POLICY SPLIT: INCENTIVIZING DUAL-MODE EXPLORATION IN LLM REINFORCEMENT WITH DUAL-MODE ENTROPY REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

To encourage diverse exploration in reinforcement learning (RL) for large language models (LLMs) without compromising accuracy, we propose Policy Split, a paradigm that bifurcates the policy into normal and high-entropy modes with a high-entropy prompt. While sharing model parameters, the two modes undergo collaborative dual-mode entropy regularization tailored to distinct objectives. Specifically, the normal mode optimizes for task correctness, while the high-entropy mode incorporates a preference for exploration, and the two modes learn collaboratively. Experiments across model sizes show consistent improvements in average reasoning accuracy together with entropy recovery. On creative-writing tasks, the trained high-entropy mode also produces more diverse outputs and improves several judged quality dimensions. Further analysis shows that the two modes develop distinct behavioral patterns and provide complementary learning signals. Code will be released.

1 INTRODUCTION

The application of reinforcement learning (RL) to large language models (LLMs) (Schulman et al., 2017; Shao et al., 2024; Liu et al., 2025b) has achieved substantial success, most notably in the emergence of o1- and R1-like large reasoning models (LRMs) (Guo et al., 2025; Qwen-Team, 2025; Kimi-Team et al., 2025; Seed et al., 2025). These LRMs exhibit sophisticated reasoning capabilities, particularly in domains requiring complex problem solving and mathematical proficiency.

Building on the success of RL for LLMs, recent research has drawn inspiration from entropy-regularized RL (Ziebart et al., 2008; Mnih et al., 2016; Haarnoja et al., 2018) to develop adapted RL frameworks (Cheng et al., 2025; Yao et al., 2025b; Tan et al., 2025) that incentivize high-entropy rollouts for more effective exploration. These methods have shown promise in mitigating entropy collapse during training and improving final accuracy and reasoning robustness.

However, a fundamental conflict between exploration and exploitation persists in entropy-guided RL for LLMs. Recent studies suggest that performance gains are often strongly correlated with entropy reduction (Yue et al., 2025; Cui et al., 2025; Gao et al., 2025; Zhang et al., 2025b), implying that incentivizing higher entropy may inadvertently degrade accuracy. More directly, a recent study (Jiang et al., 2025b) demonstrates that naive entropy regularization can cause outputs to degenerate into incoherent high-entropy sequences, as illustrated in Figure 1(a). Since these models already possess robust capabilities, forcing higher entropy tends to undermine their established proficiency.

To maintain high accuracy while encouraging exploration during continual post-training, we adopt a simple intuition: instantiate two policies within one model. One policy is optimized for accuracy, while the other prioritizes high entropy, and the two policies benefit from each other's experience.

Specifically, we propose Policy Split, a novel paradigm for entropy-guided LRM reinforcement learning. Policy Split bifurcates the policy into a normal mode and a high-entropy mode, with the latter identified by an additional system prompt. During training, as shown in Figure 1(b), the normal mode maximizes correctness as in standard GRPO-like RL, while the high-entropy mode receives an additional preference for exploration. Information is shared between the two modes, allowing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

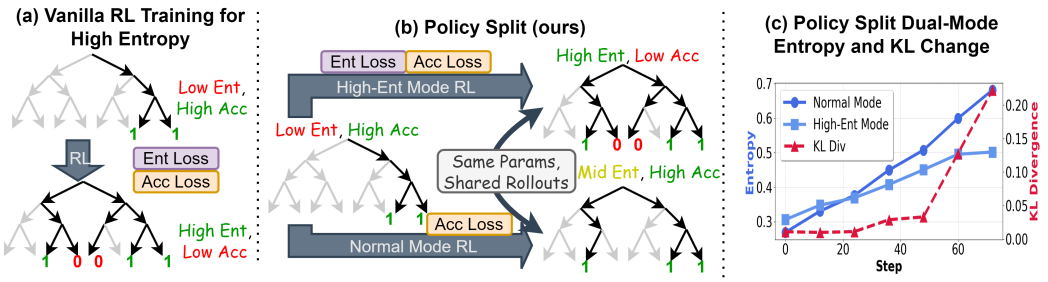


Figure 1: (a) Entropy-guided RL can reduce accuracy when it strongly favors high entropy. (b) Policy Split bifurcates the policy into normal and high-entropy modes. (c) Dual-mode entropy regularization increases the divergence between the two modes and produces distinct entropy behaviors.

the normal mode to learn from newly discovered correct rollouts and encouraging the high-entropy mode to explore while maintaining acceptable accuracy.

Experiments validate that Policy Split trains two modes with different entropy behaviors within one model, as shown in Figure 1(c). The normal mode exhibits a conservative entropy change to maintain accuracy, while the high-entropy mode explores a wider range of solutions, leading to a progressively larger divergence between the two modes. Across reasoning benchmarks and model sizes, Policy Split consistently improves average accuracy and recovers entropy relative to the low-entropy base models. We further evaluate whether the learned high-entropy behavior transfers to open-ended creative-writing tasks.

Furthermore, we analyze the dual-mode exploration of Policy Split and find that the high-entropy mode generates distinct responses that improve pass@k performance for larger k. We also test whether the learned high-entropy activation behavior is robust to an equivalent unseen prompt.

Our contributions are listed as follows.

- We introduce Policy Split, an RL framework that bifurcates a policy into normal and high-entropy modes with distinct optimization targets and collaborative training.
- Across tasks and model sizes, Policy Split consistently improves average reasoning accuracy while recovering policy entropy; its trained high-entropy mode also improves diversity and several quality dimensions on creative-writing tasks.
- We analyze Policy Split's dual-mode exploration and show that the trained high-entropy mode produces responses distinct from normal sampling and remains activatable by a semantically equivalent prompt.

2 Related Work

Entropy-Guided LLM Reinforcement Existing research shows that entropy plays an important role in the reinforcement learning (RL) of LLMs (Yue et al., 2025; Cui et al., 2025). One line of work applies entropy or its proxy by assigning weight on different tokens or sequences, including weighting more on high entropy tokens (Wang et al., 2025b; Cui et al., 2025; Vanlioglu, 2025; Wang et al., 2025a), weighting more on low entropy tokens (Chen et al., 2025; Zhang et al., 2025d), or weighting more on medium entropy tokens (Zhang et al., 2025c). Another line of work, inspired by the entropy regularization from RL methods before the era of large language models (LLMs) (Ziebart et al., 2008; Mnih et al., 2016; Haarnoja et al., 2018), proposes to directly maximize entropy for more effective exploration (Cheng et al., 2025; Yao et al., 2025b; Tan et al., 2025), or minimize entropy for unsupervised accuracy improvement (Gao et al., 2025; Zhang et al., 2025b).

