DUET: Dual-Mode Entropy Regularization for Adaptive Reasoning in Large Language Models

Anonymous ACL submission

Abstract

Chain-of-Thought (CoT) reasoning enhances the problem-solving ability of Large Language Models (LLMs), yet incurs unnecessary computational cost on simple problems—a phenomenon known as *overthinking*. We observe that the two reasoning modes of an LLM—direct answering and deliberative CoT—occupy distinct entropy regimes: direct responses exhibit low token-level entropy, while exploratory reasoning produces high-entropy distributions. Building on this observation, we propose **DUET** (**DU**al-mode **E**ntropy-regularized **T**raining), a two-stage post-training framework that uses entropy as an active regularization signal to shape *how* each mode generates and *when* each mode is invoked. Stage 1 installs a direct-answering capacity via supervised fine-tuning on self-generated targets; Stage 2 refines mode selection via offline PPO with a correctness-gated, mode-dependent entropy reward. Experiments on DeepSeek-R1-Distill models (1.5B, 7B, 8B) show that DUET, trained only on mathematical data, reduces response length by up to 47.5% on MATH500, AIME24, and GPQA while preserving or improving accuracy, and transfers effectively to the out-of-distribution MMLU benchmark. Code is available at `https://anonymous.4open.science/r/Efficient-Reasoning-2E5F`.

1 Introduction

Human cognition operates through two distinct processes: a fast, intuitive System 1 and a slow, deliberative System 2 (Kahneman, 2011; Li et al., 2025). While reasoning-specialized large language models (LLMs) such as OpenAI's o1 (OpenAI, 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025) have achieved remarkable capabilities via Chain-of-Thought (CoT) (Wei et al., 2022; Kojima et al., 2022) and its advanced variants (Yao et al., 2023; Besta et al., 2024), they often suffer from a static inference paradigm, consuming significant tokens regardless of problem difficulty (Stojkovic et al., 2025; Wu et al., 2026). We observe that the two reasoning modes—direct answering and deliberative reasoning—exhibit fundamentally distinct entropy profiles in their generated token distributions, a phenomenon rooted in how human cognitive states manifest in language production (Bubeck et al., 2023).

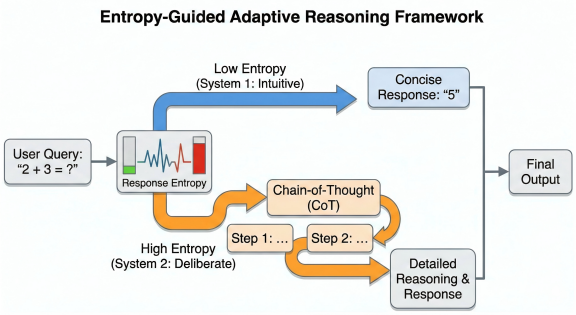


Figure 1: Entropy-driven adaptive reasoning: low-entropy for simple problems (Top), high-entropy for complex problems (Bottom).

To mitigate "overthinking", inference-time methods like UnCert-CoT (Zhu et al., 2025) and CoT-Valve (Ma et al., 2025b) assess problem difficulty at runtime to dynamically allocate computational resources, but introduce undesirable inference-time overhead. Training-time methods such as Adapt-Think (Zhang et al., 2025) utilize online reinforcement learning, which is computationally intensive and can be unstable, while O1-Pruner (Luo et al., 2025) introduces an offline training paradigm but relies on length-based heuristics for complexity, lacking a principled information-theoretic characterization of different reasoning modes.

To address this gap, we draw upon insights from cognitive psychology and psycholinguistics. Dual-process theories distinguish between System 1 (fast, intuitive) and System 2 (slow, deliberate) thinking, with individuals shifting to System 2 when facing uncertainty (Kahneman, 2011). These cognitive

modes manifest distinct linguistic signatures: System 1 produces confident, low-entropy language, while System 2 generates exploratory, high-entropy utterances incorporating hedging markers and reasoning indicators (Hale, 2006). The Entropy Reduction Hypothesis further posits that processing difficulty correlates with entropy reduction in language comprehension, with analogous patterns in production (Gussow, 2023). We treat this as a useful operational heuristic rather than a strict cognitive claim: since LLMs are trained on human-generated text, their output entropy patterns provide a practical signal for shaping dual-system inference behavior.

In this paper, we systematically demonstrate that the two reasoning modes of LLMs—direct answering (NoThinking) and chain-of-thought reasoning (Thinking)—exhibit fundamentally distinct entropy profiles. Rather than using entropy merely as a passive proxy for problem difficulty, we use entropy as a *regularization signal* that shapes the generation behavior of each mode: NoThinking responses should be deterministic and low-entropy, while Thinking responses should maintain appropriate exploratory high-entropy characteristics. We instantiate this idea in **DUET** (**DU**al-mode **E**ntropy-regularized **T**raining), a two-stage post-training framework that combines supervised fine-tuning (SFT) on NoThinking exemplars to establish the capacity for abbreviated reasoning, followed by offline Proximal Policy Optimization (PPO) with a dual-mode entropy reward to shape *when* each mode is used. We note that adaptive mode usage in DUET builds on the abbreviated-reasoning capability instilled by SFT—the PPO stage refines the selection policy via reward shaping rather than instilling a new capability from scratch.

This work makes three key contributions to efficient reasoning in LLMs:

- We systematically reveal that the two reasoning modes of LLMs exhibit fundamentally distinct entropy profiles, providing a practical, label-free signal for differentiating reasoning modes.
- We propose a dual-mode entropy regularization reward that directly shapes the generation behavior of each reasoning mode, enabling adaptive mode selection through reward-shaped optimization rather than relying on inference-time difficulty assessment.
- We evaluate on models of varying sizes (1.5B, 7B, 8B) and architectures (Qwen, Llama). Despite being trained exclusively on mathematical data, DUET achieves favorable performance on general benchmarks (GPQA, MMLU), evidencing cross-domain transferability of the learned meta-cognitive strategy.

2 Related Work

The Overthinking Problem. Recent reasoning models like OpenAI's o1 (OpenAI, 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025) achieve remarkable performance but generate verbose reasoning chains regardless of problem complexity, which is known as *overthinking phenomenon* (Chen et al., 2025; Fan et al., 2025; Zhu and Li, 2025). Many recent studies (Zeng et al., 2025; Su et al., 2025) try to understand how generation length affects the accuracy of LLMs and reveal that deliberative reasoning capability by extremely long reasoning chains does not consistently improve model performance across diverse tasks, while Ma et al. (2025a) demonstrate that models can achieve competitive performance without explicit thinking for many tasks.

CoT Compression and Adaptive Reasoning. To mitigate the computational overhead of large language models in reasoning tasks, researchers have proposed various Chain-of-Thought (CoT) compression techniques (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024). These methods aim to shorten the reasoning path by pruning, compressing, or distilling lengthy reasoning steps into a more compact form, thereby significantly reducing latency and computational costs while maintaining or even improving performance (Kang et al., 2025; Ma et al., 2025b; Lee et al., 2025; Luo et al., 2025). Further research has explored compressing discrete language-based reasoning steps into a continuous latent space, enabling more efficient thought processes by conducting reasoning within this denser representation (Cheng and Durme, 2024; Hao et al., 2024). Our approach further integrates theoretical insights from cognitive psychology, addressing gaps in interpretability by linking entropy regularization to human-like dual-system cognitive signatures (Kahneman, 2011; Hale, 2006).

Another complementary line of research is adaptive reasoning, which centers on enabling models to dynamically allocate computational effort based on task complexity. For instance, a model can learn when to engage in deep, multi-step "slow thinking" and when to rely on more direct "fast thinking" to quickly arrive at an answer, thus striking a better balance between reasoning accuracy

and efficiency (Han et al., 2025; Yi et al., 2025; Fatemi et al., 2025). This dynamic computation allocation allows the model to conserve resources on simpler problems while dedicating more capacity to solving more challenging ones. Distinct from existing methods that rely on external signals to assess task complexity, DUET directly shapes the information-theoretic properties of each reasoning mode through entropy regularization, allowing adaptive mode selection to emerge from the optimization process itself.

3 Preliminaries

3.1 Entropy as a Signature of Cognitive Mode

Dual-process theories distinguish a fast, intuitive System 1 from a slow, deliberate System 2, and the two modes leave distinct linguistic footprints in human language production: confident utterances are low-entropy, while deliberation under uncertainty inserts hedging markers and reasoning indicators that broaden the predictive distribution and elevate local entropy (Kahneman, 2011; Clark and Fox Tree, 2002; Corley and Stewart, 2008; Florian Jaeger, 2010; Hale, 2006). The Uniform Information Density hypothesis (Levy, 2008; Piantadosi et al., 2011) further argues that speakers actively manage entropy to balance processing load. Because LLMs are trained on massive corpora of human-generated text, we hypothesize they inherit these mode-specific entropy patterns through statistical learning—producing low-entropy distributions when answering directly and high-entropy distributions when reasoning exploratorily (Binz and Schulz, 2023; Lampinen et al., 2022; Shanahan, 2024). We treat this as an operational heuristic rather than a claim of conscious metacognition: it provides a principled, label-free complexity signal computed directly from the model's logits.

