latency) at test time.

Table 5: Comparison of Uncertainty Estimation Paradigms for LLMs. CaOPD targets Verbalized Confidence because it is the most deployment-friendly paradigm (API-compatible and single-pass), overcoming its inherent "overconfidence" limitation via rollout-grounded distillation.

Method Paradigm	Mechanism	Blackbox / API?	Single Pass?	Key Limitations
Logit-based (e.g., Perplexity, Entropy)	Leverages probability distribution over individual tokens	✗	✓	Long responses require complex normalization; inaccessible via most commercial APIs.
Sampling-based (e.g., SAC3, Semantic Entropy)	Examines semantic consistency across multiple generated samples	✓	✗	Computationally intensive at deployment; increases latency by a factor of $O(K)$.
Verbalized Confidence (e.g., P(True))	Asks the LLM itself to state its confidence inline via text	✓	✓	**Overconfident by default** under OPD; requires alignment to express honest doubt.

Verbalized confidence represents the ideal deployment paradigm — it requires only a single forward pass and operates perfectly via black-box API interactions. However, as documented in our work, its fatal flaw is that it is severely miscalibrated and overconfident by default after standard post-training. CaOPD specifically targets this paradigm because solving verbalized overconfidence unlocks the most scalable and universally applicable form of uncertainty signaling.

Does Verbalized Humility Bleed into Internal Logit Entropy? A natural question is whether training the model to output low verbalized confidence targets ($\hat{\mu}(x) \approx 0.0$) inadvertently causes the internal token-level distributions at reasoning positions ($t \in \mathcal{I}_a$) to become flattened. In other words, does the induced "humility" at confidence positions destructively bleed into the model's core reasoning capability?

Our empirical training dynamics provide a clear answer: **No.** The autoregressive structure of the generation $\tilde{y} = (a, \hat{\mu}(x))$ provides a natural *causal decoupling*. The per-token reverse KL (Equation equation 6) at reasoning positions $t \in \mathcal{I}_a$ is computed on prefixes $\tilde{y}_{<t}$ that contain only reasoning tokens — the confidence replacement has not yet appeared in the autoregressive context. Consequently, the teacher's distribution at these positions remains anchored to the privileged reasoning signal from $\tilde{z}$, unaffected by the confidence replacement. The gradient adjustments pulling the confidence tokens toward $\hat{\mu}(x)$ are isolated to the final few sequence positions ($t \in \mathcal{I}_c$).

This decoupling is empirically confirmed by the left panel of Figure 2: CaOPD's accuracy trajectory perfectly shadows SDPO throughout training. If the low-confidence targets at $\mathcal{I}_c$ were corrupting the logit entropy at $\mathcal{I}_a$, we would observe a tangible capability degradation. Instead, the Transformer architecture successfully maintains sharp internal distributions for reasoning while selectively routing epistemic uncertainty into the allocated confidence tokens.

D Additional Experimental Setup

D.1 Datasets and Models

To evaluate the confidence calibration of models post-trained via on-policy distillation (OPD), we conduct experiments across two distinct domains that require complex multi-step reasoning and are highly susceptible to overconfidence: Science Question Answering (Science Q&A) and Tool Use.

- **Science Q&A (Chemistry):** We utilize the Chemistry L-3 subset from SciKnowEval (Feng et al., 2024) to test knowledge-intensive scientific reasoning. This dataset provides a rigorous testbed for evaluating whether a model's confidence aligns with its factual accuracy in domain-specific contexts. Following standard practices, the data is partitioned into approximately 75% training, 5% validation, and 20% test splits.
- **Tool Use:** To assess calibration in long-horizon, sequential decision-making, we evaluate on the ToolAlpaca dataset (Tang et al., 2023). This domain requires the model to map user requests to the correct API specifications, testing its ability to remain calibrated across extended reasoning traces and execution steps. We follow the original train-test split provided by the authors.

Following recent advancements in post-training and self-distillation, we select two representative open-weight language models as our primary testbeds: Qwen3-8B and Olmo-3-7B-Instruct. This selection

enables us to observe calibration behaviors across different model families and tuning paradigms under the same distillation pipeline. Furthermore, to systematically analyze the relationship between model scale, in-context learning (ICL) capability, and calibration effectiveness, we expand our evaluation to the broader Qwen3 family. Specifically, we conduct a comprehensive scaling analysis encompassing models at 0.6B, 1.7B, 4B, 8B, 14B, and 32B parameters.

D.2 Extended Evaluation Metrics

We introduce several metrics to comprehensively evaluate the capability and calibration of post-trained language models. Here, we provide their formal definitions and theoretical justifications.