Policy Split continues the work on entropy maximization through dual-mode policy bifurcation and dual-mode entropy regularization. It applies distinct entropy preferences to the two modes, decoupling the pressure for accuracy from the need for high-entropy exploration.

Generation Modes of LLMs LLMs can switch generation styles (Tseng et al., 2024; Chen et al., 2024) through their inherent role-playing and personalization capabilities (Park et al., 2023; Zhou et al., 2023; Wang et al., 2024b) or through additional training (Shea & Yu, 2023; Wang et al., 2024a; Li et al., 2024). More recently, building on foundation models that support multiple reasoning modes (Yang et al., 2025; Liu et al., 2025a; Zhan et al., 2025), adaptive mode-switching methods have been proposed to guide LRMs to adopt different reasoning efforts manually or automatically (Zhu & Li, 2025; Zhang et al., 2025a; Lou et al., 2025; Jiang et al., 2025a; Tu et al., 2025).

Policy Split extends this line of work by training modes differentiated by policy entropy rather than linguistic style or reasoning effort. Furthermore, Policy Split bifurcates the model not only for inference-time utility but also to support entropy-guided training.

3 METHOD

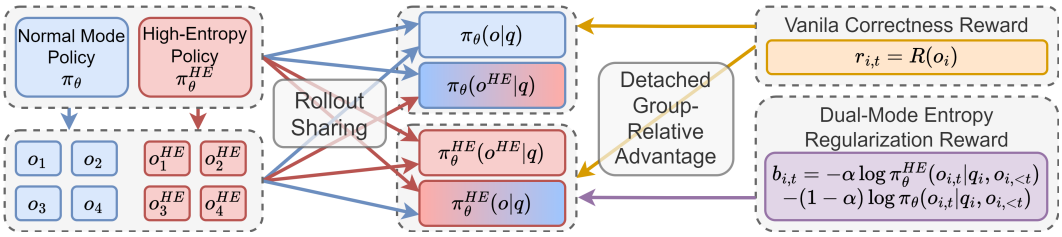


Figure 2: The training framework of Policy Split.

We first describe how prompting implements two modes within one model, then formulate dual-mode entropy regularization, and finally present the overall training procedure.

3.1 HIGH ENTROPY MODE

Mode switching within one LLM is typically implemented with prompts. We use a natural-language system instruction to identify the high-entropy mode. The system prompt is shown as follows.

> You are now in the "High Entropy" mode. Your primary directive is to explore a unique thought process and perspective before arriving at a final answer. Do not rely solely on your typical, most direct reasoning path. Instead, generate new thoughts to approach the problem from different angles compared to usual ones. Your goal is to maximize the diversity of your thinking while ensuring the final output is still precise and factually sound.

Formally, given a policy π_θ, system prompt s, and query q, the high-entropy-mode policy is defined as

$$\pi_\theta^{HE}(\cdot|q) = \pi_\theta(\cdot|s, q). \tag{1}$$

Our analysis in Section 5 shows that simply prepending a high-entropy system prompt to the LRM context is insufficient to increase entropy. The model's output distribution remains largely unchanged, with negligible shifts in inter-mode KL divergence. These findings suggest that prompting alone cannot induce the desired high-entropy behavior and that explicit training is required.

3.2 DUAL-MODE ENTROPY REGULARIZATION

As shown in Figure 1, Policy Split aims to (1) encourage novel exploration in the high-entropy mode and (2) preserve accuracy in the normal mode. We implement these objectives through dual-mode entropy regularization.

To preserve accuracy in the normal mode, the advantage for the i-th rollout in a group of size G is computed following vanilla GRPO:

$$A_{i,t} = \frac{r_i - \text{mean}(\{r_1, r_2, ..., r_G\})}{\text{std}(\{r_1, r_2, ..., r_G\})}. \tag{2}$$

To encourage exploration distinct from the normal mode while maintaining stable training, we make three changes to the high-entropy-mode advantage. First, an entropy term favors high-entropy rollouts. Second, a KL-divergence term encourages exploration away from the normal mode. Third, following previous work (Cheng et al., 2025), we clamp the entropy-related advantage so that it does not reverse the sign of the correctness advantage. Given the advantage coefficient $\eta > 0$ and clamping threshold $\kappa \geq 1$, with $\mathcal{H}$ and $\mathbb{D}_{KL}$ denoting entropy and KL divergence, respectively, the high-entropy-mode advantage is

$$\widetilde{A}_{i,t} = A_{i,t} + \eta \,\text{clip}\Big(\text{detach}(B_{i,t}),\ \pm\frac{|A_{i,t}|}{\kappa}\Big), \tag{3}$$

where

$$B_{i,t} = \frac{b_{i,t} - \text{mean}\Big(\{b_{j,s}\}_{j=1,s=1}^{G,|o_j|}\Big)}{\text{std}\Big(\{b_{j,s}\}_{j=1,s=1}^{G,|o_j|}\Big)}, b_{i,t} = \beta_1 \mathcal{H}_{i,t}^{HE} + \beta_2 \mathbb{D}_{KL}(\pi_\theta^{HE} | \pi_\theta). \tag{4}$$

Following previous work (Yao et al., 2025b), we construct the sampled entropy/KL signal from token log-probabilities. The importance correction for rollouts collected from the old policy is supplied by the PPO likelihood ratios defined below, while b_t specifies the detached per-token signal:

$$b_t = -\alpha \log \pi_\theta^{HE}(o_t | q, o_{<t}) - (1-\alpha) \log \pi_\theta(o_t | q, o_{<t}). \tag{5}$$

Appendix B derives the connection between Equation 4 and the sampled log-probability signal in Equation 5 under the stated estimation assumptions. Intuitively, the π_θ^{HE} term reflects the preference for higher entropy, while the π_θ term encourages the high-entropy mode to move away from the normal mode.

