目标是激励模型选择NoThinking模式，同时确保整体性能不会降低；（2）在政策培训过程中平衡思考和非思考样本的重要性抽样策略，从而实现冷启动，并允许模型在整个培训过程中探索和利用这两种思维模式。我们将在下面详细介绍这两个组件。

4.1约束优化目标

考虑到NoThinking模式在推理效率方面比Thinking具有显著优势，理想的选择策略应该更倾向于选择NoThinking，只要不降低整体性能。换句话说，我们应该最大限度地提高产生NoThinking反应的概率，同时确保模型的准确性不会下降。

形式上，考虑一个推理模型π_θ以及一个数据集$\mathcal{D}$.让$\pi_{\theta_{\text{ref}}}$表示参考模型，即初始模型$\pi_\theta$在训练过程中保持不变。让$R(x, y, y^*)$是奖励函数（即数学求解的准确性），其中$x, y$，以及$\hat{y}$分别表示提示、模型响应和黄金答案。它回来了$0/1$如果$y$不正确/正确。为简单起见，我们省略$\hat{y}$并将函数表示为$R(x, y)$.让$1(y1 =)$是指示符函数，如果第一个令牌为，则返回$1y$是（即。，$y$是NoThinking响应），否则返回O。那么我们的优化目标可以公式化为：

$$\begin{aligned}&\max \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}1(y_1 = \text{think})\\&s.t.\,\mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}R(x,y)\geq\\&\mathbb{E}_{x\sim\mathcal{D},y'\sim\pi_{\theta_{\text{ref}}}(\cdot|x)}R(x,y').\end{aligned}$$

为了解决这个约束优化问题，我们将约束作为惩罚项纳入目标，并赋予惩罚权重$\lambda \geq 0$*

$$\begin{aligned}\lefteqn{\max \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x),y'\sim\pi_{\theta_{\text{ref}}}(\cdot|x)}1\big(y_1 = \text{think}\big)}\\&+\lambda\big(R(x,y)-R(x,y')\big).\quad(3)\end{aligned}$$

通过将两侧除以λ，让$\delta = \frac{1}{\lambda}$，并重新组织以下术语$\pi_{\theta_{\text{ref}}}$，我们有：

$$\begin{aligned}&\max \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}1(y_1 = \text{think})\cdot\delta\\&+R(x,y)-\mathbb{E}_{y'\sim\pi_{\theta_{\text{ref}}}(\cdot|x)}R(x,y').\end{aligned}$$

实际上，$\mathbb{E}y' \sim \pi\theta_{\text{ref}.}(\cdot|x)R(x, y')$可以通过训练前的预采样来近似。具体来说，我们采样$K$答复来自$\pi_{\theta_{\text{ref}}}(\cdot|x)$对于

每个x，并计算他们的平均奖励：

$$\bar{R}_{\text{ref}}(x)=\frac{1}{K}\sum_{i=1}^{K}R(x,y'^i),y'^i\sim\pi_{\theta_{\text{ref}}}(\cdot|x).$$

然后，优化目标变为：

$$\begin{aligned}&\max \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}1(y_1 = \text{think})\cdot\delta\\&+R(x,y)-\bar{R}_{\text{ref}}(x).\end{aligned}$$

自从$1(y_1 =)$和$R(x, y)$不可微，我们采用策略梯度法来解决这个优化问题。具体来说，设Tθold是一个等于π_θ在没有梯度更新的情况下定义优势函数：$A(x, y) = 1|(y_1 =) \cdot \delta + R(x, y) - \bar{R}_{\text{ref}}(x)$然后，目标可以转化为PPO式的损失（Schulman等人，2017），而不需要KL惩罚：

$$\begin{aligned}\mathcal{L}(\theta)=-\mathbb{E}_{x\sim\mathcal{D},y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\big[\min\big(\tfrac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}A(x,y),\\\text{clip}(\tfrac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)},1-\epsilon,1+\epsilon)A(x,y)\big)\big].\quad(7)\end{aligned}$$