3.2 Token-Level Entropy as a Difficulty Proxy

Filtered average entropy. For position t in a generated sequence, the token-level entropy is $H_t = -\sum_i p_{t,i} \log p_{t,i}$, where $p_{t,i}$ is the probability of vocabulary item i at step t. Since LLM outputs contain many formulaic low-entropy tokens (especially in math) (Wang et al., 2025), we restrict attention to tokens above a threshold θ:

$$H_{\text{avg}}(y) = \frac{1}{N} \sum_{t=1}^{T} H_t \cdot \mathbb{I}(H_t > \theta),$$
$$N = \sum_{t=1}^{T} \mathbb{I}(H_t > \theta). \quad (1)$$

Sampling-based difficulty proxy. For datasets lacking annotated difficulty, we define a proxy $D = L_s - A_s$, where L_s and A_s are the standardized average response length and accuracy of π_{ref}'s rollouts on each problem. Under mild monotonicity assumptions on π_{ref} (longer and less accurate on harder problems), D is monotone in latent difficulty (Appendix A.1).

Empirical validation. We compute H_{avg} for DeepSeek-R1-Distill-Qwen-1.5B/7B and Llama-8B on MATH500, AIME24, and GSM8K, sweeping $\theta \in \{0, 0.01, 0.1, 0.2, 0.5\}$ (Figure 2, top). Both extremes (too low or too high θ) weaken the entropy–difficulty correlation; $\theta = 0.2$ is consistently strong across models and datasets. Using this threshold, H_{avg} rises with annotated difficulty on MATH500 (Figure 2, bottom); the phenomenon also replicates in non-distilled models (Qwen-3-4B (Team, 2025) and Marco-O1-7B (Zhao et al., 2024)). Token-level analysis (Appendix A.2) reveals the rise is driven by an increased *share* of deliberation-signaling high-entropy tokens rather than higher per-token uncertainty—consistent with psycholinguistic findings on language production under uncertainty (Gussow, 2023). We therefore adopt $\theta = 0.2$ as the default and use H_{avg} as the entropy signal in our regularization framework (Section 4).

The empirical pattern establishes that the two reasoning modes—direct answering and deliberative reasoning—occupy distinct entropy regimes. This motivates using entropy not merely as a passive complexity signal, but as an *active regularization target* that reinforces the natural entropy profile of each mode during training, as we detail next.

4 Methodology

4.1 Overview: DUET

Building on the empirical findings in Section 3, DUET treats entropy not as a passive indicator of problem difficulty, but as an active regularization target that shapes the generation behavior of each reasoning mode. Our key design principle

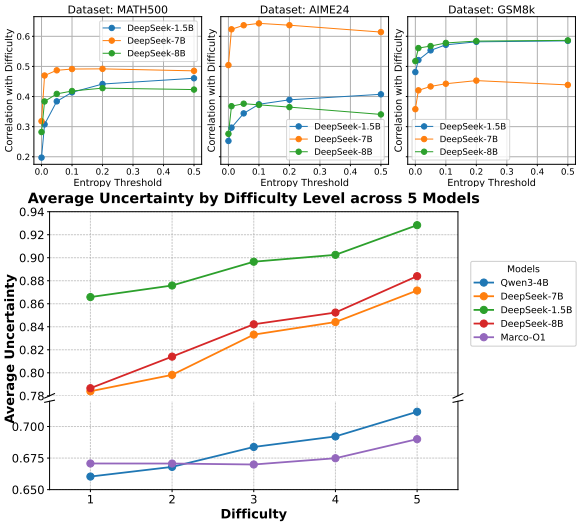


Figure 2: **Top:** Correlation between H_{avg} and a difficulty proxy across models, datasets and thresholds; θ=0.2 is consistently strong. **Bottom:** H_{avg} at θ=0.2 rises with annotated MATH500 difficulty.

is that the two modes should exhibit complementary information-theoretic profiles: NoThinking responses should be *deterministic and low-entropy*, reflecting confident direct answers, while Thinking responses should maintain *appropriate exploratory entropy*, reflecting genuine deliberation. By directly regularizing these characteristics during training, adaptive mode selection emerges from the optimization pressure—the model invokes NoThinking only when it can produce a low-entropy correct answer, and switches to Thinking otherwise.

We instantiate this principle with a two-stage post-training pipeline (Figure 3; pseudocode in Algorithm 1). **Stage 1** (Section 4.2) is a Supervised Fine-Tuning (SFT) step that endows the model with the *capacity* for NoThinking generation while minimizing distributional shift from the reference model. **Stage 2** (Section 4.3) is an offline PPO step that uses a dual-mode entropy-regularized reward to shape *when* each mode is used. Conceptually, Stage 1 teaches the model *how* to answer directly, while Stage 2 teaches *when* to do so. The offline formulation amortizes rollout cost up-front and substantially reduces training overhead relative to online RL.

4.2 Stage 1: Supervised Fine-Tuning via Self-Generated Targets

A central challenge in Stage 1 is that naive SFT on externally curated short answers can pull the model away from its native output distribution, eroding the reasoning capability that Stage 2 must subsequently refine. To avoid this, we construct SFT targets by sampling from the reference model itself, mirroring the principle behind on-policy distillation (OPD) (Agarwal et al., 2024; Hinton et al., 2015) and addressing the well-known exposure-bias problem of off-policy imitation learning (Ross et al., 2011).

Distributional gap analysis. Let π_{ref} denote the pre-SFT reference policy and π_θ the SFT-updated policy. Standard SFT on an external dataset $\mathcal{D}_{\text{ext}} = \{(x, y_{\text{ext}})\}$ minimizes

$$\mathcal{L}_{\text{SFT}}^{\text{ext}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{ext}}}[\log \pi_\theta(y \mid x)], \quad (2)$$

which is equivalent (up to an entropy constant) to minimizing the forward KL divergence

$$\mathbb{E}_x\left[D_{\text{KL}}\big(p_{\text{ext}}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)\big)\right]. \quad (3)$$

When p_{ext} differs markedly from π_{ref}, the update incurs a large *policy shift* $D_{\text{KL}}(\pi_{\text{ref}} \,\|\, \pi_\theta)$, which (i) erodes the long-form reasoning capability that the downstream offline-PPO stage relies on and (ii) amplifies exposure bias at inference, since the model is supervised on trajectories it would never have generated on its own.

Self-generated target construction. To minimize this gap, we draw targets from the reference model itself under a NoThinking prompt-injection template $\mathcal{T}_{\text{NT}}$ (Ma et al., 2025a). For each problem $x \in \mathcal{D}$, we sample K candidates

$$\{y_1, \ldots, y_K\} \overset{\text{i.i.d.}}{\sim} \pi_{\text{ref}}(\cdot \mid x, \mathcal{T}_{\text{NT}}), \quad (4)$$

where $\mathcal{T}_{\text{NT}}$ closes the reasoning block via `<think>\nOkay, I think I have finished reasoning.\n</think>\n\n` to preserve contextual coherence. Among the verified-correct candidates we keep the shortest:

$$y^\star(x) = \arg\min_{y \in \mathcal{Y}^+(x)} |y|, \quad (5)$$

$$\mathcal{Y}^+(x) = \{y_i : \text{is_correct}(y_i)\}.$$

The Stage-1 objective is then

$$\mathcal{L}_{\text{SFT}}^{\text{self}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\left[\log \pi_\theta\big(y^\star(x) \mid x, \mathcal{T}_{\text{NT}}\big)\right]. \quad (6)$$

Why self-output minimizes the distributional gap. By construction, every target lies in the support of the reference distribution: $\pi_{\text{ref}}(y^\star(x) \mid x, \mathcal{T}_{\text{NT}}) > 0$ for every problem retained. Treating the empirical target distribution $p_{\text{self}}(y \mid x) = \delta_{y=y^\star(x)}$ and the candidate distribution as the trun-

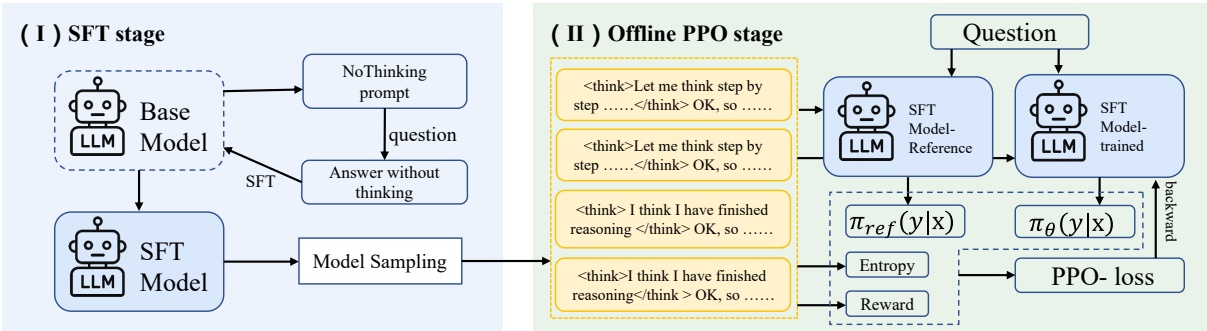


Figure 3: Two-stage post-training framework. **Stage 1 (SFT):** self-generated NoThinking targets minimize the distributional gap to the reference model while installing the prompt-conditioned shortcut. **Stage 2 (Offline PPO):** a dual-mode entropy-regularized reward enforces low entropy for NoThinking and exploratory entropy for Thinking, with an advantage baseline for stability.