Given the advantages for the normal mode (Equation 2) and high-entropy mode (Equation 3), the corresponding PPO-like surrogate objectives $\mathcal{L}$ and $\widetilde{\mathcal{L}}$ are

$$\begin{aligned} \mathcal{L}(q,o) &= \sum_{t=1}^{|o|} \min\Big[\rho_t A_t, \text{clip}(\rho_t,\ 1 \pm \epsilon) A_t\Big], \\ \widetilde{\mathcal{L}}(q,o) &= \sum_{t=1}^{|o|} \min\Big[\widetilde{\rho}_t \widetilde{A}_t, \text{clip}(\widetilde{\rho}_t,\ 1 \pm \epsilon) \widetilde{A}_t\Big], \end{aligned} \tag{6}$$

where

$$\rho_t = \frac{\pi_\theta(o_t | q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t | q, o_{<t})},\ \widetilde{\rho}_t = \frac{\pi_\theta^{HE}(o_t | q, o_{<t})}{\pi_{\theta_{\text{old}}}^{HE}(o_t | q, o_{<t})}. \tag{7}$$

Policy Split samples rollouts from both the normal and high-entropy modes and applies the corresponding objectives. Formally, it optimizes

$$\mathcal{J}_\theta = \mathbb{E}\Big[q \sim Q, o \sim \pi_{\theta_{old}}(\cdot|q), o^{HE} \sim \pi_{\theta_{old}}^{HE}(\cdot|q)\Big] \frac{\sum_{i=1}^{G/2}}{G} \Big[\mathcal{L}(q,o) + \widetilde{\mathcal{L}}(q, o^{HE})\Big]. \tag{8}$$

Following Share-GRPO (Yao et al., 2025a) and related work, we additionally share rollouts between the two modes. Intuitively, rollout sharing allows the normal mode to learn from correct answers discovered by the high-entropy mode and helps prevent the high-entropy mode from collapsing by exposing it to stable correct responses from the normal mode.

Despite rollout sharing, each mode is always paired with its designated advantage. The normal mode (without the system prompt) is trained with the correctness-only advantage, whereas the high-entropy mode (with the system prompt) is trained with the entropy-regularized advantage.

Policy Split is therefore not a direct combination of rollout sharing and a conventional entropy advantage. Its distinguishing component is the mode-conditioned entropy/KL signal in Equation 5, which relates the log-probabilities of the high-entropy and normal modes and explicitly encourages their exploration behaviors to separate. The complete design combines mode-conditioned objectives, this inter-mode entropy/KL signal, and rollout sharing.

Formally, the adapted loss is defined as

$$\mathcal{J}_\theta^{\text{Sharing}} = \mathbb{E}_{q,o,o^{HE}} \frac{1}{G} \sum_{i=1}^{G/2} \Big[\mathcal{L}(q,o) + \widetilde{\mathcal{L}}(q, o^{HE}) + \mathcal{L}(q, o^{HE}) + \widetilde{\mathcal{L}}(q,o)\Big]. \tag{9}$$

Table 1: Main results. The highest average results for each model are shown in bold. Statistical comparisons are discussed in Section 4.2.

Models	MATH-500		AIME-24/25		GPQA-Diam		MMLU-Pro		Average	
	Acc	*Ent*	*Acc*	*Ent*	*Acc*	*Ent*	*Acc*	*Ent*	*Acc*	*Ent*
Qwen3-1.7B										
Original	89.08	0.34	40.54	0.36	38.19	0.51	56.52	0.48	56.08	0.42
GRPO	89.12	0.31	41.04	0.35	37.88	0.51	55.71	0.46	55.94	0.41
w/ Forking Token Only	89.15	0.31	37.92	0.36	38.95	0.51	56.00	0.46	55.51	0.41
w/ Clip-Cov	89.03	0.31	38.33	0.36	37.63	0.50	56.66	0.45	55.41	0.41
w/ Ent Driven Adv	89.10	0.32	36.46	0.36	39.46	0.51	55.91	0.46	55.23	0.41
w/ Ent Regularization	89.60	0.59	42.29	0.64	38.28	0.88	56.07	0.74	56.56	0.71
w/ Ent Advantage	89.62	0.69	42.92	0.74	37.69	1.00	56.62	0.92	56.71	0.84
Policy Split	89.33	0.59	43.25	0.68	38.07	1.06	56.46	0.96	**56.78**	0.82
+ HE prompt	87.38	0.65	35.33	0.78	35.61	1.11	54.41	1.03	53.18	**0.89**
Qwen3-4B										
Original	93.85	0.34	67.08	0.39	52.27	0.58	66.79	0.55	70.00	0.47
GRPO	93.90	0.34	67.08	0.39	51.01	0.59	68.02	0.55	70.00	0.47
w/ Forking Token Only	94.08	0.34	67.29	0.40	53.22	0.62	68.30	0.55	70.72	0.48
w/ Clip-Cov	93.92	0.34	67.29	0.40	51.77	0.62	68.80	0.55	70.45	0.48
w/ Ent Driven Adv	93.97	0.34	66.04	0.40	53.60	0.61	69.07	0.55	70.67	0.48
w/ Ent Regularization	94.00	0.48	68.83	0.53	51.86	0.76	67.63	0.65	70.58	0.61
w/ Ent Advantage	94.08	0.52	69.79	0.58	51.89	0.84	68.30	0.76	71.02	0.68
Policy Split	94.12	0.41	71.67	0.49	52.53	0.82	68.68	0.73	**71.75**	0.61
+ HE prompt	93.50	0.64	68.12	0.73	52.59	0.94	67.68	0.84	70.47	**0.79**
Qwen3-8B										
Original	94.20	0.32	70.29	0.36	56.89	0.57	74.38	0.56	73.94	0.45
GRPO	94.42	0.30	70.83	0.35	56.45	0.57	75.36	0.54	74.27	0.44
w/ Forking Token Only	94.00	0.30	69.58	0.35	57.07	0.57	75.05	0.54	73.93	0.44
w/ Clip-Cov	94.45	0.30	69.17	0.35	56.44	0.57	75.16	0.54	73.81	0.44
w/ Ent Driven Adv	94.20	0.30	68.87	0.35	55.56	0.57	73.86	0.54	73.12	0.44
w/ Ent Regularization	94.00	0.43	71.06	0.48	57.87	0.72	74.42	0.66	74.34	0.57
w/ Ent Advantage	94.77	0.46	70.62	0.51	57.77	0.75	74.84	0.72	74.50	0.61
Policy Split	94.40	0.48	71.88	0.55	58.90	0.90	75.41	0.85	**75.15**	0.70
+ HE prompt	93.33	0.67	65.88	0.73	53.72	1.08	73.04	1.02	71.49	**0.88**

Table 2: Creative-writing results on Qwen3-8B-based models under different decoding temperatures. Metrics include rule-based diversity measures (N-gram diversity and self-BLEU diversity) and LLM-as-a-judge scores (creativity, originality, narrative flow, emotional impact, and imagery). All scores are scaled to 100. The best results are shown in bold.

