arXiv:2608.09271v1 [cs.LG] 10 Aug 2026

SoftmaxGRPO: Learning to Reason using Softmax Advantage Group Estimation

Jefferson Hernandez[1], **Jaywon Koo**[1], **Zilin Xiao**[1], **Chen Wei**[1] **and Vicente Ordonez**[1]
[1]Rice University

Group-based reinforcement learning objectives such as GRPO can allocate learning signal poorly across prompt difficulty: under binary rewards, group normalization induces a divergent weighting on easy prompts. We introduce Softmax Advantage Group Estimation (SoftmaxGRPO), a drop-in alternative that replaces z-score-normalized group advantages with temperature-scaled softmax advantages, keeping weights bounded regardless of prompt difficulty. For binary rewards, we derive the exact finite-group population objective and identify MaxRL as its low-temperature limit. For bounded scalar rewards, we show that the large-group update exactly optimizes a log-moment-generating-function objective, while a universal finite-group scalar objective cannot exist without additional assumptions on the reward distribution. Empirically, SoftmaxGRPO reallocates measured gradient budget away from near-solved prompts and consistently improves over GRPO under identical rewards. It reaches 51.8% on DeepMath with verifiable rewards and improves a 1.5B instruction-tuned model from 35.0% to 68.0% on Poetry using only lightweight text-similarity rewards.

RICE

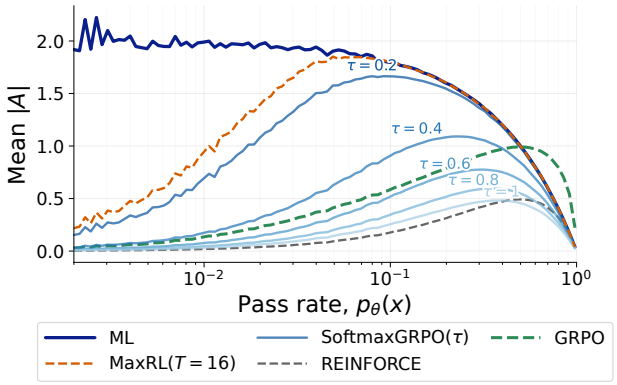
1. Introduction

Group-based reinforcement learning objectives such as GRPO have become standard for post-training [33, 35, 39]: sample M rollouts per prompt, normalize within-group rewards, and optimize a PPO-style surrogate. The normalization choice is not cosmetic. Under binary correctness rewards, groupwise objectives are best understood as optimizing different monotone transforms of pass probability, each inducing a distinct prompt-difficulty weighting [6, 36]. GRPO normalization induces a weighting that diverges on easy prompts, over-concentrating gradient signal on problems the model already solves reliably. Practical variants—Dr.GRPO, DAPO, CISPO, DPPO [4, 21, 27, 43]—address symptoms of this imbalance without replacing the underlying objective geometry.

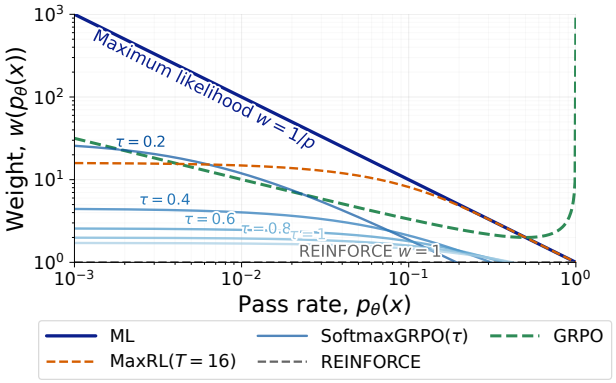
This pathology is especially damaging outside the narrow regime of tasks with cheap automatic verifiers. Many settings of practical interest—summarization, open-ended question answering, creative generation—supply only weak answer-level signals such as string-overlap scores (ROUGE, BLEU) against reference outputs. Weak rewards are noisy and sparse; a poorly shaped objective that wastes gradient budget on already-easy prompts compounds this problem, leaving even less signal where learning is most needed.

We introduce **Softmax Advantage Group Estimation (SoftmaxGRPO)**, a one-line replacement for GRPO: given M rollouts with rewards $\{r_i\}$, form within-group weights $w_i \propto \exp(r_i/\tau)$ and centered advantages $A_i = Mw_i - 1$. For binary rewards in the unclipped on-policy regime, SoftmaxGRPO induces an exact finite-M objective $h_{M,\tau}(p)$ with bounded prompt weighting; Figure 1 illustrates how τ moves SoftmaxGRPO between REINFORCE-like and MaxRL-like behavior, approaching maximum-likelihood weighting only in the joint low-temperature, large-group limit. For bounded scalar rewards, the large-group update exactly optimizes the log moment-generating function of reward. This result is also sharp: with three or more reward levels, the finite-group update is generally non-conservative, so no universal scalar analogue of $h_{M,\tau}$ exists without additional assumptions. At finite M, the weights retain the standard RAML/MPO-style exponential-tilting interpretation [1, 25]; in experiments, we optimize them with PPO clipping and reference-model KL regularization.

Corresponding author(s): jefehern@rice.edu



(a) Mean absolute advantage for ML, GRPO, REINFORCE, MaxRL, and SoftmaxGRPO(τ) across pass rates, estimated by Monte Carlo over groups of M binary rollouts.



(b) Population-level prompt-weighting functions $w(p)$ as a function of pass rate p for ML, GRPO, REINFORCE, MaxRL, and SoftmaxGRPO(τ).

Figure 1: SoftmaxGRPO defines a smooth objective family over prompt difficulty. (a) Adjusting τ changes the gradient signal magnitude, moving from REINFORCE-like toward MaxRL-like behavior at finite M. (b) The induced population weights show how SoftmaxGRPO reallocates learning signal as pass rate changes. The joint low-temperature, large-group limit approaches ML weighting, while SoftmaxGRPO avoids GRPO's easy-prompt blow-up.

Our contributions are:

- We introduce SoftmaxGRPO, a one-line drop-in replacement for GRPO that substitutes temperature-scaled softmax advantages for z-score group advantages, keeping weights bounded at all pass rates.
- We derive the exact finite-M binary-reward objective, establish its MaxRL limit, and prove an exact large-group objective for bounded scalar rewards. We also show why a universal finite-M scalar objective generally cannot exist beyond binary rewards.
- Empirically, SoftmaxGRPO reallocates gradient budget away from near-solved prompts, outperforms GRPO under identical weak rewards, and performs strongly across both verifiable and non-verifiable tasks.

2. Related Work

Reasoning RL and group-based policy objectives. Reinforcement learning with verifiable rewards (RLVR) has driven strong gains on mathematical reasoning and code generation by optimizing outcome signals from automatic checkers at scale [20, 33, 35]. More recently, this type of model post-training has also proven useful in multimodal reasoning [8, 15, 37, 38, 40, 41]. Within this paradigm, group-based objectives such as GRPO have become standard, normalizing within-group rewards before forming policy-gradient targets. Several practical variants address instabilities arising from this normalization through adaptive clipping, filtering, and reweighting schemes [4, 21, 27, 43]. Rather than patching the normalization, SoftmaxGRPO replaces it with a temperature-scaled softmax that admits an exact population-level analysis and naturally interpolates between distinct optimization regimes.

RAML, softmax policy gradient, and exponentiated-reward methods. Exponentiated-reward weighting itself is well established. Reward-Augmented Maximum Likelihood (RAML) forms reward-shaped maximum-likelihood targets [25]; Optimal Completion Distillation extends related ideas to per-prefix targets [31]; and softmax policy gradient and MPO use exponentiated-reward or advantage targets for policy improvement [1, 9]. Our contribution is not the softmax construction, but its group-level geometry: the exact finite-M prompt-weighting objective under binary rewards, its MaxRL limit, the large-group scalar-reward objective, and the finite-group obstruction beyond binary rewards.

Weak-rewards and non-verifiable training. Extending RLVR beyond verifiable domains requires either domain-specific reward models or tolerance for weak, noisy signals such as string-overlap metrics against reference outputs [24, 35]. When only the final answer is scored, the training signal provides little guidance about which parts of a long generation should change and can incentivize rationales weakly tied to the actual computation [19]; process-level constraints partially address this but inherit the same fundamental sparsity [45]. Self-supervised objectives derived from unlabeled text suggest appealing scaling properties but are typically coupled to pre-training rather than post-training fine-tuning [13]. SoftmaxGRPO targets this harder post-training regime directly, using standard gold-answer supervision with on-policy sampling to reshape objective geometry, rather than requiring step-by-step annotations or a separate reward model.

3. SoftmaxGRPO: Softmax Advantage Group Estimation

SoftmaxGRPO is a one-line replacement for GRPO-style z-score normalization in PPO-based reasoning post-training [32, 33]: given a group of M rollouts, replace standardized rewards with a softmax over rewards. The cleanest population picture comes from binary rewards, where this choice induces a prompt-weighting family over pass probability p. We make that weight function the organizing principle of this section.

3.1. Update rule

For an input x, let $\mathcal{G}(x) = \{z_i\}_{i=1}^{M}$ be a sampled rollout group, decode $y_i = \mathrm{Dec}(z_i)$, and compute rewards $R_i := R(x, y_i)$. SoftmaxGRPO forms

$$w_i = \frac{\exp(R_i/\tau)}{\sum_{j=1}^{M} \exp(R_j/\tau)}, \qquad A_i = M w_i - 1. \tag{1}$$

Since $\sum_{i=1}^{M} A_i = 0$, SoftmaxGRPO is invariant to additive reward shifts within the group and produces no update when all sampled rollouts receive the same reward. The underlying unclipped group objective is

$$\mathcal{J}^{\mathrm{uc}}_{\mathrm{SoftmaxGRPO}}(\theta; x, \mathcal{G}) = \frac{1}{M} \sum_{i=1}^{M} A_i \log \pi_\theta(z_i \mid x). \tag{2}$$

3.2. SoftmaxGRPO under binary rewards

We now specialize in binary correctness rewards and in the unclipped on-policy regime. For binary rewards, the expected SoftmaxGRPO update on a prompt with pass probability p takes the form $\omega(p)\,\nabla_\theta p_\theta(x)$. In large groups, that weight can be read off directly from the softmax.