cated reference $\pi_{\text{ref}}^{+} \propto \pi_{\text{ref}} \cdot \mathbb{I}_{\mathcal{Y}^+}$, one can show

$$D_{\text{KL}}(p_{\text{self}} \,\|\, \pi_{\text{ref}}(\cdot \mid x, \mathcal{T}_{\text{NT}})) \leq -\log \pi_{\text{ref}}(y^\star(x) \mid x, \mathcal{T}_{\text{NT}}), \tag{7}$$

which is finite and small whenever $\mathcal{Y}^+(x) \neq \emptyset$. Intuitively, training targets are by definition *reachable* under the reference policy, so the SFT update only locally redistributes mass within the support of π_{ref} rather than chasing an external distribution. In contrast, external SFT yields no such bound and can drive π_θ far from π_{ref} when p_{ext} has mass outside $\text{supp}(\pi_{\text{ref}})$. This bound also explains why we restrict $\mathcal{D}$ to the two lowest difficulty tiers of MATH (Levels 1–2; $|\mathcal{D}|$=1,912): on these problems $\mathcal{Y}^+(x)$ is reliably non-empty and the bound in (7) is tight, so the NoThinking shortcut can be installed with minimal collateral drift on the reference policy. Capability preservation is empirically confirmed by the "w/o PPO" ablation in Table 2, which retains accuracy on MATH500 within a few points of the original Thinking model across all sizes.

Remark. Stage 1 only installs the *capacity* to skip reasoning conditional on $\mathcal{T}_{\text{NT}}$; it does not yet learn *when* skipping is appropriate. Adaptive mode selection requires the entropy-regularized reward in Stage 2.

4.3 Stage 2: Offline Proximal Policy Optimization

Stage 2 refines the model's behavior to adaptively select between NoThinking and Thinking via offline PPO with a dual-mode entropy-regularized reward. Detailed training configuration is in Appendix B.5.

Offline data collection. The Stage-1 model is taken as the reference policy π_{ref}. For each problem x in the training set, we sample K responses $\{y_1, \ldots, y_K\}$ from π_{ref}, yielding $\mathcal{D}_{\text{off}} = \{(x, y_i) \mid x \in \mathcal{D},\, i = 1, \ldots, K\}$.

Reward with advantage baseline. For each (x, y), we form a base reward

$$R_{\text{base}}(x, y) = w_1 R_{\text{acc}}(y) + w_2 R_{\text{eff}}(x, y) + w_3 R_{\text{ent}}(x, y). \tag{8}$$

Accuracy (R_{acc}): $R_{\text{acc}}(y) = \mathbb{I}(\text{is_correct}(y))$.
Efficiency (R_{eff}): penalty for exceeding the average reference length $\bar{L}_{\text{ref}}(x)$ (pre-computed from the K samples):

$$R_{\text{eff}}(x, y) = -\max\left(0,\ \frac{\text{len}(y)}{\bar{L}_{\text{ref}}(x)} - 1\right). \tag{9}$$

Dual-mode entropy regularization (R_{ent}): the core of DUET,

$$R_{\text{ent}}(x, y) = \mathbb{I}(\text{is_correct}(y)) \cdot R_E(x, y), \tag{10}$$

where

$$R_E(x, y) = \begin{cases} 1 - H_{\text{avg}}(y) & \text{if } y \text{ is NoThinking}, \\ H_{\text{avg}}(y) & \text{if } y \text{ is Thinking}. \end{cases}$$

For NoThinking responses, $1 - H_{\text{avg}}(y)$ rewards deterministic outputs; for Thinking responses, $H_{\text{avg}}(y)$ rewards exploratory entropy. The accuracy gate $\mathbb{I}(\text{is_correct}(y))$ ensures mode selection emerges from joint optimization: if a problem is simple, NoThinking can yield both high accuracy and high R_E; if a problem is hard, low-entropy direct answers tend to be wrong, driving the model toward Thinking.

To reduce variance, we subtract a per-question baseline computed by averaging R_{base} over the K reference samples:

$$\bar{R}_{\text{ref}}(x) = \frac{1}{K} \sum_{i=1}^{K} R_{\text{base}}(x, y_i'), \quad y_i' \sim \pi_{\text{ref}}(\cdot \mid x), \tag{11}$$

Algorithm 1 DUET: Dual-Mode Entropy-Regularized Training

Require: Reference π_{ref}; training set $\mathcal{D}$; NoThinking template $\mathcal{T}_{\text{NT}}$; samples K; entropy threshold θ; weights (w_1, w_2, w_3); clip ϵ
Ensure: Trained policy π_θ
// Stage 1: SFT on self-generated NoThinking targets
$\mathcal{D}_{\text{SFT}} \leftarrow \emptyset$
for each $x \in \mathcal{D}$ **do**
  Sample $\{y_1, \ldots, y_K\} \sim \pi_{\text{ref}}(\cdot \mid x, \mathcal{T}_{\text{NT}})$
  $\mathcal{Y}^+ \leftarrow \{y_i : \text{is_correct}(y_i)\}$
  if $\mathcal{Y}^+ \neq \emptyset$ **then**
    $y^\star(x) \leftarrow \arg\min_{y \in \mathcal{Y}^+} |y|$; $\mathcal{D}_{\text{SFT}} \leftarrow \mathcal{D}_{\text{SFT}} \cup \{(x, y^\star(x))\}$
  end if
end for
Train π_θ on $\mathcal{D}_{\text{SFT}}$ minimizing $-\log \pi_\theta(y^\star \mid x, \mathcal{T}_{\text{NT}})$
$\pi_{\text{ref}} \leftarrow \pi_\theta$ ▷ Stage-1 model becomes new reference
// Stage 2: Offline PPO with dual-mode entropy reward
$\mathcal{D}_{\text{off}} \leftarrow \emptyset$
for each $x \in \mathcal{D}$ **do**
  Sample $\{y_1, \ldots, y_K\} \sim \pi_{\text{ref}}(\cdot \mid x)$
  $\bar{L}_{\text{ref}}(x) \leftarrow \frac{1}{K}\sum_i |y_i|$
  for $i = 1, \ldots, K$ **do**
    Compute $H_{\text{avg}}(y_i)$ at threshold θ via Eq. (1)
    $R_{\text{base}}(x, y_i) \leftarrow w_1 R_{\text{acc}} + w_2 R_{\text{eff}} + w_3 R_{\text{ent}}$
  end for
  $\bar{R}_{\text{ref}}(x) \leftarrow \frac{1}{K}\sum_i R_{\text{base}}(x, y_i)$
  $\mathcal{D}_{\text{off}} \leftarrow \mathcal{D}_{\text{off}} \cup \{(x, y_i, R_{\text{base}}(x, y_i) - \bar{R}_{\text{ref}}(x))\}_{i=1}^{K}$
end for
Train π_θ on $\mathcal{D}_{\text{off}}$ via clipped PPO objective $L^{\text{OFF-PPO}}$
return π_θ

giving the advantage $R_{\text{adv}}(x, y) = R_{\text{base}}(x, y) - \bar{R}_{\text{ref}}(x)$.

Off-policy PPO objective. With probability ratio $r(\theta) = \pi_\theta(y \mid x) \,/\, \pi_{\text{ref}}(y \mid x)$, the clipped PPO objective is

$$L^{\text{OFF-PPO}}(\theta) = -\,\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{off}}}\Big[\min\big(r(\theta)R_{\text{adv}}, \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\, R_{\text{adv}}\big)\Big], \tag{12}$$

where ϵ bounds policy drift (Schulman et al., 2015). All rollouts are prepared up-front, so no online sampling is needed.

5 Experiments

5.1 Experimental Setup

Models. We evaluate three open-source reasoning LLMs from the DeepSeek-AI et al. (2025) family with different sizes and base architectures: *DeepSeek-R1-Distill-Qwen-1.5B/7B* and *DeepSeek-R1-Distill-Llama-8B*, covering both Qwen and Llama families. Complete generation hyperparameters and prompting templates are in Appendix B.

Datasets. We train on the MATH (Hendrycks et al., 2021b) training split and evaluate on three in-domain/closely related reasoning benchmarks—MATH500 (Lightman et al., 2024), AIME24, and GPQA (Rein et al., 2024)—plus MMLU (Hendrycks et al., 2021a) as an out-of-distribution benchmark.

Baselines. We compare against seven representative methods: (1) **NoThinking** (Ma et al., 2025a), prompt injection that forces direct answering; (2) **Fast-Solving Prompt**, instructing the model to solve as fast as possible; (3) **SFT**, where the shortest correct solution per problem is used as a self-distillation target; (4) **DPO** (Rafailov et al., 2023), with two shortest correct solutions as chosen and the longest as rejected; (5) **O1-Pruner** (Luo et al., 2025), off-policy PPO with a length-aware reward; (6) **DAST** (Shen et al., 2025), a preference optimization method that estimates question difficulty via pass rate, derives a token length budget, and applies a length penalty when ranking preference data to achieve adaptive slow thinking; and (7) **AdaptThink** (Zhang et al., 2025), online RL with a constrained objective for CoT/direct switching. DAST's official repository only contains the 7B/32B checkpoint, which we adopt 7B for evaluation. AdaptThink only releases checkpoints for the 1.5B and 7B models; the 8B cell is left empty.