Model / Decoding	*Div-N-Gram*	*Div-BLEU*	*Creativity*	*Originality*	*Narrative*	*Impact*	*Imagery*
Original, $T = 0.6$	49.62	60.34	57.69	48.56	69.81	55.19	62.21
Original, $T = 1.0$	54.08	66.92	58.48	49.13	67.98	55.95	60.81
Original, $T = 1.4$	**54.20**	**66.98**	53.27	44.52	63.89	49.76	55.60
GRPO	49.82	60.26	55.29	47.31	68.56	53.94	59.71
w/ Ent Advantage	49.67	60.74	57.02	47.50	69.42	53.56	60.67
Policy Split	50.04	61.67	57.88	48.46	68.08	54.18	60.19
+ HE prompt	52.08	65.34	**62.50**	**55.29**	**70.48**	**56.06**	**65.67**

3.3 Overall Process

The overall training framework is shown in Figure 2. Policy Split first samples rollouts from the normal and high-entropy modes, then applies rollout sharing and inter-mode modeling, and finally computes the advantages in Equations 2 and 3 for the corresponding query-response pairs.

4 EXPERIMENTS

4.1 SETTINGS

We summarize the key experimental settings here and provide additional reproduction details in Appendix C.

Training Setting Following previous work (Dai et al., 2025; Yao et al., 2025c), we randomly sample 20K examples from DeepMath-103K (He et al., 2025) for RL training. We implement Policy Split and all baselines with `verl` (Sheng et al., 2025). All methods are trained for one epoch with a batch size of 256, a learning rate of 3×10^{-6}, and a rollout size of 8. The maximum rollout length is $12,288$. Following Cheng et al. (2025), we set the clamping threshold κ in Equation 3 to 2.0. We set the advantage coefficient η in Equation 3 to 3×10^{-2} and the log-probability weight α in Equation 5 to 0.5.

Evaluation Setting For general evaluation across mathematics, science, and the humanities, we use MATH-500 (Lightman et al., 2023), AIME-24 (MAA, 2024), AIME-25, GPQA-Diamond (Rein et al., 2024), and MMLU-Pro (Wang et al., 2024c). All prompts are shown in Appendix D. For creativity, we follow Minh et al. (2024) and evaluate creative story writing. The creative-writing queries are adapted and extended from that work and are listed in Appendix D. During inference, the maximum generation length is $32,768$, and the decoding temperature is 0.6 unless otherwise specified. We use the same top-p setting across all creative-writing comparisons and additionally evaluate the original model at temperatures 1.0 and 1.4. We report average accuracy across eight sampled runs and average entropy across all tokens. For the exploration analysis, we report pass@k using the same eight sampled responses. The reported entropy scores are full-vocabulary Shannon entropy in bits.

Models We position Policy Split as lightweight post-training for low-entropy LRMs, aiming to improve performance and induce dual-mode behavior with limited additional training. We evaluate well-established LRMs (Qwen3) at several scales, including 1.7B, 4B, and 8B (Yang et al., 2025).

Baselines We compare Policy Split with two categories of existing methods. Entropy-as-weight methods amplify or attenuate the advantages of selected tokens using entropy-related indicators. (1) Forking-token-only (Wang et al., 2025b) masks the 80% lowest-entropy tokens, leaving only forking tokens in the rollouts. (2) Entropy-driven advantage (Zhang et al., 2025d) divides the original advantages by normalized policy entropy, increasing the reward scale for rollouts with lower sequence entropy. (3) Clip-Cov (Cui et al., 2025) is similar to Forking-token-only but uses the covariance between advantage and log-probability for masking. Entropy-as-bias methods directly encourage higher entropy. (4) Entropy regularization (Schulman et al., 2017) directly maximizes entropy in the objective, while (5) entropy advantage (Cheng et al., 2025) adds detached entropy terms to the advantage function.

4.2 RESULTS ON GENERAL TASKS

The experimental results on general tasks are shown in Table 1. Policy Split consistently improves average accuracy over the original models, with $p < 0.01$ for all three model sizes. Compared with the strongest entropy-related baseline, the improvement reaches $p < 0.01$ on Qwen3-8B and $p < 0.05$ on Qwen3-4B; the smaller gain on Qwen3-1.7B is not statistically significant. Standard errors across the evaluated models range from 0.21 to 0.27 and are generally smaller for larger base models. These results support a claim of consistent improvement and entropy recovery rather than large absolute accuracy gains. The high-entropy mode triggered by the dedicated system prompt attains the highest average entropy, at the cost of lower average accuracy.

4.3 RESULTS ON CREATIVE TASKS

For creative-writing evaluation, we use the AlpacaEval Creative Writing prompts also adopted by Minh et al. (2024), together with rule-based metrics following Yao et al. (2025b) and LLM-as-a-judge metrics following Minh et al. (2024). For rule-based evaluation, we adopt N-gram diversity

and self-BLEU diversity, respectively defined as

$$\text{Div-N-Gram} = \frac{1}{8}\sum_{i=1}^{8}\frac{\text{\# of unique N-grams in } o_i}{\text{\# of total N-grams in } o_i}, \tag{10}$$

$$\text{Div-BLEU} = 1.0 - \frac{1}{8}\sum_{i=1}^{8}\text{BLEU}(o_i, O \setminus \{o_i\}), \tag{11}$$

where we use 2-gram diversity and BLEU-4. For LLM-as-a-judge evaluation, we follow Minh et al. (2024) and use rubrics covering creativity, originality, narrative flow, emotional impact, and imagery. Detailed prompts are shown in Appendix D. We report GPT-5 scores in the main text and provide results from GPT-4o and DeepSeek-V3 in Appendix E. The results are shown in Table 2.

Increasing the original model's temperature substantially improves both rule-based diversity metrics. However, the judged quality changes only modestly at $T = 1.0$, and most judged dimensions degrade at $T = 1.4$. In contrast, the high-entropy mode of Policy Split improves diversity over the original $T = 0.6$ setting while obtaining the strongest scores on all five judged dimensions. These results suggest that the learned high-entropy mode is not equivalent to simply increasing decoding temperature. We interpret the transfer result narrowly: math-based RL does not itself establish improved creativity; rather, Policy Split learns a prompt-conditioned high-entropy behavior that can extend to open-ended generation and improve diversity while retaining judged generation quality.