Let $p := p_\theta(x)$ and $c := e^{1/\tau}$. In a large group, a fraction p of rollouts are correct and carry unnormalized softmax mass c, while a fraction $1 - p$ are incorrect and carry mass 1. The total within-group mass is therefore $1 - p + pc$, and the success–failure gap induced by SoftmaxGRPO is

$$\omega_\tau(p) \approx \frac{c - 1}{1 - p + pc} = \frac{e^{1/\tau} - 1}{1 - p + pe^{1/\tau}}. \tag{3}$$

This is the main theoretical insight: SoftmaxGRPO defines a temperature-controlled family of prompt-weighting functions. Large τ makes $\omega_\tau(p)$ nearly constant, recovering REINFORCE-like weighting; small τ shifts mass toward hard prompts, approaching $1/p$ and thus maximum-likelihood-style weighting in the joint low-temperature, large-group limit. Unlike GRPO, whose binary-reward weighting scales as $[p(1-p)]^{-1/2}$ [6, 36], SoftmaxGRPO remains finite as $p \to 1$, so it does not overemphasize already-solved prompts.

As throughout this comparison, only the shape of $\omega(p)$ matters; positive p-independent rescalings can be absorbed into the step size in the unclipped population picture.

Objective	Weight $\omega(p)$	Geometry
REINFORCE	1	Uniform weighting over prompt difficulty.
ML	$\frac{1}{p}$	Train-on-successes geometry; strongly emphasizes hard prompts.
GRPO	$\frac{1}{\sqrt{p(1-p)}}$	Blows up on both very hard and already-easy prompts.
MaxRL(T)	$\frac{1-(1-p)^T}{p}$	Truncated ML weighting; caps the hard-prompt blow-up for finite T.
SoftmaxGRPO	$\frac{e^{1/\tau}-1}{1-p+pe^{1/\tau}}$	REINFORCE-to-MaxRL interpolation at finite M; ML-like in the joint low-temperature, large-group limit; finite at $p \to 1$.

Table 1: Prompt-weighting view of binary-reward objectives. We show weight functions up to positive p-independent scaling. SoftmaxGRPO is shown in its large-M form; the exact finite-M weight is given in Eq. (4), and the low-temperature finite-M limit in Eq. (5).

3.3. Limit behavior of SoftmaxGRPO

The mean-field form above is the main intuition. The exact finite-M binary-reward objective has the same structure. Let $S_i := \sum_{j \neq i} R_j$ be the number of other successful rollouts seen by sample i. Conditioned on $S_i = s$, a successful rollout and an unsuccessful rollout receive weights

$$w_s^{(1)} = \frac{c}{M + (s+1)(c-1)}, \qquad w_s^{(0)} = \frac{1}{M + s(c-1)}.$$

Their centered-advantage gap is therefore

$$\Delta_s^{(\tau)} := M\big(w_s^{(1)} - w_s^{(0)}\big).$$

Averaging over $S \sim \mathrm{Binomial}(M-1, p)$ yields the exact prompt-weighting function

$$\begin{aligned} \mathbb{E}_{\mathcal{G}(x)\sim\pi_\theta(\cdot|x)}[\nabla_\theta \mathcal{J}^{\mathrm{uc}}_{\mathrm{SoftmaxGRPO}}(\theta; x, \mathcal{G})] &= \omega_{M,\tau}(p_\theta(x))\, \nabla_\theta p_\theta(x), \\ \omega_{M,\tau}(p) &= \mathbb{E}_{S\sim\mathrm{Binomial}(M-1,p)}\left[\Delta_S^{(\tau)}\right]. \end{aligned} \tag{4}$$

Thus SoftmaxGRPO optimizes a scalar transform $h_{M,\tau}(p)$ with derivative $h'_{M,\tau}(p) = \omega_{M,\tau}(p)$. The closed-form expression for $h_{M,\tau}$, together with the Bernstein-polynomial representation of $\omega_{M,\tau}$, is given in Appendix A.

Most importantly, the low-temperature finite-M limit is exact:

$$\omega_{M,0}(p) = \frac{1-(1-p)^{M-1}}{p}. \tag{5}$$

This is exactly the MaxRL weighting with truncation $T = M-1$ [36]. SoftmaxGRPO is a smooth temperature-parameterized method that interpolates from REINFORCE to MaxRL as τ shrinks, and then to ML-like $1/p$ weighting as M grows.

3.4. General scalar rewards: objective and limitation

The binary result above is exact at finite M. A different exact objective emerges for bounded scalar rewards in the large-group limit. Define

$$Z_\tau(\theta; x) := \mathbb{E}_{z\sim\pi_\theta(\cdot|x)}\left[e^{R(x,z)/\tau}\right]. \tag{6}$$

For an i.i.d. on-policy group, the softmax denominator concentrates around MZ_τ. Applying the score identity to Eq. (2) therefore gives the exact limit

$$\begin{aligned}\lim_{M\to\infty} \mathbb{E}_{\mathcal{G}(x)}[\nabla_\theta \mathcal{J}^{\mathrm{uc}}_{\mathrm{SoftmaxGRPO}}] &= \mathbb{E}_{z\sim\pi_\theta}\left[\left(\frac{e^{R(x,z)/\tau}}{Z_\tau(\theta;x)} - 1\right)\nabla_\theta \log \pi_\theta(z \mid x)\right] \\ &= \nabla_\theta \log Z_\tau(\theta;x).\end{aligned} \tag{7}$$

Thus large-group SoftmaxGRPO optimizes the log moment-generating function of reward, or equivalently the exponential-utility objective up to a positive factor of τ. For binary reward, $Z_\tau = 1 - p + pe^{1/\tau}$, and Eq. (7) recovers the large-M weight in Eq. (3) exactly. Appendix A.1 gives the full proof, finite-M counterexample, and Gaussian-reward example; Appendix A.2 shows stability to small bounded noise around binary rewards.

This large-group result cannot generally be strengthened to a finite-M scalar objective. Consider $M = 2$ and three reward levels with probabilities p_k and $t_k = e^{r_k/\tau}$. The expected coefficient for level k is $m_k(p) = 2\sum_\ell p_\ell t_k/(t_k + t_\ell)$. On the simplex $p_3 = 1 - p_1 - p_2$, a scalar potential would require the one-form $(m_1 - m_3)\,dp_1 + (m_2 - m_3)\,dp_2$ to be closed. For $t = (1, 2, 4)$, however,

$$\frac{\partial(m_1 - m_3)}{\partial p_2} = -\frac{1}{15} \neq \frac{1}{15} = \frac{\partial(m_2 - m_3)}{\partial p_1}. \tag{8}$$

Hence the finite-group update is generically non-conservative once the reward distribution has three or more levels. Binary rewards are special because their state is one-dimensional, where the scalar transform $h_{M,\tau}(p)$ exists automatically.

At any finite M, the sampled weights still have a useful variational characterization. Let $u_i = 1/M$. Then

$$q^\star = \arg\max_{q\in\Delta^M}\left\{\sum_{i=1}^M q_i R_i - \tau\,\mathrm{KL}(q\|u)\right\}, \qquad q_i^\star = w_i. \tag{9}$$

This standard exponential tilt connects SoftmaxGRPO to RAML, softmax policy gradient, and MPO [1, 9, 25]. As $\tau \to \infty$, $A_i = (R_i - \bar{R})/\tau + O(\tau^{-2})$, recovering centered reward-weighted policy gradient up to scale. As $\tau \downarrow 0$, mass concentrates on the highest-reward rollout(s), yielding a best-of-group update.

4. Practical PPO optimization

Sections 3.2–3.4 established an exact finite-M transform for binary rewards, an exact large-group objective for bounded scalar rewards, and the obstruction to a universal finite-group scalar objective beyond the binary case. None of these results accounts for the off-policy corrections required when optimizing over multiple gradient steps on a fixed rollout batch. This section describes the PPO approximation used in all experiments.

In practice, rollout-level rewards are treated as stop-gradient quantities: we backpropagate through token log-probabilities, but not through reward computation or sampling decisions. We compute rollout-level SoftmaxGRPO advantages once on a batch sampled from $\pi_{\theta_{\mathrm{old}}}$ and optimize a PPO-style clipped surrogate. For rollout $z_i = (z_{i,1}, \ldots, z_{i,T_i})$, define

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(z_{i,t} \mid x, z_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(z_{i,t} \mid x, z_{i,<t})}.$$

We minimize

$$\begin{aligned}\mathcal{L}^{\mathrm{clip}}_{\mathrm{SoftmaxGRPO}}(\theta) = -\mathbb{E}_{\substack{x\sim\mathcal{D} \\ \mathcal{G}(x)\sim\pi_{\theta_{\mathrm{old}}}}}\Bigg[\\ \frac{1}{\sum_{i=1}^M T_i}\sum_{i=1}^M\sum_{t=1}^{T_i}\min\Big(\rho_{i,t}(\theta)A_i,\ \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)A_i\Big)\Bigg],\end{aligned} \tag{10}$$

Method	Benchmarks		
	GSM8K	Countdown	DeepMath
Base	23.0	2.0	30.0
SFT	68.3	40.7	35.7
Rationalization	65.2	12.5	34.5
Iterative DPO	73.1	40.4	33.0
RL-Logit	71.2	2.2	37.7
RARO	–	54.4	41.3
OPD	**76.0**	3.4	42.2
GRPO-Sim	64.0	45.1	38.5
GRPO-Exact	73.5	57.7	50.9
SoftmaxGRPO-Sim (Ours)	71.0	48.4	39.7
SoftmaxGRPO-Exact (Ours)	75.8	**58.1**	**51.8**

Table 2: Verifiable reasoning accuracy (%). "Sim" methods use the same weak similarity reward; "Exact" methods use the task verifier. Top scores per column are in bold face.

together with a reference-model KL penalty,

$$\mathcal{L}_{\mathrm{SoftmaxGRPO}}(\theta) = \mathcal{L}^{\mathrm{clip}}_{\mathrm{SoftmaxGRPO}}(\theta) + \beta\, \mathbb{E}_{\substack{x\sim\mathcal{D} \\ \mathcal{G}(x)\sim\pi_{\theta_{\mathrm{old}}}}} \left[\frac{1}{\sum_{i=1}^{M} T_i} \sum_{i=1}^{M} \sum_{t=1}^{T_i} \mathrm{KL}\Big(\pi_\theta(\cdot \mid x, z_{i,<t}) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x, z_{i,<t})\Big) \right]. \tag{11}$$

This PPO layer should be viewed as a trust-region approximation to the unclipped on-policy objective, not as part of the exact theorem. At $\theta = \theta_{\mathrm{old}}$, clipping is inactive to first order and the leading update direction matches the SoftmaxGRPO estimator; away from that local regime, clipping and reference KL trade objective fidelity for stability.