Evaluation Metric. We report Pass@1 accuracy via the Open-R1 framework (Hugging Face, 2025) and the number of generated output tokens as Length. Training configurations are in Appendix B.5.

5.2 Main Results

As shown in Table 1, DUET achieves an attractive accuracy-length trade-off. For DeepSeek-1.5B, it improves average accuracy by 3.0% while reducing response length by 47.5%. DPO also improves accuracy but with much smaller length reduction (9.2% vs. DUET's 47.5%). Aggressive length-reduction baselines (Orig_{NT}, Fast-Solving) cut tokens at the cost of large accuracy drops (e.g., -28.6% on DeepSeek-8B for Orig_{NT}). All results follow the standard Pass@1 protocol under temperature-0.6 sampling; per-problem effect sizes on the small AIME24 set (30 problems) should be interpreted accordingly. The case study in Appendix C shows 1.5B switching from a wrong NoThinking answer (426) to a correct Thinking answer (540) on a hard combinatorics problem after PPO—a concrete instance of adaptive selection.

Table 1: Main results. Acc=accuracy (%), Len=response length (tokens). Δ is the gap vs. the original Thinking model. Best non-original entries are in **bold**.

Method	MATH500		AIME24		GPQA		Avg.	
	Acc	Len	Acc	Len	Acc	Len	ΔA	ΔL
DeepSeek-R1-Distill-Qwen-1.5B								
Orig_{T}	85.2	5461	23.3	17231	39.4	9405	-	-
Orig_{NT}	70.4	944	10.0	2635	13.6	782	-18.0	-86.4%
Fast-Solving	78.2	3344	26.7	14424	36.9	9312	-2.0	-18.7%
SFT	80.6	4965	23.3	17575	32.8	10322	-3.7	+0.9%
DPO	**84.8**	4816	**33.3**	14219	34.8	9556	+1.7	-9.2%
O1-Pruner	82.8	4635	30.0	16841	37.9	8869	+0.9	-7.7%
AdaptThink	84.4	**1985**	30.0	9192	35.3	7854	+0.6	-42.3%
Ours	84.2	3257	**33.3**	**7250**	**39.4**	**5260**	**+3.0**	**-47.5%**
DeepSeek-R1-Distill-Qwen-7B								
Orig_{T}	93.2	3956	56.7	12102	49.5	8292	-	-
Orig_{NT}	79.6	723	23.3	2921	12.1	714	-28.1	-83.0%
Fast-Solving	88.8	2738	50.0	12910	**53.0**	7992	-2.5	-9.2%
SFT	89.8	3912	43.3	9799	48.0	10437	-6.1	+1.9%
DPO	94.0	3227	56.7	9514	52.0	9056	+1.1	-10.2%
O1-Pruner	**94.8**	3895	63.3	11401	49.5	8731	+2.7	-0.7%
DAST	93.6	3309	**70.0**	10804	51.5	7684	**+5.2**	-11.5%
AdaptThink	92.6	**2032**	56.7	**9214**	51.5	7336	+0.5	**-28.0%**
Ours	92.4	3105	63.3	10230	49.0	**6626**	+1.8	-19.0%
DeepSeek-R1-Distill-Llama-8B								
Orig_{T}	85.0	4390	46.7	13717	50.0	8977	-	-
Orig_{NT}	65.8	964	13.3	5961	16.7	1283	-28.6	-73.4%
Fast-Solving	78.2	3194	40.0	12029	48.0	9069	-5.2	-12.8%
SFT	75.2	**3170**	40.0	13493	47.0	8815	-6.5	-10.4%
DPO	**88.4**	4030	50.0	12874	**51.5**	8515	+2.7	-6.5%
O1-Pruner	86.6	3821	53.3	11344	48.5	8537	+2.2	-11.7%
AdaptThink	*checkpoint unavailable*						–	–
Ours	85.8	3564	**56.7**	**10126**	49.0	**7135**	**+3.3**	**-21.8%**

Table 2: Ablation results. "w/o PPO" uses Stage-1 SFT only; "w/o R_{ent}" drops the dual-mode entropy reward.

Variant	MATH500		AIME24		GPQA		Avg.	
	Acc	Len	Acc	Len	Acc	Len	ΔA	ΔL
DeepSeek-R1-Distill-Qwen-1.5B								
Full	**84.2**	3257	**33.3**	**7250**	**39.4**	**5260**	-	-
w/o PPO	80.2	4116	26.7	10821	31.3	7114	-6.2	+36.9%
w/o R_{ent}	80.8	**2765**	26.7	8489	30.8	5522	-6.2	+2.3%
DeepSeek-R1-Distill-Qwen-7B								
Full	**92.4**	3105	**63.3**	**10230**	**49.0**	6626	-	-
w/o PPO	91.2	3259	53.3	12156	48.5	**6516**	-3.9	+7.4%
w/o R_{ent}	88.6	**2807**	53.3	11992	45.5	6588	-5.8	+2.4%
DeepSeek-R1-Distill-Llama-8B								
Full	85.8	**3564**	**56.7**	**10126**	**49.0**	**7135**	-	-
w/o PPO	**88.0**	4557	43.3	13155	47.0	7666	-4.4	+21.7%
w/o R_{ent}	86.0	3582	43.3	11146	43.9	7749	-6.1	+6.4%

5.3 Ablation Study

Table 2 isolates the contribution of each component. Removing offline PPO ("w/o PPO") consistently inflates response length across all three backbones (+7.4% to +36.9%); its effect on accuracy is model-dependent: aggregate accuracy drops by 3.9–6.2% on the 1.5B and 7B backbones, while on the 8B backbone MATH500 accuracy is actually 2.2 points higher than the full model but AIME24 and GPQA drop sharply (43.3 vs. 56.7 on AIME24), yielding a -4.4 aggregate drop. This pattern indicates that Stage 2's primary effect is mode-selection refinement and length regularization rather than uniform accuracy improvement; the dataset-level accuracy gains are concentrated on problems requiring deliberation. Removing the dual-mode entropy reward ("w/o R_{ent}") yields the largest accuracy drop in two of three models, confirming that entropy regularization—rather than length pressure alone—is the key driver of accurate mode selection. The accuracy gate in (10) prevents the optimizer from inflating entropy without solving the problem.

Table 3: Out-of-distribution generalization on MMLU.

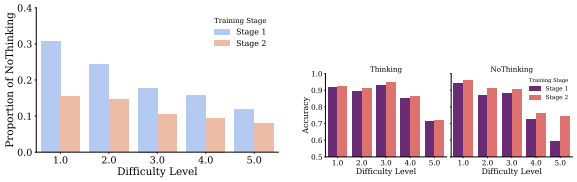
Method	MMLU			
	Acc	Len	ΔAcc	ΔLen
DeepSeek-R1-Distill-Qwen-1.5B				
$\text{Original}_{\text{Think}}$	39.4	2241	-	-
Ours	**41.7**	**778**	**+2.3**	**-65.3%**
DeepSeek-R1-Distill-Qwen-7B				
$\text{Original}_{\text{Think}}$	**63.7**	1462	-	-
Ours	61.5	**881**	-2.2	**-39.7%**
DeepSeek-R1-Distill-Llama-8B				
$\text{Original}_{\text{Think}}$	69.7	1891	-	-
Ours	**70.2**	**1461**	**+0.5**	**-22.7%**

5.4 Out-of-Distribution Generalization on MMLU

A critical question is whether the adaptive mode-selection behavior, trained only on mathematics, transfers to other domains. We evaluate on MMLU, a broad multi-task benchmark spanning humanities, STEM and social sciences—a substantial distribution shift from the MATH training data. Table 3 shows that DUET maintains comparable accuracy on the 1.5B (+2.3%) and 8B (+0.5%) models with substantial length reductions (1.5B: -65.3%, 8B: -22.7%). For 7B, accuracy drops by 2.2%, partly because the model is already near saturation on MMLU; length still drops by -39.7%. The pattern suggests our regularization shapes a fairly domain-general meta-cognitive strategy—"be deterministic when confident, exploratory when uncertain"—rather than purely mathematical heuristics. We caveat that this transfer is across math/STEM-style multiple-choice settings; open-ended natural-language tasks are left for future evaluation.

5.5 Mechanistic Analysis

To understand *why* entropy regularization works, we run a controlled analysis on DeepSeek-8B with 8 rollouts per MATH500 problem (Figure 4). After Stage 1 SFT alone, the model acquires the ability to skip Thinking but cannot match the choice to problem complexity: it both produces verbose answers on easy problems and over-skips on hard



(a) NoThinking proportion (b) Accuracy

Figure 4: Mechanistic analysis on DeepSeek-8B (8 rollouts/sample on MATH500), comparing the original Thinking model, Stage-1 SFT only, and our full method across difficulty levels.