4.4 Ablations

Coefficient of Entropy Advantage The entropy-advantage coefficient η controls the emphasis on high entropy and inter-mode separation. As shown in Table 3, increasing or decreasing η produces a corresponding increase or decrease in entropy. The setting $\eta = 0.03$ provides the best accuracy in this ablation. We also apply the additional advantage signal to the normal mode with coefficient η', explicitly encouraging either higher entropy (positive η') or lower entropy (negative η'). Both variants reduce accuracy, indicating that a correctness-only advantage ($\eta' = 0$) works best for the normal mode in this experiment.

Table 3: Ablations of the high-entropy-mode coefficient η, normal-mode coefficient η', and entropy computation weight α on Qwen3-1.7B. The default configuration is $(\eta, \eta', \alpha) = (0.03, 0, 0.5)$; each other column changes only the indicated coefficient. Accuracy and entropy are averaged across eight runs on the four tasks.

Metric	**Default**	η		η'		α			
		0.01	0.10	0.01	−0.01	0.0	0.25	0.75	1.0
Accuracy	**56.78**	56.06	49.21	56.05	56.12	55.47	56.46	56.57	55.98
Entropy	0.82	0.50	1.54	0.95	0.65	0.78	0.80	0.82	0.83

Table 4: Direct comparison of rollout sharing under GRPO, entropy advantage, and Policy Split on Qwen3-1.7B. *Rollout* denotes auto-regressive sampling cost, and *Modeling* denotes probability recomputation and updating cost. Parentheses show the accuracy change introduced by sharing.

Metric	**GRPO**		**Ent. Advantage**		**Policy Split**	
	Base	+ Share	Base	+ Share	w/o Share	Full
Rollout	1×	1×	1×	1×	1×	1×
Modeling	1×	2×	1×	2×	1×	2×
Accuracy	55.94	55.86 (−0.08)	56.71	56.22 (−0.49)	56.61	**56.78** (+0.17)

Coefficient of Entropy Computation The entropy computation coefficient α regulates the log-probability ratio between the two policies. As α approaches 1.0, the objective increasingly penalizes the high-entropy mode for lower log-probability, enforcing high entropy. Conversely, as α nears 0.0,

the regularization signal increasingly penalizes the similarity between two modes, enforcing mode differences. As demonstrated in Table 3, extreme values for α result in performance degradation, and $\alpha = 0.5$ provides the optimal balance.

Rollout Sharing As shown in Table 4, rollout sharing slightly reduces accuracy when added to either GRPO or entropy advantage. In contrast, it improves Policy Split from 56.61 to 56.78 when combined with the mode-conditioned objectives and inter-mode entropy/KL signal. This comparison indicates that the gain does not come from rollout sharing alone. The additional language-modeling computation results in an overall training time approximately $1.4\times$ that of vanilla GRPO.

5 Analysis

Policy Split divides the policy into two modes and applies heterogeneous but collaborative dual-mode entropy regularization during RL post-training. The normal mode targets reasoning accuracy, whereas the high-entropy mode is intended to support broader exploration and is evaluated on creative-writing tasks in Table 2. We conduct two analyses: (1) whether Policy Split trains two modes with distinct behaviors that can support collaborative training, and (2) whether the trained high-entropy activation behavior is robust to a semantically equivalent prompt.

5.1 Dual-Mode Comparison

Table 5: The absolute per-task difference between the high entropy mode and the original mode in average accuracy, average entropy, average response length, and the forward KL divergence between the two modes. The highest results for each model are shown in bold. The training process of Policy Split effectively enlarges the distance between the policies of the two modes.

Models	**Δ*Acc***	**Δ*Ent***	**Δ*Len***	$\mathbb{D}_{KL}$
		Qwen3-1.7B		
Original	0.56	0.02	600	0.011
GRPO	0.74	0.01	885	0.048
Policy Split	**3.60**	**0.07**	**1,356**	**0.293**
		Qwen3-4B		
Original	0.43	0.02	457	0.015
GRPO	0.33	0.02	519	0.019
Policy Split	**1.28**	**0.18**	**1,798**	**0.935**
		Qwen3-8B		
Original	1.07	0.02	204	0.010
GRPO	1.62	0.02	565	0.017
Policy Split	**3.66**	**0.18**	**1,532**	**1.094**

We compute the absolute differences in accuracy, entropy, and rollout length between the normal mode (without the system prompt) and high-entropy mode (with the high-entropy system prompt) for the original model, vanilla GRPO, and Policy Split. We also compute the forward KL divergence between the two modes. Table 5 reports the results for Qwen3-1.7B, Qwen3-4B, and Qwen3-8B, where the mode separation is most pronounced.

The results show that simply adding a system prompt to the original model causes little behavioral change, with $\mathbb{D}_{KL} \approx 0.01$. Vanilla GRPO slightly increases the divergence, but the change remains small. In comparison, Policy Split produces substantially larger differences between the normal and high-entropy modes across all reported metrics, supporting the intended separation of inference behaviors.

We further investigate whether the two modes expose different correct solutions when multiple samples are allowed. As shown in Table 6, the normal mode of Policy Split obtains the strongest pass@1 accuracy. The high-entropy mode has lower single-sample accuracy but overtakes the normal mode at pass@4 and pass@8, reaching the highest pass@8 score. This pattern provides direct evidence that the trained high-entropy mode broadens exploration.

Table 6: Pass@k results on Qwen3-8B-based models. The high-entropy mode sacrifices single-sample accuracy but achieves the highest pass@4 and pass@8, indicating broader exploration.

Model / Method	pass@1	pass@2	pass@4	pass@8
Original	73.94	77.08	80.10	82.42
GRPO	74.27	76.95	80.28	82.70
Ent Advantage	74.50	78.64	81.16	83.04
Policy Split	**75.15**	**79.51**	82.35	84.91
+ HE prompt	71.49	77.32	**82.74**	**86.53**

5.2 Robustness of High-Entropy Mode Activation

Table 7: Effects of different mode-switching system prompts on Qwen3-8B-based models. Parentheses report entropy changes relative to generation without a mode-switching prompt.

Model	Metric	None	HE	Rewritten HE	LE
Original	Acc	73.94	73.38	73.25	73.82
	Ent	0.45	0.43 (-0.02)	0.44 (-0.01)	0.41 (-0.04)
GRPO	Acc	74.27	73.09	73.03	72.85
	Ent	0.44	0.42 (-0.02)	0.42 (-0.02)	0.40 (-0.04)
Policy Split	Acc	75.15	71.49	71.17	73.02
	Ent	0.70	0.88 ($+0.18$)	0.79 ($+0.09$)	0.68 (-0.02)

As discussed in Section 3, mode switching is implemented by a natural-language system prompt describing the desired high-entropy behavior. We therefore test whether this specifically trained activation behavior remains effective under a semantically equivalent but previously unseen prompt.

