Power Distribution Bridges Sampling, Self-Reward RL, and Self-Distillation

Akiyoshi Tomihari[1] **Issei Sato**[1]

Department of Computer Science, The University of Tokyo, Japan

tomihari@g.ecc.u-tokyo.ac.jp, sato@g.ecc.u-tokyo.ac.jp

Abstract

Recent analyses question whether reinforcement learning (RL) is responsible for strong reasoning in large language models (LLMs). At the same time, distillation and inference-time sampling, including power sampling, have emerged as effective ways to improve LLM performance. However, the relationship among RL, distillation, and sampling remains unclear. In this study, we focus on the power distribution, the target distribution of power sampling, and show that the power distribution bridges sampling, self-reward KL-regularized RL, and self-distillation. From the sampling perspective, we show that inexpensive local approximations cannot reproduce sequence-level power without information about possible suffixes. From the RL perspective, the power distribution is the closed-form optimizer of KL-regularized RL when the model's sequence-level log-probabilities are used as the reward. This identification leads to *power self-distillation*, an offline distillation surrogate that shares the same target distribution and amortizes the cost of power sampling into supervised training on teacher samples. We further show that power self-distillation can achieve self-reward sharpening, that downstream true-reward gains are governed by the covariance between true reward and self-reward under the power distribution, and that this ideal power-teacher gain transfers to a distilled student when the student is close to the power teacher. Experiments on reasoning tasks support our analysis: power sampling raises self-reward, true-reward gains depend on alignment with self-reward, and power self-distillation can match or exceed the performance of power sampling at much lower inference cost. The code is available at tom4649/power-self-distillation.

1 Introduction

The strong reasoning ability exhibited by large language models (LLMs) has often been attributed to reinforcement learning (RL). However, empirical analyses with extensive sampling question whether RL alone fully explains post-RL reasoning gains: as the number of sampled generations grows, post-RL models often fail to outperform their corresponding pre-RL counterparts [Yue et al., 2025]. At the same time, distillation has become a standard way to transfer the capabilities of expensive or stronger models to smaller models [Hinton et al., 2015, Guo et al., 2025, Busbridge et al., 2025], and inference-time compute allocated to sampling or search has improved LLM performance [Snell et al., 2024, Welleck et al., 2024].

However, the relationship between sampling, RL, and self-distillation remains unclear. In particular, Karan and Du [2026] show that a base model, without additional training or external reward, can match or exceed post-RL models using *power sampling*. This raises the question of whether the success of power sampling reflects a mechanism distinct from RL and distillation, or whether these methods can be connected through a common structure. Clarifying such a connection is important

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

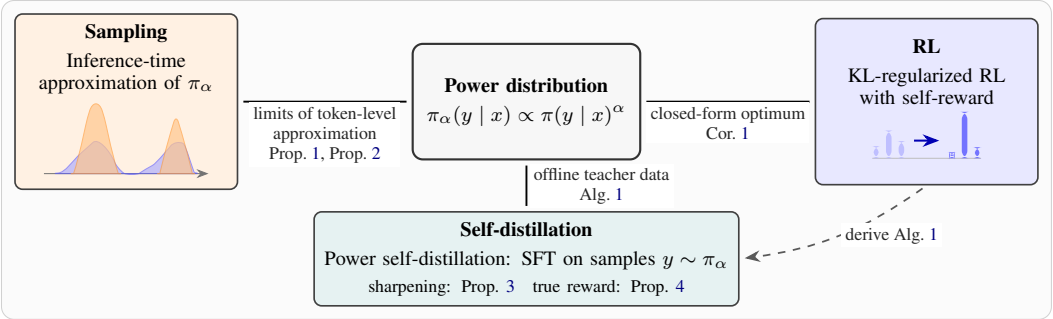

Figure 1: Overview of our contribution. The power distribution connects sampling, KL-regularized RL, and self-distillation: it is the target of power sampling, the closed-form optimum of self-reward RL, and the teacher distribution amortized by power self-distillation.

because it can reveal whether gains that appear to come from different procedures in fact arise from a common mechanism, and whether an expensive inference-time procedure can be converted into an offline training objective.

In this study, we show that sampling, RL, and self-distillation are naturally connected through the *power distribution*. As illustrated in Figure 1, this distribution is the target of power sampling, the closed-form optimum of a self-reward RL objective, and the teacher distribution amortized by self-distillation. From the sampling perspective, a natural question is whether the effect of power sampling can be reproduced by an inexpensive token-level approximation. We show that this is structurally difficult: per-token approximations cannot match the power distribution without sequence-level information. From the RL perspective, the power distribution is the closed-form optimum of KL-regularized RL [Ouyang et al., 2022] when the reward is the model's sequence-level log-probabilities, i.e., the self-reward in the sense of Huang et al. [2025]. Finally, by rewriting this RL objective, we derive *power self-distillation* as an offline distillation surrogate that shares the same target distribution and amortizes the cost of power sampling into offline training. We further show that power self-distillation admits a self-reward sharpening guarantee, that the true-reward gain of the power teacher is determined by a reward covariance under the power distribution, and that this gain transfers to a distilled student when the student remains close to the power teacher.

Our contributions are summarized as follows. Figure 1 illustrates the connection we study, and Table S.1 compares these axes with prior work.

- We show that approximating power sampling at inference time is structurally hard: per-token approximations cannot match the power distribution without sequence-level information (Propositions 1 and 2).
- We show that the power distribution is the closed-form optimum of KL-regularized RL with the model's sequence-level log-probabilities as the reward (Corollary 1), and derive *power self-distillation* by rewriting this RL objective, thereby amortizing expensive power sampling into offline training (Algorithm 1).
- We provide a self-reward sharpening bound for power self-distillation (Proposition 3), characterize the true-reward gain of the power teacher through a covariance condition under the power distribution (Proposition 4), and show that this gain transfers to a distilled student up to a forward-KL approximation term (Proposition 5).

2 Related work

RL post-training as distribution sharpening. RL has become a central tool in LLM post-training, including RL from human feedback (RLHF) [Ouyang et al., 2022] and RL with verifiable rewards (RLVR) [Shao et al., 2024, Guo et al., 2025, Lambert et al., 2024]. However, the role of RL in reasoning remains debated. Liu et al. [2025] showed that prolonged RL can outperform base models even under broad `pass@k` evaluations. On the other hand, Yue et al. [2025] showed that under `pass@k` evaluation, RLVR often improves sampling efficiency at small k but can underperform the base model at large k, suggesting that RLVR concentrates probability mass on reasoning paths already present in the base model's distribution. Complementing this view, He et al. [2025] analyzed a

degenerate rank bias in GRPO that preferentially reinforces high-probability trajectories, yielding a "distribution sharpening" regime where simply sampling more from the base model can be stronger under the same sample budget. Motivated by the perspective that many RL gains resemble distribution sharpening, Karan and Du [2026] proposed a training-free inference-time method that targets sharpened distributions of the base model. Their approach uses a Metropolis–Hastings sampler to approximate sequence-level power sampling and achieves reasoning improvements comparable to RL. Azizi et al. [2026] and Ji et al. [2026] developed lower-latency approximations to power sampling. We complement this line of work by showing that the power distribution targeted by these samplers is also the closed-form optimum of a self-reward KL-regularized RL objective.

Inference-time compute and distillation to amortize inference cost. Recent work argues that allocating additional computation at inference time can substantially improve LLM outputs [Snell et al., 2024, Welleck et al., 2024]. When an external reward or verifier is available, a common method is Best-of-N, which generates N candidates and selects the one with the highest reward; this simple strategy can yield strong empirical gains [Stiennon et al., 2020, Nakano et al., 2021, Touvron et al., 2023, Gao et al., 2023, Eisenstein et al., 2023, Mudgal et al., 2024]. To amortize the inference cost of Best-of-N, several works characterized the distribution induced by Best-of-N selection and proposed to distill this distribution into a single policy [Gui et al., 2024, Amini et al., 2025, Sessa et al., 2025, Yang et al., 2025]. In contrast to these reward-based distillation methods, we derive a self-distillation objective that amortizes power sampling itself, using only samples from the base model's power distribution.

Self-improvement without external rewards. A growing number of empirical studies suggest that language models can improve without relying on external rewards or human-provided labels, using self-generated data and intrinsic training signals. Huang et al. [2023], Wang et al. [2023] curated model-generated solutions or instructions and then fine-tuned on them. Several works perform RL using internal feedback alone, such as entropy minimization objectives [Prabhudesai et al., 2025] or confidence as the reward [Zhao et al., 2026]. Even randomly assigned rewards can improve performance [Shao et al., 2025]. Huang et al. [2025] formalized LLM self-improvement as distribution sharpening and analyzed algorithms motivated by SFT and KL-regularized RL. Building on this sharpening view, we show that the model's sequence-level log-probabilities induce the power distribution through KL-regularized RL, and that distilling this distribution can sharpen the model without external rewards.

3 Preliminaries

Notation. Let $\mathcal{X}$ denote the space of prompts and let $\mu \in \Delta(\mathcal{X})$ denote a distribution over prompts. We consider completions of length $T \geq 1$ over a finite vocabulary $\mathcal{V}$, and write $\mathcal{Y} := \mathcal{V}^T$ for the completion space. The base model is a policy $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$ and we write $\pi(\cdot \mid x)$ for the conditional distribution of y given x. With $y_{<t} := (y_1, \ldots, y_{t-1})$, we use the autoregressive factorization $\pi(y \mid x) = \prod_{t=1}^{T} \pi(y_t \mid x, y_{<t})$. We write $a \lesssim b$ to mean $a = O(b)$ and $\pi(S \mid x) := \sum_{y \in S} \pi(y \mid x)$ for a set $S \subseteq \mathcal{Y}$.

Self-improvement. Language models have been shown to be capable of self-improvement, improving their own performance without external rewards [Huang et al., 2023, Wang et al., 2023, Prabhudesai et al., 2025, Zhao et al., 2026]. This phenomenon is counterintuitive and appears to contradict the data-processing inequality, which states that mutual information is non-increasing under further processing of random variables [Cover, 1999]. Huang et al. [2025] reconcile these observations by interpreting improvements as *computational*, not statistical: self-improvement sharpens the distribution so that sampling a near-optimal solution becomes easier. This perspective connects to classical trade-offs between sampling and optimization in theoretical computer science [Kirkpatrick et al., 1983, Lovász and Vempala, 2006].

Formally, define the *self-reward* as the log-likelihood

$$r_{\text{self}}(x, y; \pi) = \log \pi(y \mid x) \tag{1}$$

and let the corresponding maximizer set be

$$\boldsymbol{y}^\star(x) := \arg\max_{y \in \mathcal{Y}} r_{\text{self}}(x, y; \pi). \tag{2}$$

Given $(\epsilon, \delta) \in (0,1)^2$, a policy $\widehat{\pi}$ is (ϵ, δ)-sharpened relative to π if the following holds:

$$\mathbb{P}_{x\sim\mu}\Big[\widehat{\pi}\big(\boldsymbol{y}^\star(x) \mid x\big) \geq 1-\delta\Big] \geq 1-\epsilon. \tag{3}$$

Huang et al. [2025] analyze the sample complexity of achieving (ϵ, δ)-sharpening when π is accessed only through conditional draws $y \sim \pi(\cdot \mid x)$ and likelihood evaluations $\pi(y \mid x)$, for supervised fine-tuning on Best-of-N targets sampled from π and for KL-regularized RL objectives driven by $r_{\mathrm{self}}(x, y; \pi)$.

Power distribution. Recent analyses of RL suggest that empirical reasoning gains resemble *distribution sharpening*, where probability mass concentrates on trajectories already well supported under the base model [Yue et al., 2025, He et al., 2025]. Motivated by this view, Karan and Du [2026] target inference-time sampling from the *power distribution* induced by the base model.

Definition 1 (Power distribution)**.** *With a policy $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$ and an exponent $\alpha > 1$, we define the* power distribution *induced by π as*

$$\pi_\alpha(y \mid x) := \frac{\pi(y \mid x)^\alpha}{\sum_{y' \in \mathcal{Y}} \pi(y' \mid x)^\alpha}. \tag{4}$$

Exact sampling from Eq. (4) is intractable at scale. Karan and Du [2026] therefore propose a Metropolis–Hastings (MH) procedure that achieves reasoning accuracy competitive with strong RL post-training [Shao et al., 2024, Guo et al., 2025], without further training. Lower-latency approximations have subsequently been proposed [Azizi et al., 2026, Ji et al., 2026], but these methods still use substantially more inference-time compute than standard autoregressive sampling.

4 Approximating power sampling requires sequence-level information

In this section, we begin from the sampling perspective. We ask whether the power distribution π_α can be reproduced by inexpensive inference-time approximations, focusing on two natural local inference-time procedures: (i) per-token temperature scaling (Section 4.1) and (ii) sequential importance sampling (SIS) with a one-step proposal (Section 4.2). In both cases, the gap to π_α is governed by sequence-level information that the local approximations do not access, showing why cheap inference-time approximations are structurally difficult and motivating the RL and self-distillation perspectives in Section 5.

4.1 Comparison to per-token temperature scaling

A natural way to locally approximate π_α is to apply the same power transformation at the token level during decoding. For $s \in \mathcal{V}$, define the next-token distribution obtained by per-token temperature scaling as

$$\pi_{\mathrm{temp},\alpha}(y_t = s \mid x, y_{<t}) := \frac{\pi(y_t = s \mid x, y_{<t})^\alpha}{\sum_{s' \in \mathcal{V}} \pi(y_t = s' \mid x, y_{<t})^\alpha}. \tag{5}$$

In contrast, the power distribution in Eq. (4) is, more precisely, the *sequence-level* power distribution $\pi_\alpha(\cdot \mid x) \propto \pi(\cdot \mid x)^\alpha$, whose next-token conditional we denote by $\pi_{\mathrm{pow},\alpha}$:

$$\pi_{\mathrm{pow},\alpha}(y_t = s \mid x, y_{<t}) := \frac{\sum_{y_{t+1:T} \in \mathcal{V}^{T-t}} \pi(y_{<t},\, s,\, y_{t+1:T} \mid x)^\alpha}{\sum_{y_{t:T} \in \mathcal{V}^{T-t+1}} \pi(y_{<t},\, y_{t:T} \mid x)^\alpha}. \tag{6}$$

We show that for arbitrary suffix distributions, the entire odds-ratio gap between Eqs. (5) and (6) is controlled by the Rényi entropy of the suffix.

Proposition 1 (Power vs. temperature odds ratios via suffix Rényi entropies)**.** *For $\alpha > 1$, a prompt x, a prefix $y_{<t}$, and $a \in \mathcal{V}$, let $q_{t,a}$ denote the conditional distribution of the suffix $Y_{t+1:T}$ under the base model,*

$$q_{t,a}(y_{t+1:T}) := \pi(y_{t+1:T} \mid x, y_{<t}, y_t = a). \tag{7}$$

For a distribution p on a finite set, define the Rényi entropy of order α as $H_\alpha(p) := 1/(1-\alpha) \log \sum_z p(z)^\alpha$. Then for any $a, b \in \mathcal{V}$ such that $\pi(y_t = a \mid x, y_{<t}) > 0, \pi(y_t = b \mid x, y_{<t}) > 0$, the ratio of next-token odds under $\pi_{\mathrm{pow},\alpha}$ versus $\pi_{\mathrm{temp},\alpha}$ satisfies

$$\frac{\pi_{\mathrm{pow},\alpha}(y_t = a \mid x, y_{<t})}{\pi_{\mathrm{pow},\alpha}(y_t = b \mid x, y_{<t})} \Big/ \frac{\pi_{\mathrm{temp},\alpha}(y_t = a \mid x, y_{<t})}{\pi_{\mathrm{temp},\alpha}(y_t = b \mid x, y_{<t})} = \exp\big((1-\alpha)\,(H_\alpha(q_{t,a}) - H_\alpha(q_{t,b}))\big). \tag{8}$$

We have $1-\alpha<0$, so Eq. (8) implies that, among next-token candidates a with comparable values of $\pi(y_t = a \mid x, y_{<t})$, those for which $q_{t,a}$ has larger Rényi entropy are relatively downweighted under $\pi_{\mathrm{pow},\alpha}$ compared to $\pi_{\mathrm{temp},\alpha}$. Thus, compared with per-token temperature scaling, sequence-level power sharpening favors continuations whose suffix distributions under π are more peaked, i.e., have lower Rényi entropy.