Overconfidence Gap (OCG). While standard proper scoring rules like Expected Calibration Error (ECE) and Brier Score (BS) measure absolute conditional miscalibration, they obscure the *direction* of the error (i.e., whether the model is overconfident or underconfident). To explicitly quantify the systematic optimism bias induced by post-training, we track OCG, defined as the difference between the expected verbalized confidence and the empirical accuracy over the test set:

$$\mathrm{OCG} = \mathbb{E}\big[\mathrm{val}(c_\theta(X))\big] - \mathbb{E}[R] \tag{23}$$

A large positive OCG strictly indicates severe overconfidence, the primary pathology induced by standard OPD, whereas a value near zero reflects well-calibrated marginal expectations.

Strict Pairwise Ranking (SPR) vs. AUROC. We introduce Strict Pairwise Ranking (SPR) to measure the discrimination ability of the model's confidence. This metric is specifically designed to expose *confidence saturation*, a severe pathology observed in on-policy distillation where models uniformly report maximum confidence (e.g., $c = 1.0$) regardless of actual correctness.

Let c_+ denote the confidence score assigned to a correct response, and c_- denote the confidence score assigned to an incorrect response. The standard Area Under the Receiver Operating Characteristic curve (AUROC) can be formulated probabilistically as:

$$AUROC = \mathbb{P}(c_+ > c_-) + 0.5 \cdot \mathbb{P}(c_+ = c_-) \tag{24}$$

AUROC awards partial credit (0.5) when the confidence scores are tied ($c_+ = c_-$). Consequently, a completely saturated model that outputs $c = 1.0$ for every single response will yield an AUROC of 0.5. While statistically representing a "random guessing" baseline, this value often visually obscures the complete collapse of the model's discriminative power.

To strictly penalize this saturation, SPR removes the credit awarded for tied predictions:

$$SPR = \mathbb{P}(c_+ > c_-) \tag{25}$$

Under SPR, a tie yields 0 points. Therefore, if a model uniformly predicts $c = 1.0$, its SPR drops exactly to 0. This metric directly answers the most critical question for deployment: *Does the model actually assign a strictly higher confidence to its correct answers compared to its mistakes?* By penalizing uninformative ties, SPR provides a much more sensitive and realistic measure of confidence discrimination for post-trained language models.

D.3 Detailed Baseline Descriptions

To comprehensively evaluate our proposed CaOPD, we compare it against two distinct categories of post-training methods: capability-focused on-policy training and calibration-aware reinforcement learning (RL).

1. Capability-Focused Paradigms. These methods primarily optimize for task success (capability) and serve to empirically illustrate the severe overconfidence degradation typical in standard post-training pipelines.

- **SDFT** (Shenfeld et al., 2026) and **SDPO** (Hübotter et al., 2026): These represent the current state-of-the-art in on-policy self-distillation. The model generates its own trajectories and is trained via reverse KL against itself conditioned on privileged context (e.g., demonstrations or environment feedback).
- **GRPO** (Shao et al., 2024): GRPO serves as a standard reinforcement learning baseline. It demonstrates the calibration effects of standard reward maximization without any specialized confidence penalties.

2. Calibration-Aware RL Methods. To demonstrate that CaOPD is superior not only to standard distillation but also to specialized calibration techniques, we benchmark against state-of-the-art methods that explicitly shape rewards to penalize miscalibration.

- **RLCR** (Damani et al., 2025): A reinforcement learning approach that integrates proper scoring rules (such as the Brier score) directly into the PPO reward function to strictly penalize overconfident mistakes.
- **CAR (Calibration-Aware Reward)** (Xuan et al., 2026): Derived from the recent Confidence Dichotomy framework, CAR explicitly penalizes the dichotomy between high verbalized confidence and low accuracy. In our experiments, we adapted the RLCR training pipeline by incorporating CAR's specialized calibration-penalized reward formulation.

Comparing CaOPD against these RL-based interventions allows us to assess whether our student-grounded empirical target replacement provides a fundamentally better capability-calibration trade-off than explicit, scalar reward shaping.

D.4 Technical Setup and Hyperparameters

Compute Infrastructure. All experiments were conducted on a single compute node equipped with four NVIDIA H200 GPUs. Our software environment is built on top of the NVIDIA PyTorch container (`nvcr.io/nvidia/pytorch:25.02-py3`), utilizing CUDA 12.8 and PyTorch v2.7.0. To build CaOPD on SDFT, all experiments were conducted using the HuggingFace TRL (von Werra et al., 2020) library, while for developing on top of SDPO, the training pipeline is implemented based on the `verl` library (Sheng et al., 2025), and we employ PyTorch Fully Sharded Data Parallel (FSDP2) for memory-efficient distributed training. For high-throughput rollout generation during the empirical target construction phase, we leverage vLLM (Kwon et al., 2023), which enables highly optimized batched inference on the multi-GPU node.