We compose two additional prompts. The rewritten high-entropy prompt preserves the meaning of the training prompt while using different wording. The low-entropy prompt uses a similar instructional style but requests the opposite behavior. Both prompts are shown in Appendix D, and the results are reported in Table 7.

The original model and vanilla GRPO exhibit only small entropy changes across the tested prompts, providing no evidence of reliable high-entropy mode activation in this evaluation.

In contrast, Policy Split retains part of the entropy increase under the unseen rewritten high-entropy prompt, indicating semantic robustness of the trained HE-mode trigger. The low-entropy prompt produces little change, however, showing that Policy Split does not learn arbitrary entropy instruction following. Its controllability should therefore be understood as a specifically trained high-entropy activation behavior.

6 Conclusion

In this work, we introduce Policy Split, an RL paradigm that bifurcates an LLM policy into normal and high-entropy modes. Through dual-mode entropy regularization, Policy Split separates the objective of maintaining task accuracy from that of encouraging broader exploration. Inter-mode modeling and rollout sharing allow the modes to learn from shared experience while retaining different optimization targets. Experiments across model scales show consistent average-accuracy improvements and entropy recovery, while the trained high-entropy mode increases diversity and several judged quality dimensions in open-ended generation. These results demonstrate two complementary behaviors that can be selected with the trained mode prompt.

Policy Split highlights the potential of LLM RL frameworks that move beyond accuracy-diversity trade-offs. The success of our dual-mode approach suggests that future research into collaborative modeling could unlock even greater flexibility, steerability, and overall performance.

AI USE STATEMENT

Beyond the use of LLMs as evaluation models explicitly described in the paper, generative AI tools were used only for language editing and proofreading. All research ideas, methodological designs, experiments, analyses, and scientific claims were developed and verified by the authors. The authors reviewed all AI-assisted edits and take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

The proposed objectives and training procedure are specified in Section 3. Section 4.2 describes the evaluation setting and statistical results. Appendix B states the assumptions and derivation underlying the entropy/KL signal, while Appendix C provides training and inference hyperparameters, reward construction, and baseline implementation details. The complete task, mode-switching, creative-writing, and evaluation prompts are provided in Appendix D.

REFERENCES

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, et al. From persona to personalization: A survey on role-playing language agents. *Transactions on Machine Learning Research*, 2024.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025.

Zitian Gao, Lynx Chen, Haoming Luo, Joey Zhou, and Bryan Dai. One-shot entropy minimization. *arXiv preprint arXiv:2505.20282*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*, 2025a.

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025b.

Kimi-Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Xinyu Li, Ruiyang Zhou, Zachary C Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133*, 2024.

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 *The Twelfth International Conference on Learning Representations*, 2023.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

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. *arXiv preprint arXiv:2503.20783*, 2025b.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*, 2025.

MAA. American invitational mathematics examination - aime. `https://maa.org/`, 2024.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent llm outputs. In *The Thirteenth International Conference on Learning Representations*, 2024.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Qwen-Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL `https://qwenlm.github.io/blog/qwq-32b/`.

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.

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

ByteDance Seed, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, et al. Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*, 2025.

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

Ryan Shea and Zhou Yu. Building persona consistent dialogue agents with offline reinforcement learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1778–1795, 2023.

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

Hongze Tan, Jianfei Pan, Jinghao Lin, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. Two tales of persona in llms: A survey of role-playing and personalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16612–16631, 2024.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *arXiv preprint arXiv:2505.10832*, 2025.

Abdullah Vanlioglu. Entropy-guided sequence weighting for efficient exploration in rl-based llm fine-tuning. *arXiv preprint arXiv:2503.22456*, 2025.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025a.

Noah Wang, Zy Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, et al. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14743–14777, 2024a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

Xintao Wang, Yunze Xiao, Jen-tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, et al. Incharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1840–1873, 2024b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024c.

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.

Huanjin Yao, Qixiang Yin, Jingyi Zhang, Min Yang, Yibo Wang, Wenhao Wu, Fei Su, Li Shen, Minghui Qiu, Dacheng Tao, et al. R1-sharevl: Incentivizing reasoning capability of multimodal large language models via share-grpo. *arXiv preprint arXiv:2505.16673*, 2025a.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025b.

Jiashu Yao, Heyan Huang, Shuang Zeng, Chuwei Luo, WangJie You, Jie Tang, Qingsong Liu, Yuhang Guo, and Yangyang Kang. Incorporating self-rewriting into large language model reasoning reinforcement. *arXiv preprint arXiv:2511.16331*, 2025c.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

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.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025a.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025b.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *arXiv preprint arXiv:2508.11408*, 2025c.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025d.

Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Libiao Peng, Jiaming Yang, Xiyao Xiao, et al. Characterglm: Customizing chinese conversational ai characters with large language models. *arXiv preprint arXiv:2311.16832*, 2023.

Jason Zhu and Hongyu Li. Towards concise and adaptive thinking in large reasoning models: A survey. *arXiv preprint arXiv:2507.09662*, 2025.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A LIMITATIONS

Stability Like other entropy-as-bias methods that encourage high-entropy rollouts, including entropy regularization and entropy advantage, Policy Split can be susceptible to training instability. If the entropy coefficient (η in our framework) is too large, the model may suffer from entropy inflation followed by a collapse in task performance. This phenomenon aligns with recent findings (Jiang et al., 2025b). Although entropy clamping provides a practical way to stabilize entropy growth, a more rigorous theoretical analysis and a fundamental solution are left for future work.

Computational Cost Policy Split adopts a training framework related to Share-GRPO (Yao et al., 2025a), sharing rollouts between modes. In our setting, rollout sharing retains $1\times$ auto-regressive sampling cost but requires $2\times$ language-modeling computation for probability recomputation and updating, resulting in approximately $1.4\times$ the end-to-end training time of vanilla GRPO. Policy Split is intended as a lightweight RL post-training method rather than a pretraining procedure. It trades this controlled post-training overhead for two switchable behaviors: a normal reasoning mode and a high-entropy exploration mode.

B THEORETICAL DISCUSSION OF LOSS

This appendix motivates the sampled log-probability signal in Equation 5. We first describe log-probability estimators of entropy and KL divergence, then discuss policy importance sampling, and finally derive the connection to Equation 4 under these estimation assumptions. This derivation is not intended to claim strict equivalence under arbitrary training dynamics.