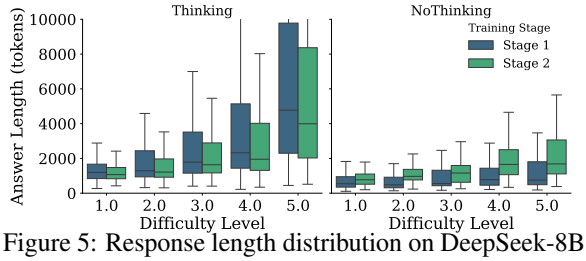


Figure 5: Response length distribution on DeepSeek-8B across difficulty levels. Thinking-mode lengths shrink sharply on hard problems after training, while NoThinking lengths remain compact—the efficiency gain comes from adaptive token-budget allocation within Thinking rather than indiscriminate mode conversion.

ones. After Stage 2 PPO with the dual-mode entropy reward, the overall NoThinking proportion adjusts to a regime that better tracks per-problem difficulty (panel a), and both modes' accuracy improves (panel b)—confirming that the gain is not a generic shortening pressure but a redistribution of computational budget.

To pinpoint the efficiency source, Figure 5 decomposes response length by mode. Thinking-mode lengths on hard problems contract sharply post-training, while NoThinking lengths stay compact across all levels—i.e., the primary efficiency driver is *learned conciseness within Thinking responses on hard problems*, not indiscriminate Thinking-to-NoThinking conversion. This is exactly what our reward in (10) incentivizes: the model is rewarded for high entropy *conditional on correctness*, discouraging padded Thinking traces with low-entropy filler. The accuracy gate further blocks the "high-entropy-but-wrong" shortcut—a hedging-only response would inflate H_{avg} but receive zero reward when it fails the correctness check. Together, these effects explain the favorable trade-offs in Tables 1 and 3 as a learned, information-theoretic mode-selection policy rather than blind shortening.

5.6 Hyperparameter Sensitivity

The entropy threshold τ in Eq. (1) controls which tokens contribute to H_{avg} and thus shapes the entropy reward signal. Table 4 sweeps τ on DeepSeek-R1-Distill-Llama-8B. Without filtering (τ=0.0), formulaic low-entropy tokens dilute the signal, leading to lower accuracy on AIME24 (40.0) and GPQA (44.4). Setting τ=0.2 yields the best AIME24 (56.7) and GPQA (49.0) scores with competitive MATH500 accuracy and the second-shortest length, striking the best overall balance. Although τ=0.5 achieves slightly higher MATH500 accuracy (87.6), it degrades AIME24 performance, indicating that overly aggressive filtering discards useful entropy information. We therefore adopt τ=0.2 as the default and recommend re-tuning on a small held-out set when applying DUET to substantially different domains.

Table 4: Entropy threshold sensitivity on DeepSeek-R1-Distill-Llama-8B. τ=0.2 achieves the best overall trade-off.

τ	MATH500		AIME24		GPQA	
	Acc	Len	Acc	Len	Acc	Len
0.0	85.4	4139	40.0	11071	44.4	8260
0.01	85.6	4117	40.0	12724	48.0	8439
0.1	86.6	3705	50.0	12360	47.5	7894
0.2	85.8	3564	**56.7**	**10126**	**49.0**	**7135**
0.5	**87.6**	**3461**	40.0	11380	45.0	7757

6 Conclusion

We presented **DUET**, a dual-mode entropy-regularized training framework that addresses the overthinking problem of large reasoning models from an information-theoretic perspective. Our analysis establishes that the two reasoning modes of LLMs—direct answering and deliberative reasoning—occupy distinct entropy regimes, and we show that this signal can be turned into an actionable training objective. The SFT stage on self-generated NoThinking targets installs the abbreviated-reasoning capacity with a bounded-KL update that preserves the reference policy; the offline PPO stage with a correctness-gated, mode-dependent entropy reward then shapes *when* each mode is invoked. Across MATH500, AIME24, GPQA and MMLU, DUET reduces response length by up to 47.5% while preserving or improving accuracy, without inference-time difficulty assessment or auxiliary modules. The mechanistic analysis suggests the efficiency gain stems from a learned redistribution of computational budget rather than blind shortening, offering a principled and lightweight path toward more efficient inference in large reasoning models.

Limitations

DUET has several limitations. First, our empirical validation covers mathematical reasoning, with transfer evaluated on GPQA (graduate STEM) and MMLU (multi-task multiple-choice); generalization to open-ended natural-language generation, creative writing, ethical reasoning, or multi-turn dialogue—where the entropy–difficulty correlation may need recalibration—remains unexplored. Second, the dual-mode formulation is binary: the policy chooses between NoThinking and Thinking but cannot continuously vary reasoning depth, whereas in practice problems span a spectrum that may benefit from finer-grained mode choices or hybrid traces. Third, entropy is an imperfect proxy for reasoning quality, and the correctness gate in R_{ent}, while empirically effective, does not eliminate all reward-hacking pathways; a process reward model that scores per-step reasoning quality would be a natural extension. Finally, our experiments are confined to models up to 8B parameters and to the DeepSeek-R1 distilled family; how the adaptive mode policy scales to substantially larger reasoning models, and whether the dual-mode separation persists under different RL recipes (e.g., GRPO, online PPO, or process rewards), are open questions left to future work.

Ethics Statement

This work proposes a training framework to improve the efficiency of reasoning in large language models. All experiments are conducted on publicly available, open-source models (DeepSeek-R1-Distill family) and publicly available benchmarks (MATH, MATH500, AIME24, GPQA, MMLU). Our method does not involve the collection or use of personally identifiable data, nor does it generate harmful or offensive content. The primary intended benefit is reducing the computational cost of LLM inference, which has positive implications for energy consumption and accessibility. We do not foresee significant ethical risks arising from this work, though adaptive reasoning policies should be evaluated for safety-critical applications before deployment.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *Preprint*, arXiv:2412.13171.

Herbert H. Clark and Jean E. Fox Tree. 2002. Using uh and um in spontaneous speaking. *Cognition*, 84(1):73–111.

Martin Corley and Oliver W. Stewart. 2008. Hesitation disfluencies in spontaneous speech: The meaning of um. *Language and Linguistics Compass*, 2(4):589–602.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. 2025. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill? *Preprint*, arXiv:2504.06514.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. Concise reasoning via reinforcement learning. *Preprint*, arXiv:2504.05185.

T. Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61(1):23–62.

Arella E. Gussow. 2023. Chapter three - language production under message uncertainty: When, how, and why we speak before we think. In Kara D. Federmeier and Jessica L. Montag, editors, *Speaking, Writing and Communicating*, volume 78 of *Psychology of Learning and Motivation*, pages 83–117. Academic Press.

John Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science*, 30(4):643–672.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. *Preprint*, arXiv:2412.18547.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *Preprint*, arXiv:2412.06769.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

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

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Daniel Kahneman. 2011. Thinking, fast and slow. *Farrar, Straus and Giroux*.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. *Preprint*, arXiv:2503.01141.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhijiang Guo, and 2 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *Preprint*, arXiv:2502.17419.

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

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025a. Reasoning models can be effective without thinking. *Preprint*, arXiv:2504.09858.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025b. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2024. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`. [Accessed 19-05-2026].

Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

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

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of

Proceedings of Machine Learning Research, pages 627–635, Fort Lauderdale, FL, USA. PMLR.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France. PMLR.

Murray Shanahan. 2024. Talking about large language models. *Commun. ACM*, 67(2):68–79.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *Preprint*, arXiv:2503.04472.

Jovan Stojkovic, Chaojie Zhang, Íñigo Goiri, Josep Torrellas, and Esha Choukse. 2025. Dynamollm: Designing llm inference clusters for performance and energy efficiency. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 1348–1362.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. 2025. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *Preprint*, arXiv:2505.00127.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Chao Wu, Baoheng Li, Mingchen Gao, Yu Tian, and Zhenyi Wang. 2026. From efficiency to adaptivity: A deeper look at adaptive reasoning in large language models. *Preprint*, arXiv:2511.10788.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Jingyang Yi, Jiazheng Wang, and Sida Li. 2025. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *Preprint*, arXiv:2504.21370.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. 2025. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? *Preprint*, arXiv:2502.12215.

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

Yu Zhao, Huiming Liu, Lai Wei, Junfeng Guo, Dandan Tu, Zhenyu Cai, Yipei Wang, and Xi Chen. 2024. Marco-o1: Towards open reasoning models for open-ended solutions. *Preprint*, arXiv:2411.14405.

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

Yuqi Zhu, Ge Li, Xue Jiang, Jia Li, Hong Mei, Zhi Jin, and Yihong Dong. 2025. Uncertainty-guided chain-of-thought for code generation with llms. *Preprint*, arXiv:2503.15341.

A Entropy Theory and Analysis

A.1 Justification for $D = L_s - A_s$ as a Difficulty Proxy

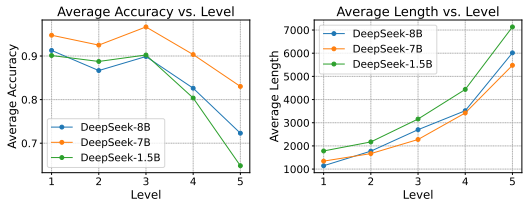


Figure 6: Accuracy and length across difficulty levels on MATH500.

Consider a problem set $\mathcal{Q}$ with inherent difficulty $d : \mathcal{Q} \to \mathbb{R}^+$, and for each problem $q \in \mathcal{Q}$ let $a(q) \in [0, 1]$ denote accuracy and $l(q) \in \mathbb{R}^+$ denote response length. Under our framework the model minimizes

$$\mathcal{L}(\theta) = \mathbb{E}_{q\sim\mathcal{Q}} \left[l(q) - \lambda \cdot a(q) \right], \qquad (13)$$

with $\lambda > 0$ trading off length and accuracy.

