Reinforcement Learning for LLMs

Karl Stratos

Last updated: November, 2026

Contents

1 Generative Story **2**

2 REINFORCE **2**
2.1 Surrogate Objective . . . 2
2.2 Relation to Importance Sampling . . . 3
2.3 Relation to Rejection Sampling . . . 3

3 Trust-Region Methods **3**
3.1 Surrogate Objective . . . 3
3.2 First-Order Exactness . . . 4

4 Trust Region Policy Optimization (TRPO) **4**
4.1 Natural Policy Gradient . . . 4
4.2 Algorithm . . . 5

5 Proximal Policy Optimization (PPO) **5**
5.1 Asymmetric Clipping Effect . . . 5
5.2 Relative Clipping Effect . . . 6
5.3 Length Normalization . . . 6
5.4 KL Variant . . . 7

6 Mirror Descent Policy Optimization (MDPO) **7**
6.1 Kimi k1.5 Variant . . . 7

7 Divergence Proximal Policy Optimization (DPPO) **8**
7.1 TRPO Motivation . . . 8
7.2 Average Bound . . . 9

8 Training Inference Mismatch (TIM) **9**
8.1 Causes of TIM . . . 10

9 Score Centering (SC) **10**
9.1 Top-k Approximation . . . 11
9.2 Importance Sampling Hybrid . . . 11

10 Advantage Estimation **12**
10.1 Critic-Based . . . 12
10.1.1 Generalized advantage estimator (GAE) . . . 12
10.2 Critic-Free . . . 13

A Markov Decision Process (MDP) **14**
A.1 Training . . . 15

B KL Estimation **15**

C Lemmas **16**

1 Generative Story

An environment issues an initial state $s_1 \sim \mathbf{env}$ consisting of system/user prompts. For $t = 1 \dots T$ where $T \le T_{\max}$ is a bounded random termination step, do

$$\begin{aligned} x_t &\sim p_\theta(\cdot|s_t) && \text{(token } x_t \in \mathcal{V}\text{)} \\ (s_t, x_t) &\mapsto s_{t+1} && \text{(state transition)} \end{aligned} \tag{1}$$

The state s_{t+1} contains all generated tokens $x_1 \dots x_t$ as well as all information present in the past states $s_1 \dots s_t$ so that (1) remains autoregressive. The environment issues a final reward for the terminal state $R = R(s_{T+1}) \in \mathbb{R}$. See Appendix A for how this is implemented in practice. The goal is to maximize

$$J(\theta) = \mathop{\mathbf{E}}_{\substack{s_1 \sim \mathbf{env} \\ \tau \sim p_\theta(\cdot|s_1)}} [R(s_{T+1})] \tag{2}$$

where $\tau = (x_1 \dots x_T)$ is the token trajectory (rollout) generated from s_1 yielding a state trajectory $s_2 \dots s_{T+1}$.[1] We sometimes write the reward $R(\tau) = R(s_{T+1})$ in rollouts. Most methods are **policy-gradient** methods, i.e., maximize (2) by following its gradient.

Endpoints. We distinguish the rollout endpoint $K \le T$ from the termination step, where $K = T$ if the rollout is terminated and $K < T$ if truncated (e.g., due to time/length limit). We assume the reward is always zero for nonterminal states, thus $R(s_{K+1}) = 0$ if truncated. For steps beyond genuine termination, we assume that the state is frozen: $s_t = s_{T+1}$ for $T < t \le T_{\max}$.

Denoting random variables. Failing to specify what is random leads to poor readability, but marking every random variable overwhelms notation. As a middle ground, we will only mark what is helpful in reminding us of the generative story at hand. For instance, if we write $\tau \sim p_\theta$ it is understood that the initial state is drawn as usual (i.e., $p_\theta(\tau) = \mathbf{E}_{s_1 \sim \mathbf{env}}[p_\theta(\tau|s_1)]$). If we write $\mathbf{E}_\theta[R|s_t]$, it means the expected final reward R if the model θ starts the session from s_t.