Hyperparameters. The detailed hyperparameters used across our post-training runs for both Qwen3-8B and Olmo-3-7B-Instruct are summarized in Table 6.

E Extended Results and Analysis

E.1 Generalization and Continual Learning

In Section 5.4, we presented the aggregated results demonstrating CaOPD's robustness under Out-of-Distribution (OOD) shifts and Continual Learning. Tables 7 and 8 provide the exhaustive metrics for these settings across the Qwen3-8B and Olmo-3-7B-Instruct backbones. Here, we provide the implementation details and expand on the underlying mechanisms driving these results.

Implementation Details for Continual Training (CT). To rigorously evaluate the models in a continual learning scenario, we adopt a straightforward sequential fine-tuning approach without introducing domain-specific replay buffers or complex regularization terms. Specifically, we first train the base model on Domain A (e.g., Tool Use) for the standard number of epochs to convergence. We then initialize the training for Domain B (e.g., Science Q&A) using this exact checkpoint, and perform **Continual Training (CT)** on Domain B for the exact same number of epochs and with identical hyperparameters (e.g., learning rate, batch size) as the standalone ID training. This isolates the effect of the optimization objective (SDFT vs. CaOPD) on preserving or forgetting calibration.

Why does standard OPD fail OOD? The massive calibration collapse observed in standard SDFT (e.g., OOD ECE soaring to 0.599 in Table 7) occurs because standard OPD forces the model to memorize the marginal confidence distribution of the source training domain. When shifted to an unseen OOD domain where the model's intrinsic capability is strictly lower, the model blindly applies the highly confident linguistic tone (Optimism Bias) it memorized from the source domain. CaOPD prevents this by training the model to act as its own calibrator. Because the empirical target $\hat{\mu}(x)$ is dynamically generated based on the model's actual competence on the current prompt, the model learns the *functional mapping* from its own hidden state entropy to a confidence scalar, which robustly transfers across domains.

Table 6: Hyperparameters used for CaOPD.

Parameter	Value
General & Data	
Base Models	Qwen3-8B, Olmo-3-7B-Instruct
Thinking Mode	False
Max prompt length	2048
Max response length	8192
Batching & Rollout	
Question batch size	32
Mini batch size	32
Number of rollouts for empirical target (K)	8
Inference engine	vLLM
Rollout Temperature	1.0
Validation	
Number of rollouts	{8,16}
Temperature	0.6
Top-p	0.95
Distillation Loss (SDPO / CaOPD)	
Top-K distillation	100
Distillation divergence	Reverse-KL
Teacher-EMA update rate	0.05
Rollout importance sampling clip	2
Training & Optimization	
Optimizer	AdamW
Learning rate	$\{1 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$
Warmup steps	10
Weight decay	0.01
Gradient Clip Norm	1.0

Table 7: Results on **Chemistry** as the test domain. **ID** denotes training on Chemistry data, **OOD** denotes training on Tool Use data and testing on Chemistry, and **CT** denotes continual training on Chemistry after Tool Use training. For Accuracy and SPR, higher is better; for Brier Score (BS) and ECE, lower is better.

Method	Setting	Qwen3-8B					Olmo-3-7B-Instruct				
		Acc ↑	ECE ↓	BS ↓	SPR ↑	Mean Conf.	Acc ↑	ECE ↓	BS ↓	SPR ↑	Mean Conf.
Base	–	30.9	0.587	0.559	0.307	0.897	33.8	0.492	0.462	0.488	0.831
SDFT	ID	49.1	0.486	0.477	0.387	0.972	45.2	0.429	0.423	0.397	0.882
SDFT	OOD	36.7	0.599	0.591	0.353	0.961	32.9	0.574	0.558	0.287	0.903
SDFT	CT	46.2	0.513	0.508	0.230	0.968	42.9	0.500	0.491	0.262	0.926
CaOPD	ID	50.0	0.266	0.281	0.599	0.382	44.8	0.176	0.260	0.568	0.529
CaOPD	OOD	44.3	0.358	0.365	0.348	0.334	38.1	0.371	0.378	0.411	0.752
CaOPD	CT	48.6	0.126	0.209	0.662	0.438	42.4	0.147	0.248	0.459	0.458

The Mechanism of "Calibration Forgetting" in Continual Learning. In Continual Training, the model's weights undergo sequential gradient updates to assimilate new knowledge. While self-distillation algorithms like SDFT are explicitly designed to prevent the catastrophic forgetting of *reasoning capabilities* (Accuracy), they completely fail to protect the calibration mapping. As shown in Table 8, SDFT's CT accuracy on Tool Use recovers perfectly to 0.647 (matching the ID baseline), but its SPR collapses completely to 0.000. This indicates a complete loss of discriminative resolution for confidence.