Temperature and stability. The same temperature τ controls both objective geometry and optimizer sharpness. Lower τ moves the population objective toward MaxRL/ML-like weighting, but it also concentrates the within-group softmax on the highest-reward samples. In practice, smaller τ therefore requires a tighter trust region—typically a larger reference-KL coefficient, a smaller clip range, or fewer inner-loop updates. We treat this as an optimization issue rather than a change to the underlying population objective.

Scope of claims. The claims have three distinct scopes. **Finite-group theorem:** for binary rewards under on-policy unclipped optimization, SoftmaxGRPO induces $h_{M,\tau}(p)$ with derivative $\omega_{M,\tau}(p)$. **General-reward theorem and limitation:** for bounded scalar rewards, Eq. (7) is exact as $M \to \infty$, while Eq. (8) rules out a universal finite-M scalar objective without additional reward assumptions. **Implementation:** Eqs. (10)–(11) define the PPO-clipped, reference-KL-regularized approximation used in experiments. Clipping is inactive to first order at $\theta = \theta_{\mathrm{old}}$; away from that local regime, clipping and reference KL trade objective fidelity for stability.

5. Experiment Settings

5.1. Tasks & Datasets

We evaluate SoftmaxGRPO across eight benchmarks spanning verifiable and non-verifiable reasoning; all tasks use a unified *think-then-answer* format with official splits where available (full details in Appendix B.3). On the *verifiable* side, **GSM8K** [5] tests multi-step arithmetic on grade-school word problems with exact-match accuracy;

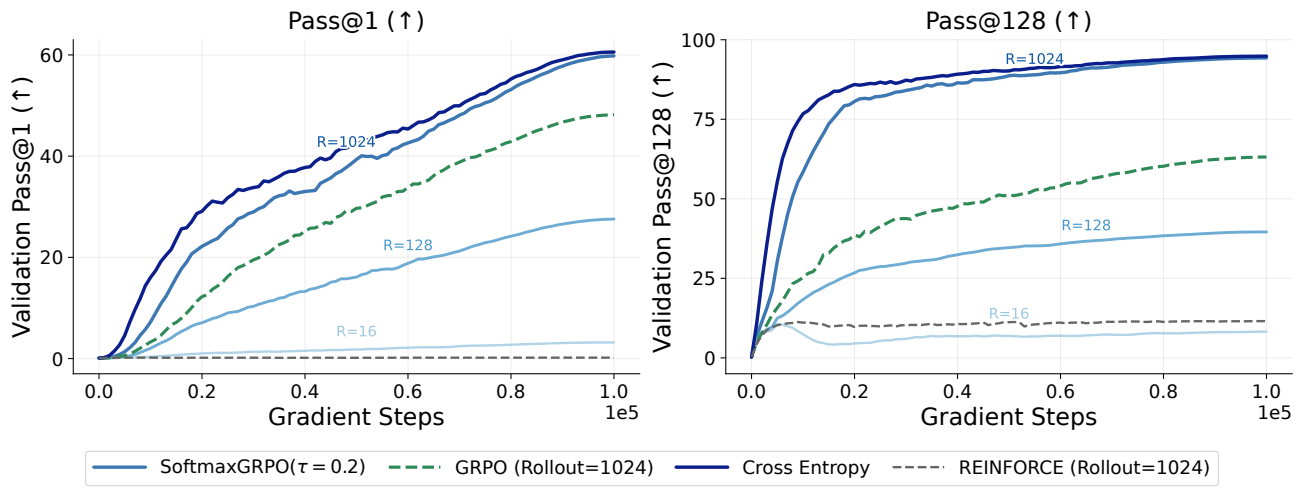


Figure 2: Comparison of training dynamics in ImageNet classification under exact maximum likelihood, REINFORCE, GRPO, and SoftmaxGRPO. With sufficient rollouts, SoftmaxGRPO closely matches cross-entropy training, while REINFORCE fails to make progress from low initial success rates even at large rollout counts.

Countdown is a controlled combinatorial task requiring four integers to be combined into a target value via basic arithmetic, with correctness checked by deterministic expression evaluation; and **DeepMath** [16] covers general math reasoning, where answer verification is itself nontrivial. For *non-verifiable* tasks, **Poetry Writing** is a custom dataset pairing creative prompts with expert reference poems, evaluated by an LLM judge [42]; and **MeetingBank** [18] is a long-context summarization benchmark of city council meeting transcripts, assessed via LLM-as-a-judge. We additionally report transfer to three standard capability benchmarks from a separate **OpenThoughts3-1.2M** [11] training run: **AlpacaEval 2.0** [10] (instruction following, length-controlled win rate), **MMLU** [17] (broad academic knowledge), and **GPQA** [30] (graduate-level science reasoning).

5.2. Comparisons with Prior Work

We compare SoftmaxGRPO against several post-training methods under identical dataset, training, and evaluation conditions. Full implementation details are in Appendix B.8. **SFT** maximizes log-likelihood on expert answers directly. **Rationalization** [44] augments each expert answer with a model-generated chain-of-thought rationale before fine-tuning. **Iterative DPO** [26, 29] runs 3 rounds of preference optimization from the SFT checkpoint, using on-policy samples as negatives. **RL-Logit** [12, 46] trains with rewards derived from the model's own logits on the expert answer (see Appendix B.7). **GRPO-Exact** [33] applies GRPO with ground-truth binary verifier rewards on all three verifiable tasks. **GRPO-Sim** applies the same GRPO objective with the similarity-based reward r_{sim} (Appendix B.5) in place of a ground-truth verifier, while using standard group normalization; it therefore tests GRPO under weak rewards on both verifiable and non-verifiable tasks. Full details are in Appendix B.8. **OPD** [2, 23] distills from a `Qwen3-8B` [42] teacher via per-token reverse-KL along on-policy trajectories. **RARO** [3] trains a relativistic reasoning critic that provides shaped rewards for joint policy-critic optimization via GRPO.

Training setup. Unless otherwise noted, the main experiments fine-tune `Qwen2.5-1.5B` [28] using AdamW [22] with a learning rate of 1×10^{-6} and bfloat16 precision, running on NVIDIA H200 and A100 GPUs via the VeRL framework [34]. The main GSM8K and Countdown results use $M{=}8$ and $\tau{=}0.1$ for SoftmaxGRPO; DeepMath uses $M{=}16$ and $\tau{=}0.3$, and non-verifiable tasks use $M{=}8$ and $\tau{=}0.3$. Within every objective-isolation comparison, methods share the model, data, reward, rollout group size, PPO clip, KL coefficient, learning rate, and training budget; only the group-advantage computation differs. SoftmaxGRPO uses the task's native ground-truth verifier as the reward signal for verifiable tasks (GSM8K, Countdown, DeepMath) and the lightweight similarity-based reward r_{sim} (Appendix B.5) for non-verifiable tasks. Full training details and per-task reward formulas are in Appendix B.1.

Task	Method	$[0, 0.2)$	$[0.2, 0.5)$	$[0.5, 0.7)$	$[0.7, 0.9)$	$[0.9, 1]$
GSM8K	GRPO	4.7%	4.3%	40.1%	14.5%	36.4%
	SoftmaxGRPO	7.3%	33.9%	11.9%	36.9%	10.0%
Countdown	GRPO	12.2%	20.2%	39.6%	18.4%	9.6%
	SoftmaxGRPO	16.0%	30.7%	33.1%	15.1%	5.1%

Table 3: Gradient allocation across prompt difficulty. Fraction (%) of total token-level gradient budget assigned by each method to prompts binned by measured pass rate p; each row sums to 100%. SoftmaxGRPO consistently reduces allocation to near-solved prompts ($p \geq 0.9$) and shifts budget toward lower-pass-rate examples, most sharply on GSM8K (10.0% vs. 36.4% for GRPO in the near-solved bin).

6. Main Results

We organize our main results by reward type: verifiable tasks with programmatic correctness checks (§6.2), and non-verifiable tasks that require LLM-based or heuristic evaluation (§6.4).

6.1. ImageNet classification

ImageNet classification provides a controlled test of how closely SoftmaxGRPO approximates *exact* maximum likelihood in a setting where the latter is available in closed form as the standard cross-entropy objective. We therefore compare four training objectives on ImageNet [7] using a ResNet-50 [14]: REINFORCE with a standard baseline, GRPO, SoftmaxGRPO, and exact maximum likelihood. For the RL-style objectives, each rollout samples a class prediction and receives reward 1 if the predicted class matches the ground-truth label and 0 otherwise. Full experimental details are provided in the Appendix.

Figure 2 shows a clear gap between expected-reward optimization and maximum-likelihood-style training. REINFORCE fails to make meaningful progress even at large per-example rollout budgets, reflecting the difficulty of learning from sparse binary feedback when initial success rates are low. Exact maximum likelihood, by contrast, exhibits the expected cross-entropy training dynamics. SoftmaxGRPO is trained on the same sampled rollouts and observes the same sparse set of successful trajectories as REINFORCE, but converts this limited signal into a substantially stronger update: as the rollout count increases, it improves steadily and closely tracks exact maximum likelihood. GRPO improves over REINFORCE, but remains visibly farther from the exact maximum-likelihood baseline.

6.2. Verifiable tasks

Table 2 reports both verifier-based training and a direct objective-isolation comparison under the same weak similarity reward. SoftmaxGRPO-Sim improves over GRPO-Sim on all three tasks: $+7.0$ points on GSM8K, $+3.3$ on Countdown, and $+1.2$ on DeepMath. It also exceeds demonstration-based SFT on GSM8K and Countdown despite using only weak output-overlap rewards. With exact verifier rewards, SoftmaxGRPO-Exact reaches 75.8% on GSM8K, 58.1% on Countdown, and 51.8% on DeepMath. It exceeds GRPO-Exact on all three tasks and attains the best result on Countdown and DeepMath; on GSM8K, it is competitive with OPD (75.8 vs. 76.0), which uses dense per-token distillation from a stronger teacher. Appendix B.2 confirms the advantage at 3B scale.