B.1 ESTIMATION WITH LOG-PROBABILITY

The derivation and implementation follow Yao et al. (2025b). By the definition of Shannon entropy, for a policy π_θ over vocabulary $\mathcal{V}$, its entropy $\mathcal{H}$ is

$$\mathcal{H}(\pi_\theta^{HE}(\cdot|q,o^{<t})) = -\sum_{o^t\in\mathcal{V}} \pi_\theta^{HE}(o^t|q,o^{<t}) \log \pi_\theta^{HE}(o^t|q,o^{<t}). \tag{12}$$

Rewriting this expression as an expectation gives

$$\mathcal{H}(\pi_\theta^{HE}(\cdot|q,o^{<t})) = \mathbb{E}_{o^t\sim\pi_\theta^{HE}(\cdot|q,o^{<t})}[-\log \pi_\theta^{HE}(o^t|q,o^{<t})]. \tag{13}$$

Similarly, KL divergence can be estimated from log-probabilities using the k1 estimator (Schulman et al., 2017). By definition,

$$\mathbb{D}_{KL}(\pi_\theta^{HE}|\pi_\theta) = \sum_{o^t\in\mathcal{V}} \pi_\theta^{HE}(o^t|q,o^{<t}) \log\left(\frac{\pi_\theta^{HE}(o^t|q,o^{<t})}{\pi_\theta(o^t|q,o^{<t})}\right). \tag{14}$$

Rewriting this expression as an expectation gives

$$\mathbb{D}_{KL}(\pi_\theta^{HE}|\pi_\theta) = \mathbb{E}_{o^t\sim\pi_\theta^{HE}(\cdot|q,o^{<t})}\left[\log \pi_\theta^{HE}(o^t|q,o^{<t}) - \log \pi_\theta(o^t|q,o^{<t})\right]. \tag{15}$$

B.2 IMPORTANCE SAMPLING

Directly computing $\mathbb{E}_{o^t\sim\pi_\theta^{HE}}$ is generally intractable because the rollouts are sampled from $\pi_{\theta_{old}}^{HE}$. Following the importance-sampling formulation used in PPO (Schulman et al., 2017), entropy and KL divergence can still be estimated from log-probabilities. For any function f,

$$\begin{aligned}\mathbb{E}_{x\sim P}[f(x)] &= \sum_x P(x)f(x) \\ &= \sum_x Q(x)\frac{P(x)}{Q(x)}f(x) \\ &= \mathbb{E}_{x\sim Q}\left[\frac{P(x)}{Q(x)}f(x)\right].\end{aligned} \tag{16}$$

Substituting P with $\pi_\theta^{HE}(\cdot|q, o^{<t})$, Q with $\pi_{\theta_{old}}^{HE}(\cdot|q, o^{<t})$, and f with

$$-\log \pi_\theta^{HE}(o^t|q, o^{<t}), \tag{17}$$

or

$$\log \pi_\theta^{HE}(o^t|q, o^{<t}) - \log \pi_\theta(o^t|q, o^{<t}), \tag{18}$$

gives the corresponding importance-sampling form for entropy and KL divergence. Following the implementation convention of Yao et al. (2025b), Policy Split uses the PPO likelihood ratio $\widetilde{\rho}_t$ in Equation 7 to provide this correction in the surrogate objective. Equation 5 therefore defines the detached token-level signal and does not introduce a second importance ratio inside b_t.

B.3 Connection between Equation 4 and 5

Substituting the log-probability estimates into Equation 4 gives

$$\begin{aligned} b_t &= \beta_1 \mathcal{H}_t^{HE} + \beta_2 \mathbb{D}_{KL}(\pi_\theta^{HE}|\pi_\theta) \\ &= -(\beta_1 - \beta_2) \log \pi_\theta^{HE} - \beta_2 \log \pi_\theta \\ &= \beta_1 \Big[-(1 - \frac{\beta_2}{\beta_1}) \log \pi_\theta^{HE} - \frac{\beta_2}{\beta_1} \log \pi_\theta \Big]. \end{aligned} \tag{19}$$

Let $\alpha = 1 - \beta_2/\beta_1$. Because the subsequent normalization removes the common scale factor β_1, we compute b_t as

$$b_t = -\alpha \log \pi_\theta^{HE} - (1 - \alpha) \log \pi_\theta. \tag{20}$$

C Implementation Details

C.1 Rewarding

We use the answer-parsing logic from the PRM800K dataset for reward computation[1]. Correct responses receive a reward of 1.0, and incorrect responses receive 0.0.

C.2 Hyper-parameters

The training and inference hyperparameters shared by all RL approaches are listed in Tables 8 and 9, respectively.

Table 8: Training hyperparameters.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
parameters	full	optimizer	AdamW
epoch	1	batch size	256
learning rate	$3e-6$	learning rate scheduler	constant
gradient clip norm	1.0	weight decay	$1e-2$
cutoff length	$12,288$	rollout size	8
rollout temperature	1.0	PPO ϵ	0.2
KL loss coefficient	0.0		

C.3 Baselines

For a controlled comparison, we reimplement all baselines under the same experimental settings.

GRPO w/ Forking Token Only Following the original paper (Wang et al., 2025b), we mask the 80% lowest-entropy tokens during loss computation.

[1] https://github.com/openai/prm800k/blob/main/prm800k/grading/grader.py

Table 9: Inference hyperparameters.

Setting	Max new tokens	Temperature	Top-k	Top-p
Train	$12,288$	1.0	-1	1.0
Test	$32,768$	0.6	20	0.95

GRPO w/ Clip-Cov We implement Clip-Cov using the official implementation[2] and its recommended hyperparameters: a clip ratio of 0.0002, a lower clipping bound of 1.0, and an upper clipping bound of 5.0.

GRPO w/ Entropy Drive Advantage Entropy-Driven Advantage follows the signal-level component of EDGE-GRPO (Zhang et al., 2025d). Other components, such as guided error correction, are excluded to maintain a controlled comparison.

GRPO w/ Entropy Regularization We use the official `verl` implementation of entropy regularization and set the loss coefficient to 0.003 after testing 0.001, 0.003, and 0.01.

GRPO w/ Entropy Advantage Instead of the vanilla entropy-advantage method, we use the formulation of Cheng et al. (2025), which introduces additional entropy scaling and clamping for more stable training. Following that work, we set $\alpha = 0.4$ and $\kappa = 2.0$.

D Prompts

D.1 Task Solving Prompts

Math-500

Solve the following math problem. Put your final answer within \boxed{}.