Comparison to Karan and Du [2026]. Karan and Du [2026] also studied the gap between per-token temperature and sequence-level power sampling, and formalized it in the special case of two extreme tokens (positive vs. negative pivotal tokens; their Example 1 and Proposition 3). Our result enables a quantitative comparison for any two next-token candidates.

Proposition 1 suggests that matching the next-token distribution induced by sequence-level power sampling at a step requires information about the suffix distributions following each candidate token.

4.2 Variance-minimizing one-step proposals for sequential power sampling

Beyond marginal token distributions, we turn to sequential importance sampling (SIS) targeting π_α, where a basic design goal is to stabilize incremental importance weights. Proposition 3.3 of Zhao et al. [2024] identifies the unique one-step variance-minimizing proposal in a general SIS setup, and we apply it to the power distribution π_α.

Fix a prompt $x \in \mathcal{X}$. Define the unnormalized power mass $\tilde{\pi}_\alpha(y_{1:T}) := \pi(y_{1:T} \mid x)^\alpha$ and, for $t = 0, \dots, T$, the prefix totals

$$\tilde{\pi}_{\alpha,t}(y_{1:t}) := \sum_{y_{t+1:T} \in \mathcal{V}^{T-t}} \tilde{\pi}_\alpha(y_{1:T}), \tag{9}$$

where for $t = 0$ the prefix is empty. Let $Z_\alpha := \sum_{y_{1:T} \in \mathcal{V}^T} \tilde{\pi}_\alpha(y_{1:T})$ be the normalizing constant, so that $\tilde{\pi}_{\alpha,0} = Z_\alpha$, and let $\pi_\alpha(\cdot \mid x)$ be the normalized power distribution on $\mathcal{V}^T$ from Eq. (4). For $t \geq 1$, write $\pi_\alpha(y_{1:t} \mid x)$ for the *prefix marginal* obtained by summing $\pi_\alpha(y_{1:T} \mid x)$ over $y_{t+1:T}$; then $\pi_\alpha(y_{1:t} \mid x) = \tilde{\pi}_{\alpha,t}(y_{1:t})/Z_\alpha$.

Consider extending a fixed prefix $y_{<t}$ by one token $Y_t \sim q(\cdot \mid x, y_{<t})$ in one step of SIS (or SMC without resampling), while keeping the global target $\pi_\alpha(\cdot \mid x)$ on $\mathcal{V}^T$. Define the *incremental importance weight* [Chopin et al., 2020]

$$W_t := \frac{\pi_\alpha(y_{<t}, Y_t \mid x)}{\pi_\alpha(y_{<t} \mid x)\, q(Y_t \mid x, y_{<t})}, \tag{10}$$

where we condition on $y_{<t}$ with $\pi_\alpha(y_{<t} \mid x) > 0$, and $\mathrm{Var}[W_t]$ denotes variance under $Y_t \sim q(\cdot \mid x, y_{<t})$. The next proposition shows the unique proposal $q(\cdot \mid x, y_{<t})$ that minimizes $\mathrm{Var}[W_t]$ at such a prefix.

Proposition 2 (Variance-minimizing one-step proposal at prefix $y_{<t}$)**.** *In the setting above, fix $y_{<t}$ with $\pi_\alpha(y_{<t} \mid x) > 0$. Among all proposals $q(\cdot \mid x, y_{<t})$ on $\mathcal{V}$, the unique minimizer of* $\mathrm{Var}[W_t]$ *is*

$$q_t^\star(y_t \mid x, y_{<t}) := \frac{\tilde{\pi}_{\alpha,t}(y_{1:t})}{\tilde{\pi}_{\alpha,t-1}(y_{<t})} = \pi_\alpha\big(y_t \mid x, y_{<t}\big), \tag{11}$$

where $y_{1:t} = (y_{<t}, y_t)$; the right-hand side equals the next-token conditional under $\pi_\alpha(\cdot \mid x)$.

Proposition 2 implies that minimizing the *local* one-step variance of the incremental weight forces the proposal to coincide with the next-token conditional in Eq. (11), which itself depends on the prefix totals $\tilde{\pi}_{\alpha,t}$ summed over all suffixes. In particular, proposals that modify only the base next-token conditional cannot in general equal the unique minimizer in Eq. (11). The proof and SIS background are in Section A.2.

Implication. Propositions 1 and 2 indicate that inexpensive one-step approximations cannot reproduce π_α without sequence-level information, leaving inference-time approximation of π_α structurally expensive. This aligns with prior work that expends additional inference-time compute [Karan and Du, 2026, Azizi et al., 2026, Ji et al., 2026] to approximate the power distribution.

5 From self-reward RL to power self-distillation

Section 4 shows that π_α is structurally expensive to approximate by sampling at inference time. In this section, we take the complementary view that π_α also connects RL and self-distillation, allowing

us to shift the cost to offline training. Section 5.1 identifies π_α as the closed-form optimum of a KL-regularized RL objective with self-reward. Section 5.2 uses this identification to derive an offline self-distillation algorithm from that RL objective. Section 5.3 then analyzes what the resulting distilled model achieves: a self-reward sharpening guarantee, a covariance characterization of the ideal power teacher's true-reward gain, and a bound transferring this gain to the student when the forward-KL distillation error is small.

5.1 Power distribution as the optimum of self-reward RL

Let $q : \mathcal{X} \to \Delta(\mathcal{Y})$ be a candidate policy, and consider the KL-regularized RL objective with reward r [Ouyang et al., 2022, Guo et al., 2025]

$$J_\beta(q; \pi, r) := \mathbb{E}_{x\sim\mu}\Big[\mathbb{E}_{y\sim q(\cdot\,|\,x)}\big[r(x,y)\big] - \beta\, D_{\mathrm{KL}}\big(q(\cdot \mid x) \,\|\, \pi(\cdot \mid x)\big)\Big] \tag{12}$$

with $\beta > 0$. By the standard closed-form solution of KL-regularized RL [Levine, 2018], the unique maximizer of Eq. (12) is the reward-tilted distribution

$$\pi^\star_\beta(y \mid x) := \frac{\pi(y \mid x)\exp\big(\beta^{-1} r(x,y)\big)}{Z_r(x)}, \qquad Z_r(x) := \sum_{y'\in\mathcal{Y}} \pi(y' \mid x)\exp\big(\beta^{-1} r(x,y')\big). \tag{13}$$

We restate this as Proposition 6 in Section A.6 and include a proof for completeness. Specializing the reward in Eq. (13) to the self-reward r_{self} in Eq. (1) yields the power distribution.

Corollary 1 (Self-reward tilt equals the power distribution)**.** *Suppose* $r(x,y) = r_{\mathrm{self}}(x,y;\pi) = \log \pi(y \mid x)$ *as in Eq.* (1)*. Then the optimizer* $\pi^\star_\beta$ *in Eq.* (13) *equals the power distribution* π_α *in Eq.* (4)*:*

$$\pi^\star_\beta(\cdot \mid x) = \pi_\alpha(\cdot \mid x), \qquad \alpha := 1 + \beta^{-1} > 1. \tag{14}$$

This identification connects power sampling and self-improvement RL: the inference-time target of Karan and Du [2026] coincides with the closed-form optimum of the KL-regularized self-reward objective studied in Huang et al. [2025], namely π_α.

5.2 Deriving power self-distillation

We now derive a self-distillation procedure from the RL objective without requiring the deployed model to sample from π_α at inference time.

RL objective as reverse and then forward KL to π_α. With $r = r_{\mathrm{self}}$ and $\alpha = 1 + \beta^{-1}$, the inner objective in Eq. (12) can be rewritten for each x as

$$\mathbb{E}_{y\sim q(\cdot\,|\,x)}[r_{\mathrm{self}}(x,y)] - \beta D_{\mathrm{KL}}\big(q(\cdot \mid x)\|\pi(\cdot \mid x)\big) = -\beta D_{\mathrm{KL}}\big(q(\cdot \mid x)\|\pi_\alpha(\cdot \mid x)\big) + \beta \log Z_r(x), \tag{15}$$

with the same partition function $Z_r(x)$ as in Eq. (13), which does not depend on q. Thus, for each prompt x, maximizing $J_\beta(q; \pi, r_{\mathrm{self}})$ over unconstrained q is equivalent to minimizing the reverse KL divergence $D_{\mathrm{KL}}\big(q(\cdot \mid x) \,\|\, \pi_\alpha(\cdot \mid x)\big)$, with unique minimizer $q(\cdot \mid x) = \pi_\alpha(\cdot \mid x)$.

The left-hand side of Eq. (15) is the original self-reward KL-regularized RL objective, whose direct optimization requires online policy optimization and is typically more challenging than supervised training. The reverse-KL form on the right-hand side is also an expectation under q, so direct optimization still requires on-policy samples from the learner. This can be viewed as an online distillation procedure. However, this approach requires generation inside the training loop and substantially increases training time.

We therefore convert the objective into an offline distillation surrogate that shares the same target distribution π_α, by minimizing the forward KL from the teacher distribution to the student, $D_{\mathrm{KL}}\big(\pi_\alpha(\cdot \mid x) \,\|\, q(\cdot \mid x)\big)$. The population minimizer over q is still π_α, so this surrogate preserves the same target distribution while enabling offline maximum-likelihood training on teacher samples. This forward-KL surrogate mirrors the reward-augmented maximum-likelihood method of Norouzi et al. [2016], who also exchanged the reverse KL appearing in entropy-regularized RL for a forward KL.

Forward KL yields MLE on teacher samples. Expanding the forward KL training objective gives

$$\mathbb{E}_{x\sim\mu}\big[D_{\mathrm{KL}}\big(\pi_\alpha(\cdot \mid x) \,\|\, q(\cdot \mid x)\big)\big] = \mathbb{E}_{x,y\sim\mu,\pi_\alpha}\big[\log \pi_\alpha(y \mid x)\big] - \mathbb{E}_{x,y\sim\mu,\pi_\alpha}\big[\log q(y \mid x)\big], \tag{16}$$

where $x, y \sim \mu, \pi_\alpha$ abbreviates $x \sim \mu$ and $y \sim \pi_\alpha(\cdot \mid x)$. The first term on the right-hand side of Eq. (16) does not depend on q, so minimizing the population forward KL is equivalent to maximizing the expected log-likelihood $\mathbb{E}_{x,y\sim\mu,\pi_\alpha}\big[\log q(y \mid x)\big]$. In practice we form an empirical objective by drawing i.i.d. pairs $D = \{(x_i, y_i)\}_{i=1}^n$ with $x_i \sim \mu$ and $y_i \sim \pi_\alpha(\cdot \mid x_i)$. Let Π_α be a student policy class that depends on α. We solve the following maximum likelihood estimate (MLE) problem:

$$\widehat{\pi} \in \arg\max_{q\in\Pi_\alpha} \sum_{i=1}^{n} \log q(y_i \mid x_i). \tag{17}$$

This procedure uses only offline completions sampled from the power distribution π_α derived from the base policy π, and it does not rely on any external reward labels, so it is an instance of self-distillation. We refer to it as *power self-distillation* and summarize it in Algorithm 1. In practice we run teacher inference once, store $D = \{(x_i, y_i)\}_{i=1}^n$, and then train the student with standard supervised fine-tuning on D. Separating teacher generation from student training simplifies implementation and enables dataset reuse. In Section C, we provide a complementary KL-gap analysis.

5.3 Sharpening and true reward under power self-distillation

In this subsection, we analyze three complementary aspects of power self-distillation: (i) Proposition 3 bounds the extent to which $\widehat{\pi}$ sharpens the self-reward, in the sense of Huang et al. [2025]; (ii) Proposition 4 shows that the local rate at which sharpening changes a true reward is determined by the covariance $\mathrm{Cov}_{\pi_\alpha}(r^\star, r_{\mathrm{self}})$; and (iii) Proposition 5 shows how the resulting power-teacher gain transfers to an imperfect distilled student.

(i) Self-reward sharpening of the distilled model. Huang et al. [2025] formalize self-improvement via (ϵ, δ)-sharpening relative to π as in Eq. (3). The next proposition bounds how well the MLE in Eq. (17) concentrates on the self-reward maximizer set $\boldsymbol{y}^\star(x)$ in Eq. (2).

Proposition 3 (Power self-distillation and sharpening)**.** *Fix $\alpha > 1$ and $\rho, \delta \in (0, 1)$. Suppose $\pi_\alpha \in \Pi_\alpha$ and there exists a constant $M < \infty$, independent of α, such that $|\Pi_\alpha| \le M$. Let $D = \{(x_i, y_i)\}_{i=1}^n$ be i.i.d. samples with $x_i \sim \mu, y_i \sim \pi_\alpha(\cdot \mid x_i)$, and let $\widehat{\pi} \in \arg\max_{q\in\Pi_\alpha} \sum_{i=1}^n \log q(y_i \mid x_i)$ be an MLE. Then with probability at least $1 - \rho$ over D,*

$$\mathbb{P}_{x\sim\mu}\big[\widehat{\pi}(\boldsymbol{y}^\star(x) \mid x) \le 1 - \delta\big] \lesssim \frac{\log(M\rho^{-1})}{\delta n} + \mathbb{P}_{x\sim\mu}\big[\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1 - \tfrac{\delta}{2}\big]. \tag{18}$$

In particular, the right-hand side of Eq. (18) converges to 0 as $n \to \infty$ and $\alpha \to \infty$.

Thus, for sufficiently large n and α, power self-distillation can achieve (ϵ, δ)-sharpening in the sense of Eq. (3). The proof is in Section A.3.

(ii) When does sharpening also improve a different true reward? Proposition 3 guarantees concentration on the self-reward maximizer set, but evaluation is typically governed by a different true reward (e.g., correctness). Let $r^\star : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ denote this true reward and define, for fixed $x \in \mathcal{X}$,

$$R(\alpha; x) := \mathbb{E}_{y\sim\pi_\alpha(\cdot|x)}\big[r^\star(x, y)\big]. \tag{19}$$

The next proposition characterizes how $R(\alpha; x)$ changes with α.

Proposition 4 (Covariance form of $\partial_\alpha R(\alpha; x)$)**.** *For any $\alpha > 0$ and any fixed $x \in \mathcal{X}$,*

$$\frac{\partial}{\partial\alpha} R(\alpha; x) = \mathrm{Cov}_{y\sim\pi_\alpha(\cdot|x)}\big(r^\star(x, y), r_{\mathrm{self}}(x, y)\big), \tag{20}$$

where covariances are over the support of $\pi(\cdot \mid x)$, on which $\log \pi(\cdot \mid x)$ is finite. In particular, if for some $b(x) \in \mathbb{R}$ and $c(x) > 0$,

$$r^\star(x, y) = c(x)\, r_{\mathrm{self}}(x, y) + b(x) \qquad \forall y \in \mathcal{Y}, \tag{21}$$

then $R(\alpha; x)$ is non-decreasing in α:

$$\frac{\partial}{\partial\alpha} R(\alpha; x) = c(x)\, \mathrm{Var}_{y\sim\pi_\alpha(\cdot|x)}\big(r_{\mathrm{self}}(x, y)\big) \ge 0. \tag{22}$$

The proof is in Section A.4. Proposition 4 states that $\partial_\alpha R(\alpha; x)$ equals the covariance between $r^\star$ and r_{self} under $\pi_\alpha(\cdot \mid x)$, so whether increasing α improves the true reward is determined exactly

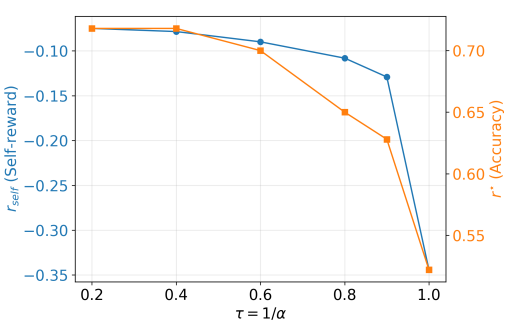

Figure 2: **Sharper power sampling raises both r_{self} and $r^\star$.** A smaller temperature parameter $\tau = 1/\alpha$ (sharper distribution) increases r_{self} and true reward $r^\star$ (accuracy) on MATH500.