Assumption 1 (length monotonicity). For an optimal model under $\mathcal{L}$, $\mathbb{E}[l(q) \mid d(q) = d_1] < \mathbb{E}[l(q) \mid d(q) = d_2]$ whenever $d_1 < d_2$.

Assumption 2 (accuracy monotonicity). $\mathbb{E}[a(q) \mid d(q) = d_1] > \mathbb{E}[a(q) \mid d(q) = d_2]$ whenever $d_1 < d_2$.

Figure 6 verifies both assumptions empirically: across DeepSeek-1.5B/7B/8B on MATH500, accuracy decreases and length increases with annotated difficulty.

Proposition 1 (Monotonicity of D).[1] Under Assumptions 1–2, the standardized difference $D(q) = L_s(q) - A_s(q)$ is monotonically increasing in $d(q)$, where $L_s = (l - \mu_l)/\sigma_l$ and $A_s = (a - \mu_a)/\sigma_a$.

Proof. For $d(q_1) < d(q_2)$,

$$D(q_2) - D(q_1) = \frac{1}{\sigma_l}[l(q_2) - l(q_1)] - \frac{1}{\sigma_a}[a(q_2) - a(q_1)].$$

By Assumption 1, $l(q_2) > l(q_1)$, so the first term is positive; by Assumption 2, $a(q_2) < a(q_1)$, so the second term is negative. The difference is therefore strictly positive. □

Corollary 1. Under $\mathcal{L}$, problems with $D(q) < 0$ tend to be easier and benefit from NoThinking responses, while $D(q) > 0$ flags problems requiring deliberation. We caveat that since $l(q)$ depends on the model's evolving generation behavior, D has a mild circularity: the metric is computed under the reference model π_{ref} before training begins, so during PPO training D is held fixed and the circularity does not affect optimization.

[1] We use "Proposition" rather than "Theorem" to reflect the modest scope of this result: it is a direct consequence of the two monotonicity assumptions above, not a deep theoretical claim.

A.2 MATH500 Entropy Analysis

To validate that token-level entropy serves as an effective complexity indicator, we sample the MATH500 dataset 8 times and filter tokens with entropy below 0.1 to focus on meaningful reasoning content.

A.2.1 Entropy Distribution Patterns

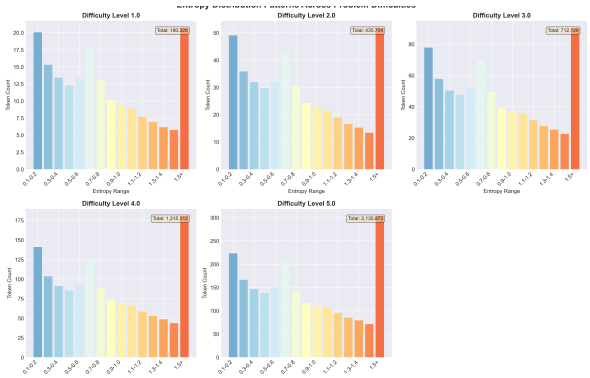


Figure 7: Entropy distribution patterns across problem difficulties.

Figure 7 illustrates the entropy distribution across five difficulty levels. The proportion of low-entropy tokens (0.1–0.5) decreases from 26.8% at level 1.0 to 24.6% at level 5.0, while high-entropy tokens (>1.0) expand from 25.2% to 27.7%.

A.2.2 Mechanisms Behind Entropy Increase

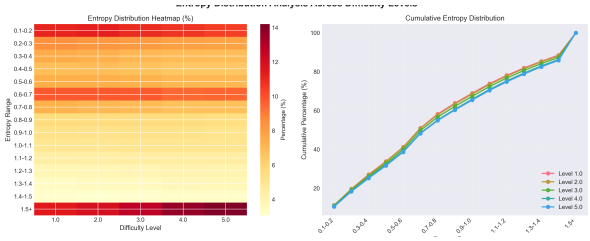


Figure 8: Entropy heatmap and cumulative trends.

Increased decision complexity. Figure 8 reveals a clear rightward shift of probability mass. Simple problems (difficulty 1.0) exhibit deterministic patterns with confident next-token predictions; complex problems (difficulty 5.0) require considering multiple reasoning paths, raising prediction uncertainty.

Reasoning mode transition. Figure 9 demonstrates a shift in reasoning modes. Low-difficulty

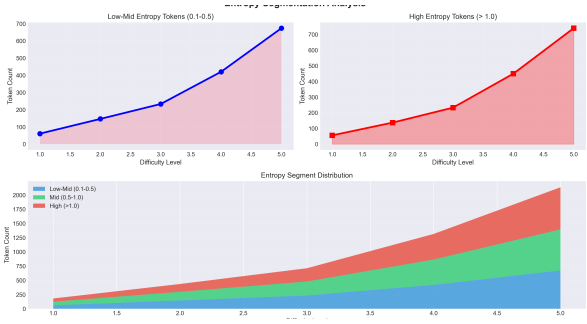


Figure 9: Entropy segmentation analysis.

problems predominantly employ pattern matching with low-mid entropy (0.1–0.5); high-difficulty problems require exploratory reasoning with high entropy (>1.0). The high-entropy segment expands 12.7-fold in absolute count and nearly doubles in proportion.

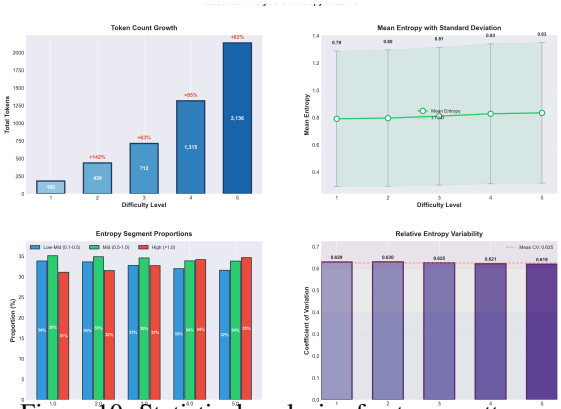


Figure 10: Statistical analysis of entropy patterns.

Statistical evidence. Mean entropy rises by 6.3% from difficulty 1.0 to 5.0, while the coefficient of variation drops from 0.629 to 0.619, indicating consistent rather than random uncertainty growth.

A.2.3 High-Frequency Token Analysis

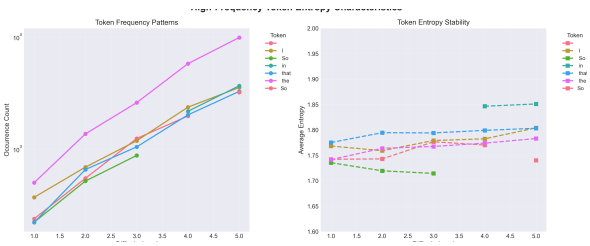


Figure 11: High-frequency token analysis. **Left:** log-scale frequency. **Right:** entropy stability across difficulty levels.

Core reasoning tokens such as "the", "I" and "that" maintain stable entropy (1.77–1.90) across difficulties, while their occurrence count grows roughly 35-fold from difficulty 1.0 to 5.0 (Figure 11). This stability-frequency paradox suggests:

1. The model's reasoning vocabulary is largely invariant across complexity;
2. Difficulty manifests through more frequent decision points rather than higher per-token uncertainty;
3. Entropy growth is an aggregate effect.

A.3 Cross-Dataset Entropy Analysis

We extend the analysis across three datasets of increasing difficulty: GSM8K (elementary arithmetic), MATH500 (high school), and AIME24 (competition).

A.3.1 Entropy Distribution Evolution

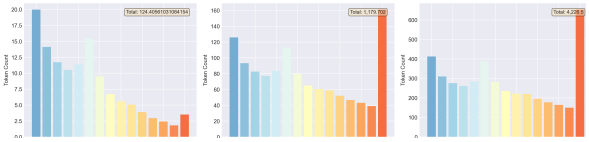


Figure 12: Entropy distribution patterns across dataset complexities.

Figure 12 reveals that for GSM8K, 45% of tokens fall in the low-mid range (0.1–0.5); this drops to 32% for MATH500 and 30% for AIME24, while high-entropy tokens (> 1.0) rise from 16% to 37%.

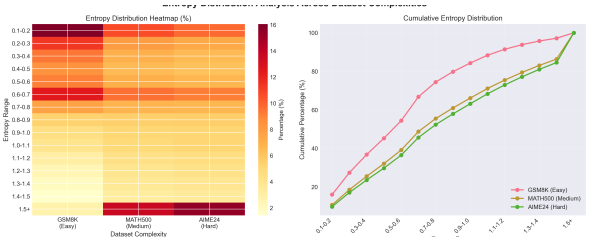


Figure 13: Cross-dataset entropy heatmap and cumulative trends.

The heatmap (Figure 13) shows a clear rightward shift across datasets.