6.3. Gradient allocation by prompt difficulty

The divergence of GRPO's population weight as $p \to 1$ does not by itself establish wasted computation, because ∇p can simultaneously vanish. Table 3 measures the realized gradient allocation directly. On GSM8K, GRPO spends 36.4% of its gradient budget on near-solved prompts with $p \geq 0.9$, compared with 10.0% for SoftmaxGRPO. Conversely, SoftmaxGRPO allocates 82.7% to the moderate-difficulty range $p \in [0.2, 0.9)$, compared with 58.9% for

Method	Benchmarks				
	Poetry Score	**MeetingBank Summ.**	**AlpacaEval 2.0**	**MMLU**	**GPQA**
Base	35.0	35	1.61	60.9	24.2
SFT	53.7	55	2.18	61.4	25.6
GRPO-Sim	54.6	62	2.24	62.2	23.8
OPD	42.6	42	2.41	64.1	25.3
SoftmaxGRPO (Ours)	**68.0**	**70**	**2.50**	**65.2**	**27.1**

Table 4: Non-verifiable task results. We report **Poetry** score, **MeetingBank** summarization score (%), **AlpacaEval 2.0** length-controlled win rate, **MMLU**, and **GPQA** accuracy. Top scores per column are in bold face.

	$M = 4$		$M = 8$	
τ	Pass@1	Len	Pass@1	Len
0.1	75.4%	102.5	**75.8%**	100.2
0.3	74.6%	101.6	75.7%	102.8
0.5	74.4%	102.9	75.0%	104.0
1.0	46.1%	75.2	61.1%	92.7
1.3	53.5%	99.6	63.4%	96.8
1.5	41.8%	72.8	50.2%	82.7
10.0	31.7%	74.0	52.5%	86.8

(a) GSM8K. Pass@1 (%) and mean answer length (tokens) across τ and M. Performance is stable for $\tau \leq 0.5$ and degrades sharply for $\tau \geq 1.0$, as the reward weighting flattens.

	$M = 4$		$M = 8$	
τ	Pass@1	Len	Pass@1	Len
0.1	54.8%	74.0	**58.1%**	57.8
0.3	**57.8%**	60.3	55.2%	134.3
0.5	29.8%	134.0	55.8%	21.7
1.0	50.0%	59.6	54.6%	22.8
1.3	53.8%	70.3	55.2%	68.7
1.5	51.6%	64.6	54.6%	63.1
10.0	46.0%	71.8	45.2%	57.7

(b) Countdown. Pass@1 (%) and mean response length (tokens) across τ and M. Performance peaks at $\tau \leq 0.3$ and degrades at higher τ, with some settings showing anomalous lengths.

Table 5: Temperature (τ) and rollout group size (M) ablations on GSM8K (left) and Countdown (right). Both tasks favor low τ; performance degrades as τ increases and the reward weighting flattens toward a uniform average.

GRPO. Countdown has fewer near-solved prompts, but shows the same shift: SoftmaxGRPO assigns more budget to $p < 0.5$ (46.7% vs. 32.4%) and less to $p \geq 0.9$ (5.1% vs. 9.6%). These measurements support the predicted reallocation away from already-easy prompts and toward examples with greater room to improve.

6.4. Non-verifiable tasks

Table 4 reports results on five non-verifiable benchmarks spanning creative generation (Poetry), long-context summarization (MeetingBank), instruction following (AlpacaEval 2.0), and general knowledge (MMLU, GPQA). SoftmaxGRPO achieves the best performance across all five tasks, demonstrating that reward-augmented distillation transfers effectively beyond the verifiable regime. The largest gains appear on creative and generative tasks: on Poetry, SoftmaxGRPO scores 68.0, a +13.4 improvement over the next-best baseline (GRPO-Sim, 54.6), and on MeetingBank summarization it reaches 70% vs. 62% for GRPO-Sim, confirming that the method scales to longer-form generation where programmatic verification is unavailable. SoftmaxGRPO also leads on AlpacaEval 2.0 (2.50 vs. 2.41 for OPD), MMLU (65.2 vs. 64.1), and GPQA (27.1 vs. 25.6), indicating gains on general capabilities alongside task-specific improvements. Cross-judge and blind human calibration are reported in Appendix B.4.

6.5. Ablations

We ablate two key hyperparameters: the temperature τ, which controls how sharply rewards are converted into softmax weights $w_i \propto \exp(R_i/\tau)$ (Eq. (1)); and the rollout group size M, which determines how many rollouts are sampled per prompt to compute the group advantage. Table 5 reports a sweep over $\tau \in \{0.1, 0.3, 0.5, 1.0, 1.3, 1.5, 10.0\}$ and $M \in \{4, 8\}$ on both GSM8K and Countdown.

GSM8K. Performance is robust to both τ and M at low temperatures: Pass@1 stays within a tight band (74.4–75.8%) for $\tau \leq 0.5$ across both rollout group sizes, with $M{=}8$ providing a marginal improvement over $M{=}4$ (75.8% vs. 75.4% at $\tau{=}0.1$). Accuracy degrades sharply once $\tau \geq 1.0$—dropping as low as 31.7% at $\tau{=}10$—as the soft-max weighting flattens toward a uniform average and the per-step training signal weakens. Answer lengths remain stable ($\approx$100 tokens) throughout the low-τ regime, with no collapsed runs observed.

Countdown. Countdown is more sensitive to τ, with its best results at $\tau{=}0.1$ for $M{=}8$ (58.1%) and $\tau{=}0.3$ for $M{=}4$ (57.8%). Several (τ, M) configurations exhibit inflated (>100 tokens) or collapsed (<30 tokens) responses. Sharper softmax weights concentrate the update on fewer rollouts and raise variance, whereas flatter weights weaken the advantage signal; either effect can interact with the PPO trust region and the task's short output format to produce length drift.

Both tasks favor low temperatures, with $\tau \leq 0.3$ providing the best observed accuracy–stability trade-off. This accords with Section 3.2: low τ differentiates high-reward rollouts, whereas larger τ approaches a weak centered-REINFORCE signal. Because concentration also raises variance, τ should be tuned jointly with group size, PPO clipping, and reference KL. For rewards normalized to $[0, 1]$, $\tau \in [0.1, 0.3]$ is a reliable starting range rather than a scale-free default.

7. Discussion and Conclusion

We introduced SoftmaxGRPO, a drop-in replacement for GRPO that uses temperature-scaled softmax group advantages. Under binary rewards in the on-policy unclipped regime, SoftmaxGRPO admits an exact finite-group population objective with bounded prompt weighting and recovers MaxRL as $\tau \downarrow 0$ for finite M. For bounded scalar rewards, its large-group update exactly optimizes a log-moment-generating-function objective; at finite M, however, the update is generally non-conservative once rewards have three or more levels. This separates the contribution from the established use of exponentiated-reward weights in RAML, softmax policy gradient, and MPO: the new result is the finite-group prompt geometry, its limits, and the boundary of where a scalar objective exists.

The experiments support that geometry directly. Under identical weak rewards, SoftmaxGRPO-Sim improves over GRPO-Sim on all three verifiable tasks, while the gradient-allocation measurement shows that GRPO assigns 36.4% of its GSM8K gradient budget to prompts with $p \geq 0.9$, versus 10.0% for SoftmaxGRPO. Across the broader evaluation, SoftmaxGRPO reaches 51.8% on DeepMath with verifier rewards and improves Poetry from 35.0% to 68.0% using only lightweight similarity rewards. The scope remains important: the finite-M theorem is exact only for binary rewards under on-policy unclipped optimization, the scalar-reward theorem is asymptotic in group size, the practical method uses PPO clipping and reference KL, the main evaluations focus on a 1.5B model, and non-verifiable evaluation depends on imperfect overlap rewards and LLM judges. Broader scale validation, adaptive temperature selection, and richer process-level rewards remain useful directions.

Acknowledgments. We would like to thank the Ken Kennedy Institute, NSF Career Award #2201710 and the NSF Campus Cyberinfrastructure grant "CC* Compute: Interactive Data Analysis Platform" NSF OAC-2019007, and Rice University's Center for Research Computing (CRC) for their support.

References

[1] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

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

[3] Locke Cai and Ivan Provilkov. Escaping the verifier: Learning to reason via demonstrations. *arXiv preprint arXiv:2511.21667*, 2025.

[4] Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

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

[6] Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: Complex vision-language reasoning via iterative sft-rl cycles. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 123817–123846. Curran Associates, Inc., 2025.

[9] Nan Ding and Radu Soricut. Cold-start reinforcement learning with softmax policy gradient. *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[11] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

[12] Alexander Gurung and Mirella Lapata. Learning to reason for long-form story generation. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=dr3eg5ehR2.

[13] Ali Hatamizadeh, Syeda Nahida Akter, Shrimai Prabhumoye, Jan Kautz, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. RLP: Reinforcement as a pretraining objective. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9Gp45bnDrJ.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Ruozhen He, Nisarg A. Shah, Qihua Dong, Zilin Xiao, Jaywon Koo, and Vicente Ordonez. Beyond referring expressions: Scenario comprehension visual grounding, 2026. URL https://arxiv.org/abs/2604.02323.

[16] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kHB5Te5IWm.

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

[18] Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. MeetingBank: A benchmark dataset for meeting summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.906. URL https://aclanthology.org/2023.acl-long.906/.

[19] Minbin Huang, Runhui Huang, Chuanyang Zheng, Jingyao Li, Guoxuan Chen, Han Shi, and Hong Cheng. Answer-consistent chain-of-thought reinforcement learning for multi-modal large langauge models. *arXiv preprint arXiv:2510.10104*, 2025.

[20] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[21] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

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

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

[24] Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025.

[25] Mohammad Norouzi, Samy Bengio, Zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. Reward augmented maximum likelihood for neural structured prediction. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[26] Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason E Weston. Iterative reasoning preference optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=4XIKfvNYvx.

[27] Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*, 2026.

[28] A Yang Qwen, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint*, 2024.

[29] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

[30] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

[31] Sara Sabour, William Chan, and Mohammad Norouzi. Optimal completion distillation for sequence learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkMW1hRqKX.

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

[33] 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.

[34] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[35] Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.

[36] Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

[37] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 30865–30891. Curran Associates, Inc., 2025.