在这里，$\text{clip}(\cdot)$表示剪切函数，提高了训练的稳定性。

4.2重要性抽样

在优化的每一步$\mathcal{L}(\theta)$使用on策略训练，我们对一批进行采样$\mathcal{D}b$从数据集中$\mathcal{D}$，然后采样K响应$y^i i = 1^K$来自

$\pi_{\theta_{\text{old}}}(\cdot|x)$对于每一个$x \in Db$估计$\mathcal{L}(\theta)$然而，自最初$\pi\theta$自然地将思维应用于所有问题，不可能从中获得无思维样本$\pi_{\theta_{\text{old}}}$从训练开始（即。，$\pi_{\theta_{\text{old}}}(y_1 = |x) \approx 0\rangle$因此，该模型只能从思维样本中学习，永远不会产生无思维反应。

为了解决这一冷启动挑战，我们采用了重要性抽样技术。具体来说，我们定义了一个新的分布$\pi_{\text{IS}}(\cdot|x)$

$$ \pi_{\mathrm{IS}}(y_{t}=a|x,y_{;}\\ {0.5,\mathrm{~if~}t=1,a=w_{\mathrm{start}};}\\ {\pi_{\theta_{\mathrm{old}}}}(y_{t}=a|x,y_{1.}\end{array}\right. $$

在这里，w_{start}是开始长时间思考的常用词，比如“好的”。在训练过程中，我们对反应进行采样$y^i i = 1^K$来自$\pi\text{IS}(\cdot|x)$而不是$\pi_{\theta_{\text{old}}}(\cdot|x)$，这样一批样本中有一半处于思考模式，另一半处于无思考模式。这允许模型从训练开始就从这两种模式中学习，并最终自适应地

算法1自适应思维

输入：策略模型π_θ；数据集$\mathcal{D}$；超参数K, δ, ϵ

初始化：参考模型$\pi_{\theta_{\text{ref}}} \leftarrow \pi_\theta$
1：样品K响应$y'i i = 1^K \sim \pi\theta_{\text{ref}}(\cdot|x)$并计算$\bar{R}\text{ref}(x)$对于每一个$x \in \mathcal{D}$（方程式5）
2：为step $= 1, \ldots, M$做
3：更新旧策略模型$\pi\theta_{\text{old}} \leftarrow \pi_\theta$以及重要性抽样分布Tis（方程式8）
4：抽样一批$\mathcal{D}b$从$\mathcal{D}$
5：样品K响应$y^i i = 1^K \sim \pi_{\text{IS}}(\cdot|x)$对于每一个$x \in \mathcal{D}b$并估算$\mathcal{L}\text{AT}(\theta)$（方程式9。……的一半$y^i$是思考反应，另一半是无思考反应。）
6：更新策略模型π_θ通过最小化$\dot{\mathcal{L}}_{\text{AT}}(\theta)$

能够选择适当的模式。因此，AdaptThink的最终损失函数变为：

$$\mathcal{L}_{\text{AT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D},y\sim\pi_{\text{IS}}(\cdot|x)}\big[\min\big(\tfrac{\pi_\theta(y|x)}{\pi_{\text{IS}}(y|x)}A(x,y),\\ \text{clip}(\tfrac{\pi_\theta(y|x)}{\pi_{\text{IS}}(y|x)}, 1-\epsilon, 1+\epsilon)A(x,y)\big)\big]. \qquad (9)$$

除了支持冷启动外，重要性抽样还保留了在整个培训过程中探索和利用思维和非思维模式的机会。这可以防止π_θ避免永远陷入一种思维模式而完全忽视另一种，即使后者在未来可能会显示出更大的优势。最后，我们在算法1中总结了我们的AdaptThink算法。

4.3理解损失的新视角

在本小节中，我们提供了另一个视角来理解我们的损失函数$\mathcal{L}\text{AT}(\theta)$通过比较优势$A(x,y)$思考和非思考样本$\pi\text{IS}(\cdot|x)$.给予提示$x$，我们将Thinking和NoThinking样本的平均通过率表示为$\bar{R}\text{think}(x)$和$\bar{R}\text{nothink}(x)$分别。那么他们的平均优势是：

$$\bar{A}_{\text{think}}(x) = \bar{R}_{\text{think}}(x) - \bar{R}_{\text{ref}}(x),\\ \bar{A}_{\text{nothink}}(x) = \delta + \bar{R}_{\text{nothink}}(x) - \bar{R}_{\text{ref}}(x).$$