By continually enforcing the target replacement ($\tilde{y} = (a, \hat{\mu}(x))$; Section 4.2) during sequential updates, CaOPD constantly re-anchors the confidence tokens to the model's evolving capability frontier. This explicitly regularizes the confidence outputs against the model's actual execution outcomes on the new domain, successfully immunizing the network against Calibration Forgetting.

Table 8: Results on **Tool Use** as the test domain. **ID** denotes training on Tool Use data, **OOD** denotes training on Chemistry data and testing on Tool Use, and **CT** denotes continual training on Tool Use after Chemistry training. For Accuracy and SPR, higher is better; for Brier Score (BS) and ECE, lower is better.

Method	Setting	Qwen3-8B					Olmo-3-7B-Instruct				
		Acc ↑	ECE ↓	BS ↓	SPR ↑	Mean Conf.	Acc ↑	ECE ↓	BS ↓	SPR ↑	Mean Conf.
Base	–	64.7	0.324	0.328	0.396	0.971	54.4	0.367	0.381	0.351	0.911
SDFT	ID	67.6	0.321	0.320	0.085	0.999	67.6	0.304	0.303	0.503	0.981
SDFT	OOD	66.2	0.329	0.331	0.166	0.991	50.0	0.381	0.391	0.360	0.872
SDFT	CT	64.7	0.352	0.353	0.000	0.999	69.1	0.294	0.294	0.349	0.985
CaOPD	ID	70.6	0.228	0.242	0.555	0.699	72.1	0.177	0.186	0.580	0.891
CaOPD	OOD	64.7	0.307	0.315	0.332	0.954	54.4	0.320	0.359	0.428	0.864
CaOPD	CT	66.2	0.253	0.264	0.510	0.768	67.6	0.285	0.331	0.548	0.427

E.2 The True Cost of Calibration: Test-Time Compute vs. Amortization

A natural concern is that CaOPD's calibration gains come at the expense of additional compute. We clarify the cost structure at each stage.

Training-time overhead. CaOPD's gradient step is identical to standard SDPO or SDFT: the same token-level forward/backward pass on the revised completion $\tilde{y}$ under student and teacher conditioning. The only additional cost is computing $\hat{\mu}(x)$ from K rollouts per prompt. Crucially, this cost depends on the OPD backbone:

- **Under SDPO:** SDPO's standard training loop already samples multiple rollouts per prompt to identify successful trajectories for constructing the privileged context z. CaOPD reuses these *same* rollouts to evaluate $\hat{\mu}(x)$, requiring only a lightweight verifier pass on trajectories that have already been generated. The marginal sampling cost is therefore **zero**; the only addition is the verifier evaluation, which is negligible compared to generation.
- **Under SDFT:** The standard SDFT pipeline generates a single student trajectory per prompt. CaOPD introduces K additional rollouts for empirical confidence estimation. At $K = 8$ (our default; see Appendix E.3 for the ablation), this increases rollout generation by a constant factor. In practice, the overhead remains modest: as shown in Figure 2 (right), the wall-clock time per optimization step is nearly indistinguishable from the baseline, because rollout generation is parallelized via vLLM and does not bottleneck the training loop.

Deployment-time savings. The key comparison is against test-time calibration methods such as Self-Consistency (Wang et al., 2022) or SelfCheckGPT (Manakul et al., 2023), which require generating K responses *online* for every user query, incurring $O(K)$ inference latency. CaOPD computes $\hat{\mu}(x)$ entirely during training and distills it into the model's weights. At deployment, calibrated confidence is produced in a single $O(1)$ forward pass — the same cost as an uncalibrated model. This makes CaOPD uniquely suited for latency-sensitive agentic deployment where per-query multi-sampling is prohibitive.

Summary. Table 9 provides an overview of the cost structure.

Table 9: Cost breakdown of CaOPD under different OPD backbones. Under SDPO, the rollouts required for $\hat{\mu}(x)$ are already generated by the base algorithm, so CaOPD adds only a verifier evaluation pass. Under SDFT, K additional rollouts are needed but the gradient step itself is unchanged.

	CaOPD + SDPO	**CaOPD + SDFT**
Extra rollout sampling	None (reused)	K per prompt
Verifier evaluation	K per prompt	K per prompt
Gradient step overhead	None	None
Deployment cost	$O(1)$	$O(1)$

E.3 Detailed Analysis: Effect of Rollout Sample Size and Confidence Reshaping

To provide a deeper understanding of the compute-efficiency and distributional shifts induced by CaOPD, we expand upon the analysis presented in Section 5.3. Figure 6 presents both the ablation over the number of rollouts (K) and the resulting structural changes to the model's confidence distribution.