2 REINFORCE

REINFORCE (Williams, 1992) gives a direct gradient estimator for (2). By the log-derivative trick and a property of the score function, for any "baseline" function $b(s_t)$ independent of x_t we can write (Lemma C.1)

$$\nabla J(\theta) = \mathop{\mathbf{E}}_{\tau \sim p_\theta} \left[\sum_{t=1}^{T} (R(s_{T+1}) - b(s_t)) \nabla \log p_\theta(x_t|s_t) \right] \tag{3}$$

The gradient is now samplable and weighted at a token level. We should make $R(s_{T+1}) - b(s_t)$ large if the eventual outcome is unexpectedly good, so it is natural to define

$$b(s_t) = \mathbf{E}_\theta[R|s_t] \tag{4}$$

(the **value function**). It is also approximately the minimizer of per-step total variance (Lemma C.2).

2.1 Surrogate Objective

To use the gradient estimator (3), we can construct a surrogate objective at each θ as

$$J_{\text{rein}}(\theta) = \mathop{\mathbf{E}}_{\tau \sim p_{\text{sg}(\theta)}} \left[\sum_{t=1}^{T} \text{sg}\left(R(s_{T+1}) - b(s_t)\right) \log p_\theta(x_t|s_t) \right] \tag{5}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. It is evident that $\nabla J_{\text{rein}}(\theta) = \nabla J(\theta)$. If we choose (4) as the baseline, by per-step conditioning we can rewrite (5) as

$$J_{\text{rein}}(\theta) = \mathop{\mathbf{E}}_{\tau \sim p_{\text{sg}(\theta)}} \left[\sum_{t=1}^{T} \text{sg}(A_\theta(s_t, x_t)) \log p_\theta(x_t|s_t) \right] \tag{6}$$

where $A_\theta(s_t, x_t) = \mathbf{E}_\theta[R|s_t, x_t] - \mathbf{E}_\theta[R|s_t]$ is the **advantage function** (Section 10).

[1] We assume deterministic state transition for simplicity, but the stochastic case can be immediately handled by taking an outer expectation.

2.2 Relation to Importance Sampling

To remove the model from the expectation in (2), we can simply use importance sampling. Then the derivation of the gradient is more straightforward:

$$J(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\frac{p_\theta(\tau|s_1)}{p_\psi(\tau|s_1)}R(s_{T+1})\right] \quad \Rightarrow \quad \nabla J(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\frac{p_\theta(\tau|s_1)}{p_\psi(\tau|s_1)}(R(s_{T+1})-b(s_t))\nabla\log p_\theta(x_t|s_t)\right] \tag{7}$$

We see that the gradient reduces to REINFORCE (3) at $\psi = \mathrm{sg}(\theta)$.

2.3 Relation to Rejection Sampling

If $R \in \{0,1\}$ and $b(s_t) = 0$, the surrogate objective (5) becomes

$$J_{\text{rein}}(\theta) = \Pr_{\tau\sim p_{\mathrm{sg}(\theta)}}(R=1) \times \mathop{\mathbf{E}}_{\tau\sim p_{\mathrm{sg}(\theta)}:\, R=1}[\log p_\theta(\tau|s_1)]$$

This is rejection sampling (i.e., supervised fine-tuning on successful trajectories).

3 Trust-Region Methods

REINFORCE (6) is exact but fully on-policy. In reality, the rollouts come from a slightly lagging version of the model ("policy staleness"):

1. An inference engine (**sampler**) uses the current checkpoint $\psi \leftarrow \theta$ to generate a large batch $\mathcal{B}$ of rollouts.
2. A training engine (**trainer**) updates θ on the frozen $\mathcal{B}$ in minibatches, possibly making many epochs. Thus later updates use stale rollouts.