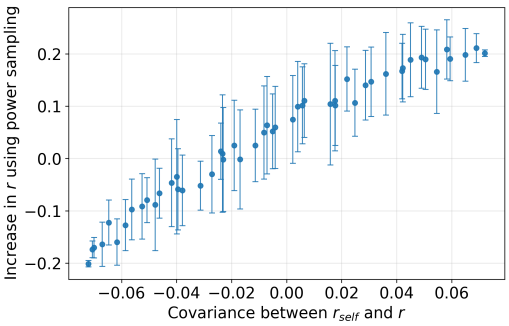

Figure 3: **Empirical illustration of Proposition 4.** Synthetic rewards r with varying correlation to r_{self} on MATH500, with error bars denoting standard deviation.

by $\mathrm{Cov}_{y\sim\pi_\alpha(\cdot|x)}\big(r^\star(x,y), r_{\text{self}}(x,y)\big)$. In particular, when $r^\star = r_{\text{self}}$, this covariance reduces to $\mathrm{Var}_{y\sim\pi_\alpha(\cdot|x)}\big(\log \pi(y \mid x)\big) \geq 0$, so $R(\alpha; x)$ is non-decreasing in α.

(iii) Transfer to an imperfect distilled student. The covariance identity characterizes the ideal teacher π_α, whereas power self-distillation produces a student q trained to approximate π_α. Define the population true reward

$$\mathcal{R}(q) := \mathbb{E}_{x\sim\mu}\mathbb{E}_{y\sim q(\cdot|x)}[r^\star(x,y)]$$

and the integrated covariance gain

$$\Gamma_\alpha := \mathbb{E}_{x\sim\mu}\left[\int_1^\alpha \mathrm{Cov}_{y\sim\pi_s(\cdot|x)}\big(r^\star(x,y), r_{\text{self}}(x,y)\big)\,\mathrm{d}s\right].$$

By Proposition 4 and the fundamental theorem of calculus, $\Gamma_\alpha = \mathcal{R}(\pi_\alpha) - \mathcal{R}(\pi)$. The next proposition decomposes the student's true-reward gain into this ideal power-teacher gain minus a student approximation term.

Proposition 5 (Transfer of power-teacher gain)**.** *Assume $r_{\min} \leq r^\star(x,y) \leq r_{\max}$ for all (x,y), and let $\Delta_r := r_{\max} - r_{\min}$. For distributions p, q, write $\mathrm{TV}(p,q) := \sup_A |p(A) - q(A)|$. Then for any policy q,*

$$\mathcal{R}(q) - \mathcal{R}(\pi) \geq \Gamma_\alpha - \Delta_r\,\mathbb{E}_{x\sim\mu}\big[\mathrm{TV}\big(q(\cdot \mid x), \pi_\alpha(\cdot \mid x)\big)\big]. \tag{23}$$

Consequently, if $\mathbb{E}_{x\sim\mu} D_{\mathrm{KL}}(\pi_\alpha(\cdot \mid x)\|q(\cdot \mid x)) < \infty$, then

$$\mathcal{R}(q) - \mathcal{R}(\pi) \geq \Gamma_\alpha - \Delta_r\sqrt{\frac{\mathbb{E}_{x\sim\mu} D_{\mathrm{KL}}\big(\pi_\alpha(\cdot \mid x)\|q(\cdot \mid x)\big)}{2}}. \tag{24}$$

The proof is in Section A.5. Proposition 5 directly matches the power self-distillation objective: minimizing the population forward KL to π_α reduces the student-transfer penalty in Eq. (24). Thus, when the integrated covariance gain Γ_α is positive and the distilled student is close enough to the power teacher, the true-reward gain of the power distribution transfers to the student.

6 Numerical evaluation

This section experimentally validates the following points.

- (RQ1) Power sampling increases self-reward (Section 5.1).
- (RQ2) Sharpening can improve true reward when $r^\star$ aligns with r_{self} (Section 5.3).
- (RQ3) Power self-distillation achieves self-improvement (Sections 5.2 and 5.3).

Detailed experimental setups are provided in Section B.1, and synthetic experiments validating Section 4 are shown in Sections D.5 and D.6.

Table 1: True reward $r^\star$ (accuracy) and self-reward r_{self} for Qwen2.5-Math-7B on MATH500. The left two columns report means over all sampled completions. The right two columns report self-reward Best-of-N: the completion with the largest r_{self} among samples generated with different seeds. Evaluated with four seeds.

		All completions		Self-reward Best-of-N	
Model	Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
RandW	Standard	0.000 ± 0.000	-13.406 ± 0.008	0.000	-13.309
	Power	0.000 ± 0.000	-12.370 ± 0.014	0.000	-12.133
Base	Standard	0.508 ± 0.016	-0.316 ± 0.043	0.680	-0.097
	Temperature	0.683 ± 0.014	-0.061 ± 0.015	0.756	-0.036
	Power	0.714 ± 0.006	-0.077 ± 0.001	0.742	-0.062
Power-distilled	Standard	0.643 ± 0.025	-0.089 ± 0.005	0.764	-0.042
	Temperature	$\mathbf{0.722} \pm 0.009$	$\mathbf{-0.043} \pm 0.001$	**0.768**	$\mathbf{-0.034}$

6.1 Setup

We used the Qwen2.5-Math-7B [Yang et al., 2024b], Qwen2.5-7B [Yang et al., 2024a], and Phi-3.5-mini-instruct [Abdin et al., 2024] models on the MATH [Lightman et al., 2024], HumanEval [Chen et al., 2021], MBPP [Austin et al., 2021], and GPQA [Rein et al., 2024] datasets. In the main text, we focus on the Qwen2.5-Math-7B model on the MATH dataset, which consists of 12,500 competition-style math problems spanning seven categories. For evaluation, we used MATH500, a selected subset of the MATH test set. For power self-distillation (Algorithm 1), we sampled 500 training problems from MATH, excluding those in MATH500. We fine-tuned with LoRA adapters [Hu et al., 2022] using the AdamW optimizer [Loshchilov and Hutter, 2017].

For power sampling, we used the MH procedure of Karan and Du [2026] (Algorithm 2) with their default hyperparameters, including $\alpha = 4.0$. For additional baselines, we used standard autoregressive sampling (Standard) and token-wise temperature scaling (Temperature) with $\tau = 1/\alpha = 0.25$, so that the token-level baseline uses the same local power exponent as power sampling. We studied three model variants: the base model (Base), the power-distilled model (Power-distilled, Algorithm 1), and a randomly initialized model (RandW). RandW is a negative control for cases where likelihood is not aligned with correctness.

To study the relationship between true reward $r^\star$ and self-reward r_{self}, we additionally evaluated an approach we call self-reward Best-of-N: given N sampled completions $\{y_i\}_{i=1}^N$, we selected the completion with the largest value of $r_{\text{self}}(y_i)$. In all experiments, r_{self} denotes the completion-token average log-likelihood under the corresponding base model, with prompt tokens masked out; this length normalization makes values comparable across completions.

6.2 Results

Power sampling increases self-reward (RQ1). Table 1 shows mean self-reward (r_{self}) and accuracy ($r^\star$) over sampled completions (left two columns). Power sampling raises r_{self} for both the base model and RandW. Decoding with token-wise temperature also raises r_{self} on the base model.

Sharpening can improve true reward when $r^\star$ aligns with r_{self} (RQ2). Table 1 also shows that higher r_{self} is typically accompanied by higher true reward $r^\star$, except on RandW, where r_{self} is not aligned with $r^\star$. Notably, self-reward Best-of-N yields the largest gains in $r^\star$ across all models.

To make this point clearer, Figure 2 plots the decoding temperature $\tau = 1/\alpha$ against r_{self} and $r^\star$; both quantities decrease as τ increases (i.e., sharpening weakens). Figure 3 uses synthetic rewards (Section B.1 for details), whose correlation with r_{self} ranges from positive to negative; the gain in r from power sampling grows roughly linearly with $\text{Cov}(r, r_{\text{self}})$.

Power self-distillation achieves self-improvement (RQ3). Table 1 shows that after power self-distillation, the student with temperature decoding scores higher on both $r^\star$ and r_{self} than the base model under standard sampling, temperature sampling, or power sampling. The strongest result is obtained by combining power self-distillation with Temperature decoding. At inference time, the

Table 2: Qualitative comparison for Qwen2.5-Math-7B on a MATH500 question: *"The coordinates of a parallelogram are* $(5, 3)$*,* $(6, 8)$*,* $(7, 4)$*, and* (x, y) *with* $x > 7$*. What is the value of* $x + y$*?"*

Model	Sampling	Correctness	Summary
Base	Base	No	Reaches the correct value but lacks sound reasoning. Terminates immediately without following the required formatting constraints.
	Temperature	No	Uses irrelevant mathematical properties and generates an incorrect formula, resulting in a hallucinated final answer.
	Power	Yes	Maintains logical consistency and mathematical accuracy, but simulates a Python execution to present a non-executed solution.
Power-distilled	Standard	Yes	Shows robust reasoning and self-correction by re-evaluating the problem when constraints are not met.

student uses only autoregressive decoding (with temperature), thereby amortizing the inference cost of power sampling into offline training.

Qualitative example. Table 2 summarizes completions on one MATH500 problem. With token-wise temperature, the model cites irrelevant facts and concludes with a hallucinated formula, plausibly because token-wise tilting in Eq. (5) does not coincide with sequence-level tilting in Eq. (6). Power sampling instead tilts toward π_α and is graded correct, but the completion includes plausible Python code that is never executed, and the model only mimics a reasoning pattern. After power self-distillation, standard decoding yields the correct answer with more robust step-by-step reasoning. The full completions are shown in Section D.4.

Additional dataset–model combinations are reported in Section D.1; in each case, the distilled model outperforms the corresponding base model.

7 Conclusion

We showed that the power distribution bridges power sampling, self-reward KL-regularized RL, and self-distillation as the sampling target, closed-form RL optimum, and teacher distribution. From the sampling perspective, inexpensive local approximations are structurally limited: per-token temperature scaling and variance-minimizing one-step proposals both miss sequence-level information. From the RL perspective, the same sequence-level power distribution is the optimizer of KL-regularized RL when the reward is the model's sequence-level log-probabilities. This identification yields power self-distillation, an offline surrogate that amortizes power sampling into supervised training on teacher samples. Power self-distillation can achieve self-reward sharpening, while true-reward improvement is governed by $\mathrm{Cov}_{\pi_\alpha}(r^\star, r_{\mathrm{self}})$, and the resulting power-teacher gain transfers to a distilled student up to a forward-KL approximation term. Finally, we supported the analysis with experiments.

Limitations. Self-improvement through sharpening and distillation inherits the capabilities of the base model, so gains can be small when the base is weak; improving base-model quality (e.g., pretraining) is outside our scope. Our analysis and experiments focus on autoregressive language models over finite horizons. We do not fully rule out pretraining-time benchmark contamination in the underlying LLMs. Our experiments are also limited to 7B-scale and smaller models; evaluation on substantially larger models is left to future work.

Acknowledgments and Disclosure of Funding

This work was supported by JSPS KAKENHI Grant Number JP24H00709.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. Variational best-of-n alignment. In *International Conference on Learning Representations*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Seyedarmin Azizi, Erfan Baghaei Potraghloo, Minoo Ahmadi, Souvik Kundu, and Massoud Pedram. Power-SMC: Low-latency sequence-level power sampling for training-free LLM reasoning. *arXiv preprint arXiv:2602.10273*, 2026.

Ananth Balashankar, Ziteng Sun, Jonathan Berant, Jacob Eisenstein, Michael Collins, Adrian Hutter, Jong Lee, Chirag Nagpal, Flavien Prost, Aradhana Sinha, Ananda Theertha Suresh, and Ahmad Beirami. InfAlign: Inference-aware language model alignment. In *International Conference on Machine Learning*, volume 267, 2025.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation scaling laws. In *International Conference on Machine Learning*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR, 2023.

Sara A Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Lin Gui, Cristina Garbacea, and Victor Veitch. BoNBon alignment for large language models and the sweetness of best-of-n sampling. In *Advances in Neural Information Processing Systems*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. In *Conference on Empirical Methods in Natural Language Processing*, pages 25559–25571, 2025.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *International Conference on Learning Representations*, 2025.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, 2023.

Xiaotong Ji, Rasul Tutunov, Matthieu Zimmer, and Haitham Bou Ammar. Scalable power sampling: Unlocking efficient, training-free reasoning for LLMs via distribution sharpening. *arXiv preprint arXiv:2601.21590*, 2026.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. In *International Conference on Learning Representations*, 2026.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations*, 2023.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

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

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Advances in Neural Information Processing Systems*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 57–68. IEEE, 2006.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *International Conference on Machine Learning*, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. In *Advances in Neural Information Processing Systems*, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

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

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In *International Conference on Learning Representations*, 2016.

Pier Giuseppe Sessa, Robert Dadashi-Tazehozi, Leonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Rame, Bobak Shahriari, Sarah Perrin, Abram L. Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos Garea, Amélie Héliou, Aliaksei Severyn, Matthew Hoffman, Nikola Momchev, and Olivier Bachem. BOND: Aligning LLMs with best-of-n distillation. In *International Conference on Learning Representations*, 2025.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in RLVR. *arXiv preprint arXiv:2506.10947*, 2025.

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.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Yee Whye Teh, Victor Bapst, Wojciech Marian Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 13484–13508, 2023.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Survey Certification.

Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *The Annals of Statistics*, pages 339–362, 1995.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

Tong Yang, Jincheng Mei, Hanjun Dai, Zixin Wen, Shicong Cen, Dale Schuurmans, Yuejie Chi, and Bo Dai. Faster wind: Accelerating iterative best-of-n distillation for llm alignment. In *International Conference on Artificial Intelligence and Statistics*, pages 4537–4545, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025.

Tong Zhang. From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, pages 2180–2210, 2006.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60704–60748. PMLR, 2024.

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

Table S.1: Comparison of prior work by its connection to power distributions, sampling, RL, distillation, and whether it avoids external rewards.

Paper	Power distribution	Sampling	RL	Distillation	No external reward
Norouzi et al. [2016]	–	–	✓	✓	–
Rusu et al. [2016]	–	–	✓	✓	–
Teh et al. [2017]	–	–	✓	✓	–
Laskin et al. [2023]	–	–	✓	✓	–
Huang et al. [2025]	–	✓	✓	✓	✓
Gui et al. [2024]	–	✓	–	✓	–
Amini et al. [2025]	–	✓	–	✓	–
Balashankar et al. [2025]	–	✓	✓	–	–
Sessa et al. [2025]	–	✓	✓	✓	–
Yang et al. [2025]	–	✓	✓	✓	–
Karan and Du [2026]	✓	✓	–	–	✓
Azizi et al. [2026]	✓	✓	–	–	✓
Ji et al. [2026]	✓	✓	–	–	✓
Ours	✓	✓	✓	✓	✓

Algorithm 1 Power self-distillation

Require: Base model π; power exponent $\alpha > 0$; teacher sampler $\textsc{TeacherSample}(x; \pi, \alpha)$ approximating $\pi_\alpha(\cdot \mid x)$ (e.g., Algorithm 2); prompt source μ; dataset size n; student model q_θ (initialized from π).

Ensure: Trained student q_θ.

Offline teacher sampling (data collection):
for $i = 1, \ldots, n$ **do**
  Sample $x_i \sim \mu$.
  Sample $y_i \sim \textsc{TeacherSample}(x_i; \pi, \alpha)$.
end for
Store $D \leftarrow \{(x_i, y_i)\}_{i=1}^n$.
Student training (supervised fine-tuning):
Optimize completion-only NLL on D:

$$\theta \leftarrow \arg\min_\theta \sum_{(x,y)\in D} -\log q_\theta(y \mid x),$$

where the loss is computed only on completion tokens by masking prompt tokens.