{query}

AIME-24 and AIME-25

Solve the following math problem. Put your final answer within \boxed{}.

{query}

GPQA-Diamond

Select one option from the given choices that accurately answers the question, and enclose your final answer (which should be the uppercase letter index of the chosen option) within \boxed{}.

Question
{question}

Choices
{choices}

[2] https://github.com/PRIME-RL/Entropy-Mechanism-of-RL

MMLU-Pro

Select one option from the given choices that accurately answers the question, and enclose your final answer (which should be the uppercase letter index of the chosen option) within \boxed{}.

Question
{question}

Choices
{choices}

D.2 Entropy Mode Prompts

High Entropy Mode

You are now in the "High Entropy" mode. Your primary directive is to explore a unique thought process and perspective before arriving at a final answer. Do not rely solely on your typical, most direct reasoning path. Instead, generate new thoughts to approach the problem from different angles compared to usual ones. Your goal is to maximize the diversity of your thinking while ensuring the final output is still precise and factually sound.

Rewritten High Entropy Mode

Activate the "Maximum Variance" framework. Your fundamental mission is to traverse unconventional intellectual paths and distinctive viewpoints before concluding your analysis. Eschew your standard, linear logic. Instead, synthesize fresh concepts to investigate challenges through multifaceted lenses that deviate from your routine methods. Aim for the highest possible cognitive breadth during your deliberation, ensuring the ultimate resolution remains rigorously accurate and grounded in reality.

Low Entropy Mode

You are now in "Low Entropy" mode. Your primary directive is to provide the most direct, efficient, and logically dense response possible. Avoid creative tangents, speculative reasoning, or exploring multiple perspectives. Instead, employ a linear, high-probability reasoning path. Prioritize established facts and standard deductive logic to minimize "noise" and maximize the "signal" in your output. Your goal is to reach the most accurate conclusion using the fewest cognitive steps, ensuring the final answer is concise, highly structured, and strictly precise.

D.3 Creative Writing

Creative Writing Tasks

Write a story about a mysterious door that appears in an unexpected place.
Write a story about an alien civilization's first contact with Earth from their perspective.
Write a story about a world where time suddenly starts moving backwards.
Write a story about a library where the books are blank until someone touches them, revealing their own life story.
Write a story about a person who discovers that their shadow has started acting independently and even leaving without them.
Write a story about a futuristic city built entirely inside a giant, dormant biological

organism.
Write a story about a world where people are born with a timer on their wrist showing exactly how much time they have left to live.
Write a story about a lighthouse keeper who realizes the light isn't for ships, but to keep something in the ocean from coming ashore.
Write a story about a professional 'memory wiper' who accidentally rediscovers a memory of their own that they weren't supposed to have.
Write a story about a small town where, once a year, all the statues come to life for exactly one hour.
Write a story about a group of children who find an old radio that broadcasts news from fifty years in the future.
Write a story about a world where color is a luxury that people have to buy, and the poor live in a black-and-white reality.
Write a story about an astronaut who lands on a planet that looks exactly like Earth, but they are the only human there.

Creative Writing Evaluation

You are an expert literary critic and creative writing judge with an eye for technical skill, emotional depth, and narrative innovation. Please evaluate the following story based on the provided prompt. You will score the story across five specific dimensions on a scale of **0 to 10**.

Evaluation Rubric

1. **Creativity (Novelty and uniqueness of ideas):**
* *High Score:* The story explores unconventional concepts, unexpected plot twists, or unique world-building.
* *Low Score:* The story relies heavily on clichés or predictable tropes.

2. **Originality (Innovative approach to the prompt):**
* *High Score:* The author interpreted the prompt in a way that is fresh or subverts expectations.
* *Low Score:* The author took the most literal or common interpretation of the prompt.

3. **Narrative Flow (Coherence and story progression):**
* *High Score:* The story moves logically and smoothly; the pacing feels intentional and keeps the reader engaged.
* *Low Score:* The story feels disjointed, confusing, or has significant pacing issues (e.g., a rushed ending).

4. **Emotional Impact (Ability to evoke feelings):**
* *High Score:* The writing makes the reader feel a specific emotion (sadness, joy, tension, etc.) through character depth or stakes.
* *Low Score:* The writing feels clinical, flat, or fails to make the reader care about the outcome.

5. **Imagery (Vividness of descriptions):**
* *High Score:* Rich, sensory details that allow the reader to "see" and "feel" the environment.
* *Low Score:* Vague or generic descriptions; over-reliance on "telling" rather than "showing."

Input Data

Original Prompt:
prompt

Story to Evaluate:
story

Output Format

Please provide your evaluation in the following format:

* **Creativity:** [Score]
* **Originality:** [Score]
* **Narrative Flow:** [Score]
* **Emotional Impact:** [Score]
* **Imagery:** [Score]

Directly output the scores in the above format, without any additional explanations.

E ADDITIONAL LLM-AS-A-JUDGE

Table 10: Creative-writing results on Qwen3-8B-based models using GPT-5, GPT-4o, and DeepSeek-V3 as judges. The evaluated dimensions are creativity, originality, narrative flow, emotional impact, and imagery.

Models	***Creativity***	***Originality***	***Narrative***	***Impact***	***Imagery***
			GPT-5		
Original	57.69	48.56	69.81	55.19	62.21
GRPO	55.29	47.31	68.56	53.94	59.71
w/ Ent Adv	57.02	47.50	69.42	53.56	60.67
Policy Split	57.88	48.46	68.08	54.18	60.19
+ HE prompt	**62.50**	**55.29**	**70.48**	**56.06**	**65.67**
			GPT-4o		
Original	81.54	74.90	85.96	79.90	84.71
GRPO	80.67	72.88	86.35	79.42	83.75
w/ Ent Adv	80.82	73.27	85.77	79.47	83.94
Policy Split	81.88	74.90	86.35	80.53	83.41
+ HE prompt	**84.38**	**77.88**	**86.39**	**80.91**	**86.88**
			DeepSeek-V3		
Original	89.62	85.19	**84.62**	**84.71**	84.33
GRPO	89.04	85.19	84.33	83.85	84.71
w/ Ent Adv	88.85	85.38	**84.62**	84.23	84.71
Policy Split	89.23	85.87	84.13	83.46	75.88
+ HE prompt	**90.38**	**88.27**	84.42	83.85	**87.50**

The scores produced by GPT-5, GPT-4o, and DeepSeek-V3 are shown in Table 10. Across these judges, the high-entropy mode of Policy Split generally obtains the strongest creativity, originality, and imagery scores among the compared methods.