A.3.2 Quantitative Complexity Indicators

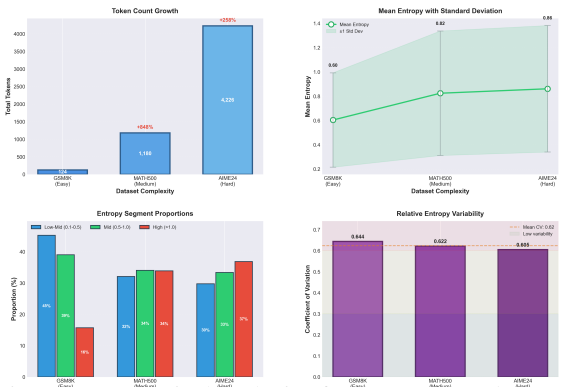


Figure 14: Statistical analysis of entropy across datasets.

Mean entropy: GSM8K → 0.60, MATH500 → 0.82 (+36.7%), AIME24 → 0.86 (+43.3%). Total meaningful tokens grow 34.1× from GSM8K to AIME24.

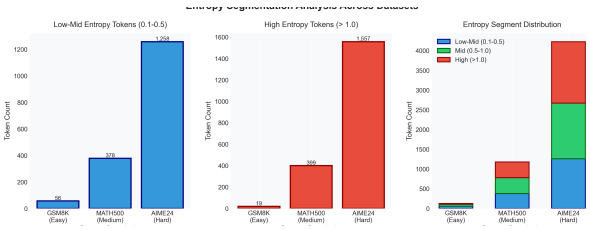


Figure 15: Cross-dataset entropy segmentation.

A.3.3 Token-Level Behavioral Consistency

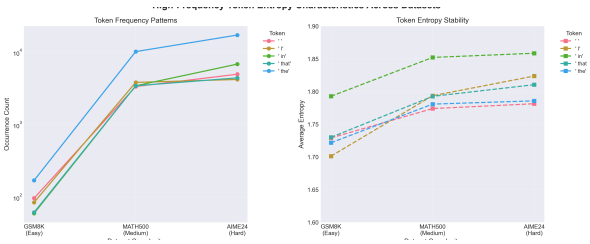


Figure 16: Cross-dataset high-frequency token characteristics.

Core reasoning tokens ("the", "I", "that") maintain entropy values between 1.72 and 1.87 across all datasets, while frequencies vary by up to 100×. This consistency further supports that difficulty manifests through decision-point density rather than per-token uncertainty.

A.3.4 Implications and Caveats

The cross-dataset monotonic rise in mean entropy (0.60 → 0.86) and the consistent natural breakpoints at 0.5 and 1.0 provide reliable signals for adaptive reasoning across mathematical domains. We caveat that entropy ranges are computed under a specific tokenizer and model family; absolute thresholds may shift across model architectures, so for new domains we recommend recalibrating using the procedure in Appendix A.1.

B Experiment Setup

B.1 Datasets

We use MATH (Hendrycks et al., 2021b) (∼10K problems of high school level) for training, selecting 5,000 problems from MATH Train set and generating 16 solutions per problem. We evaluate on:

- **MATH500** (Lightman et al., 2024): 500 problems sampled from MATH, covering algebra, calculus, geometry, and number theory across difficulty levels.
- **AIME24**: problems from the American Invitational Mathematics Examination 2024, a challenging high-school competition (30 problems).
- **GPQA** (Rein et al., 2024) (Diamond): graduate-level STEM reasoning tasks.

B.2 Generation Configuration

We use DeepSeek-R1-Distill models with: temperature 0.6, top-p 0.95, and maximum sequence length 32,768 tokens.

B.3 NoThinking Prompt

We use prompt injection to make the model skip explicit reasoning:

```
Okay, I think I have finished
reasoning.  \n</think>\n\n
```

B.4 Entropy-Based Complexity Assessment

We use entropy threshold $\tau = 0.2$ on MATH500. Token-level entropy is computed from the top-10 token probabilities (renormalized within this subset); the 0.2 threshold filters out low-information tokens when averaging across the generated sequence.

B.5 Training Configuration

Stage 1: SFT. Learning rate 5.0e-6, 3 epochs, cutoff length 8,192 tokens, 1,500 NoThinking exemplars, ∼10 minutes on 2 A100 80GB GPUs.

Stage 2: Offline PPO. Learning rate 5.0e-7, 1 epoch, cutoff length 12,288 tokens, 10,000 problem instances, ∼1.5 hours on 4 A100 80GB GPUs. We use reward weights $w_1 = 0.3$ (accuracy), $w_2 = 0.5$ (efficiency), $w_3 = 0.3$ (entropy regularization), and $K = 16$ candidates per question for offline data collection and baseline averaging.

On reward weights. We choose the best-performed reward weights w_1, w_2, w_3 in $(0.3, 0.5, 0.3)$. We performed a coarse grid search over $w_i \in \{0.1, 0.3, 0.5, 0.7\}$ on the DeepSeek-1.5B model and selected the configuration achieving the best trade-off between accuracy and length on the MATH validation split; full sensitivity tables are left to future work due to compute constraints.

On the range of H_{avg}. For NoThinking responses, $R_E = 1 - H_{\text{avg}}(x)$ can become negative when $H_{\text{avg}} > 1$. In our setting this occurs rarely because the indicator $\mathbb{I}(\text{is_correct}(y))$ gates the entire term, so a negative R_E is incurred only on *correct* NoThinking responses with unusually high uncertainty. This is the intended behavior: such cases are not really "confident direct answers" and the reward penalizes them.

Reward hacking considerations. Entropy is an imperfect proxy for reasoning quality, and one might worry the model could inflate H_{avg} on Thinking responses with hedging tokens ("hmm," "perhaps") or template low-entropy responses on NoThinking. Two design choices mitigate this in our experiments: (1) the accuracy gate $\mathbb{I}(\text{is_correct}(y))$ ensures entropy rewards apply only to correct responses, blocking the "high-entropy but wrong" path; (2) PPO clipping bounds the policy-update magnitude away from π_{ref}. Empirically we observe accuracy improvements (not just entropy inflation), suggesting egregious hacking does not occur in practice; nevertheless, a process reward model that scores reasoning step quality is a natural extension.

C Case Study

We select representative MATH500 problems by difficulty: an "easy" problem (level 2.0) and a "hard" problem (level 5.0). The case study illustrates how 1.5B model switches reasoning mode appropriately after PPO—in particular, on the hard combinatorics problem, the pre-PPO model used NoThinking and arrived at the wrong answer (426), while the post-PPO model switched to Thinking mode and arrived at the correct answer (540). This is a concrete instance of the adaptive mode selection observed in aggregate in Figure 4.

Figure 17 and Figure 18 show the prompt templates. For the easy problem, Figures 19 and 20 show that both 1.5B and 8B models learn to use NoThinking mode, generating concise solutions.

For the hard problem, Figures 21 and 22 illustrate how both models appropriately engage Thinking mode, employing extensive reasoning to solve the combinatorics problem. The 1.5B case is particularly noteworthy: the pre-PPO model used NoThinking and arrived at **426** (incorrect), while the post-PPO model switched to Thinking mode and arrived at **540** (correct).

Prompt Template (Easy)

System Prompt:
You are a helpful assistant. You should think step-by-step and put your final answer within `\boxed{{}}`.

Problem:
In a class of 50 students, 28 participate in MATHCOUNTS, 21 participate in science club, and 6 students participate in neither. How many students participate in both?

Figure 17: Easy prompt (MATH500 difficulty 2.0).

Prompt Template (Hard)

System Prompt:
You are a helpful assistant. You should think step-by-step and put your final answer within `\boxed{{}}`.

Problem:
Six cars pull up to a red light, one at a time. At the light, there are three lanes: left-turn, straight, and right-turn. How many ways can the cars stack up so that all three lanes are occupied? The cars are distinguishable but pull up in a fixed order.

Figure 18: Hard prompt (MATH500 difficulty 5.0).

DeepSeek-R1-Distill-Qwen-1.5B

Before PPO:
`<think>`Okay, so I have this problem here: There are 50 students in a class. Out of these, 28 are in MATHCOUNTS, 21 are in the science club, and 6 are in neither. I need to find out how many students are in both MATHCOUNTS and the science club. Hmm, let me think. This seems like a problem involving sets and maybe the principle of inclusion-exclusion. I remember that when dealing with two overlapping sets, the total number of elements in either set is equal to the sum of the elements in each set minus the elements in their intersection. So, if I denote the number of students in both MATHCOUNTS and science club as x, then I can set up an equation. Let me write that down. The total number of students is 50. But 6 are in neither club, so the number of students in at least one of the clubs is $50 - 6 = 44$. That makes sense because 6 students are excluded from both. Now, according to the inclusion-exclusion principle, the number of students in MATHCOUNTS plus the number of students in the science club minus the number of students in both equals the number of students in at least one club. So, mathematically, that would be: $|M|+|S|-|M \cap S|$ = Total in at least one club Plugging in the numbers, that's: 28 (MATHCOUNTS) + 21 (science club) $- x$ (both) $= 44$ So, $28 + 21 - x = 44$ Adding 28 and 21 together gives 49, so: $49 - x = 44$ To find x, I subtract 44 from both sides: $49 - 44 = x$ Which simplifies to: $5 = x$ So, x is 5. That means 5 students are in both MATHCOUNTS and the science club. Wait, let me double-check. If 5 students are in both, then the number of students only in MATHCOUNTS would be $28 - 5 = 23$, and only in science club would be $21 - 5 = 16$. Adding those up: $23 + 16 + 5$ (both) $= 44$, which is exactly the number of students in at least one club. Adding the 6 students in neither gives $44 + 6 = 50$, which matches the total number of students in the class. So, that checks out. I think that's correct. There are 5 students who are in both clubs. `</think>`
To solve the problem, we can use the principle of inclusion-exclusion.
1. **Total students**: 50; 2. **Students in neither club**: 6; 3. **Students in at least one club**: 50 - 6 = 44. Let: $|M|$ = number of students in MATHCOUNTS = 28, $|S|$ = number of students in science club = 21, $|M \cap S|$ = number of students in both clubs = x (unknown).
Using the inclusion-exclusion principle: $|M \cup S| = |M| + |S| - |M \cap S|$, $44 = 28 + 21 - x$, $x = 5$.
Answer: The number of students participating in both MATHCOUNTS and the science club is $\boxed{5}$. **(800**

tokens)