In principle, there is nothing that prevents us from going off-policy. Importance sampling (7) allows us to use arbitrary ψ without losing consistency. The problem is *variance*.[2]

3.1 Surrogate Objective

Trust-region (TR) methods rely on approximation to trade off bias against variance. The target objective is the "improvement" of the current policy $J(\theta) - J(\psi)$. We write it in a few different forms for convenience:

$$J(\theta)-J(\psi) = \mathop{\mathbf{E}}_{\tau\sim p_\theta}\left[\sum_{t=1}^{T}A_\psi(s_t,x_t)\right] = \sum_{t=1}^{T_{\max}}\mathop{\mathbf{E}}_{\substack{s\sim d_{\theta,t}\\ x\sim p_\psi(\cdot|s)}}\left[\frac{p_\theta(x|s)}{p_\psi(x|s)}A_\psi(s,x)\right] = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\left(\prod_{i=1}^{t}\frac{p_\theta(x_i|s_i)}{p_\psi(x_i|s_i)}\right)A_\psi(s_t,x_t)\right] \tag{8}$$

where $d_{\theta,t}(s) = \Pr_\theta(s_t = s)$. The first equality is known as the **performance difference lemma** (Kakade and Langford, 2002).[3] Note that (8) is just $J(\theta)$ shifted by a constant, so it has the same true gradient $\nabla J(\theta)$ as REINFORCE. But the estimator's variance blows up if ψ is very different from θ. To reduce the variance, TR assumes

$$d_{\theta,t} \approx d_{\psi,t} \tag{9}$$

at every t. Plugging this assumption in the forms above, we have the surrogate objective:

$$J_\psi(\theta) = \sum_{t=1}^{T_{\max}}\mathop{\mathbf{E}}_{\substack{s\sim d_{\psi,t}\\ x\sim p_\psi(\cdot|s)}}\left[\frac{p_\theta(x|s)}{p_\psi(x|s)}A_\psi(s,x)\right] = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)}A_\psi(s_t,x_t)\right] \tag{10}$$

Intuitively, we approximate the probability ratio of the whole history by the current token, banking on $\theta \approx \psi$.

[2] This is mainly because it involves unbounded probability ratios. REINFORCE kills them by assuming $\psi = \mathrm{sg}(\theta)$.

[3] Verify easily by noting that the sum of advantages telescopes.

3.2 First-Order Exactness

A property of the TR approximation is that it is first-order exact at $\theta = \psi$. Define

$$\texttt{Bias}_\psi(\theta) = J_\psi(\theta) - \big(J(\theta) - J(\psi)\big) \tag{11}$$

We can easily check $\texttt{Bias}_\psi(\psi) = 0$ and $\nabla\texttt{Bias}_\psi(\psi) = 0_d$. This implies (Taylor-expand around ψ)

$$\texttt{Bias}_\psi(\psi + \Delta) = O(||\Delta||_2^2) \tag{12}$$

This gives (10) a favorable bias-variance tradeoff (i.e., variance reduction at mild inconsistency) since the bias shrinks quadratically rather than linearly as $\Delta \to 0_d$. The bound's quadratic dependence on policy change cannot be improved in general.

4 Trust Region Policy Optimization (TRPO)

TRPO (Schulman *et al.*, 2015b) gives an explicit bound on the bias (11) based on total variation (TV):

$$\mathrm{TV}(P, Q) = \frac{1}{2}\sum_x |P(x) - Q(x)| = 1 - \sum_x \min(P(x), Q(x)) = \min_{(X,Y):\, X\sim P,\, Y\sim Q} \Pr(X \neq Y)$$

The second form follows from the identity $|a - b| = a + b - 2\min(a, b)$. The last form is the classical "coupling" characterization of TV (Lemma C.3). TV is useful for bounding the expectation of a function g under different distributions (exactly what a bias is). Using the first form, by triangle inequality

$$|\mathbf{E}_P[g] - \mathbf{E}_Q[g]| \leq 2\,||g||_\infty\,\mathrm{TV}(P, Q) \tag{13}$$

We can use (13) to easily show, for some constants $C, C' > 0$ (Lemma C.5)

$$\begin{aligned} |\texttt{Bias}_\psi(\theta)| &\leq C \times \Big(\max_s \mathrm{TV}(p_\psi(\cdot|s), p_\theta(\cdot|s))\Big)^2 \\ &\leq C' \times \Big(\max_s \mathrm{KL}(p_\psi(\cdot|s), p_\theta(\cdot|s))\Big) \end{aligned} \tag{14}$$