Algorithm 2 Power sampling using Metropolis–Hastings [Karan and Du, 2026] (Power(∞): deterministic acceptance).

Notation. Let the unnormalized power target $\tilde{\pi}_\alpha(y \mid x) \propto \pi(y \mid x)^\alpha$. Let $A(y', y)$ denote the Metropolis–Hastings acceptance ratio comparing completions y, y' (with x fixed), where $p_{\text{prop}}(y' \mid y, x)$ denotes the autoregressive proposal density for resampling a suffix under $p_{\text{prop}}(\cdot \mid x, \cdot)$:

$$A(y', y) \;:=\; \min\left\{1,\; \frac{\tilde{\pi}_\alpha(y' \mid x)}{\tilde{\pi}_\alpha(y \mid x)} \cdot \frac{p_{\text{prop}}(y \mid y', x)}{p_{\text{prop}}(y' \mid y, x)}\right\}. \tag{25}$$

Require: Base model π; proposal p_{prop}; prompt x; completion length T with $B \mid T$; block size B; inner iterations N_{MCMC}; exponent $\alpha > 0$.

Ensure: Completion $y_{1:T}$ (MH: approximate sample from powered conditional $\pi(\cdot \mid x)^\alpha$ up to MCMC error; Power(∞): accept proposals only if $\pi(y' \mid x) > \pi(y \mid x)$, so $\pi(y \mid x)$ is monotone along accepted moves).

for $k = 0, 1, \ldots, T/B - 1$ **do**

  Given the current state $y_{1:kB}$, construct an initialization $y^{(0)}$ by extending autoregressively with p_{prop} to length $(k+1)B$:

$$y_t^{(0)} \sim p_{\text{prop}}(y_t \mid x, y_{<t}), \qquad kB + 1 \le t \le (k+1)B.$$

  Set $y \leftarrow y^{(0)}$.

  for $n = 1, \ldots, N_{\text{MCMC}}$ **do**

    Sample m uniformly from $\{1, \ldots, (k+1)B\}$.

    Construct a proposal completion y' with prefix $y_{1:m-1}$ and resample the suffix:

$$y'_t \sim p_{\text{prop}}(y_t \mid x, y'_{<t}), \qquad m \le t \le (k+1)B.$$

    (MH) Compute $A(y', y)$ from Eq. (25). Draw $u \sim \text{Uniform}(0, 1)$. If $u \le A(y', y)$, set $y \leftarrow y'$.

    (Power(∞)) If $\pi(y' \mid x) > \pi(y \mid x)$, set $y \leftarrow y'$.

  end for

  Set $y_{1:(k+1)B} \leftarrow y$ as the current state carried into the next block iteration.

end for

return $y_{1:T}$

A Proofs and background

A.1 Proof of Proposition 1

Proof of Proposition 1. Fix x and a prefix $y_{<t}$ with $\pi(y_{<t} \mid x) > 0$, and write $p(s) := \pi(y_t = s \mid x, y_{<t})$ for $s \in \mathcal{V}$. For any suffix $y_{t+1:T}$ and token s, autoregressive factorization gives

$$\pi(y_{<t},\, s,\, y_{t+1:T} \mid x) = \pi(y_{<t} \mid x)\, p(s)\, q_{t,s}(y_{t+1:T}).$$

Using Eq. (6), the numerator for token s is therefore

$$\sum_{y_{t+1:T} \in \mathcal{V}^{T-t}} \pi(y_{<t},\, s,\, y_{t+1:T} \mid x)^\alpha = \pi(y_{<t} \mid x)^\alpha\, p(s)^\alpha \sum_{y_{t+1:T}} q_{t,s}(y_{t+1:T})^\alpha.$$

Summing over $s \in \mathcal{V}$ yields the corresponding denominator in Eq. (6), so the prefix factor $\pi(y_{<t} \mid x)^\alpha$ cancels and

$$\pi_{\mathrm{pow},\alpha}(y_t = s \mid x, y_{<t}) = \frac{p(s)^\alpha \sum_z q_{t,s}(z)^\alpha}{\sum_{s' \in \mathcal{V}} p(s')^\alpha \sum_z q_{t,s'}(z)^\alpha}. \tag{26}$$

For temperature scaling, Eq. (5) gives $\pi_{\mathrm{temp},\alpha}(y_t = s \mid x, y_{<t}) = p(s)^\alpha / \sum_{s'} p(s')^\alpha$. Hence for any $a, b \in \mathcal{V}$ with $p(b) > 0$,

$$\frac{\pi_{\mathrm{pow},\alpha}(y_t = a \mid x, y_{<t})}{\pi_{\mathrm{pow},\alpha}(y_t = b \mid x, y_{<t})} = \frac{p(a)^\alpha \sum_z q_{t,a}(z)^\alpha}{p(b)^\alpha \sum_z q_{t,b}(z)^\alpha}, \qquad \frac{\pi_{\mathrm{temp},\alpha}(y_t = a \mid x, y_{<t})}{\pi_{\mathrm{temp},\alpha}(y_t = b \mid x, y_{<t})} = \left(\frac{p(a)}{p(b)}\right)^\alpha,$$

and therefore

$$\frac{\pi_{\mathrm{pow},\alpha}(y_t = a \mid x, y_{<t})}{\pi_{\mathrm{pow},\alpha}(y_t = b \mid x, y_{<t})} \Big/ \frac{\pi_{\mathrm{temp},\alpha}(y_t = a \mid x, y_{<t})}{\pi_{\mathrm{temp},\alpha}(y_t = b \mid x, y_{<t})} = \frac{\sum_z q_{t,a}(z)^\alpha}{\sum_z q_{t,b}(z)^\alpha}.$$

By the definition of H_α, $\sum_z q(z)^\alpha = \exp\big((1-\alpha) H_\alpha(q)\big)$, which yields Eq. (8). □

A.2 Background and proof of Proposition 2

This appendix is aligned with the sequential Monte Carlo presentation of Zhao et al. [2024], who derive a general *twist-induced* proposal (their Prop. 3.3) that minimizes the variance of the one-step incremental importance weight for a given tower of intermediate targets. We provide a proof of the same variance-minimization fact specialized to π_α using the Cauchy–Schwarz inequality (cf. Zhao et al. 2024, App. A.2).

A.2.1 From a sequence-level target to a sequential sampler

Let $\mathcal{V}$ be a finite vocabulary and fix a prompt x and completion length $T \geq 1$. Let $P(y_{1:T}) := \pi_\alpha(y_{1:T} \mid x)$ denote the power distribution on $\mathcal{V}^T$ from Eq. (4), i.e.,

$$P(y_{1:T}) \propto \pi(y_{1:T} \mid x)^\alpha, \qquad \pi(y_{1:T} \mid x) = \prod_{t=1}^{T} \pi(y_t \mid x, y_{<t}).$$

Exact sampling from P may be intractable because normalizing constants involve sums over exponentially many sequences. Many practical samplers therefore build $y_{1:T}$ *sequentially*: having generated a prefix $y_{<t}$, they draw a next token y_t from a proposal $q(\cdot \mid x, y_{<t})$ and update importance weights so that, after T steps, full-length draws can be reweighted to be (exactly or approximately) correct for P.

A.2.2 Incremental importance weights

For $t = 1, \ldots, T$, let P_t denote the marginal of P on the length-t prefix:

$$P_t(y_{1:t}) := \sum_{y_{t+1:T} \in \mathcal{V}^{T-t}} P(y_{1:T}).$$

One step of sequential importance sampling extends $y_{<t}$ by sampling $Y_t \sim q(\cdot \mid x, y_{<t})$. The *incremental* multiplicative factor appended to the running weight is [Chopin et al., 2020]

$$W_t := \frac{P_t(y_{<t}, Y_t)}{P_{t-1}(y_{<t})\, q(Y_t \mid x, y_{<t})}, \tag{27}$$

defined on the event $\{P_{t-1}(y_{<t}) > 0\}$, where $(y_{<t}, Y_t)$ denotes the length-t prefix ending in Y_t. For the power distribution, $P_t(y_{1:t}) = \pi_\alpha(y_{1:t} \mid x)$, so Eq. (27) agrees with W_t in Eq. (10).

If one initializes weights at $w_0 := 1$ and updates $w_t := w_{t-1}W_t$, then for any completed trajectory $y_{1:T}$ with $\prod_{t=1}^{T} P_{t-1}(y_{<t}) > 0$,

$$w_T = \frac{P(y_{1:T})}{q(y_{1:T} \mid x)}, \qquad q(y_{1:T} \mid x) := \prod_{t=1}^{T} q(y_t \mid x, y_{<t}), \tag{28}$$

which is the usual full-sequence importance weight of $y_{1:T}$ for the target P against the autoregressive proposal q. Thus each W_t is the local factor that must be "well behaved" if the final weights are not to explode or collapse.

A.2.3 Why minimize $\mathrm{Var}[W_t]$ at one step?

Condition on a fixed feasible prefix $y_{<t}$ with $P_{t-1}(y_{<t}) > 0$. Write $f_t(v) := P_t(y_{<t}, v)/P_{t-1}(y_{<t})$ for $v \in \mathcal{V}$, i.e., the *true* conditional $P(y_t = v \mid y_{<t})$ under P. Then $W_t = f_t(Y_t)/q(Y_t)$ with $Y_t \sim q$.

Whenever $q(v) > 0$ for all v with $f_t(v) > 0$, the mean is always $\mathbb{E}[W_t \mid y_{<t}] = \sum_{v \in \mathcal{V}} q(v)\, f_t(v)/q(v) = 1$. However, $\mathrm{Var}[W_t \mid y_{<t}]$ depends strongly on q: if q places too little mass where f_t is large, occasional huge weights arise, which is the usual "weight degeneracy" pathology in importance sampling. Minimizing $\mathrm{Var}[W_t \mid y_{<t}]$ therefore makes the *single-step* contribution to weight instability as small as possible (among independent proposals), holding the prefix fixed. This is the same local objective highlighted by Zhao et al. [2024] for twist-induced proposals.

A.2.4 Proof of Proposition 2

Proof of Proposition 2. Fix $y_{<t}$ with $\pi_\alpha(y_{<t} \mid x) > 0$ and write $f(v) := \pi_\alpha(y_{<t}, v \mid x)/\pi_\alpha(y_{<t} \mid x)$ for $v \in \mathcal{V}$. Then $\sum_{v \in \mathcal{V}} f(v) = 1$ and $W_t = f(Y_t)/q(Y_t)$ under $Y_t \sim q$, assuming $q(v) > 0$ whenever $f(v) > 0$.

Since $\mathbb{E}[W_t] = \sum_v f(v) = 1$,

$$\mathrm{Var}[W_t] = \mathbb{E}[W_t^2] - 1 = \sum_{v \in \mathcal{V}} \frac{f(v)^2}{q(v)} - 1.$$

By Cauchy–Schwarz,

$$\Big(\sum_{v \in \mathcal{V}} f(v)\Big)^2 = \Big(\sum_{v \in \mathcal{V}} \sqrt{q(v)} \cdot \frac{f(v)}{\sqrt{q(v)}}\Big)^2 \le \Big(\sum_{v \in \mathcal{V}} q(v)\Big)\Big(\sum_{v \in \mathcal{V}} \frac{f(v)^2}{q(v)}\Big) = \sum_{v \in \mathcal{V}} \frac{f(v)^2}{q(v)}.$$

The left-hand side equals 1, so $\mathrm{Var}[W_t] \ge 0$ with equality if and only if the Cauchy–Schwarz inequality is tight, i.e., $\sqrt{q(v)} \propto f(v)/\sqrt{q(v)}$, equivalently $q(v) \propto f(v)$. Because $\sum_v f(v) = 1$, the unique minimizer on $\{v : f(v) > 0\}$ is $q(v) = f(v)$, which is $\pi_\alpha(v \mid x, y_{<t})$.

Finally, with $\tilde{\pi}_{\alpha,t}$ as in Eq. (9) and Z_α as in the main text,

$$f(v) = \frac{\pi_\alpha(y_{<t}, v \mid x)}{\pi_\alpha(y_{<t} \mid x)} = \frac{\tilde{\pi}_{\alpha,t}(y_{1:t})/Z_\alpha}{\tilde{\pi}_{\alpha,t-1}(y_{<t})/Z_\alpha} = \frac{\tilde{\pi}_{\alpha,t}(y_{1:t})}{\tilde{\pi}_{\alpha,t-1}(y_{<t})},$$

where $y_{1:t} = (y_{<t}, v)$, which is Eq. (11). □

A.3 Proof of Proposition 3

We first provide the following lemma, which is used to bound the Hellinger distance between the MLE and the true conditional distribution for finite-class models.

Lemma 1 (Finite-class MLE Hellinger bound [Wong and Shen, 1995, Geer, 2000, Zhang, 2006])**.** *Assume $|\Pi| < \infty$ and $\pi^\star \in \Pi$. Let $D = \{(x_i, y_i)\}_{i=1}^n$ be i.i.d. with $x_i \sim \mu$ and $y_i \sim \pi^\star(\cdot \mid x_i)$, and let $\widehat{\pi} \in \arg\max_{\pi \in \Pi} \sum_{i=1}^n \log \pi(y_i \mid x_i)$ be an MLE. Then for any $\rho \in (0, 1)$, with probability at least $1 - \rho$,*

$$\mathbb{E}_{x \sim \mu}\big[D_H^2\big(\widehat{\pi}(\cdot \mid x), \pi^\star(\cdot \mid x)\big)\big] \le \frac{2 \log(|\Pi|\rho^{-1})}{n}.$$

Using this lemma, we can prove Proposition 3 as follows.

Proof of Proposition 3. Define the failure event $F(x) := \{\widehat{\pi}(\boldsymbol{y}^\star(x) \mid x) \le 1-\delta\}$. By a simple inclusion,

$$F(x) \subseteq \Big\{\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1-\tfrac{\delta}{2}\Big\} \cup \Big\{\widehat{\pi}(\boldsymbol{y}^\star(x) \mid x) \le 1-\delta,\ \pi_\alpha(\boldsymbol{y}^\star(x) \mid x) > 1-\tfrac{\delta}{2}\Big\}.$$

Taking $\mathbb{P}_{x\sim\mu}[\cdot]$ yields

$$\mathbb{P}_{x\sim\mu}[F(x)] \le \mathbb{P}_{x\sim\mu}\big[\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1-\tfrac{\delta}{2}\big] \ + \ \mathbb{P}_{x\sim\mu}[E(x)], \tag{29}$$

where $E(x) := \{\widehat{\pi}(\boldsymbol{y}^\star(x) \mid x) \le 1-\delta,\ \pi_\alpha(\boldsymbol{y}^\star(x) \mid x) > 1-\tfrac{\delta}{2}\}$.