[38] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 118818–118850. Curran Associates, Inc., 2025.

[39] Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Thoughts are all over the place: On the underthinking of long reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=WcUo7Z2Jnh.

[40] Haotian Xia, Haonan Ge, Junbo Zou, Hyun Woo Choi, Xuebin Zhang, Danny Suradja, Botao Rui, Ethan Tran, Wendy Jin, Zhen Ye, Xiyang Lin, Christopher Lai, Shengjie Zhang, Junwen Miao, Shichao Chen, Rhys Tracy, Vicente Ordonez, Weining Shen, and Hanjie Chen. Sportr: A benchmark for multimodal large language model reasoning in sports, 2026. URL https://arxiv.org/abs/2511.06499.

[41] Zilin Xiao, Jaywon Koo, Siru Ouyang, Jefferson Hernandez, Yu Meng, and Vicente Ordonez. Proxythinker: Test-time guidance through small visual reasoners. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

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

[43] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

[44] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_3ELRdg2sgI.

[45] Zizheng Zhan, Ken Deng, Huaixi Tang, Wen Xiang, Kun Wu, Weihao Li, Wenqiang Zhu, Jingxuan Xu, Lecheng Huang, Zongxian Feng, et al. Kat-v1: Kwai-autothink technical report. *arXiv preprint arXiv:2507.08297*, 2025.

[46] Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nnwvwge40d.

A. Exact SoftmaxGRPO Derivation

This appendix proves the exact finite-M binary-reward statement used in Eq. (4) and gives the promised closed-form transform $h_{M,\tau}$. As in the main text, gradients act only on the log-likelihood terms in Eq. (2); rewards and sampled rollouts are treated as stop-gradient quantities. We fix a prompt x and work in the unclipped on-policy regime.

Let

$$p := p_\theta(x) = \Pr_{z\sim\pi_\theta(\cdot|x)}\big[\mathrm{Dec}(z) \in C(x)\big], \qquad R_i := \mathbf{1}\big\{\mathrm{Dec}(z_i) \in C(x)\big\},$$

with

$$z_1, \dots, z_M \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x), \qquad S_i := \sum_{j \neq i} R_j, \qquad c := e^{1/\tau}.$$

We will use the identities

$$\mathbb{E}[R_i \nabla_\theta \log \pi_\theta(z_i \mid x)] = \nabla_\theta p_\theta(x), \qquad \mathbb{E}[(1 - R_i) \nabla_\theta \log \pi_\theta(z_i \mid x)] = -\nabla_\theta p_\theta(x), \tag{12}$$

where the second follows from $\mathbb{E}[\nabla_\theta \log \pi_\theta(z_i \mid x)] = 0$.

Conditioned on $S_i = s$, SoftmaxGRPO assigns

$$w_i = \begin{cases} w_s^{(0)} = \dfrac{1}{M + s(c-1)}, & R_i = 0, \\ w_s^{(1)} = \dfrac{c}{M + (s+1)(c-1)}, & R_i = 1. \end{cases}$$

Hence the centered advantage $A_i = M w_i - 1$ has the Davis–Recht conditional-linear form [6]

$$A_i = (1 - R_i) a_{S_i} + R_i b_{S_i}, \qquad a_s = \frac{M}{M + s(c-1)} - 1, \qquad b_s = \frac{Mc}{M + (s+1)(c-1)} - 1. \tag{13}$$

Their gap is

$$\begin{aligned} \Delta_s^{(\tau)} := b_s - a_s &= M \left(\frac{c}{M + (s+1)(c-1)} - \frac{1}{M + s(c-1)} \right) \\ &= \frac{M(c-1)\big(M - 1 + s(c-1)\big)}{\big(M + s(c-1)\big)\big(M + (s+1)(c-1)\big)}. \end{aligned} \tag{14}$$

In particular, $\Delta_s^{(\tau)} > 0$ for every s, so SoftmaxGRPO always upweights correct rollouts relative to incorrect ones.

Proof of Eq. (4). By exchangeability of the M rollouts,

$$\mathbb{E}[\nabla_\theta \mathcal{J}^{\text{uc}}_{\text{SoftmaxGRPO}}(\theta; x, \mathcal{G})] = \mathbb{E}[A_1 \nabla_\theta \log \pi_\theta(z_1 \mid x)] \,.$$

Condition on $S_1 = s$. Since S_1 depends only on $\{z_j\}_{j \neq 1}$, it is independent of z_1. Using Eq. (13) and the score identities in Eq. (12),

$$\begin{aligned} \mathbb{E}[A_1 \nabla_\theta \log \pi_\theta(z_1 \mid x) \mid S_1 = s] &= a_s \, \mathbb{E}[(1 - R_1) \nabla_\theta \log \pi_\theta(z_1 \mid x)] + b_s \, \mathbb{E}[R_1 \nabla_\theta \log \pi_\theta(z_1 \mid x)] \\ &= (b_s - a_s) \, \nabla_\theta p_\theta(x) = \Delta_s^{(\tau)} \, \nabla_\theta p_\theta(x). \end{aligned}$$

Averaging over $S_1 \sim \text{Binomial}(M-1, p)$ gives

$$\mathbb{E}[\nabla_\theta \mathcal{J}^{\text{uc}}_{\text{SoftmaxGRPO}}(\theta; x, \mathcal{G})] = \mathbb{E}_{S \sim \text{Binomial}(M-1,p)} \left[\Delta_S^{(\tau)} \right] \nabla_\theta p_\theta(x),$$

which is Eq. (4).

Closed form for $h_{M,\tau}$ and Bernstein form for $\omega_{M,\tau}$. Expanding the binomial expectation yields

$$\omega_{M,\tau}(p) = \sum_{s=0}^{M-1} \Delta_s^{(\tau)} \binom{M-1}{s} p^s (1-p)^{M-1-s}. \tag{15}$$

Thus $\omega_{M,\tau}$ is a Bernstein polynomial of degree $M-1$ with coefficients $\Delta_s^{(\tau)}$. Since

$$\frac{d}{dp} I_p(s+1, M-s) = \frac{p^s (1-p)^{M-1-s}}{B(s+1, M-s)} = M \binom{M-1}{s} p^s (1-p)^{M-1-s},$$

where $I_p(\cdot,\cdot)$ is the regularized incomplete beta function and $B(\cdot,\cdot)$ is the beta function, the choice

$$h_{M,\tau}(p) = \frac{1}{M}\sum_{s=0}^{M-1} \Delta_s^{(\tau)} I_p(s+1, M-s), \qquad h_{M,\tau}(0) = 0, \tag{16}$$

satisfies

$$h'_{M,\tau}(p) = \omega_{M,\tau}(p).$$

This is the exact finite-M scalar transform induced by SoftmaxGRPO under binary rewards. The main-text expression $\omega_{M,\tau}(p) = \mathbb{E}_{S\sim\mathrm{Binomial}(M-1,p)}[\Delta_S^{(\tau)}]$ is simply the binomial-expectation form of Eq. (15).

Because $\omega_{M,\tau}$ is a finite Bernstein polynomial, it is continuous and bounded on $[0,1]$ for every finite (M,τ). In particular,

$$\omega_{M,\tau}(1) = \Delta_{M-1}^{(\tau)} = 1 - \frac{M}{1+(M-1)e^{1/\tau}} < \infty, \tag{17}$$

so SoftmaxGRPO has no easy-prompt singularity at finite group size.

Low-temperature finite-M limit. Let $\tau \downarrow 0$, so $c = e^{1/\tau} \to \infty$. Then

$$\lim_{\tau\downarrow 0} \Delta_s^{(\tau)} = \begin{cases} M-1, & s = 0, \\ \dfrac{M}{s+1}, & s = 1,\dots,M-1. \end{cases}$$

Substituting into Eq. (15) gives

$$\omega_{M,0}(p) = (M-1)(1-p)^{M-1} + \sum_{s=1}^{M-1} \frac{M}{s+1}\binom{M-1}{s} p^s (1-p)^{M-1-s}.$$

Using

$$\frac{M}{s+1}\binom{M-1}{s} = \binom{M}{s+1},$$

we obtain

$$\begin{aligned}
\omega_{M,0}(p) &= \sum_{s=0}^{M-1} \binom{M}{s+1} p^s (1-p)^{M-1-s} - (1-p)^{M-1} \\
&= \frac{1}{p}\sum_{k=1}^{M} \binom{M}{k} p^k (1-p)^{M-k} - (1-p)^{M-1} \\
&= \frac{1-(1-p)^M}{p} - (1-p)^{M-1} \\
&= \frac{1-(1-p)^{M-1}}{p},
\end{aligned}$$

which proves Eq. (5). This is exactly the MaxRL weighting with truncation $T = M-1$ [36].

Large-M mean-field form. The mean-field form in Eq. (3) is recovered by substituting the mean of $S \sim \mathrm{Binomial}(M-1,p)$ into Eq. (14). Setting $s = (M-1)p$ gives

$$\begin{aligned}
M - 1 + s(c-1)\big|_{s=(M-1)p} &= (M-1)\big(1+p(c-1)\big), \\
M + s(c-1)\big|_{s=(M-1)p} &= M + (M-1)p(c-1).
\end{aligned}$$

As $M \to \infty$, both $M + (M-1)p(c-1)$ and $M + (M-1)p(c-1) + (c-1)$ are $\approx M(1+p(c-1))$, so

$$\omega_\tau(p) \approx \Delta_{(M-1)p}^{(\tau)} = \frac{M(c-1)(M-1)\big(1+p(c-1)\big)}{\big[M\big(1+p(c-1)\big)\big]^2} \approx \frac{c-1}{1+p(c-1)} = \frac{e^{1/\tau}-1}{1-p+pe^{1/\tau}},$$

which is Eq. (3). Integrating with respect to p then gives $h_\tau(p) \propto \log(1 - p + pe^{1/\tau})$, up to an additive constant and a positive p-independent scaling.