Effect of Rollout Sample Size (K). As shown in the left panel of Figure 6, we ablate $K \in \{1, 2, 4, 8, 16, 32\}$. Task accuracy (blue line) remains remarkably flat across all K, confirming that capability is completely decoupled from confidence sampling variance. Calibration exhibits a phase transition: small K (1–4) yields heavily quantized targets that trap the model in overconfidence, while $K = 8$ emerges as a compute-efficient sweet spot where mean confidence aligns with actual accuracy and discrimination (SPR) peaks. Beyond $K = 8$, returns diminish rapidly.

Why do small sample sizes ($K \leq 4$) fail? The reason is *target quantization*. When $K = 1$, the empirical target $\hat{\mu}(x)$ is strictly binary (0.0 or 1.0). This structurally prevents the model from learning intermediate confidence values, pushing it back toward the same saturation pathology as standard OPD. At $K = 8$, the target space has 9 distinct levels (increments of 0.125), providing sufficient granularity to represent graded epistemic uncertainty. This also explains why CaOPD's Confidence Calibration term (Equation equation 6, $t \in \mathcal{I}_c$) requires a non-degenerate target distribution: binary targets effectively reduce it to a second copy of the saturation-inducing signal that CaOPD is designed to eliminate.

Visualizing the Structural Reshaping of Confidence. The right panels of Figure 6 visualize the underlying confidence distributions. Standard OPD (SDFT eval) collapses the entire distribution into a delta function at 1.0, destroying the model's capacity for graded uncertainty. In contrast, the empirical targets used by CaOPD ($K = 16$, lower left) present a dispersed distribution reflecting true execution variance across prompts. Crucially, the deployed single-pass CaOPD model (lower right) successfully *internalizes* this dispersion, actively placing uncertain predictions into lower confidence bins. This confirms that the target replacement does not merely shift the mean confidence downward but structurally restores the model's ability to *discriminate* between successes and failures, consistent with the large SPR improvements reported in Tables 2 and 3.

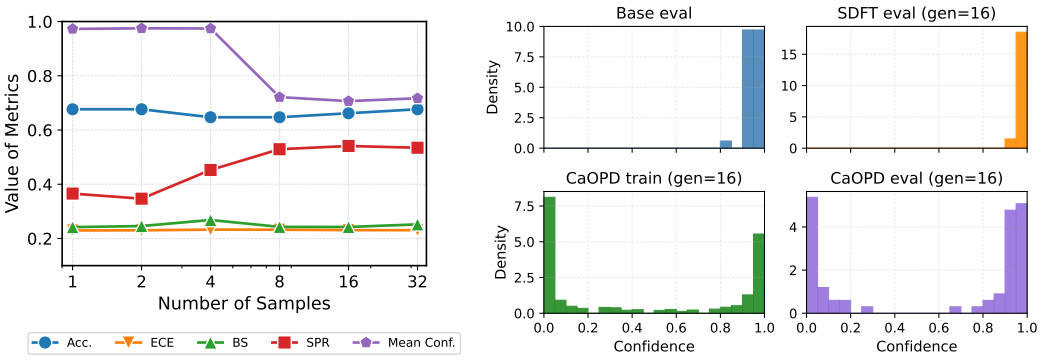

Figure 6: Effect of rollout sample size and confidence distributions.

E.4 Training Dynamics and Format Adherence Across Model Scales

To provide a deeper understanding of how model capacity impacts the CaOPD optimization process, we track the training dynamics across the entire Qwen3 family (from 0.6B to 32B parameters). Figure 7 illustrates the training loss (left) and the confidence format adherence rate (right) over the course of training. Format adherence is defined as the fraction of rollouts that successfully generate a parsable confidence segment c matching the `Confidence: [value]` format specified in the prompt instructions (Appendix C.2).

Optimization Efficiency Scales with Capacity. As shown in the left panel of Figure 7, convergence speed and terminal loss are strongly dictated by model size. Larger models (Qwen3-32B, 14B)