注意，选择NoThinking的概率（即。，$\pi_\theta(y_1 = |x))$和思考（即。，$\pi_\theta(y_1 = w^*|x))$具有竞争力。因此，在优化时$\mathcal{L}\text{AT}(\theta)$，$\pi\theta(y_1 = |x)$只有当$\bar{A}\text{nothink}(x) > 0$和$\bar{A}\text{nothink}(x) > \bar{A}_{\text{think}}(x)$，这给了我们：

$$\bar{R}_{\text{nothink}}(x) + \delta > \bar{R}_{\text{ref}}(x),\\ \bar{R}_{\text{nothink}}(x) + \delta > \bar{R}_{\text{think}}(x).$$

换句话说，只有当问题足够简单，使得NoThinking和Thinking以及参考模型之间的精度差距小于δ。$\mathcal{L}\text{AT}(\theta)$会支持NoThinking并鼓励$\pi\theta$以直接生成最终解决方案。对于NoThinking远远落后于其他两个更具挑战性的问题，$\mathcal{L}\text{AT}(\theta)$将优先考虑绩效并提供指导$\pi\theta$更频繁地进行思考。因此，$\mathcal{L}_{\text{AT}}(\theta)$这与我们在第3.2节中对难度适应性思维的期望非常一致。

5实验

5.1设置

模型。我们选择DeepSeek-R1-Distill-Qwen1.5B和DeepSeek-R2-Distill-Kwen-7B作为初始策略模型，这两种流行的推理模型在数学问题解决方面表现出了令人印象深刻的性能。

数据集和度量。我们使用的训练数据集是DeepScaleR（Luo等人，2025b）数据集，它由来自AIME 1983-2023、AMC、Omni math（Gao等人，2024）和STILL（Min等人，2024）的4oK数学问题组成。为了进行评估，我们使用了三个难度越来越大的数学数据集：GSM8K（Cobbe等人，2021）测试集（1319道小学数学题）、MATH500（Lightman等人，2024）（500道高中竞赛数学题）和AIME 2024（30道奥赛级数学题）。对于评估指标，我们同时考虑准确性和响应长度。我们还报告了所有测试数据集的平均精度变化和平均长度缩短率。考虑到AIME 2024的规模有限，我们对每个病例重复抽样16份回复，并报告平均结果。对于所有模型，我们将评估上下文大小设置为16K，并将温度设置为DeepSeek模型卡中建议的0.6。

实施细节。我们基于VeRL（Sheng等人，2024）框架构建代码

Method	GSM8K			MATH 500			AIME 2024			Average	
	Acc	Length	RatiONT	Acc	Length	RatiONT	Acc	Length	RatiONT	△Acc	△Length
DeepSeek-Rl-Distill-Qwen-1.5B											
OriginalThinking	79.0	978	0.0%	80.6	4887	0.0%	29.4	12073	0.0%		
OriginalNoThinking	69.8	280	100.0%	67.2	658	100.0%	14.0	2190	100.0%	-12.7	-79.9%
DPOShortest	78.3	804	0.0%	82.4	3708	0.0%	30.7	10794	0.0%	+0.8	-17.5%
OverThink	77.2	709	0.0%	81.2	4131	0.0%	28.3	11269	0.0%	-0.8	-16.5%
DAST	77.2	586	0.0%	83.0	2428	0.0%	26.9	7745	0.0%	-0.6	-42.1%
O1-Pruner	74.8	458	0.0%	82.2	3212	0.0%	28.9	10361	0.0%	-1.0	-33.9%
TLMRE	80.7	863	0.0%	85.0	3007	0.0%	29.2	8982	0.0%	+2.0	-25.3%
ModelMerging	79.7	603	0.0%	63	2723	0.0%	18.1	10337	0.0%	-9.4	-32.3%
RFTMixThinking	76	1077	8.8%	72.4	4341	33.4%	25.2	11157	21.0%	-5.1	-2.9%
AdaptThink	83.1	480	86.9%	82.0	1782	76.8%	31.0	6679	40.4%	+2.4	-53.0%
DeepSeek-Rl-Distill-Qwen-7B											
OriginalThinking	87.9	682	0.0%	90.2	3674	0.0%	53.5	10306	0.0%	-	-
OriginalNoThinking	85.1	283	100.0%	80.6	697	100.0%	24.2	1929	100.0%	-13.9	-73.6%
DPOShortest	85.7	402	0.0%	91.6	2499	0.0%	52.5	8699	0.0%	-0.6	-29.5%
OverThink	86.3	426	0.0%	89.4	2435	0.0%	53.1	8744	0.0%	-0.9	-28.8%
DAST	86.7	459	0.0%	89.6	2162	0.0%	45.6	7578	0.0%	-3.2	-33.4%
O1-Pruner	87.6	428	0.0%	86.6	2534	0.0%	49.2	9719	0.0%	-2.7	-24.7%
TLMRE	88.9	756	0.0%	91.8	2899	0.0%	54.0	8633	0.0%	+1.0	-8.8%
ModelMerging	88.4	531	0.0%	72.6	2280	0.0%	36.9	8624	0.0%	-11.2	-25.5%