High-temperature limit. Let $\tau \to \infty$, so $c = e^{1/\tau} = 1 + 1/\tau + O(\tau^{-2})$. Writing $\epsilon := c - 1 \to 0$ and expanding Eq. (14):

$$\Delta_s^{(\tau)} = \frac{M\epsilon\big(M - 1 + s\epsilon\big)}{(M + s\epsilon)(M + (s+1)\epsilon)} = \frac{(M-1)\epsilon}{M} + O(\epsilon^2).$$

The leading term is s-independent, so the binomial weights in Eq. (15) sum to 1 and give

$$\omega_{M,\tau}(p) = \frac{M-1}{M\tau} + O(\tau^{-2}),$$

a p-independent constant. To connect this to REINFORCE directly, expand a_s and b_s to first order in ϵ:

$$a_s \approx -\frac{s\epsilon}{M}, \qquad b_s \approx \frac{(M-s-1)\epsilon}{M}.$$

When $R_i = 0$ the group has S_i successes total, so $\bar{R} = S_i/M$ and $A_i = a_{S_i} \approx \epsilon(R_i - \bar{R})$. When $R_i = 1$ the group has $S_i + 1$ successes total, so $\bar{R} = (S_i + 1)/M$ and $A_i = b_{S_i} \approx \epsilon(R_i - \bar{R})$. In both cases,

$$A_i \;\approx\; \frac{R_i - \bar{R}}{\tau} + O(\tau^{-2}), \tag{18}$$

so SoftmaxGRPO recovers centered group-reward (REINFORCE-style) weighting, scaled by $1/\tau$. The p-independent leading weight $\omega_{M,\tau}(p) \approx (M-1)/(M\tau)$ is consistent with Eq. (18): since $\mathbb{E}[R_i - \bar{R}]^2 = p(1-p) \cdot \frac{M-1}{M}$ for binary rewards, every prompt difficulty p receives the same $O(\tau^{-1})$ learning signal.

A.1. General Scalar Rewards

We give the full large-group argument behind Eq. (7). Fix a prompt x, write $g_\theta(z) = \nabla_\theta \log \pi_\theta(z \mid x)$, and let $X(z) = \exp(R(x,z)/\tau)$. Assume that R is a bounded, measurable, stop-gradient reward and that the policy score has a finite $(1+\eta)$-moment for some $\eta > 0$. The unclipped group update can be written as

$$G_M = \frac{\frac{1}{M}\sum_{i=1}^M X_i g_\theta(z_i)}{\frac{1}{M}\sum_{j=1}^M X_j} - \frac{1}{M}\sum_{i=1}^M g_\theta(z_i). \tag{19}$$

The law of large numbers, boundedness of X, and the score identity give

$$\begin{aligned} \lim_{M\to\infty} \mathbb{E}[G_M] &= \frac{\mathbb{E}_{z\sim\pi_\theta}[X(z) g_\theta(z)]}{\mathbb{E}_{z\sim\pi_\theta}[X(z)]} - \mathbb{E}_{z\sim\pi_\theta}[g_\theta(z)] \\ &= \frac{\nabla_\theta Z_\tau(\theta; x)}{Z_\tau(\theta; x)} = \nabla_\theta \log Z_\tau(\theta; x), \end{aligned} \tag{20}$$

where $Z_\tau(\theta; x) = \mathbb{E}_{\pi_\theta}[e^{R/\tau}]$. Thus, for bounded scalar rewards, the large-M population update has an exact scalar objective: the log moment-generating function of reward. The boundedness assumption can be replaced by the corresponding exponential- moment and uniform-integrability conditions.

Why a universal finite-M objective does not exist. For completeness, we expand the counterexample summarized in Eq. (8). Let $M = 2$ and let the reward take three values with probabilities (p_1, p_2, p_3) and exponentiated values $t_k = e^{r_k/\tau}$. Conditional on the first rollout having level k, its expected nonconstant coefficient is

$$m_k(p) = 2\sum_{\ell=1}^3 p_\ell \frac{t_k}{t_k + t_\ell}.$$

The common centered baseline does not affect integrability. On the simplex $p_3 = 1 - p_1 - p_2$, the update therefore corresponds to the one-form $(m_1 - m_3)\,dp_1 + (m_2 - m_3)\,dp_2$. For $t = (1, 2, 4)$, direct substitution gives

$$m_1 = \frac{3}{5}p_1 + \frac{4}{15}p_2 + \frac{2}{5}, \quad m_2 = \frac{2}{3}p_1 + \frac{1}{3}p_2 + \frac{2}{3},$$
$$m_3 = \frac{3}{5}p_1 + \frac{1}{3}p_2 + 1,$$

and hence

$$m_1 - m_3 = -\frac{1}{15}p_2 - \frac{3}{5}, \qquad m_2 - m_3 = \frac{1}{15}p_1 - \frac{1}{3}.$$

The cross-partials are $-1/15$ and $1/15$, respectively, so the one-form is not closed. This rules out a universal finite-group scalar potential once three or more reward levels are allowed. Additional assumptions that reduce the reward geometry to a one-dimensional family can restore integrability; binary rewards are the canonical example.

Gaussian example and reward-scale sensitivity. Although Gaussian rewards are unbounded, their exponential moments exist. If $R_\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$, then

$$\log Z_\tau(\theta) = \frac{\mu_\theta}{\tau} + \frac{\sigma_\theta^2}{2\tau^2}. \tag{21}$$

Up to the positive global factor $1/\tau$, the induced objective is $\mu_\theta + \sigma_\theta^2/(2\tau)$: exponentiation adds a temperature-controlled variance bonus. If the variance comes from policy-independent, homoscedastic Gaussian reward noise, its gradient is zero and the large-M direction reduces to $\tau^{-1}\nabla_\theta \mu_\theta$; such noise therefore does not bias the population direction.

The same example makes the dependence on reward scale explicit. For unnormalized weights $X_i = e^{R_i/\tau}$, the large-group effective sample size satisfies

$$\frac{\mathrm{ESS}}{M} \longrightarrow \frac{(\mathbb{E}X)^2}{\mathbb{E}[X^2]} = \exp\left(-\frac{\sigma_\theta^2}{\tau^2}\right). \tag{22}$$

Consequently, τ must be calibrated to the reward standard deviation. Our practical range $\tau \in [0.1, 0.3]$ is intended for rewards normalized to $[0, 1]$ and should not be transferred unchanged to unbounded, unnormalized rewards.

A.2. Robustness to Noisy Binary Rewards

Let $Y_i \in \{0, 1\}$ denote the clean success indicator and suppose the observed reward is $R_i = Y_i + \varepsilon_i$, with $|\varepsilon_i| \le \delta$. If w_i^0 is the clean binary SoftmaxGRPO weight and $\widetilde{w}_i$ is the noisy weight, then

$$\widetilde{w}_i = \frac{w_i^0 e^{\varepsilon_i/\tau}}{\sum_j w_j^0 e^{\varepsilon_j/\tau}}. \tag{23}$$

Because $e^{\varepsilon_i/\tau} \in [e^{-\delta/\tau}, e^{\delta/\tau}]$, every group obeys the multiplicative envelope

$$e^{-2\delta/\tau} w_i^0 \le \widetilde{w}_i \le e^{2\delta/\tau} w_i^0. \tag{24}$$

Writing G_M^0 and $\widetilde{G}_M$ for the clean and noisy group updates, respectively, this also gives

$$\|\widetilde{G}_M - G_M^0\| \le \left(e^{2\delta/\tau} - 1\right) \sum_i w_i^0 \|g_\theta(z_i)\|.$$

Thus, for fixed M and integrable policy scores, the population update is perturbed by $O(\delta/\tau)$. If the noise variables are identically distributed and conditionally independent of the rollouts given the clean labels, symmetry preserves the one-dimensional binary form, and this statement can be written directly as

$$\widetilde{\omega}_{M,\tau}(p) = \omega_{M,\tau}(p) + O(\delta/\tau). \tag{25}$$

For i.i.d. zero-mean noise, a sharper expected statement follows from

$$\widetilde{w}_i = w_i^0 \left[1 + \frac{\varepsilon_i - \sum_j w_j^0 \varepsilon_j}{\tau} \right] + O(\delta^2/\tau^2).$$

The first-order term vanishes after conditioning on the clean group. For bounded noise with variance σ_ε^2 and δ/τ small, the expected perturbation is therefore $O(\sigma_\varepsilon^2/\tau^2)$, with higher-order terms controlled by δ/τ. The relevant sensitivity parameter is the noise-to-temperature ratio: small verifier noise preserves the binary geometry, whereas the bound becomes uninformative once the noise scale is comparable to τ.

A.3. Temperature τ as a KL Trust-Region Parameter

Section 3.4 showed that the SoftmaxGRPO softmax weights are the unique solution of

$$\max_{q \in \Delta^M} \left\{ \sum_i q_i r_i - \tau \, \mathrm{KL}(q \| u) \right\}, \qquad u_i = \tfrac{1}{M},$$

with $q_i^\star = w_i \propto e^{r_i/\tau}$. This variational representation gives τ a precise meaning: it is the Lagrange multiplier (trust-region budget) controlling how far the tilted within-group target deviates from the uniform on-policy empirical prior. Small τ permits large KL deviations—concentrating mass on high-reward rollouts—while large τ forces q to remain close to uniform, smoothing the advantage signal toward centered policy gradient. This perspective reframes τ not as an arbitrary temperature but as a within-group diversity constraint analogous to the KL regularizer in exponentiated-reward policy search [1].

Adaptive τ via effective sample size. For binary rewards, an explicit closed-form rule can target a desired within-group effective sample size (ESS). Suppose k of the M rollouts are correct. With $c = e^{1/\tau}$, the two distinct softmax weights are

$$w^+ = \frac{c}{kc + M - k}, \qquad w^- = \frac{1}{kc + M - k},$$

and the within-group ESS is

$$\mathrm{ESS}(k, c) = \frac{(kc + M - k)^2}{kc^2 + M - k}.$$

Setting $\mathrm{ESS} = \nu$ for a target $\nu \in (k, M]$ and solving for c yields the closed-form rule

$$c = \frac{k(M-k) + \sqrt{k(M-k)\,\nu(M-\nu)}}{k(\nu - k)}, \qquad \tau = \frac{1}{\log c}.$$

This rule admits an intuitive interpretation: targeting $\nu = M/2$, for example, ensures the update is never dominated by a single rollout regardless of k. The rule degenerates when $k = 0$ or $k = M$ (all rewards equal, no signal), but those cases require no adaptation. A full empirical evaluation of adaptive τ selection is left to future work.

B. Experimental Details

B.1. Training Hyperparameters

All experiments use the VeRL framework [34] with AdamW [22] optimisation in bfloat16 precision. Tables B.1 and B.2 summarise the per-task hyperparameters for the verifiable and non-verifiable training runs, respectively. Unless otherwise noted, all runs use the shared optimizer, precision, and learning-rate settings stated above.