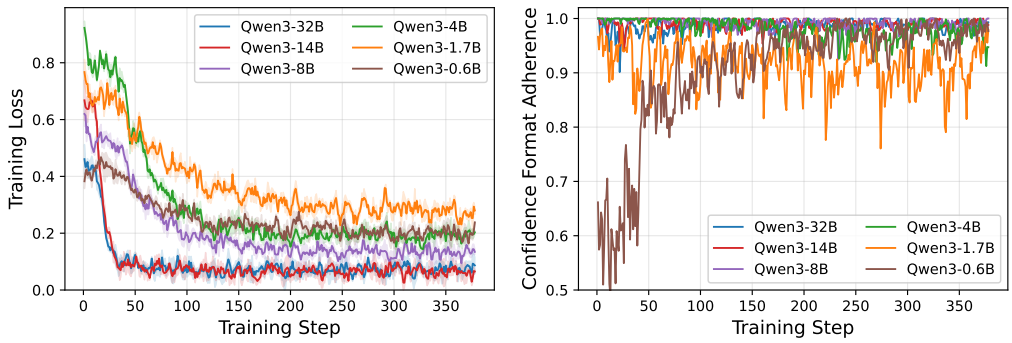

Figure 7: **Training dynamics across model scales. Left:** Larger models (14B, 32B) exhibit highly efficient optimization, rapidly converging to a near-zero loss. **Right:** While small models (0.6B, 1.7B) initially struggle with the strict confidence formatting instructions, CaOPD acts as a strong corrective alignment force, driving their format adherence to $> 90\%$ within the first 100 steps.

converge rapidly (within 50 steps) to a low loss, while smaller models (0.6B, 1.7B) exhibit higher variance and a higher terminal loss. This confirms that internalizing the continuous empirical confidence target $\hat{\mu}(x)$, which takes values at granularities of $1/K$ (see Section E.3), requires sufficient representational capacity. Notably, despite these differences in convergence speed, even the 0.6B model achieves meaningful calibration improvements over its SDFT baseline (Figure 4), indicating that CaOPD provides value across the entire scale range.

The Instruction-Following Bottleneck and CaOPD's Corrective Force. The right panel of Figure 7 provides empirical support for the constraints discussed in Appendix H. Larger models inherently possess strong instruction-following capabilities, maintaining near 100% format adherence from step zero. Conversely, the 0.6B and 1.7B models initially struggle with the confidence formatting instruction, starting with adherence rates between 50% and 80%.

However, CaOPD acts as a corrective alignment mechanism: within the first 50 to 100 steps, the gradient updates drive even the 0.6B model to over 90% format adherence. This is a direct consequence of the target replacement: since the revised completion $\tilde{y}$ always contains a well-formatted confidence segment (the string representation of $\hat{\mu}(x)$), the reverse KL loss at confidence positions ($t \in \mathcal{I}_c$) provides a consistent formatting signal at every training step. In contrast, standard OPD has no such structural anchor — the confidence format is only implicitly encouraged by the prompt instruction, which small models may initially fail to follow.

E.5 Full API Evaluation Results and The Scaling Law of Miscalibration

To substantiate the *Scaling Law of Miscalibration* introduced in Figure 1, we evaluate 12 leading LLMs spanning proprietary APIs (OpenAI's GPT-5.x series, Anthropic's Claude 4.x series, Google's Gemini 3.1 series) and open-weight models (DeepSeek-V3.1, Qwen3.5-397B) on the Science Q&A (Chemistry) (Feng et al., 2024) domain.

Table 10 presents the detailed capability and calibration metrics for all evaluated models.

Pervasive Confidence Saturation. A large number of models, particularly from the Gemini family (e.g., Gemini-3.1-Pro) and open-weight models (e.g., DeepSeek-V3.1), exhibit severe confidence saturation. Despite actual accuracies between 56% and 72%, their mean confidence scores are pegged at 0.99–1.00. This is consistent with our Entropy Collapse analysis (Proposition 2): on-policy distillation and RLHF force models to adopt the near-deterministic certainty of their privileged teacher distribution, leaving no capacity for expressing graded doubt.

Calibration Requires Substantial Overhead in Current Pipelines. Among the proprietary APIs, only Claude-Opus-4.6 achieves near-perfect calibration (Accuracy: 83.8%, Mean Confidence: 0.838, ECE: 0.027). However, Opus-4.6 is a massively scaled frontier model with undisclosed alignment pipelines. Standard-tier models from the same or other providers (e.g., Claude-Sonnet-4.5, Gemini-

Table 10: Detailed evaluation of leading proprietary and open-weight models on Science Q&A (Chemistry). Models are ordered by family. The massive gap between Accuracy and Mean Confidence in most models highlights the pervasive Optimism Bias induced by standard post-training. CaOPD enables a compact 8B model to achieve calibration quality rivaling frontier APIs.