(the latter is Pinsker's inequality $\mathrm{TV}(P, Q)^2 \leq \frac{1}{2}\mathrm{KL}(P, Q)$). Thus to justify $J_\psi(\theta)$ as a valid lower bound (i.e., $J(\theta) - J(\psi) \geq J_\psi(\theta) - \text{const}$), it is sufficient to have the state-wise bounds

$$\Big(\mathrm{TV}(p_\psi(\cdot|s), p_\theta(\cdot|s)) \leq \epsilon \quad \forall s\Big) \qquad \bigvee \qquad \Big(\mathrm{KL}(p_\psi(\cdot|s), p_\theta(\cdot|s)) \leq \epsilon \quad \forall s\Big) \tag{15}$$

TRPO uses the KL-based constraint but approximates the max with mean for computational reasons. Define[4]

$$\mathrm{KL}_\psi(\theta) = \mathop{\mathbf{E}}_{s_1\sim\mathbf{env}}[\mathrm{KL}(p_\psi(\cdot|s_1), p_\theta(\cdot|s_1))] = \mathop{\mathbf{E}}_{\substack{s_1\sim\mathbf{env} \\ \tau\sim p_\psi(\cdot|s_1)}}\left[\sum_{t=1}^{T} \mathrm{KL}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t))\right] \tag{16}$$

The KL-based TRPO objective is

$$\max_{\theta:\, \mathrm{KL}_\psi(\theta)\leq\epsilon} J_\psi(\theta) \tag{17}$$

4.1 Natural Policy Gradient

(17) can be beautifully simplified by local approximations $\theta = \psi + \Delta$:

$$J_\psi(\psi + \Delta) \approx \nabla J(\psi)^\top \Delta \tag{18}$$

$$\mathrm{KL}_\psi(\psi + \Delta) \approx \frac{1}{2}\Delta^\top I(\psi)\Delta \tag{19}$$

[4]This is different from defining it as $\mathrm{KL}(p_\psi, p_\theta)$ between the marginals. But it is the right divergence since we care about the policies' behavioral similarities given an prompt, not the similarities between their marginalized versions. It is also more computationally convenient since we can use the linearity of expectation. Qi *et al.* (2026) retrospectively justifies constraining this KL term in the average bound (34).

(18) uses the first-order exactness of $J_\psi(\theta)$ at $\theta = \psi$. (19) uses $\nabla \mathrm{KL}_\psi(\psi) = 0_d$ and $\nabla^2 \mathrm{KL}_\psi(\psi) = I(\psi)$ where $I(\psi) = \mathbf{E}_{s_1 \sim \mathbf{env}}[\mathrm{Cov}_{\tau \sim p_\psi(\cdot|s_1)}(\nabla \log p_\psi(\tau|s_1))]$ is the expected conditional Fisher matrix of ψ (easy exercise). This is now steepest ascent with a Fisher-ball constraint, and we get the usual exact update

$$\Delta^\star = \underset{\Delta:\ \frac{1}{2}||\Delta||^2_{I(\psi)} \le \epsilon}{\arg\max} \nabla J(\psi)^\top \Delta = \eta I(\psi)^{-1} \nabla J(\psi) \qquad \left(\eta = \sqrt{\frac{2\epsilon}{||\nabla J(\psi)||^2_{I(\psi)^{-1}}}}\right) \tag{20}$$

The direction $v = I(\psi)^{-1}\nabla J(\psi)$ coincides with the **natural policy gradient** of Kakade (2001) which replaces the parameter space with coordinate-free information geometry.