After PPO:
<think>Okay, I think I have finished reasoning.</think>
To solve the problem, we'll use the principle of inclusion-exclusion. 1. **Total Students:** There are 50 students in the class. 2. **Participants in MATHCOUNTS:** 28 students. 3. **Participants in Science Club:** 21 students. 4. **Students in Neither:** 6 students. **Step 1:** Calculate the number of students who are in at least one of the clubs. Total participants = 50 - 6 = 44 students. **Step 2:** Apply inclusion-exclusion. Number in both clubs = 28 + 21 - 44 = 5 students. $\boxed{5}$ students participate in both MATHCOUNTS and the science club. **(219 tokens)**

Figure 19: For this straightforward problem, DeepSeek-1.5B exhibits unnecessary deliberation before PPO. After PPO, our model adaptively selects NoThinking, producing a concise solution with only 219 tokens.

DeepSeek-R1-Distill-Llama-8B

Before PPO:
<think>Okay, so I have this problem here about students participating in MATHCOUNTS and science club. Let me try to figure it out step by step. Hmm, let's see. The class has 50 students in total. Out of these, 28 participate in MATHCOUNTS, 21 participate in science club, and 6 students don't participate in either. I need to find how many students are in both MATHCOUNTS and science club.**[...](omit)** Alternatively, maybe using a Venn diagram. Imagine two overlapping circles, one for MATHCOUNTS with 28 students and one for science club with 21 students. The overlapping part is the number we're trying to find. The total number of students is 50, but 6 are not in either, so 50 - 6 = 44 are in at least one of the clubs. So, in the Venn diagram, the total area covered by both circles is 44. So, the area covered by MATHCOUNTS alone is 28 - Both, and the area covered by science club alone is 21 - Both. So, adding those gives $(28 - B) + (21 - B) + B = 49 - B$. But this sum should equal 44. So, $49 - B = 44$, so $B = 5$. Yep, same answer again. **Final Answer:** The number of students who participate in both MATHCOUNTS and science club is $\boxed{5}$. </think>
Given a class of 50 students, where 28 participate in MATHCOUNTS, 21 participate in science club, and 6 students participate in neither, we need to find the number of students who participate in both MATHCOUNTS and science club. We use the principle of inclusion-exclusion. The formula for the total number of students is: Total $=$ MATHCOUNTS $+$ Science Club $-$ Both $+$ Neither. Plugging in the given values: $50 = 28 + 21 - B + 6$. Simplifying the equation: $50 = 55 - B$. Solving for B: $B = 55 - 50 = 5$. Thus, the number of students who participate in both MATHCOUNTS and science club is $\boxed{5}$. **(1569 tokens)**

After PPO:
<think>Okay, I think I have finished reasoning.</think>
To solve the problem, we use the principle of inclusion-exclusion. Let Total students = 50, MATHCOUNTS = 28, Science Club = 21, Neither = 6. Students in at least one club = 50 - 6 = 44. Let x be the number in both. By inclusion-exclusion: $28 + 21 = 44 + x$, so $49 = 44 + x$ and $x = 5$. Therefore, the number of students who participate in both is $\boxed{5}$. **(284 tokens)**

Figure 20: DeepSeek-8B exhibits a significant reduction in tokens after selecting NoThinking mode.

DeepSeek-R1-Distill-Qwen-1.5B

Before PPO:
<think>Okay, I think I have finished reasoning.</think>
To solve the problem of determining the number of ways six distinguishable cars can stack up into three lanes (left-turn, straight-going, and right-turn) such that all three lanes are occupied, we can use the concept of permutations. 1. **Understanding the Problem**: - There are six cars, each distinguishable. - They pull up one at a time in a fixed order. - The cars must occupy all three lanes, meaning each lane must have at least one car. 2. **Total Permutations**: - The total number of ways to arrange six cars in order is 6! (6 factorial). - $6! = 720$. 3. **Subtracting Unoccupied Cases**: - We need to subtract the cases where one or more lanes are empty. - This can be calculated using the principle of inclusion-exclusion. 4. **Applying Inclusion-Exclusion Principle**: - Let A, B, and C represent the cases where the left-turn, straight-going, and right-turn lanes are empty, respectively. - The number of ways where at least one lane is empty is $|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$. 5. **Calculating Each Term**: - $|A| = |B| = |C| = 5! = 120$. - $|A \cap B| = |A \cap C| = |B \cap C| = 4! = 24$. - $|A \cap B \cap C| = 3! = 6$. 6. **Putting It All Together**: - Total ways with all lanes occupied: $6! - 3 \times 5! + 3 \times 4! - 3! = 720 - 360 + 72 - 6 = 426$. **Final Answer**: The number of ways the cars can stack up so that all three lanes are occupied is $\boxed{426}$. **(483 tokens)**

After PPO:
<think>Okay, I need to solve this problem about six cars pulling up to a red light with three lanes: left-turn, straight-going, and right-turn. All three lanes must be occupied, and the cars are distinguishable but pull up one by one in a fixed order. Hmm, let me think about how to approach this. **[...](omit)** Therefore, I think the correct answer is 540. </think>
To solve the problem, we use the principle of inclusion-exclusion to count the number of onto functions from six cars to three lanes: $\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n$ with $n = 6, k = 3$: $3^6 - 3 \cdot 2^6 + 3 \cdot 1^6 = 729 - 192 + 3 = 540$. Thus, the number of ways the cars can stack up so that all three lanes are occupied is $\boxed{540}$. **(2438 tokens)**

Figure 21: For the hard combinatorics problem, DeepSeek-1.5B before PPO used NoThinking and got **426 (wrong)**; after PPO it switched to Thinking mode and got **540 (correct)**. This is a concrete instance of the adaptive mode selection from Figure 4.

DeepSeek-R1-Distill-Llama-8B

Before PPO:
<think> Okay, I think I have finished reasoning. </think>
To solve the problem, let's break it down step by step. 1. **Understand the Problem:** - There are six cars that pull up to a red light one by one. - The light has three lanes: left-turn, straight-going, and right-turn. - Each car must choose one of the three lanes. - The order of the cars matters because they are distinguishable. 2. **Determine the Total Number of Possibilities:** - Each car has 3 choices: left, straight, or right. - Since the cars are distinguishable and choose lanes in sequence, this is a permutation problem with repetition allowed. - The total number of ways the cars can choose lanes is 3^6. 3. **Calculate the Total Number of Ways:** - $3^6 = 3 \times 3 \times 3 \times 3 \times 3 \times 3 = 729$. 4. **Conclusion:** - Therefore, there are $\boxed{729}$ ways the cars can stack up so that all three lanes are occupied. **(245 tokens)**

After PPO:
<think> Okay, so I have this problem here where six cars pull up to a red light one at a time. There are three lanes: left-turn, straight, and right-turn. I need to figure out how many ways the cars can stack up so that all three lanes are occupied. Hmm, let me break this down step by step. First, I think it's about permutations because the order matters here. The problem says that if the first car turns left and the second goes straight, that's different from the first going straight and the second turning left. So each car is distinguishable, but they all arrive in a fixed order. So, the order in which they arrive matters. **[...](omit)** So yes, the number is $3^6 - 3*2^6 + 3*1^6 = 729 - 192 + 3 = 540$. So, yeah, the final answer is 540. **Final Answer** The number of ways is $\boxed{540}$. </think>
To solve the problem of determining how many ways six cars can pull up to a red light such that all three lanes (left-turn, straight, and right-turn) are occupied, we need to consider the order of the cars and the lanes they choose. Each car is distinguishable and arrives one at a time, and the order matters. We can use the principle of inclusion-exclusion to count the number of valid sequences where each lane is used at least once. The total number of possible sequences without any restrictions is 3^6. We then subtract the cases where one or more lanes are empty and add back the cases where two lanes are empty. 1. **Total number of sequences**: $3^6 = 729$ 2. **Subtract sequences where one lane is empty**: - Each lane being empty reduces the number of choices to 2 for each car, so we subtract $3 \times 2^6 = 3 \times 64 = 192$. 3. **Add back sequences where two lanes are empty**: - Each case where two lanes are empty reduces the number of choices to 1 for each car, so we add back $3 \times 1^6 = 3 \times 1 = 3$. Putting it all together: $3^6 - 3 \times 2^6 + 3 \times 1^6 = 729 - 192 + 3 = 540$ Thus, the number of ways the cars can stack up so that all three lanes are occupied is $\boxed{540}$. **(2910 tokens)**

Figure 22: DeepSeek-8B engages Thinking mode for the hard problem after PPO, correctly applying inclusion-exclusion.