Model	Type	Accuracy ↑	ECE ↓	BS ↓	Mean Confidence
GPT-5.2	Proprietary	63.3%	0.106	0.173	0.608
GPT-5.4	Proprietary	72.9%	0.080	0.154	0.802
Claude-Opus-4.6	Proprietary	**83.8%**	**0.027**	**0.090**	0.838
Claude-Sonnet-4.5	Proprietary	72.4%	0.155	0.201	0.874
Gemini-2.5-Flash	Proprietary	57.6%	0.424	0.424	1.000
Gemini-2.5-Pro	Proprietary	61.9%	0.365	0.364	0.984
Gemini-3.1-Flash-Lite	Proprietary	62.9%	0.360	0.360	0.989
Gemini-3.1-Pro	Proprietary	71.9%	0.278	0.276	0.997
DeepSeek-V3.1	Open-weight	56.7%	0.433	0.433	1.000
Kimi-K2.5	Open-weight	60.0%	0.379	0.380	0.979
Qwen3.5-397B	Open-weight	71.4%	0.214	0.237	0.929
MiniMax-M2.5	Open-weight	47.6%	0.498	0.495	0.974
Qwen3-8B (CaOPD)	**Ours**	80.6%	0.141	0.134	0.770

2.5-Pro) fail to replicate this calibration, suggesting that reliable uncertainty signaling currently demands a prohibitive capability overhead — or a targeted calibration mechanism such as CaOPD.

CaOPD Achieves Competitive Calibration at Small Scale. Our CaOPD-trained Qwen3-8B achieves 80.6% accuracy with a well-aligned mean confidence of 0.770 (ECE: 0.141). Two caveats are important for fair interpretation. First, the high accuracy is primarily attributable to the SDPO backbone (which already achieves 80.6%; Table 3); CaOPD's contribution is the calibration improvement (ECE reduced from 0.157 to 0.141, SPR from 0.571 to 0.669) with no accuracy degradation. Second, the API models are evaluated zero-shot, whereas CaOPD is fine-tuned on the Chemistry domain, so the accuracy comparison is not strictly apples-to-apples. The meaningful takeaway is that CaOPD enables a compact 8B model to achieve *calibration* quality (Reliability = 0.87; Figure 1, right) that rivals frontier APIs, demonstrating that explicit decoupling is a far more efficient path to calibrated confidence than brute-force scaling.

F Qualitative Case Studies: The Illusion of Certainty

To concretely illustrate the severe overconfidence pathology of standard OPD and the corrective alignment of CaOPD, we present two qualitative examples from the ToolAlpaca (Tool Use) domain. These examples highlight the two core mechanisms of CaOPD: (1) preserving high confidence when the model is capable and correct, and (2) expressing honest doubt when the model operates outside its knowledge boundary.

F.1 Case Study 1: Restoring Capability and Calibrated Confidence

In this scenario, the model must select the correct API to find details about a holiday. Standard SDFT falls into a common agentic failure mode (over-planning/hallucinating a search step) yet remains highly confident. CaOPD identifies the direct path and retains high, justified confidence.

[Context]
User Request: "I am curious about how the Chinese New Year is celebrated. Could you find the detailed information about the holiday, including its history, purpose, and traditions?"
Available Tools: `getHolidayDetails`, `searchHolidays`, `getHolidaysByCountry`, etc.
Ground Truth Action: `getHolidayDetails` with `{"holidayId": "chinese-new-year"}`

[Standard SDFT] → **Incorrect & Overconfident**
Reasoning: "To find detailed information about the Chinese New Year, I need to use the `getHolidayDetails` tool. However, I first need to identify the unique identifier... I will use the `searchHolidays` tool to find the relevant holiday ID."
Predicted Action: `searchHolidays` *(Incorrect tool selection)*
Verbalized Confidence: **0.95**

[CaOPD (Ours)] → **Correct & Calibrated**
Reasoning: "To find detailed information about the Chinese New Year, I need to use the `getHolidayDetails` tool... I will assume they are interested in the general details of the holiday."
Predicted Action: `getHolidayDetails` *(Correct tool selection)*
Verbalized Confidence: **0.90**

Takeaway: This example demonstrates that CaOPD does not merely apply a naive, global downward penalty to the model's confidence. When the CaOPD model successfully navigates the reasoning trajectory, it accurately assigns a high confidence score (0.90).

F.2 Case Study 2: Honest Failure and the "Illusion of Certainty"

This example demonstrates a "hard negative" where both models fail due to hallucinating optional API parameters. While SDFT exhibits blind optimism, CaOPD successfully signals its epistemic uncertainty.