4.2 Algorithm

We can solve for v in $I(\psi)v = \nabla J(\psi)$ without inverting $I(\psi)$ by conjugate gradient, which only requires an efficient mapping $v \mapsto I(\psi)v$. Since $I(\psi) = \nabla G_\psi(\psi)$ is the Jacobian of the gradient function $G_\psi(\theta) = \nabla \mathrm{KL}_\psi(\theta)$ at $\theta = \psi$,

$$I(\psi)v = \mathbf{JVP}(G_\psi, \psi, v)$$

can be computed by the Jacobian-vector product (JVP).[5] Thus a basic implementation may resemble

1. Estimate $g_\psi = \nabla J(\psi)$, e.g., by REINFORCE (6) with samples $\tau \sim p_\psi$.
2. Estimate $v = I(\psi)^{-1} g_\psi$ by conjugate gradient with the mapping $v \mapsto I(\psi)v = \mathbf{JVP}(G_\psi, \psi, v)$.
3. Compute $\eta = \sqrt{2\epsilon / g_\psi^\top v}$ and update $\theta \leftarrow \psi + \eta v$.

In practice, TRPO additionally performs backtracking line search along v that ensures the actual surrogate objective $J_\psi(\theta)$ improves and the KL constraint is satisfied. Unfortunately, conjugate gradient still requires iteratively computing $v \mapsto I(\psi)v$ per update, so the algorithm remains computationally expensive compared to first-order methods.

5 Proximal Policy Optimization (PPO)

PPO (Schulman *et al.*, 2017) directly maximizes $J_\psi(\theta)$ without hard constraints, but promoting a trust region implicitly. The objective uses per-step clipping:

$$J^{\mathrm{ppo}}_\psi(\theta) = \mathop{\mathbf{E}}_{\tau \sim p_\psi}\left[\sum_{t=1}^{T} \min\left(\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t), \mathrm{clip}^{1+\epsilon}_{1-\epsilon}\left(\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)}\right) A_\psi(s_t, x_t)\right)\right] \tag{21}$$

It is easy to check that (21) is a lower bound on $J_\psi(\theta)$ and remains first-order exact (Section 3.2). Crucially, the operation $\mathrm{clip}^{1+\epsilon}_{1-\epsilon}(r_t) \in [1-\epsilon, 1+\epsilon]$ passes no gradient through r_t if $r_t = \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)}$ falls out of the trust region. The objective combines two separate clipping effects: (i) asymmetric, and (ii) relative.

5.1 Asymmetric Clipping Effect

Because the advantage can be negative, the contribution of the t-th token to training is dropped iff r_t moves excessively in the *right* direction

$$\Big(A_\psi(s_t, x_t) > 0 \quad \bigwedge \quad r_t > 1+\epsilon\Big) \qquad \bigvee \qquad \Big(A_\psi(s_t, x_t) < 0 \quad \bigwedge \quad r_t < 1-\epsilon\Big) \tag{22}$$

In constrast, t is *not* dropped if r_t moves excessively in the wrong direction. This asymmetric design has the effect of pulling θ back into the trust region only when it is mistaken (Figure 1). An intuition is that we should not penalize the right move even when it is outside the trust region, but we should not keep amplifying it either.

[5] Recall that it avoids explicitly computing the Jacobian $\nabla G_\psi(\psi) \in \mathbb{R}^{d \times d}$ by computing the directional derivatives of $G_\psi : \mathbb{R}^d \to \mathbb{R}^d$ at $\theta = \psi$ in $v \in \mathbb{R}^d$ (at the cost of doubling the computation graph).



Figure 1: PPO clipping (21) updates $p_\theta(x_t|s_t)$ only if $r_t = \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)}$ is (i) wrong or (ii) still inside the trust region. When $A_\psi(s_t, x_t) > 0$, it is increased iff $r_t \leq 1+\epsilon$. When $A_\psi(s_t, x_t) < 0$, it is decreased iff $r_t \geq 1-\epsilon$.