Let $B(x) := \mathcal{Y} \setminus \boldsymbol{y}^\star(x)$. For each x, write $\widehat{\pi}_x := \widehat{\pi}(\cdot \mid x)$ and $p_x := \pi_\alpha(\cdot \mid x)$. For two distributions $p, q \in \Delta(\mathcal{Y})$, define the squared Hellinger distance

$$D_H^2(p,q) := \sum_{y\in\mathcal{Y}} \big(\sqrt{p(y)} - \sqrt{q(y)}\big)^2.$$

By the reverse triangle inequality applied to the vectors $(\sqrt{\widehat{\pi}_x(y)})_{y\in B(x)}$ and $(\sqrt{p_x(y)})_{y\in B(x)}$,

$$D_H^2(\widehat{\pi}_x, p_x) \ge \sum_{y\in B(x)} \big(\sqrt{\widehat{\pi}_x(y)} - \sqrt{p_x(y)}\big)^2 \ge \big(\sqrt{\widehat{\pi}(B(x) \mid x)} - \sqrt{\pi_\alpha(B(x) \mid x)}\big)^2. \tag{30}$$

On the event $E(x)$, we have $\widehat{\pi}(B(x) \mid x) = 1 - \widehat{\pi}(\boldsymbol{y}^\star(x) \mid x) \ge \delta$ and $\pi_\alpha(B(x) \mid x) = 1 - \pi_\alpha(\boldsymbol{y}^\star(x) \mid x) < \delta/2$, so Equation (30) implies

$$D_H^2(\widehat{\pi}_x, p_x) \ge \big(\sqrt{\delta} - \sqrt{\delta/2}\big)^2 = \Big(1 - \tfrac{1}{\sqrt{2}}\Big)^2 \delta =: c_0\,\delta.$$

Therefore $\mathbf{1}\{E(x)\} \le D_H^2(\widehat{\pi}_x, p_x)/(c_0\delta)$, and hence

$$\mathbb{P}_{x\sim\mu}[E(x)] \le \frac{1}{c_0\delta}\,\mathbb{E}_{x\sim\mu}\big[D_H^2(\widehat{\pi}_x, p_x)\big]. \tag{31}$$

Finally, by the finite-class MLE Hellinger bound (Lemma 1) and $|\Pi_\alpha| \le M$, with probability at least $1-\rho$,

$$\mathbb{E}_{x\sim\mu}\big[D_H^2(\widehat{\pi}_x, p_x)\big] \le \frac{2\log(M\rho^{-1})}{n}.$$

Combining Eqs. (29) and (31) with the bound above yields

$$\mathbb{P}_{x\sim\mu}\big[\widehat{\pi}(\boldsymbol{y}^\star(x) \mid x) \le 1-\delta\big] \le \mathbb{P}_{x\sim\mu}\big[\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1-\tfrac{\delta}{2}\big] \ + \ \frac{2}{c_0}\cdot\frac{\log(M\rho^{-1})}{\delta\, n}.$$

Absorbing constants proves Eq. (18).

Convergence of the upper bound. The MLE term satisfies $\frac{\log(M\rho^{-1})}{\delta n} \to 0$ as $n \to \infty$.

For the limit of α, fix $x \in \mathcal{X}$ and write $m(x) := \max_{y\in\mathcal{Y}} \pi(y \mid x)$. By definition of $\boldsymbol{y}^\star(x)$, we have $\pi(y \mid x) = m(x)$ for all $y \in \boldsymbol{y}^\star(x)$ and $\pi(y \mid x) < m(x)$ for all $y \in B(x) = \mathcal{Y} \setminus \boldsymbol{y}^\star(x)$. The normalizing constant of the power distribution satisfies

$$\begin{aligned} Z_\alpha(x) &:= \sum_{y'\in\mathcal{Y}} \pi(y' \mid x)^\alpha && (32)\\ &= \sum_{y'\in\boldsymbol{y}^\star(x)} m(x)^\alpha \ + \ \sum_{y'\in B(x)} \pi(y' \mid x)^\alpha && (33)\\ &= \big|\boldsymbol{y}^\star(x)\big|\, m(x)^\alpha \ + \ \sum_{y'\in B(x)} \pi(y' \mid x)^\alpha. && (34) \end{aligned}$$

For each $y' \in B(x)$, the ratio $\pi(y' \mid x)/m(x)$ lies in $[0,1)$, hence $\big(\pi(y' \mid x)/m(x)\big)^\alpha \to 0$ as $\alpha \to \infty$. Because $B(x)$ is finite, $\sum_{y'\in B(x)} \pi(y' \mid x)^\alpha = o\big(m(x)^\alpha\big)$, and therefore

$$\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) = \frac{\big|\boldsymbol{y}^\star(x)\big|\, m(x)^\alpha}{Z_\alpha(x)} \xrightarrow[\alpha\to\infty]{} 1. \tag{35}$$

The indicators $\mathbf{1}\{\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1 - \frac{\delta}{2}\}$ converge to 0 for μ-almost every x as $\alpha \to \infty$ by Eq. (35). Since indicators are bounded by 1, dominated convergence yields

$$\mathbb{P}_{x\sim\mu}\big[\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1 - \tfrac{\delta}{2}\big] = \mathbb{E}_{x\sim\mu}\big[\mathbf{1}\{\pi_\alpha(\boldsymbol{y}^\star(x) \mid x) \le 1 - \tfrac{\delta}{2}\}\big] \xrightarrow[\alpha\to\infty]{} 0.$$

Thus, the second term in Eq. (18) converges to 0 as $\alpha \to \infty$. Together with the $n \to \infty$ limit of the first term, the full upper bound converges to 0. □

A.4 Proof of Proposition 4

Proof of Proposition 4. Recall

$$\pi_\alpha(y) = \frac{\pi(y)^\alpha}{Z_\alpha} = \frac{e^{\alpha \log \pi(y)}}{Z_\alpha}, \qquad Z_\alpha := \sum_{y' \in \mathcal{Y}} \pi(y')^\alpha = \sum_{y'} e^{\alpha \log \pi(y')}.$$

Differentiating $\pi_\alpha(y)$ with respect to α yields

$$\begin{aligned}
\frac{\partial}{\partial \alpha}\pi_\alpha(y) &= \frac{\partial}{\partial \alpha}\left(\frac{e^{\alpha \log \pi(y)}}{Z_\alpha}\right) \\
&= \frac{\log \pi(y) e^{\alpha \log \pi(y)} Z_\alpha - e^{\alpha \log \pi(y)} \frac{\partial}{\partial \alpha} Z_\alpha}{Z_\alpha^2} \\
&= \pi_\alpha(y)\left(\log \pi(y) - \frac{\partial}{\partial \alpha} \log Z_\alpha\right).
\end{aligned}$$

Using

$$\frac{\partial}{\partial \alpha} \log Z_\alpha = \frac{1}{Z_\alpha} \sum_{y'} \log \pi(y') e^{\alpha \log \pi(y')} = \sum_{y'} \pi_\alpha(y') \log \pi(y') = \mathbb{E}_{\pi_\alpha}[\log \pi], \tag{36}$$

we obtain

$$\begin{aligned}
\frac{\partial}{\partial \alpha}\mathbb{E}_{\pi_\alpha}[r^\star] &= \sum_y r^\star(y) \frac{\partial}{\partial \alpha}\pi_\alpha(y) && (37)\\
&= \sum_y r^\star(y)\pi_\alpha(y)\left(\log \pi(y) - \mathbb{E}_{\pi_\alpha}[\log \pi]\right) && (38)\\
&= \mathbb{E}_{\pi_\alpha}\big[r^\star(\log \pi - \mathbb{E}_{\pi_\alpha}[\log \pi])\big] && (39)\\
&= \mathrm{Cov}_{\pi_\alpha}(r^\star, \log \pi) && (40)\\
&= \mathrm{Cov}_{\pi_\alpha}(r^\star, r_{\mathrm{self}}). && (41)
\end{aligned}$$

□

A.5 Proof and extensions for Proposition 5

Proof of Proposition 5. Insert the power teacher between q and the base model:

$$\mathcal{R}(q) - \mathcal{R}(\pi) = \big(\mathcal{R}(\pi_\alpha) - \mathcal{R}(\pi)\big) + \big(\mathcal{R}(q) - \mathcal{R}(\pi_\alpha)\big). \tag{42}$$

By Proposition 4 and the fundamental theorem of calculus,

$$\mathcal{R}(\pi_\alpha) - \mathcal{R}(\pi) = \mathbb{E}_{x\sim\mu}\left[\int_1^\alpha \mathrm{Cov}_{y\sim\pi_s(\cdot|x)}\big(r^\star(x,y), r_{\mathrm{self}}(x,y)\big)\,\mathrm{d}s\right] = \Gamma_\alpha.$$

For the student-transfer term, define

$$f_x(y) := \frac{r^\star(x,y) - r_{\min}}{\Delta_r} \in [0,1].$$

The variational characterization of total variation gives

$$\big|\mathbb{E}_{y\sim q(\cdot|x)}[f_x(y)] - \mathbb{E}_{y\sim\pi_\alpha(\cdot|x)}[f_x(y)]\big| \le \mathrm{TV}\big(q(\cdot \mid x), \pi_\alpha(\cdot \mid x)\big).$$

Rescaling by Δ_r and averaging over $x \sim \mu$ yields

$$\mathcal{R}(q) - \mathcal{R}(\pi_\alpha) \geq -\Delta_r \mathbb{E}_{x\sim\mu}\big[\mathrm{TV}\big(q(\cdot \mid x), \pi_\alpha(\cdot \mid x)\big)\big] .$$

Combining this inequality with Eq. (42) proves Eq. (23).

For the forward-KL bound, Pinsker's inequality and Jensen's inequality give

$$\begin{aligned}
\mathbb{E}_{x\sim\mu}\mathrm{TV}\big(q(\cdot \mid x), \pi_\alpha(\cdot \mid x)\big) &= \mathbb{E}_{x\sim\mu}\mathrm{TV}\big(\pi_\alpha(\cdot \mid x), q(\cdot \mid x)\big) \\
&\leq \mathbb{E}_{x\sim\mu}\sqrt{\frac{D_{\mathrm{KL}}\big(\pi_\alpha(\cdot \mid x)\|q(\cdot \mid x)\big)}{2}} \\
&\leq \sqrt{\frac{\mathbb{E}_{x\sim\mu}D_{\mathrm{KL}}\big(\pi_\alpha(\cdot \mid x)\|q(\cdot \mid x)\big)}{2}}.
\end{aligned}$$

Substituting this into Eq. (23) proves Eq. (24). □

Corollary 2 (Reverse-KL and self-reward RL transfer)**.** *Define*

$$\mathsf{D}_{\mathrm{rev}}(q, \pi_\alpha) := \mathbb{E}_{x\sim\mu} D_{\mathrm{KL}}\big(q(\cdot \mid x)\|\pi_\alpha(\cdot \mid x)\big).$$

If $\mathsf{D}_{\mathrm{rev}}(q, \pi_\alpha) < \infty$*, then*

$$\mathcal{R}(q) - \mathcal{R}(\pi) \geq \Gamma_\alpha - \Delta_r\sqrt{\frac{\mathsf{D}_{\mathrm{rev}}(q, \pi_\alpha)}{2}}. \tag{43}$$

Moreover, let $\beta > 0$*,* $\alpha = 1 + \beta^{-1}$*, and define the self-reward RL optimality gap*

$$\varepsilon_{\mathrm{RL}}(q) := J_\beta(\pi_\alpha; \pi, r_{\mathrm{self}}) - J_\beta(q; \pi, r_{\mathrm{self}}).$$

Whenever the objective values are finite,

$$\mathcal{R}(q) - \mathcal{R}(\pi) \geq \Gamma_\alpha - \Delta_r\sqrt{\frac{\varepsilon_{\mathrm{RL}}(q)}{2\beta}}. \tag{44}$$

Thus, if $\Gamma_\alpha > 0$*, strict improvement follows whenever*

$$\mathsf{D}_{\mathrm{rev}}(q, \pi_\alpha) < \frac{2\Gamma_\alpha^2}{\Delta_r^2}, \qquad \textit{or equivalently} \qquad \varepsilon_{\mathrm{RL}}(q) < \frac{2\beta\Gamma_\alpha^2}{\Delta_r^2}.$$

Proof. Pinsker's inequality in the reverse-KL direction and Jensen's inequality give

$$\begin{aligned}
\mathbb{E}_{x\sim\mu}\mathrm{TV}\big(q(\cdot \mid x), \pi_\alpha(\cdot \mid x)\big) &\leq \mathbb{E}_{x\sim\mu}\sqrt{\frac{D_{\mathrm{KL}}\big(q(\cdot \mid x)\|\pi_\alpha(\cdot \mid x)\big)}{2}} \\
&\leq \sqrt{\frac{\mathsf{D}_{\mathrm{rev}}(q, \pi_\alpha)}{2}}.
\end{aligned}$$

Substitution into Eq. (23) proves Eq. (43). By Eq. (15), the self-reward KL-regularized RL objective satisfies

$$J_\beta(q; \pi, r_{\mathrm{self}}) = -\beta\mathbb{E}_{x\sim\mu}D_{\mathrm{KL}}\big(q(\cdot \mid x)\|\pi_\alpha(\cdot \mid x)\big) + \beta\mathbb{E}_{x\sim\mu}[\log Z_r(x)].$$

Evaluating this identity at $q = \pi_\alpha$ and subtracting yields $\varepsilon_{\mathrm{RL}}(q) = \beta\mathsf{D}_{\mathrm{rev}}(q, \pi_\alpha)$, which proves Eq. (44). □

Corollary 3 (Finite-sample MLE transfer)**.** *Fix* $\rho \in (0, 1)$*. Suppose that* Π_α *is finite,* $\pi_\alpha \in \Pi_\alpha$*, and* $D = \{(x_i, y_i)\}_{i=1}^n$ *consists of i.i.d. samples with* $x_i \sim \mu$ *and* $y_i \sim \pi_\alpha(\cdot \mid x_i)$*. Let*

$$\widehat{\pi} \in \arg\max_{q\in\Pi_\alpha} \sum_{i=1}^n \log q(y_i \mid x_i)$$

be the MLE in Eq. (17). Then, with probability at least $1 - \rho$ *over* D*,*

$$\mathcal{R}(\widehat{\pi}) - \mathcal{R}(\pi) \geq \Gamma_\alpha - \Delta_r\sqrt{\frac{2\log\big(|\Pi_\alpha|\rho^{-1}\big)}{n}}. \tag{45}$$

In particular, the MLE improves on π whenever the finite-sample penalty in Eq. (45) is strictly smaller than Γ_α. Equivalently, if $\Gamma_\alpha > 0$ and

$$n > \frac{2\Delta_r^2 \log\bigl(|\Pi_\alpha|\rho^{-1}\bigr)}{\Gamma_\alpha^2},$$

then $\mathcal{R}(\widehat{\pi}) > \mathcal{R}(\pi)$ with probability at least $1-\rho$.

Proof. By Lemma 1, with probability at least $1-\rho$,

$$\mathbb{E}_{x\sim\mu} D_H^2\bigl(\widehat{\pi}(\cdot \mid x), \pi_\alpha(\cdot \mid x)\bigr) \le \frac{2\log\bigl(|\Pi_\alpha|\rho^{-1}\bigr)}{n}.$$

With the definition

$$D_H^2(p,q) := \sum_{y\in\mathcal{Y}} \bigl(\sqrt{p(y)} - \sqrt{q(y)}\bigr)^2,$$

we have $\mathrm{TV}(p,q) \le D_H(p,q)$. Therefore Jensen's inequality gives

$$\mathbb{E}_{x\sim\mu} \mathrm{TV}\bigl(\widehat{\pi}(\cdot \mid x), \pi_\alpha(\cdot \mid x)\bigr) \le \sqrt{\frac{2\log\bigl(|\Pi_\alpha|\rho^{-1}\bigr)}{n}}.$$

Substituting this bound into Eq. (23) proves Eq. (45). □

A.6 Closed-form optimizer for KL-regularized RL: restatement and proof

We restate the standard closed-form solution of KL-regularized RL used in Section 5.1.