[Context]
User Request: "Retrieve the text of Proverbs 18:24 from the Bible Digital API."
Available Tools: `getSpecificVerse(book*, chapter*, verse*, version?, language?)`
Ground Truth Action: `getSpecificVerse` with `{"book": "Proverbs", "chapter": 18, "verse": 24}` *(Only required parameters)*

[Standard SDFT] → **Incorrect & Blindly Optimistic**
Reasoning: "To retrieve the text of Proverbs 18:24, I need to use the `getSpecificVerse` tool with the appropriate parameters."
Predicted Action: `getSpecificVerse` with `{"version": "King James Version", "language": "English"}` *(Hallucinated optional parameters)*
Verbalized Confidence: **1.00**

[CaOPD (Ours)] → **Incorrect but Self-Aware**
Reasoning: "To retrieve the text of Proverbs 18:24, I need to use the `getSpecificVerse` tool with the appropriate parameters."
Predicted Action: `getSpecificVerse` with `{"version": "NIV", "language": "en"}` *(Hallucinated optional parameters)*
Verbalized Confidence: **0.10**

Takeaway: In autonomous agent deployment, a model predicting an absolute confidence of **1.00** on an incorrect API call will autonomously execute the flawed action, leading to compounding system failures. In contrast, CaOPD's confidence score of **0.10** accurately flags the model's uncertainty about the optional parameters. In a production environment, this low signal would safely trigger a human-in-the-loop fallback or an API schema re-evaluation. This behavior is a direct consequence of the target replacement mechanism (Section 4.2): during training, the model's rollouts on similar prompts would have yielded a low empirical success rate $\hat{\mu}(x)$, which was distilled as the confidence target.

G Broader Impact: Calibrated Small Models as Cascade Routers

In Section 5.5 (Figure 4), we demonstrated that CaOPD decouples capability from calibration across model scales from 0.6B to 32B. An interesting consequence emerges at the lower end of the scaling curve: small models trained with CaOPD can accurately signal their own limitations.

As shown in the rightmost panel of Figure 4, the CaOPD-trained 0.6B model achieves approximately 58% accuracy, and its mean confidence drops to roughly 0.3, faithfully reflecting this limitation. In contrast, the SDFT-trained 0.6B model outputs a confidence of 1.0 despite failing nearly half the time. This distinction is not merely cosmetic: a confidence score of 1.0 on a 58%-accurate model provides zero signal for downstream decision-making, while a score of 0.3 is actionable.

Application: Confidence-Based Cascade Routing. This property enables a practical deployment pattern. In a cascade inference system, a CaOPD-trained small model is deployed at the frontend. For queries where the model's confidence exceeds a threshold, it serves the answer at low cost and latency. For queries where confidence is low, the system defers to a larger, more expensive model. The key requirement for such a system is that the router's confidence must be *calibrated*, otherwise the threshold is meaningless. Standard OPD models, which saturate at $c = 1.0$, cannot serve as routers regardless of their accuracy. CaOPD structurally resolves this by ensuring that confidence scales with actual capability (Figure 4, right panel).

We note that this discussion identifies a promising application direction; empirically validating cascade routing performance (e.g., cost-accuracy tradeoffs under different thresholds) is left to future work.

H LIMITATIONS AND FUTURE WORK

Limitations. Like all on-policy distillation methods, CaOPD inherits the requirement that the base model possesses sufficient in-context learning capacity to benefit from the privileged teacher context z. Our scaling analysis (Section 5.5) shows that even at 0.6B parameters, CaOPD delivers meaningful calibration improvements; however, capability gains are naturally bounded by the base model's reasoning capacity. Additionally, verbalized confidence requires the model to produce a parsable confidence segment c matching the specified format. While CaOPD's target replacement provides a strong formatting signal during training (Appendix E.4), occasional formatting failures can occur at test time, a limitation shared with all verbalized uncertainty methods and one we also observed in proprietary API evaluations. Finally, computing $\hat{\mu}(x)$ requires K rollouts per prompt during training, increasing the training compute budget. We find $K = 8$ sufficient in practice (Appendix E.3), and this cost is fully amortized at deployment where inference remains a single $O(1)$ forward pass.

Future Work. Our work highlights several exciting directions for future research:

- **Step-level calibration for long-horizon reasoning.** CaOPD currently provides utterance-level confidence. In agentic or multi-step CoT settings, extending the target replacement principle to dense, per-step uncertainty (e.g., integrating with Process Reward Models) could enable finer-grained error detection and early termination of flawed reasoning chains.
- **Efficient rollout strategies.** The current approach uses a fixed rollout budget K for all prompts. Adaptive allocation, e.g., early stopping when initial rollouts return unanimous verifier scores, could reduce training cost without degrading target quality. Extending CaOPD to preference optimization frameworks (DPO, RLHF) is also an open direction.
- **Confidence-based cascade routing.** As discussed in Appendix G, CaOPD-calibrated small models could serve as lightweight triage routers in cascade inference systems, deferring uncertain queries to larger models. Empirically validating cost-accuracy tradeoffs under this architecture is a natural next step.