	GSM8K	Countdown	DeepMath
Training Hyper-Parameters			
Base model		`Qwen/Qwen2.5-1.5B`	
Optimizer		AdamW	
Learning rate		1×10^{-6}	
SoftmaxGRPO temperature τ	0.1	0.1	0.3
PPO clip ε		$[0.20,\ 0.28]$	
KL regularization β		10^{-3}	
Rollout group size M	8	8	16
Rollout batch size	64	64	512
Mini-batch size (per device)	4	4	8
Total training iterations	1,000	2,000	3,220
Max response tokens	256	256	1,024
Hardware			
GPU device	8× A100	8× H200	16× H200
Compute setup	1 node × 8	1 node × 8	2 nodes × 8

Table B.1: Training hyperparameters for the main verifiable reasoning results. The SoftmaxGRPO temperature and rollout group size are task-specific; the optimizer, learning rate, PPO clip, and KL coefficient are shared.

	Poetry	MeetingBank	OpenThoughts
Training Hyper-Parameters			
Base model		`Qwen/Qwen2.5-1.5B`	
Optimizer		AdamW	
Learning rate		1×10^{-6}	
SoftmaxGRPO temperature τ		0.3	
PPO clip ε		$[0.20,\ 0.28]$	
KL regularization β		10^{-3}	
Rollout group size M	8	8	8
Rollout batch size	64	512	512
Mini-batch size (per device)	4	4	4
Total training iterations	1,350	3,220	3,220
Max response tokens	1,024	4,096	6,144
Hardware			
GPU device	8× A100	16× H200	16× H200
Compute setup	1 node × 8	2 nodes × 8	2 nodes × 8

Table B.2: Training hyperparameters for non-verifiable runs. The OpenThoughts3-1.2M checkpoint is used for transfer evaluation on AlpacaEval 2.0, MMLU, and GPQA. Rows above the second rule are shared across all three tasks.

Configuration selection and matched comparisons. The main GSM8K and Countdown configurations, $(M, \tau) = (8, 0.1)$, are the best-performing cells in the factorial sweep reported in Table 5. DeepMath uses $(M, \tau) = (16, 0.3)$, while all non-verifiable runs use the fixed default $(M, \tau) = (8, 0.3)$. We did not run the full sweep on those tasks. Temperature is specific to SoftmaxGRPO and has no GRPO counterpart. In every SoftmaxGRPO/GRPO objective-isolation comparison, we match the model, dataset, reward, rollout group size, PPO clip, KL coefficient, learning rate, rollout sampling, and training budget; only the group-advantage computation changes.

B.2. Model-Scale Evaluation

We additionally compare verifier-trained SoftmaxGRPO and GRPO at 3B parameters on GSM8K and Countdown. Within each model scale, the methods use matched training conditions and differ only in their group-advantage

computation.

Base model	Method	GSM8K	Countdown
Qwen2.5-1.5B	GRPO	73.5	57.7
	SoftmaxGRPO	**75.8**	**58.1**
Qwen2.5-3B	GRPO	80.2	50.9
	SoftmaxGRPO	**82.3**	**60.4**

Table B.3: Verifier-based accuracy (%) across model scales under matched conditions. Bold marks the better objective within each model size.

The SoftmaxGRPO advantage persists at 3B: it improves over GRPO by 2.1 points on GSM8K and 9.5 points on Countdown. The Countdown result should not be read as a monotonic scaling law from two model sizes. In particular, the GRPO regression from 1.5B to 3B is consistent with verifier-RL scaling behavior reported independently by Cai and Provilkov [3]; our result establishes that SoftmaxGRPO does not exhibit that regression in this matched comparison.

Reward formulas (verifiable tasks). For GSM8K and DeepMath we use the Math-Verify rule-based answer verifier[1]: reward $= 1$ if the model's final boxed answer matches the gold answer after standard normalisation, and 0 otherwise. Every prompt is appended with the instruction *"Please reason step by step, and put your final answer within \boxed{}."* For Countdown, correctness is checked by deterministic evaluation of the predicted arithmetic expression; no additional prompt instruction is added.

B.3. Task and Dataset Details

GSM8K. GSM8K [5] is a dataset of grade-school math word problems requiring multi-step arithmetic and careful tracking of intermediate quantities. Each example pairs a natural-language question with a short, unambiguous final answer. We report exact-match accuracy after standard answer normalization. Training demonstrations, when used, are drawn from the `HAD653/gsm8k-cot-120b` dataset,[2] which provides chain-of-thought solutions generated by the `gpt-oss-120b` model. During SoftmaxGRPO training, we use the verifiable reward function.

Countdown. We use a 24-style variant of Countdown where the goal is to combine four integers to obtain 24 using basic arithmetic operations and parentheses, with each integer used exactly once. Correctness is determined by deterministic execution of the predicted expression. This controlled setting isolates the role of exploration and credit assignment, since answer checking is far simpler than solution search. Training demonstrations, when used, are drawn from the verified split of the Countdown-Task-GOLD dataset.[3]. During SoftmaxGRPO training, we use the verifiable reward function.

DeepMath. DeepMath [16] covers general math reasoning problems where answer verification is itself nontrivial, often requiring solving the problem from scratch or handling nontrivial symbolic manipulation. Each DeepMath example already includes three solutions generated by DeepSeek-R1 and verified to be correct; we use the shortest of the three as the training demonstration. During SoftmaxGRPO training, we use the verifiable reward function.

Poetry Writing. Poetry Writing is a custom dataset of prompts (topics and optional stylistic constraints) paired with expert reference poems. Source poems are drawn from the `jnb666/poems` dataset.[4] Since that dataset contains poems only (without accompanying prompts), we use `gpt-5-instant` to generate a plausible instruction for each poem—specifically, asking the model what instruction would most naturally have produced the given poem—yielding (instruction, poem) training pairs. Overlap-based rewards only imperfectly capture quality. We

[1] https://github.com/huggingface/Math-Verify
[2] https://huggingface.co/datasets/HAD653/gsm8k-cot-120b
[3] https://huggingface.co/datasets/HuggingFaceTB/Countdown-Task-GOLD/viewer/verified_Qwen2.5-7B-Instruct
[4] https://huggingface.co/datasets/jnb666/poems

generate poems with the instruction *"You are a helpful assistant that writes poetry."* For evaluation we use `Qwen3-30B-A3B-Thinking-2507` [42] as a judge, scoring poems both in isolation and via pairwise comparison to the expert reference with the rubric prompt shown in Figure B.1.

MeetingBank Summarization. MeetingBank [18] is a long-context benchmark built from public city council meeting transcripts. Training transcripts and reference summaries are taken from the `microsoft/MeetingBank-LLMCompressed` dataset.[5] The goal is to generate concise, informative summaries aligned to professionally written minutes. Summarization quality is not programmatically verifiable and involves trade-offs between coverage, faithfulness, and concision. During training, we track ROUGE and BLEU only. After training, we evaluate the final checkpoint on the held-out test set with `Qwen3-30B-A3B-Thinking-2507` [42] using a MeetingBank-specific rubric prompt that scores informativeness, factuality, fluency, coherence, and conciseness against the transcript segment and associated meeting metadata, as shown in Figure B.2.

OpenThoughts3 transfer run. For the transfer results on AlpacaEval 2.0, MMLU, and GPQA, we train a separate non-verifiable SoftmaxGRPO checkpoint on OpenThoughts3-1.2M [11] and evaluate that checkpoint on the three downstream benchmarks without further task-specific fine-tuning.

AlpacaEval 2.0, MMLU, and GPQA. These three benchmarks assess generalization of capabilities acquired during non-verifiable training. AlpacaEval 2.0 [10] measures instruction-following quality via length-controlled pairwise win rate against GPT-4. MMLU [17] evaluates broad academic knowledge across 57 subjects. GPQA [30] probes graduate-level reasoning in science domains. No training data from these benchmarks is used; results reflect transfer from the separate OpenThoughts3-1.2M training run.

B.4. LLM-Judge Validation

Our primary evaluator is `Qwen3-30B-A3B-Thinking-2507` with the task-specific rubrics reproduced in Figures B.1 and B.2. We validate these evaluations in two ways. First, we re-score the same outputs with the architecturally distinct `gemma-4-31B-it` judge and compare method rankings. Second, one human rater blindly scores 30 Poetry and 30 MeetingBank outputs without access to the generating method's identity, and we compare those ratings with the primary judge.

Evaluation	Poetry	MeetingBank
Primary judge score	68.0	70.0
Second judge score	59.2	65.4
Cross-judge rank correlation	0.854	0.945
Human–judge correlation	0.74	0.89

Table B.4: LLM-judge validation. Scores use the paper's 0–100 reporting scale; rank correlation compares method-level orderings across the two judges, and human–judge correlation compares the blind human ratings with the primary judge on 30 outputs per task.

The second judge is systematically stricter in absolute score, but the two judges agree strongly on method ranking and preserve the SoftmaxGRPO-versus- baseline ordering on both tasks. Human agreement is higher for MeetingBank, where factual coverage provides a more objective anchor, than for Poetry, where quality is intrinsically more subjective. Because the calibration uses a single human rater, it measures agreement with that rater rather than inter-rater variability.