	Unnorm.	PPO	GRPO	DAPO	Dr. GRPO
population objective	$\mathbf{E}_\psi[J]$	$\frac{\mathbf{E}_\psi[J]}{\mathbf{E}_\psi[T]}$	$\mathbf{E}_\psi[\frac{J}{T}]$	$\mathbf{E}_{s_1}[\frac{\mathbf{E}_\psi[J\|s_1]}{\mathbf{E}_\psi[T\|s_1]}]$	$\frac{\mathbf{E}_\psi[J]}{T_{\text{const}}}$
empirical objective	$\frac{1}{N}\sum_i J^{(i)}$	$\frac{1}{\sum_i T^{(i)}}\sum_i J^{(i)}$	$\frac{1}{N}\sum_{p,j}\frac{J^{(p,j)}}{T^{(p,j)}}$	$\frac{1}{P}\sum_p \frac{\sum_j J^{(p,j)}}{\sum_j T^{(p,j)}}$	$\frac{1}{NT_{\text{const}}}\sum_i J^{(i)}$
equal weights for	toks/batch	toks/batch	rollouts/prompt	toks/prompt	toks/batch
longer rollout ↑ weight	◯	◯	✗	◯ (within prompt)	◯
change in grad. dir.	✗	✗	△	△	✗

Figure 2: Summary of length normalization schemes. J denotes the total contribution of a rollout with length T. We index $i = 1 \ldots N$ (all rollouts), $j = 1 \ldots G$ (rollouts in a group), and $p = 1 \ldots P$ (all prompts) in a batch of $N = GP$ rollouts.

5.2 Relative Clipping Effect

PPO clipping is based on relative probability change. As a result, if the advantage at step t is positive, we drop the token iff

$$p_\theta(x_t|s_t) > (1+\epsilon)p_\psi(x_t|s_t) \quad \Leftrightarrow \quad p_\theta(x_t|s_t) - p_\psi(x_t|s_t) > \epsilon p_\psi(x_t|s_t) \tag{23}$$

We see that high-confidence positive tokens are disproportionately spared ($\rightarrow$ exploitation).[6] For instance, if $p_\psi(x_t|s_t) = 0.9$ and $\epsilon = 0.2$, $x_t|s_t$ never gets clipped and thus continues to be reinforced even when $p_\theta(x_t|s_t) \approx 1$. In contrast, if the advantage is negative, we drop the token iff

$$p_\theta(x_t|s_t) < (1-\epsilon)p_\psi(x_t|s_t) \quad \Leftrightarrow \quad p_\psi(x_t|s_t) - p_\theta(x_t|s_t) > \epsilon p_\psi(x_t|s_t) \tag{24}$$

Here low-confidence negative tokens are disproportionately spared ($\rightarrow$ exploration). Researchers have proposed modifications to address PPO's disproportionate clipping effects, for example

- DAPO (Yu *et al.*, 2026) uses a higher upper threshold $\epsilon' > \epsilon$ in $\text{clip}_{1-\epsilon}^{1+\epsilon'}(r_t)$ to promote reinforcing more positive tokens (i.e., low-confidence). Note that it still allows for exploiting high-confidence positive tokens.
- DPPO (Qi *et al.*, 2026) uses an adaptive threshold $\epsilon_t = \frac{\epsilon}{p_\psi(x_t|s_t)}$ corresponding to absolute clipping, leading to *both* stronger reinforcement of low-confidence positive tokens *and* stronger suppression of low-confidence negative tokens (Section 7).

5.3 Length Normalization

Figure 2 summarizes various length normalization schemes developed for PPO, motivated along two orthogonal axes:

1. **Inter-batch scaling.** Without normalization, a batch with longer rollouts makes a larger update. The original PPO paper normalizes by the total number of tokens in each batch. Dr. GRPO avoids modifying the objective by assuming a constant rollout length T_{const}.
2. **Intra-batch weighting.** Since the objective weights all tokens in the batch equally, a longer rollout makes a larger update. GRPO weights all rollouts equally within a prompt. DAPO weights all tokens equally within a prompt (thus a longer rollout makes a larger update—within a prompt).

[6] This is suspected to be related to the empirical phenomenon of "entropy collapse" (θ becomes almost deterministic) which kills exploration in the next round of PPO.