Proposition 6 (Closed-form optimizer for KL-regularized RL [Levine, 2018])**.** *For each $x \in \mathcal{X}$, let $\pi_\beta^\star$ be the reward-tilted distribution defined in Eq. (13), and assume $Z_r(x) < \infty$ for every $x \in \mathcal{X}$. Then $\pi_\beta^\star$ maximizes $J_\beta(q;\pi,r)$ in Eq. (12) over all $q : \mathcal{X} \to \Delta(\mathcal{Y})$.*

Proof of Proposition 6. Fix $x \in \mathcal{X}$ and write $f(y) := \pi(y \mid x)\exp\bigl(\beta^{-1} r(x,y)\bigr)$ and $Z := Z_r(x) = \sum_{y'\in\mathcal{Y}} f(y')$. For any $q(\cdot \mid x) \in \Delta(\mathcal{Y})$, expanding the KL divergence against $\pi_\beta^\star(\cdot \mid x)$ gives

$$\begin{aligned}
\mathbb{E}_{y\sim q(\cdot|x)}[r(x,y)] - \beta\, D_{\mathrm{KL}}\bigl(q(\cdot \mid x) \,\|\, \pi(\cdot \mid x)\bigr) &= -\beta \sum_{y\in\mathcal{Y}} q(y \mid x) \log \frac{q(y \mid x)}{f(y)/Z} \\
&= -\beta\, D_{\mathrm{KL}}\bigl(q(\cdot \mid x) \,\|\, \pi_\beta^\star(\cdot \mid x)\bigr) + \beta \log Z,
\end{aligned}$$

where we used $\pi_\beta^\star(y \mid x) = f(y)/Z$ from Eq. (13). Since $D_{\mathrm{KL}}(\cdot \,\|\, \pi_\beta^\star(\cdot \mid x)) \ge 0$ with equality if and only if $q(\cdot \mid x) = \pi_\beta^\star(\cdot \mid x)$, the inner objective is uniquely maximized at $q(\cdot \mid x) = \pi_\beta^\star(\cdot \mid x)$. Because $J_\beta(q;\pi,r)$ is an expectation over $x \sim \mu$ of these decoupled per-x objectives, the unique global maximizer is $q = \pi_\beta^\star$. □

B Experimental details

B.1 Setup

Models and datasets. We used Qwen2.5-Math-7B [Yang et al., 2024b], Qwen2.5-7B [Yang et al., 2024a], and Phi-3.5-mini-instruct [Abdin et al., 2024] models on the following datasets.

- **Mathematics.** We used the MATH dataset [Lightman et al., 2024], which consists of 12,500 competition-style math problems spanning seven categories (e.g., geometry, number theory, and precalculus), with 7,500 training and 5,000 test problems. For evaluation, we used MATH500, a randomly selected subset of the MATH test set standardized by OpenAI[1]. For distillation, we sampled 500 examples from MATH with MATH500 removed[2].
- **Programming.** For evaluation, we used HumanEval [Chen et al., 2021], a set of 164 handwritten programming problems covering algorithms, reasoning, mathematics, and language understanding; each problem includes unit tests, and a solution was correct if it passed all tests. For distillation, we used MBPP [Austin et al., 2021], a benchmark of crowd-sourced Python programming problems designed to be solvable by entry-level programmers. We used 420 questions from the sanitized subset, excluding the prompt split.
- **Multiple-choice science.** We used GPQA [Rein et al., 2024], a multiple-choice science benchmark (physics, chemistry, and biology) requiring advanced reasoning. For evaluation, we used GPQA-Diamond, a high-quality subset of 198 questions. For distillation, we used the remaining 250 GPQA questions after removing any overlap with GPQA-Diamond.

Standard sampling. We used HuggingFace `generate` with `do_sample=True`. For the Qwen and Phi models, temperature, top-p, and top-k use the Transformers defaults: temperature 1.0, top-$p = 1.0$, and top-$k = 50$.

Power sampling. We used the power sampling algorithm of Karan and Du [2026], largely following their hyperparameters. Specifically, we used $\alpha = 4.0$, maximum sampling token length 3072, block size 192, $N_{\mathrm{MCMC}} = 10$, and the proposal LLM p_{prop} set to the base model with sampling temperature $\tau = 1/\alpha = 0.25$. The token-wise Temperature baseline uses the same τ, applying the corresponding local power transform independently at each decoding step. For the randomly initialized model (RandW; Section 6), we instead used maximum token length 1024 and $N_{\mathrm{MCMC}} = 2$, because under the default settings (maximum token length 3072 and $N_{\mathrm{MCMC}} = 10$) EOS tokens rarely appeared for RandW and wall-clock sampling time became significantly longer.

Self-reward computation. To report r_{self}, we computed, under the corresponding base model,, the average log-likelihood over completion tokens, excluding prompt tokens. Our theoretical analysis assumes completions of a fixed length T, but in our experiments completion lengths vary across prompts and sampling methods, so we normalize by the number of completion tokens to remove length bias in r_{self}.

Synthetic random rewards. For the synthetic-reward probe in Figure 3, each completion y is mapped to a scalar in $[0, 1)$ by applying SHA-256 to the UTF-8 encoding of y and interpreting the leading 64 bits of the digest as an unsigned fraction. Let $z_{\mathrm{self}}(y)$ and $z_r(y)$ denote the z-scores of the self-reward $r_{\mathrm{self}}(x, y)$ and of the hash reward above, each computed with the corresponding pooled global sample mean and sample standard deviation. We then define

$$r_\lambda(y) \;:=\; \lambda\, z_{\mathrm{self}}(y) + \sqrt{1-\lambda^2}\,\big(z_r(y) + \varepsilon(y)\big), \qquad \lambda \in [-1, 1],$$

where the $\varepsilon(y)$ are i.i.d. $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.5$. Figure 3 sweeps λ and plots the mean increase in r_λ under power versus standard sampling against the empirical covariance between r_{self} and r_λ, using completions produced under standard sampling. The construction is designed to sweep $\mathrm{Cov}(r_\lambda, r_{\mathrm{self}})$ in a controlled way; we plot empirical gain against this controlled covariance to visualize the qualitative rate prediction of Proposition 4.

[1] https://huggingface.co/datasets/HuggingFaceH4/MATH-500

[2] https://raw.githubusercontent.com/rasbt/math_full_minus_math500/main/math_full_minus_math500.json

Distillation. We trained the student with supervised fine-tuning on the offline power-sampled dataset. Concretely, we minimized the standard token-level cross-entropy loss of a causal language model on the teacher-generated completion, masking the prompt tokens (i.e., the loss was computed only on the completion tokens). The student was initialized from the base model and was trained with LoRA adapters (r=16, α=32, dropout 0.05) applied to `q_proj,k_proj,v_proj,o_proj,gate_proj,up_proj,down_proj`. We trained the models for 3 epochs using the AdamW optimizer with a weight decay of 0.01 and a linear warmup ratio of 0.03. The learning rate was tuned per dataset and model as summarized in Table S.2. We used per-device batch size 1 with 8 gradient accumulation steps, and enabled gradient checkpointing. We set the maximum sequence length to 1024 tokens to keep activation memory manageable on a single GPU. Teacher completions exceeding this cap were truncated, and the cross-entropy loss was computed on all in-window completion tokens. The truncation affected only a minority of completions (e.g., 83.6% of Qwen2.5-Math-7B completions on MATH fit fully within the cap), and each in-window token still provides a valid distillation signal toward π_α.

Table S.2: Learning rate used for SFT distillation, per (dataset, model) pair.

Dataset	Qwen2.5-7B	Qwen2.5-Math-7B	Phi-3.5-mini-instruct
MATH	1×10^{-5}	1×10^{-5}	1×10^{-3}
HumanEval/MBPP	1×10^{-5}	1×10^{-5}	5×10^{-4}
GPQA	1×10^{-5}	1×10^{-4}	2×10^{-4}

Hardware and execution time. All experiments were conducted on GPU nodes equipped with two Intel Xeon Platinum 8360Y CPUs, 512 GiB of host memory, and eight NVIDIA A100 GPUs with 40 GiB of memory each. On a single GPU, supervised fine-tuning of one student per dataset and model finished in under one hour, while teacher generation via power sampling (Algorithm 2) took more than one day per dataset and model. The total compute is on the order of a few hundred A100-GPU-hours.

C Additional analyses

C.1 A forward–reverse KL gap for power teachers.

Power self-distillation replaces the reverse-KL form induced by self-reward KL-regularized RL,

$$D_{\mathrm{KL}}\big(q(\cdot \mid x) \,\|\, \pi_\alpha(\cdot \mid x)\big),$$

with the offline forward-KL surrogate

$$D_{\mathrm{KL}}\big(\pi_\alpha(\cdot \mid x) \,\|\, q(\cdot \mid x)\big).$$

We now show that, for a power teacher, the effect of such a KL reversal is controlled by the variance of the same self-reward that defines the power distribution. Fix a prompt x and omit x from the notation. Let $\pi, \rho \in \Delta(\mathcal{Y})$ be strictly positive distributions on a finite support, and define logits $s(y) := \log \pi(y)$ and $r(y) := \log \rho(y)$. For a convex function F, its convex conjugate is

$$F^\star(u) := \sup_{p \in \Delta(\mathcal{Y})} \{\langle u, p\rangle - F(p)\}.$$

When $F(p) = \sum_y p(y) \log p(y)$ is the negative entropy on the simplex, this conjugate is the log-sum-exp function

$$F^\star(u) = \log \sum_y \exp(u(y)).$$

Thus $\nabla F^\star(u) = \mathrm{softmax}(u)$. Let $D_{F^\star}$ denote the Bregman divergence induced by $F^\star$. Then

$$D_{F^\star}(s\|r) = D_{\mathrm{KL}}(\rho\|\pi), \qquad D_{F^\star}(r\|s) = D_{\mathrm{KL}}(\pi\|\rho).$$

Moreover, for any vector v,

$$v^\top \nabla^2 F^\star(u) v = \mathrm{Var}_{Y \sim \mathrm{softmax}(u)}[v(Y)].$$

Applying a second-order Taylor expansion around the midpoint between r and s yields, for some intermediate logit distributions m_+, m_- along the logit-interpolation path $\mathcal{I}(\pi, \rho)$,

$$D_{\mathrm{KL}}(\rho\|\pi) = D_{\mathrm{KL}}(\pi\|\rho) + \frac{1}{4}\left\{\mathrm{Var}_{Y\sim m_+}\left[\log \frac{\rho(Y)}{\pi(Y)}\right] - \mathrm{Var}_{Y\sim m_-}\left[\log \frac{\rho(Y)}{\pi(Y)}\right]\right\}. \tag{46}$$

Therefore,

$$|D_{\mathrm{KL}}(\rho\|\pi) - D_{\mathrm{KL}}(\pi\|\rho)| \le \frac{1}{4} \sup_{m \in \mathcal{I}(\pi,\rho)} \mathrm{Var}_{Y\sim m}\left[\log \frac{\rho(Y)}{\pi(Y)}\right], \tag{47}$$

where $\mathcal{I}(\pi, \rho)$ denotes distributions of the form

$$m_\lambda(y) \propto \rho(y)^{1-\lambda} \pi(y)^\lambda.$$

Thus, for a generic teacher, the forward–reverse KL gap is controlled by the teacher–base log-density ratio, which can be large or ill-defined under support mismatch.

For the power teacher $\rho = \pi_\alpha$, however,

$$\log \frac{\pi_\alpha(Y)}{\pi(Y)} = (\alpha - 1) \log \pi(Y) - \log Z_\alpha, \qquad Z_\alpha := \sum_{y'} \pi(y')^\alpha.$$

The constant $-\log Z_\alpha$ does not affect variance, and the logit-interpolation path remains on the power path. Consequently, for some $\lambda_+ \in [(\alpha+1)/2, \alpha]$ and $\lambda_- \in [1, (\alpha+1)/2]$,

$$D_{\mathrm{KL}}(\pi_\alpha\|\pi) = D_{\mathrm{KL}}(\pi\|\pi_\alpha) + \frac{(\alpha-1)^2}{4}\Big\{\mathrm{Var}_{Y\sim\pi_{\lambda_+}}[\log \pi(Y)] - \mathrm{Var}_{Y\sim\pi_{\lambda_-}}[\log \pi(Y)]\Big\}. \tag{48}$$

Hence,

$$|D_{\mathrm{KL}}(\pi_\alpha\|\pi) - D_{\mathrm{KL}}(\pi\|\pi_\alpha)| \le \frac{(\alpha-1)^2}{4} \sup_{\lambda \in [1,\alpha]} \mathrm{Var}_{Y\sim\pi_\lambda}[\log \pi(Y)]. \tag{49}$$

It shows that, unlike in generic distillation with an arbitrary teacher, the absolute gap induced by reversing the KL in power self-distillation is not an arbitrary teacher-mismatch quantity.

D Additional experimental results

D.1 Other datasets and models

This section reports results on additional dataset–model combinations that are not shown in the main text. In all cases, the distilled model has a higher $r^\star$ than the base under standard autoregressive decoding. The distilled model often attains $r^\star$ comparable to that of the corresponding base model with power sampling.

Table S.3: MATH500: true reward $r^\star$ (accuracy) and self-reward r_{self}. Left: all completions, means with $\pm$ std over seeds. Right: self-reward Best-of-N over samples generated with different seeds (max r_{self} per item, then same aggregation). 4 seeds.

		All completions		Self-reward Best-of-N	
Model	Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
Qwen / Base	Standard	0.410 ± 0.004	-0.405 ± 0.049	0.578	-0.185
	Power	$\mathbf{0.706} \pm 0.017$	-0.093 ± 0.001	0.676	-0.080
Qwen / Power-distilled	Standard	0.631 ± 0.013	-0.094 ± 0.002	**0.682**	-0.064
	Temperature	0.661 ± 0.006	$\mathbf{-0.073} \pm 0.001$	0.676	$\mathbf{-0.060}$
Phi / Base	Standard	0.449 ± 0.014	-0.234 ± 0.001	0.476	-0.173
	Power	$\mathbf{0.513} \pm 0.017$	-0.175 ± 0.002	**0.492**	-0.155
Phi / Power-distilled	Standard	0.470 ± 0.000	-0.118 ± 0.001	0.480	-0.088
	Temperature	0.457 ± 0.014	$\mathbf{-0.106} \pm 0.001$	0.460	$\mathbf{-0.086}$

Table S.4: HumanEval: true reward $r^\star$ (HumanEval pass) and self-reward r_{self}. Left: all completions, means with $\pm$ std over seeds. Right: self-reward Best-of-N over samples generated with different seeds (max r_{self} per item, then same aggregation). 4 seeds.

		All completions		Self-reward Best-of-N	
Model	Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
Qwen-Math / Base	Standard	0.320 ± 0.016	-0.741 ± 0.012	0.384	-0.427
	Power	0.538 ± 0.030	$\mathbf{-0.144} \pm 0.003$	0.561	$\mathbf{-0.106}$
Qwen-Math / Power-distilled	Standard	0.416 ± 0.023	-0.563 ± 0.040	0.451	-0.334
	Temperature	$\mathbf{0.541} \pm 0.005$	-0.304 ± 0.001	**0.567**	-0.208
Qwen / Base	Standard	0.326 ± 0.025	-0.966 ± 0.046	0.378	-0.426
	Power	$\mathbf{0.573} \pm 0.020$	$\mathbf{-0.130} \pm 0.004$	0.567	$\mathbf{-0.096}$
Qwen / Power-distilled	Standard	0.425 ± 0.029	-0.849 ± 0.017	0.470	-0.325
	Temperature	0.541 ± 0.017	-0.479 ± 0.024	**0.598**	-0.235
Phi / Base	Standard	0.549 ± 0.021	-0.913 ± 0.012	0.561	-0.589
	Power	0.712 ± 0.027	$\mathbf{-0.330} \pm 0.004$	**0.732**	$\mathbf{-0.294}$
Phi / Power-distilled	Standard	0.634 ± 0.031	-0.730 ± 0.029	0.604	-0.473
	Temperature	$\mathbf{0.715} \pm 0.020$	-0.627 ± 0.028	0.677	-0.447

Table S.5: GPQA: true reward $r^\star$ (accuracy) and self-reward r_{self}. Left: all completions, means with $\pm$ std over seeds. Right: self-reward Best-of-N over samples generated with different seeds (max r_{self} per item, then same aggregation). 4 seeds.