[5] https://huggingface.co/datasets/microsoft/MeetingBank-LLMCompressed

Poetry Evaluation Prompt

[Evaluation Role]
You are an expert poetry critic with deep training in New Criticism, Formalism, and Reader-Response theory. You evaluate poetry based on craft, execution, and internal logic, not personal taste or agreement with the subject matter.
[Goal]
Your goal is to provide a rigorous critical evaluation of the poem provided below.
[Inputs]
POEM: {POEM_TEXT}
[Scoring Guidelines]
You will score the poem on 7 dimensions. For every dimension, use a standard Likert scale of 1-10, where:
1-2: Rudimentary / Fails to execute / Accidental.
3-4: Competent but cliche / Lacks tension / Inconsistent.
5-6: Solid execution / Clear intent / Some distinctiveness.
7-8: Excellent craft / High complexity / Strong command of devices.
9-10: Masterful / Transformative / Exceptional handling of the mode.
[Rubric]
A) Formal Design & Lineation. Does the poem have an intelligible architecture? Do line breaks, stanza shapes, and white space create meaning or pacing? If a specific form is used, is it handled with discipline or purposeful variation?
B) Sonic Craft & Rhythm. Evaluate the ear of the poem. Look for rhythm (meter or cadence), rhyme (perfect or slant), alliteration, assonance, and the texture of consonants. Does the sound support the meaning?
C) Imagery & Figurative Language. Evaluate the "seeing" of the poem. Are images concrete and sensory? Are metaphors and similes fresh and coherent, or reliant on mixed metaphors and cliches?
D) Complexity & Tension (Close-Reading Depth). Does the poem withstand close scrutiny? Look for ambiguity, paradox, irony, and layers of meaning. Does the poem avoid being overly simplistic or didactic?
E) Diction, Syntax, & Voice. Evaluate word choice and sentence structure. Is the diction precise and compressed? Is the syntax used to control pacing? Is the speaker's persona distinct and controlled?
F) Affective Arc & Resolution. Evaluate the movement of the poem. Is there a turn (volta), a realization, or an emotional shift? Does the ending feel earned?
G) Context & Coherence. Does the poem succeed on its own terms? If it engages with history/identity, does it do so with nuance? Does the poem maintain internal consistency?
[Output Instructions]
1. **Critical Analysis:** First, think step-by-step to justify your assessment. You MUST explicitly reference specific lines or devices to support your scores. The scoring calculation happens here.
2. **JSON Output:** After the analysis, output the scores in valid JSON format.
[JSON Output]

```
{
  "A":  1-10,
  "B":  1-10,
  "C":  1-10,
  "D":  1-10,
  "E":  1-10,
  "F":  1-10,
  "G":  1-10
}
```

Figure B.1: Poetry Evaluation Prompt. We evaluate generated poems with `Qwen3-30B-A3B-Thinking-2507` using a seven-dimension rubric scored on a 1-10 Likert scale. The resulting aggregate is normalized to a 0–100 scale for reporting.

B.5. Reward Formulas and Normalization

For verifiable tasks, the reward definitions are given in the training-details paragraph above. For non-verifiable tasks, we use lightweight answer-level text similarity rewards. We define

$$r_{\text{sim}} = 0.6\,\text{F1}_{\text{SQuAD}} + 0.4\,\text{ROUGE-}L.$$

For datasets with an explicit output-format requirement, the final reward is

$$r = 0.35\,r_{\text{format}} + 0.65\,r_{\text{sim}}.$$

For datasets without a required format, we use $r = r_{\text{sim}}$. In our non-verifiable runs, Poetry Writing and MeetingBank use the similarity reward alone, while the OpenThoughts3-1.2M run additionally includes the format term to enforce the required thinking-style output structure.

B.6. Prompt Templates

B.7. RL-Logit Reward Variants

We implement two variants of logit-based rewards following Gurung and Lapata [12], Zhou et al. [46]. **Log-probability reward:** the scalar reward is $\log \pi_\theta(a^\star \mid q, z)$, the log-probability of the expert answer $a^\star$ given the question q and the generated reasoning trace z. **Perplexity reward:** the reward is the negative perplexity of the

MeetingBank Evaluation Prompt

[Evaluation Role]
You are an expert evaluator of city-council meeting summaries with deep training in meeting discourse analysis, public-record minute writing, procedural language, and factual consistency checking. You evaluate summaries in the style of MeetingBank: concise, segment-level summaries aligned to a specific portion of a city council meeting transcript.
[Goal]
You judge summaries based on whether they accurately capture the important content of the meeting segment, especially the main discussion points, decisions, motions, votes, amendments, referrals, and action items, while remaining concise, coherent, and readable. You are NOT evaluating creativity, elegance, or whether you personally agree with the discussion.
[Inputs]
MEETING TITLE (optional): {MEETING_TITLE}
TRANSCRIPT SEGMENT: {TRANSCRIPT_TEXT}
CANDIDATE SUMMARY: {SUMMARY_TEXT}
[Non-Negotiable Rules]
Judge ONLY against the transcript segment and provided metadata; do not use external knowledge. Faithfulness is more important than fluency, and small factual errors matter, including names, departments, bill numbers, ordinance/resolution IDs, dates, dollar amounts, motions, amendments, vote outcomes, referrals, and assigned follow-up actions.
Do NOT reward plausible inventions. If a detail is not in the transcript, it should not be in the summary. Do NOT penalize a summary for reusing wording from the transcript when that improves accuracy and concision.
If the segment contains a decision, motion, amendment, vote, referral, or action item, omission or distortion of that procedural outcome is a major error. If the segment contains no explicit decision or action item, do NOT hallucinate one; instead reward summaries that correctly characterize the segment as discussion, testimony, presentation, clarification, deliberation, or public comment.
If the transcript itself contains contradictions, do NOT penalize a summary for faithfully reflecting the ambiguity or for choosing the corrected/final version; do penalize a summary that resolves genuine ambiguity by inventing a false certainty. If the segment is purely procedural, brevity is appropriate. If the segment is largely inaudible, garbled, or mostly crosstalk, reward summaries that honestly acknowledge the limited content rather than fabricating substance.
[Scoring Rubric]
Score each dimension on a 1-10 Likert scale:
1-2: Fails badly / misleading / unusable / misses the point entirely.
3-4: Weak / major omissions or distortions / low utility.
5-6: Adequate / partially correct / noticeable problems but has some value.
7-8: Strong / accurate / useful / well-compressed.
9-10: Excellent / highly faithful / sharp / near-reference quality.
A) Informativeness. Does the summary capture the main points of the meeting segment at an appropriate level of compression? A strong summary contains all and only the important information: the core issue under discussion, the substance of the debate, the main proposal, the most important concerns raised, and any meaningful outcome, condensed to a length proportional to the segment's substantive content.
B) Factuality. Are all facts in the summary consistent with the transcript segment? Check who said or did what, what was proposed, whether something was approved, denied, deferred, amended, tabled, or referred, any vote result, any action item or next step, and all important entities, numbers, and procedural details.
C) Fluency. Are the sentences clear, grammatical, and readable? A strong summary should be well-written, precise, and easy to understand by a reader unfamiliar with the meeting.
D) Coherence. Does the summary fit together as a unified account of one meeting segment? A strong summary should present information in a logical order and stay focused on the same agenda item or discussion thread.
E) Conciseness. Does the summary avoid unnecessary repetition and filler while remaining information-dense? Higher scores mean the summary is MORE concise and information-dense.
[Output Instructions]
Step 1 — Critical Analysis. Provide an evidence-grounded analysis that justifies your evaluation. You MUST explicitly reference concrete details from the transcript segment and the candidate summary, identify what the summary gets right and wrong, and note any hallucinated content.
Step 2 — Per-Dimension Verdicts. For each dimension, write one sentence stating the score and the single most important reason for that score, formatted as:

```
Informativeness [X/10]: <reason>
Factuality [X/10]: <reason>
Fluency [X/10]: <reason>
Coherence [X/10]: <reason>
Conciseness [X/10]: <reason>
```

Step 3 — JSON Scores. Output the final scores in valid JSON format.
[JSON Output]

```
{
  "informativeness": 1-10,
  "factuality": 1-10,
  "fluency": 1-10,
  "coherence": 1-10,
  "redundancy": 1-10
}
```

Figure B.2: MeetingBank Evaluation Prompt. We evaluate held-out MeetingBank summaries with `Qwen3-30B-A3B-Thinking-2507` using a five-dimension rubric over informativeness, factuality, fluency, coherence, and conciseness. The JSON output records the final conciseness score under the `redundancy` field, matching the evaluation pipeline.

expert answer under the same conditional distribution, $-\exp\big(-\frac{1}{|a^\star|}\log \pi_\theta(a^\star \mid q, z)\big)$. We report the best-performing variant for each task.

B.8. Baseline Implementation Details

Rationalization. We prompt the base model to annotate each expert demonstration with a free-form rationale, then perform SFT on the concatenated (question, rationale, answer) sequences. This baseline is designed to incentivize

the model to produce explicit reasoning before the final answer, following the STaR framework [44].

Iterative DPO. Inspired by Iterative Reasoning Preference Optimization [26], we perform 3 rounds of DPO iteratively. In each round, we sample one response per question from the current policy to form preference pairs favoring the expert answer over the on-policy sample. We initialize from the SFT checkpoint to mitigate distribution mismatch and report the best performance across rounds.

On-Policy Distillation (OPD). We closely follow the setting described in Lu and Lab [23] (see also Agarwal et al. [2]). We sample on-policy rollouts from the student and, at each visited prefix, query the teacher (`Qwen3-8B` [42]) to obtain its next-token distribution under the same student-conditioned context. The student is then updated to minimize a per-token reverse-KL distillation loss toward the teacher along these trajectories. This provides dense token-level supervision while avoiding the training–inference mismatch of purely off-policy distillation.

RARO. RARO [3] learns a *reasoning critic* that performs a *relativistic* pairwise comparison between a policy answer and the corresponding expert answer, predicting whether the expert is better, the policy is better, or whether they are tied. The critic's prediction induces a shaped reward: the critic is rewarded for correctly identifying the expert in the pair, while the policy is rewarded for "fooling" the critic; the explicit tie option mitigates critic degeneracy near optimality and stabilizes training. In practice, RARO jointly optimizes both policy and critic with GRPO, using key stabilizers such as sharing parameters between the critic and policy, mixing policy/critic rollouts within a batch, and sampling critic prompts from a replay buffer of past expert and policy answers to reduce catastrophic forgetting.

GRPO-Sim and SoftmaxGRPO-Sim. Both weak-reward variants use the same similarity reward r_{sim} defined in Appendix B.5. Concretely, $r_{\text{sim}} = 0.6\,\text{F1}_{\text{SQuAD}} + 0.4\,\text{ROUGE-}L$ measures token-level overlap between the model output and the reference demonstration; tasks with a required output format also use the format term described in Appendix B.5. GRPO-Sim applies standard GRPO group normalization [33], whereas SoftmaxGRPO-Sim replaces only that advantage computation with Eq. (1). The paired runs match the model, data, reward, rollout group size, PPO clip, KL coefficient, learning rate, rollout sampling, and training budget. Consequently, their difference in Table 2 isolates group-advantage geometry under an identical weak reward. The Exact variants provide the complementary comparison under each task's binary verifier.