RFTMixThinking	86.2	365	66.5%	84.8	2411	64.8%	49.4	9969	10.0%	-3.7	-28.0%
AdaptThink	91.0	309	99.6%	92.0	1875	76.6%	55.6	8599	6.3%	+2.3	-40.1%

表1：准确度（Acc）、反应长度（length）和NoThinking反应的比率(Ratio_{NT})在三个数学基准上使用不同的方法。最佳结果和第二结果分别用粗体和下划线表示。

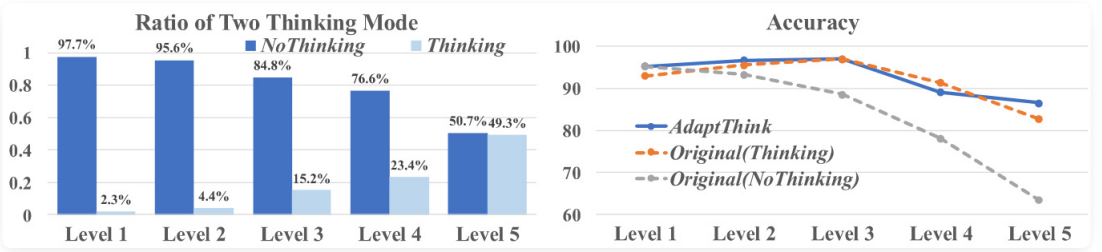

图3：左：AdaptThink-7B在不同数学水平上选择思考或无思考的比例。右图：AdaptThink-7B和DeepSeek-R1-Distill-Qwen-7B在不同数学水平上使用思考和无思考的准确性比较。

速率分别设置为16K、128和2e-6。超参数K, δ，以及ϵAdaptThink中的值分别设置为16、0.05和0.2。不同使用方式的比较δ如第5.4节所示。我们训练1个历元的模型，总共314步。对于1.5B型号，我们使用一个$8 \times \mathrm{H}800$节点，花费约32小时。对于7B模型，我们使用四个$8\times\mathrm{H}800$节点，花费约28小时。最后，我们分别为1.5B和7B模型选择了300步和150步的检查点，其中模型的准确性和响应长度达到了良好的平衡。

5.2基线

我们将AdaptThink与以下具有代表性的高效推理方法进行了比较：

·DPOshortest通过对训练数据集中的每个问题的多个响应进行采样，并将最短的正确响应和最长的响应配对来构建偏好数据，然后使用DPO（Rafailov等人，2023）来微调模型。

·OverThink（Chen等人，2O24）首先通过将每个训练问题的原始长思维反应作为反例，并保留在思维中得出正确答案的前两次尝试作为正例来构建偏好数据，然后使用SimPO（Meng等人，2024）来缓解模型的过度思考行为。

·DAST（Shen等人，2025）首先通过使用基于长度的奖励函数对预采样的响应进行排名来构建偏好数据，然后