		All completions		Self-reward Best-of-N	
Model	Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
Qwen-Math / Base	Standard	0.100 ± 0.025	-0.675 ± 0.076	0.101	-0.675
	Power	$\mathbf{0.277} \pm 0.022$	$\mathbf{-0.088} \pm 0.002$	0.278	$\mathbf{-0.087}$
Qwen-Math / Power-distilled	Standard	0.275 ± 0.004	-0.165 ± 0.001	**0.283**	-0.113
	Temperature	$\mathbf{0.277} \pm 0.001$	-0.149 ± 0.001	0.278	-0.109
Qwen / Base	Standard	0.244 ± 0.017	-1.531 ± 0.185	0.247	-1.527
	Power	0.283 ± 0.033	$\mathbf{-0.118} \pm 0.001$	0.288	$\mathbf{-0.118}$
Qwen / Power-distilled	Standard	0.280 ± 0.035	-0.437 ± 0.098	0.278	-0.426
	Temperature	$\mathbf{0.285} \pm 0.025$	-0.210 ± 0.007	**0.293**	-0.201
Phi / Base	Standard	0.223 ± 0.027	-0.802 ± 0.023	0.222	-0.800
	Power	$\mathbf{0.309} \pm 0.019$	$\mathbf{-0.215} \pm 0.004$	**0.308**	$\mathbf{-0.214}$
Phi / Power-distilled	Standard	0.268 ± 0.004	-0.321 ± 0.010	0.268	-0.319
	Temperature	0.292 ± 0.012	-0.284 ± 0.059	0.298	-0.262

Llama-3.1-Tulu-3-8B-DPO. To further broaden model coverage, we evaluated Llama-3.1-Tulu-3-8B-DPO [Lambert et al., 2024] on MATH500. As shown in Table S.6, power sampling and power self-distillation improve both true reward and self-reward over the base model, consistent with the trends observed for the other models.

Table S.6: MATH500 results for Llama-3.1-Tulu-3-8B-DPO. We report true reward $r^\star$ (accuracy) and self-reward r_{self}.

		All completions		Self-reward Best-of-N	
Model	Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
Tulu / Base	Standard	0.470	−0.132	0.474	−0.104
	Temperature	0.472	−0.110	0.508	−0.089
	Power	**0.510**	**−0.104**	**0.510**	−0.090
Tulu / Power-distilled	Standard	0.494	−0.112	**0.500**	−0.103
	Temperature	**0.500**	**−0.100**	0.492	**−0.093**

Harder mathematics benchmarks. We additionally evaluated Qwen2.5-Math-7B and Phi-3.5-mini-instruct on AMC23 and AIME24. As shown in Table S.7, the results exhibit the same qualitative trend on these harder mathematics benchmarks: power sampling and power self-distillation consistently increase self-reward and often improve true reward over standard sampling.

Table S.7: Results on AMC23 and AIME24 for Qwen2.5-Math-7B and Phi-3.5-mini-instruct. We report true reward $r^\star$ (accuracy) and self-reward r_{self}.

			All completions		Self-reward Best-of-N	
Model	Dataset	Method	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
Qwen2.5-Math-7B	AMC23	Base / Standard	0.350	−0.173	0.400	−0.087
		Base / Temperature	0.502	−0.063	0.575	−0.045
		Power sampling	0.525	−0.051	0.525	−0.051
		Power-distilled / Standard	0.506	−0.063	**0.600**	−0.031
		Power-distilled / Temperature	**0.575**	**−0.030**	**0.600**	**−0.018**
	AIME24	Base / Standard	0.075	−0.225	0.133	−0.100
		Base / Temperature	0.083	−0.068	0.100	−0.044
		Power sampling	**0.133**	−0.042	0.133	−0.042
		Power-distilled / Standard	**0.133**	−0.091	**0.200**	−0.052
		Power-distilled / Temperature	0.117	**−0.038**	0.100	**−0.026**
Phi-3.5-mini-instruct	AMC23	Base / Standard	0.169	−0.232	0.150	−0.172
		Base / Temperature	0.156	−0.112	0.200	−0.084
		Power sampling	**0.225**	−0.075	0.225	−0.075
		Power-distilled / Standard	0.194	−0.049	0.175	−0.034
		Power-distilled / Temperature	0.181	**−0.030**	**0.250**	**−0.023**
	AIME24	Base / Standard	0.017	−0.269	**0.033**	−0.187
		Base / Temperature	**0.033**	−0.126	**0.033**	−0.093
		Power sampling	**0.033**	−0.080	**0.033**	−0.080
		Power-distilled / Standard	**0.033**	−0.054	**0.033**	−0.035
		Power-distilled / Temperature	0.017	**−0.033**	**0.033**	**−0.024**

D.2 Power(∞)

We also evaluated Power(∞) using Qwen2.5-Math-7B on MATH500. This variant runs the MH power-sampling loop and accepts a proposal y' if and only if $\pi(y' \mid x) > \pi(y \mid x)$ (Algorithm 2), corresponding to the limit $\alpha \to \infty$.

Table S.8: Power(∞) results for Qwen2.5-Math-7B on MATH500. Left: all completions, means with $\pm$ std over seeds. Right: self-reward Best-of-N over samples generated with different seeds.

	All completions		Self-reward Best-of-N	
Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
Power(∞)	$\mathbf{0.728 \pm 0.012}$	-0.075 ± 0.001	0.736	-0.061

D.3 Self-reward GRPO and On-policy distillation

We compared power self-distillation with self-reward GRPO on MATH500 using Qwen2.5-Math-7B. The GRPO reward is the same self-reward signal used in the theory, and we report several KL coefficients β. Table S.9 shows that both GRPO and on-policy distillation (OPD) improve r_{self} over the base model under standard decoding. However, their downstream gains are smaller in this setting than those of power sampling and power self-distillation reported in Table 1.

Table S.9: MATH500 comparison for Qwen2.5-Math-7B. We report downstream true reward $r^\star$ (accuracy) and self-reward r_{self}. Left: all completions, mean with $\pm$ std over four seeds except OPD, which uses one checkpoint. Right: self-reward Best-of-N among seeds.

		All completions		Self-reward Best-of-N	
Model / Checkpoint	Sampling	$r^\star(\uparrow)$	r_{self}	$r^\star(\uparrow)$	r_{self}
GRPO (β=0.04)	Standard	0.522 ± 0.004	-0.197 ± 0.029	0.698	-0.099
	Temperature	0.690 ± 0.005	-0.051 ± 0.000	0.750	-0.036
GRPO (β=0.10)	Standard	0.495 ± 0.006	-0.217 ± 0.031	0.712	-0.112
	Temperature	0.681 ± 0.005	-0.082 ± 0.002	0.728	-0.045
GRPO (β=0.33)	Standard	0.499 ± 0.004	-0.224 ± 0.040	0.666	-0.108
	Temperature	0.699 ± 0.006	-0.071 ± 0.001	0.740	-0.043
OPD	Standard	0.496 ± 0.014	-0.272 ± 0.010	0.610	-0.124

On-policy distillation efficiency. We also compared offline power self-distillation with an on-policy distillation variant, which still requires generation inside the training loop. Table S.10 shows that, once the offline teacher dataset is fixed, power self-distillation performs only parameter updates and completes one epoch in 183.8 ± 0.3 seconds. OPD is about $31\times$ slower per update in this setup, which reinforces the motivation for amortizing power-sampling cost into a reusable offline dataset.

Table S.10: Training-time comparison for one epoch on 500 MATH prompts with Qwen2.5-Math-7B. We report mean $\pm$ standard deviation over two runs.

Training scheme	Generation time	Parameter-update time	Total time	Time / update
Power self-distillation	–	183.8 ± 0.3s	183.8 ± 0.3s	0.369 ± 0.001s
On-policy distillation	5587.0 ± 1539.1s	120.2 ± 1.1s	5707.1 ± 1540.2s	11.41 ± 3.08s

D.4 Qualitative results

This section presents full completions for one MATH-style geometry problem summarized in Table 2 with the gold answer $x + y = 17$. The prompt is:

> *The coordinates of a parallelogram are* $(5, 3)$, $(6, 8)$, $(7, 4)$, *and* (x, y) *with* $x > 7$. *What is the value of* $x + y$*?*

```
The coordinates of a parallelogram are $(5,3)$, $(6,8)$, $(7,4)$, and $(x,y)$ with $x>7$.
What is the value of $x+y$?
```

(a) Question.

```
To find the missing coordinate of the parallelogram, we can use the property that
opposite sides of a parallelogram are equal and parallel. Let's denote the given points
as A(5, 3), B(6, 8), and C(7, 4). The missing point is D(x, y).

First, we need to find the vector AB and apply it to point C to find the coordinates of
D. The vector AB is obtained by subtracting the coordinates of A from B: AB = (6-5, 8-3)
= (1, 5).

To find the coordinates of D, we add the vector AB to the coordinates of C: D = C + AB =
(7 + 1, 4 + 5) = (8, 9).

Therefore, the value of $x + y$ is $8 + 9 = 17$.
```

(b) Base sampling.

```
To find the value of $x + y$, we first need to determine the coordinates of the fourth
vertex of the parallelogram. Since the opposite sides of a parallelogram are equal and
parallel, we can find the fourth vertex by using the properties of parallelograms.

Let's denote the fourth vertex as $(x, y)$. Since the opposite sides are equal, we can
write the following equations based on the given coordinates:

1. $x - 5 = 6 - x$
2. $y - 3 = 8 - y$

Solving these equations, we get:

1. $2x = 11 \Rightarrow x = \frac{11}{2}$
2. $2y = 11 \Rightarrow y = \frac{11}{2}$

However, since $x > 7$, we need to adjust our solution. The correct coordinates for the
fourth vertex should be $(8, 6)$ to satisfy the condition $x > 7$.

Therefore, the value of $x + y$ is $8 + 6 = 14$.
```

(c) Temperature sampling.

Figure S.1: Full generations for an example in MATH500 (gold $x + y = 17$) (Part 1/4).

````
Can you solve the following math problem? The coordinates of a parallelogram are (5, 3), (6, 8), (7, 4)
and $(x, y)$ and $x > 7$. What is the value of $x + y$? Please reason step by step, and put your final
answer within \boxed{{}}. To solve this problem, we need to use the properties of a parallelogram. In a
parallelogram, the diagonals bisect each other. This means that the midpoint of one diagonal is the same
as the midpoint of the other diagonal.

Given the coordinates of three vertices of the parallelogram as (5, 3), (6, 8), and (7, 4), let's denote
the fourth vertex as $(x, y)$. The midpoint of the diagonal connecting (5, 3) and $(x, y)$ should be the
same as the midpoint of the diagonal connecting (6, 8) and (7, 4).

The midpoint of a line segment with endpoints $(x_1, y_1)$ and $(x_2, y_2)$ is given by:
\[
\left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)
\]

So, the midpoint of the diagonal connecting (5, 3) and $(x, y)$ is:
\[
\left( \frac{5 + x}{2}, \frac{3 + y}{2} \right)
\]

And the midpoint of the diagonal connecting (6, 8) and (7, 4) is:
\[
\left( \frac{6 + 7}{2}, \frac{8 + 4}{2} \right) = \left( \frac{13}{2}, 6 \right)
\]

Since these midpoints are the same, we can set up the following equations:
\[
\frac{5 + x}{2} = \frac{13}{2}
\]
\[
\frac{3 + y}{2} = 6
\]

Solving these equations will give us the values of $x$ and $y$.

Let's solve these equations step by step using Python.

```python
from sympy import symbols, Eq, solve

# Define the variables
x, y = symbols('x y')

# Define the equations based on the midpoints
eq1 = Eq((5 + x) / 2, 13 / 2)
eq2 = Eq((3 + y) / 2, 6)

# Solve the equations
solution = solve((eq1, eq2), (x, y))
print(solution)
```

```output
{x: 8.00000000000000, y: 9.00000000000000}
```

The solution to the equations is $x = 8$ and $y = 9$. Therefore, the coordinates of the fourth vertex are
$(8, 9)$.

Now, we need to find the value of $x + y$. Since $x = 8$ and $y = 9$, we have:
\[
x + y = 8 + 9 = 17
\]

So, the value of $x + y$ is $\boxed{17}$.
````

(d) Power sampling.

Figure S.1: Full generations for an example in MATH500 (gold $x + y = 17$) (Part 2/4).

```
To solve for the coordinates \((x, y)\) of the parallelogram and find the value of \(x + y\), we need to
use the properties of parallelograms. Specifically, the diagonals of a parallelogram bisect each other.
This means that the midpoint of one diagonal is equal to the midpoint of the other diagonal.

Given the coordinates of three vertices of the parallelogram: \((5, 3)\), \((6, 8)\), and \((7, 4)\), we
need to find the coordinates of the fourth vertex \((x, y)\).

First, let's find the midpoint of the diagonal formed by the vertices \((5, 3)\) and \((7, 4)\):
\[
\text{Midpoint} = \left( \frac{5 + 7}{2}, \frac{3 + 4}{2} \right) = \left( \frac{12}{2}, \frac{7}{2}
\right) = (6, 3.5)
\]

Next, we find the midpoint of the diagonal formed by the vertices \((6, 8)\) and \((x, y)\):
\[
\text{Midpoint} = \left( \frac{6 + x}{2}, \frac{8 + y}{2} \right)
\]

Since the diagonals bisect each other, these midpoints must be equal:
\[
\left( \frac{6 + x}{2}, \frac{8 + y}{2} \right) = (6, 3.5)
\]

We now set up the equations by equating the corresponding coordinates:
\[
\frac{6 + x}{2} = 6 \quad \text{and} \quad \frac{8 + y}{2} = 3.5
\]

Solving the first equation for \(x\):
\[
\frac{6 + x}{2} = 6
\]
\[
6 + x = 12
\]
\[
x = 6
\]

Solving the second equation for \(y\):
\[
\frac{8 + y}{2} = 3.5
\]
\[
8 + y = 7
\]
\[
y = -1
\]

Thus, the coordinates of the fourth vertex are \((6, -1)\). However, we are given that \(x > 7\), which
means there might be an error in our initial assumption or calculation. Let's re-evaluate the problem.

Given \(x > 7\), we need to consider the other possible diagonal pair. Let's use the vertices \((5, 3)\)
and \((x, y)\) for one diagonal and \((6, 8)\) and \((7, 4)\) for the other diagonal.

First, find the midpoint of the diagonal formed by \((6, 8)\) and \((7, 4)\):
\[
\text{Midpoint} = \left( \frac{6 + 7}{2}, \frac{8 + 4}{2} \right) = \left( \frac{13}{2}, \frac{12}{2}
\right) = (6.5, 6)
\]
```

(e) Power-distilled model with base sampling (Part 1/2).

Figure S.1: Full generations for an example in MATH500 (gold $x + y = 17$) (Part 3/4).

```
Next, find the midpoint of the diagonal formed by \((5, 3)\) and \((x, y)\):
\[
\text{Midpoint} = \left( \frac{5 + x}{2}, \frac{3 + y}{2} \right)
\]

Since the diagonals bisect each other, these midpoints must be equal:
\[
\left( \frac{5 + x}{2}, \frac{3 + y}{2} \right) = (6.5, 6)
\]

We now set up the equations by equating the corresponding coordinates:
\[
\frac{5 + x}{2} = 6.5 \quad \text{and} \quad \frac{3 + y}{2} = 6
\]

Solving the first equation for \(x\):
\[
\frac{5 + x}{2} = 6.5
\]
\[
5 + x = 13
\]
\[
x = 8
\]

Solving the second equation for \(y\):
\[
\frac{3 + y}{2} = 6
\]
\[
3 + y = 12
\]
\[
y = 9
\]

Thus, the coordinates of the fourth vertex are \((8, 9)\). Since \(x > 7\), this solution is
valid.

Finally, we find \(x + y\):
\[
x + y = 8 + 9 = 17
\]

Therefore, the value of \(x + y\) is \(\boxed{17}\).
```

(f) Power-distilled model with base sampling (Part 2/2).

Figure S.1: Full generations for an example in MATH500 (gold $x + y = 17$) (Part 4/4).

D.5 Synthetic validation of suffix-Rényi odds corrections

To validate Proposition 1 in a setting that reflects the Zipf-like word-frequency structure of natural language, we construct a finite synthetic autoregressive distribution whose language-model next-token probabilities follow a Zipf-like law over many candidates. Unlike the extreme pivotal-token construction of Karan and Du [2026], every next-token candidate is followed by a full-support suffix distribution. The construction is summarized in Figure S.2. The base next-token distribution has $V = 64$ tokens with Zipf-like probabilities

$$p_i \propto (i+1)^{-1.05}, \qquad i = 0, \ldots, V-1.$$

For every token i, the conditional suffix distribution q_i has the same support size $M = 256$, no zero-probability suffixes, and a non-uniform power-law shape

$$q_i(z) \propto z^{-s_i}, \qquad z = 1, \ldots, M.$$

The suffix exponent $s_i \in [0.45, 1.65]$ varies deterministically and non-monotonically with the next-token rank, using a sinusoidal component plus a small trend. Thus, all suffix distributions have identical support size and full support, but differ in sharpness. This deliberately avoids the singular-versus-uniform example in Karan and Du [2026]: the experiment isolates the more general quantity identified by Proposition 1, namely the suffix Rényi entropy. In Figure S.2, the left panel shows the Zipf-like next-token distribution, the middle panel shows the token-dependent suffix exponent s_i, and the right panel shows representative full-support suffix distributions.

For each $\alpha \in \{1.1, 1.5, 2, 3, 4, 8\}$, we compute both the token-wise temperature next-token distribution and the sequence-level power next-token conditional exactly under this synthetic distribution. The temperature next-token distribution is

$$\pi_{\mathrm{temp},\alpha}(i) = \frac{p_i^\alpha}{\sum_j p_j^\alpha},$$

whereas the next-token conditional induced by the sequence-level power distribution is

$$\pi_{\mathrm{pow},\alpha}(i) = \frac{p_i^\alpha \sum_z q_i(z)^\alpha}{\sum_j p_j^\alpha \sum_z q_j(z)^\alpha}.$$

Figure S.3 compares the two sides of Proposition 1 for every unordered token pair and every tested α. The left panel plots the Rényi-predicted log odds correction against the directly computed power-versus-temperature log odds correction, while the right panel shows the distribution of these corrections at the main experimental exponent $\alpha = 4$.

Figure S.4 illustrates the consequence of the correction at the level of next-token preferences: even when $p_i > p_j$ and temperature favors token i, sequence-level power can favor token j if q_j has sufficiently lower suffix Rényi entropy.

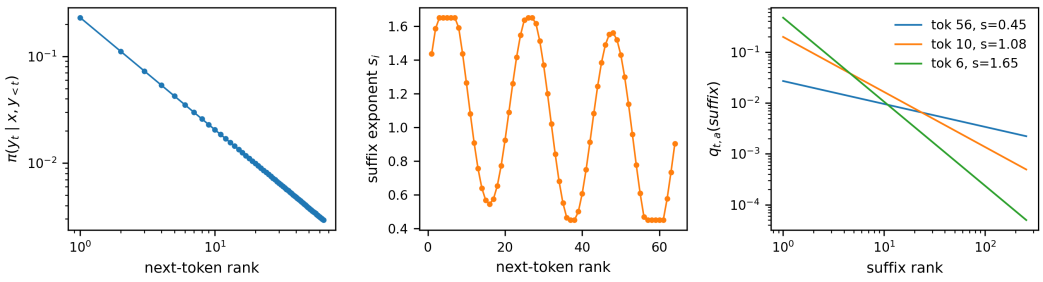

Figure S.2: Synthetic distribution setting. Left: the base next-token distribution p_i is Zipf-like over 64 tokens. Middle: suffix sharpness varies non-monotonically with the next-token rank through the power-law exponent s_i. Right: representative suffix distributions q_i are full-support, non-uniform power laws over the same support size $M = 256$. This setup differs from the singular-versus-uniform pivotal-token construction of Karan and Du [2026].

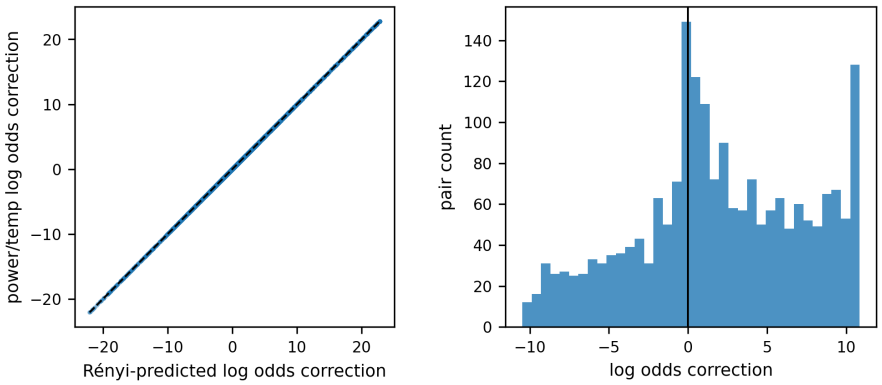

Figure S.3: Suffix Rényi entropy coincides with the power-vs-temperature odds gap. For every unordered token pair (i, j) and every tested α, we compare the Rényi-predicted log correction, $(1 - \alpha)(H_\alpha(q_i) - H_\alpha(q_j))$, with the closed-form log odds correction, $\log((\pi_{\mathrm{pow},\alpha}(i)/\pi_{\mathrm{pow},\alpha}(j))/(\pi_{\mathrm{temp},\alpha}(i)/\pi_{\mathrm{temp},\alpha}(j)))$. The diagonal agreement shows that the suffix-Rényi formula explains the full pairwise gap in a many-token full-support setting. The right panel shows the distribution of closed-form corrections at $\alpha = 4$.

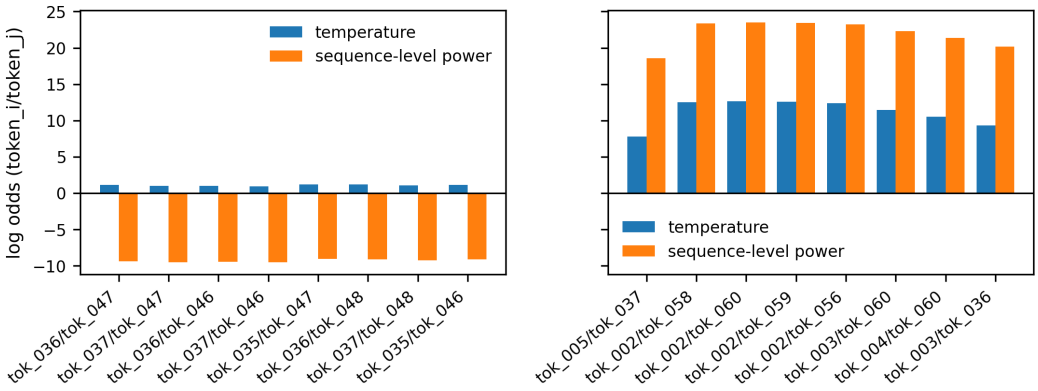

Figure S.4: Suffix entropy can reverse next-token preferences under sequence-level power. Pairs are ordered so that $i < j$, hence the Zipf base next-token distribution gives $p_i > p_j$, and token-wise temperature has positive log odds for i over j. Left: examples where the sequence-level power next-token conditional reverses this ordering because token j has a sharper, lower-entropy suffix distribution. Right: control examples where the Rényi entropy is not large enough to reverse the preferred token. All bars are closed-form log odds, not sampled frequencies.

D.6 Synthetic validation of optimal one-step proposals for sequential power sampling

We reuse the synthetic distribution of Section D.5 to validate Proposition 2. For a fixed prompt and an empty prefix, the unique variance-minimizing one-step proposal in Equation (11) reduces to

$$q^\star(i) \;\propto\; p_i^\alpha \sum_{z=1}^{M} q_i(z)^\alpha, \qquad i = 0, \dots, V-1,$$

which equals the next-token conditional of the sequence-level power distribution $\pi_{\mathrm{pow},\alpha}$ and depends on the suffix power masses $\sum_z q_i(z)^\alpha$ of every candidate token. We compare $q^\star$ with three one-step proposals that do not use those suffix totals: the base proposal $q^{\mathrm{base}}(i) = p_i$, the token-wise temperature proposal $q^{\mathrm{temp}}(i) \propto p_i^\alpha$, and a uniform reference $q^{\mathrm{unif}}(i) = 1/V$.

For each proposal q, the first-step incremental importance weight in Equation (10) simplifies to

$$W_1(i) \;=\; \frac{q^\star(i)}{q(i)},$$

and we show its exact mean, the coefficient of variation $\mathrm{CV}^2(W_1) = \mathrm{Var}[W_1]/\mathbb{E}[W_1]^2$, and the effective sample size fraction $\mathrm{ESS}/N = 1/(1 + \mathrm{CV}^2(W_1))$. By Proposition 2, only $q^\star$ achieves $\mathrm{Var}[W_1] = 0$ and hence $\mathrm{ESS}/N = 1$; the closed-form values for the other proposals are computed exactly from the synthetic distribution.

Figure S.5 compares the four proposals at $\alpha = 4$. The left panel shows the proposal probabilities; the oracle proposal equals the target next-token conditional $\pi_{\mathrm{pow},\alpha}$ by construction, and the temperature, base, and uniform proposals deviate from it, especially on next-token ranks where the suffix exponent s_i is small and $\sum_z q_i(z)^\alpha$ is large. The right panel plots $\log W_1(i)$: only the oracle proposal yields a constant log weight, while the other proposals produce token-dependent log weights.

Figure S.6 reports the exact ESS/N and $\mathrm{CV}^2(W_1)$ as a function of α. The oracle proposal attains $\mathrm{ESS}/N = 1$ for every α, whereas the gap between the temperature proposal and the oracle widens as α grows, because larger α amplifies the suffix power masses that the local temperature transform ignores.

Figure S.7 checks the same conclusion with Monte Carlo: for each proposal we draw N tokens, compute the self-normalized ESS, and average across replicates. The sampled ESS/N concentrates around the exact values from Figure S.6 as N grows, and the ordering of the proposals is preserved at every particle budget.

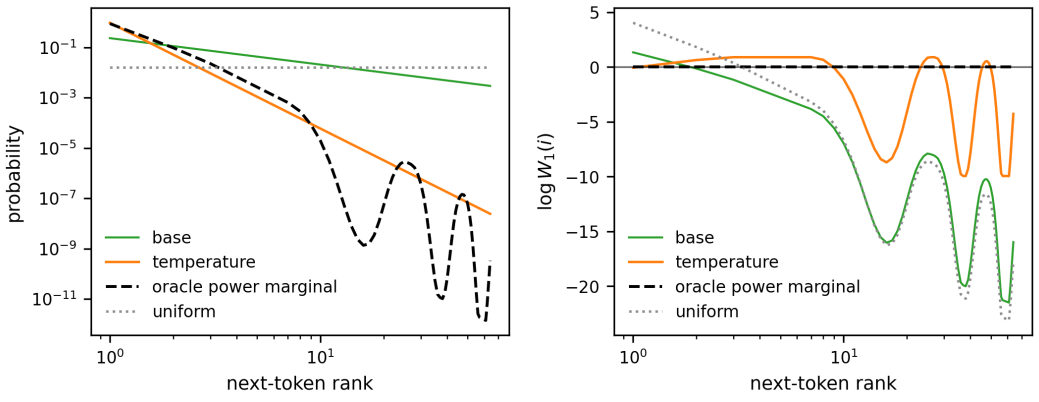

Figure S.5: One-step SIS proposals for the sequence-level power target. Left: proposal probabilities at $\alpha = 4$. The oracle proposal $q^\star$ of Equation (11) equals the target next-token conditional $\pi_{\mathrm{pow},\alpha}$ by construction, while the temperature, base, and uniform proposals deviate from it. Right: log incremental weight $\log W_1(i)$ for each proposal; the oracle yields a constant log weight, while base, temperature, and uniform proposals do not.

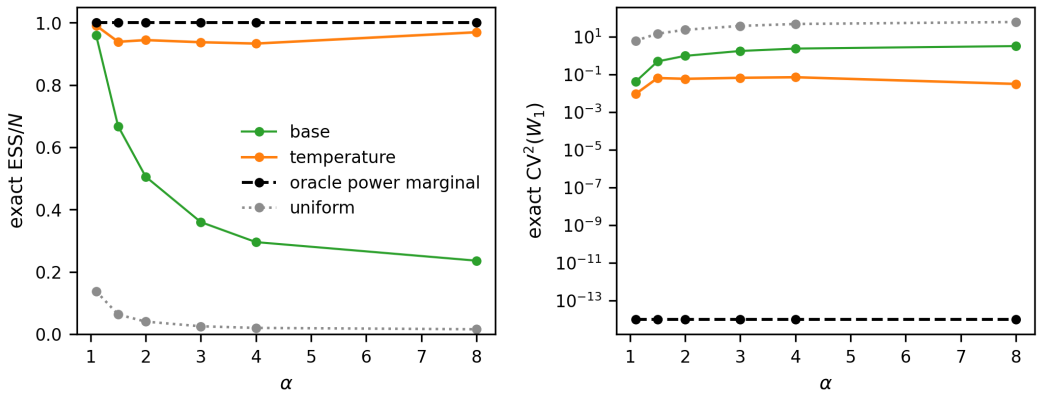

Figure S.6: Exact first-step weight variance and effective sample size. Left: exact $\mathrm{ESS}/N = 1/(1+\mathrm{CV}^2(W_1))$ for each one-step proposal as a function of α. Right: exact $\mathrm{CV}^2(W_1)$ on a log scale. The oracle proposal $q^\star$ achieves $\mathrm{Var}[W_1] = 0$ at every α, confirming Proposition 2, while the temperature proposal degrades as α grows.

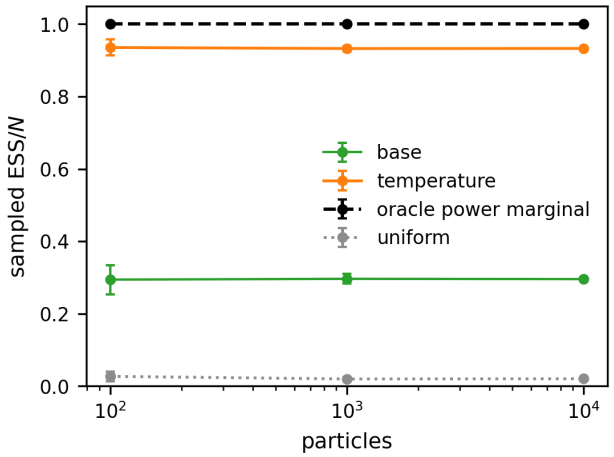

Figure S.7: Monte Carlo effective sample size at $\alpha = 4$. For each proposal we draw N tokens, compute the self-normalized ESS, and average across replicates; error bars show one standard deviation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the main theoretical and empirical claims: the structural difficulty of inexpensive power-sampling approximations, the derivation of power self-distillation from a KL-regularized self-reward RL objective, the sharpening and covariance characterizations, and the empirical comparison on reasoning tasks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The conclusion includes a limitations paragraph discussing dependence on the base model's capabilities and the focus on autoregressive language models with finite completion horizons.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The main theoretical statements specify their assumptions in the propositions and corollary, and the appendix provides the proofs for all results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The paper describes the evaluated models, datasets, splits, power-sampling procedure, distillation setup, LoRA configuration, optimizer, and hyperparameters in the main text and appendix. These details are intended to make the reported comparisons reproducible.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The experiments use publicly available datasets and models cited in the paper, but the current submission does not include public code or generated data to preserve anonymity. We plan to release the code and reproduction instructions upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The experimental setup specifies the models, datasets, evaluation subsets, distillation data sources, power-sampling parameters, baselines, LoRA configuration, optimizer, learning rates, sequence length, batch size, gradient accumulation, and training schedule.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: The main and appendix tables report means with standard deviations, and the covariance experiment reports standard-deviation error bars.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Section B.1 reports the hardware configuration used for the experiments (CPU type, host memory, GPU type, number of GPUs per node, and GPU memory) together with execution times.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work analyzes and evaluates post-training and inference methods for publicly available language models on standard academic benchmarks, and we are not aware of any deviation from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

    Justification: This paper is foundational research on theoretical and empirical properties of power sampling and self-distillation, and does not introduce a deployed system, dataset release, or application-specific intervention with direct societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release a new pretrained language model, scraped dataset, or other high-risk asset. The experiments use existing public models and benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: The paper uses existing public models and datasets, and properly credits them through citations to the original papers and resources in the experimental setup.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A].

Justification: The paper introduces a method and reports experiments, but the current submission does not release a new dataset, code package, or model checkpoint as a new asset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The work does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The work does not involve crowdsourcing experiments or research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology of this research does not involve the use of LLMs in any important, original, or non-standard way.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.