5.4 KL Variant

PPO also considers a more classical soft-constraint variant that maximizes the Lagrangian $L(\theta, \lambda) = J_\psi(\theta) - \lambda(\mathrm{KL}_\psi(\theta) - \epsilon)$ of (17) over θ, without letting the opponent minimize it over $\lambda > 0$. Equivalently, we maximize the TR objective with a KL penalty

$$J_\psi^{\text{ppo-kl}}(\theta) = J_\psi(\theta) - \lambda \mathrm{KL}_\psi(\theta) = \mathop{\mathbf{E}}_{\tau \sim p_\psi} \left[\sum_{t=1}^{T} \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t) - \lambda \mathrm{KL}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t)) \right] \tag{25}$$

So this is just the raw TR objective with a per-step KL regularization. We can use any variance-stabilized KL estimator (Appendix B). Since the penalty is always active, the original paper proposes an adaptive scheme where λ is doubled when the KL is $> 1.5\epsilon$, halved when $< \frac{\epsilon}{1.5}$. However, the empirical observation is that the KL variant is less competitive than the clipping variant, whose per-token regularization is sample-dependent.

6 Mirror Descent Policy Optimization (MDPO)

MDPO (Tomar *et al.*, 2020) revisits the KL variant of PPO:

$$J_\psi^{\text{mdpo}}(\theta) = \mathop{\mathbf{E}}_{\tau \sim p_\psi} \left[\sum_{t=1}^{T} \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t) - \lambda_{\text{iter}} \mathrm{KL}(p_\theta(\cdot|s_t), p_\psi(\cdot|s_t)) \right] \tag{26}$$

The only practical differences from (25) are: the per-step KL direction is reversed, and the regularization strength is scheduled over policy iterations.[7] But (26) is motivated more specifically from a classical tabular setting. The key observation is that the TR surrogate (10) is already linear in the model's token distribution $q_s(x) := p_\theta(x|s)$ for a fixed s. Shown again without importance sampling,

$$J_\psi(\theta) = \sum_{t=1}^{T_{\max}} \mathop{\mathbf{E}}_{s \sim d_{\psi,t}} \left[\sum_x q_s(x) A_\psi(s, x) \right] \tag{27}$$

So the *per-state* objective is $q_s^\top A_\psi(s, \cdot)$ where the gradient is literally the advantages $A_\psi(s, \cdot) \in \mathbb{R}^V$. In a tabular setting, at each s we use the optimal token distribution

$$q_s^\star = \mathop{\arg\max}_{q_s \in \Delta^{V-1}} \sum_x q_s(x) A_\psi(s, x) - \lambda \mathrm{KL}(q_s, p_\psi(\cdot|s)) \tag{28}$$

(28) is equivalent to exponentiated gradient descent (i.e., mirror descent with probability simplex + KL) on a trivial (linear) loss. The fact that $\sum_x q_s^\star(x) A_\psi(s, x) \geq 0$ implies $J(q^\star) - J(\psi) = \sum_t \mathbf{E}_{q^\star}[A_\psi(s, x)] \geq 0$. MDPO (26) is motivated as taking gradient steps toward $q_s^\star$ instead of computing the well-known "tilting" solution $q_s^\star(x) \propto e^{A_\psi(s,x)/\lambda} p_\psi(x|s)$. With a bit of analysis, we can show the exact relationship:

$$J_\psi^{\text{mdpo}}(\theta) = \sum_{t=1}^{T_{\max}} \mathop{\mathbf{E}}_{s \sim d_{\psi,t}} [C_s^\star - \lambda \mathrm{KL}(p_\theta(\cdot|s), q_s^\star)] \tag{29}$$

where $C_s^\star$ is the maximum of (28).[8]

6.1 Kimi k1.5 Variant

Kimi k1.5 (Team *et al.*, 2025a) applies the mirror descent approach to a sequence level and minimizes a regression loss toward the tilting solution. It observes that the original objective $J(\theta) = \mathbf{E}_{\tau \sim p_\theta}[R(\tau)]$ (2) is also linear in the model's *rollout* distribution $r_{s_1}(\tau) = p_\theta(\tau|s_1)$. So by the same line of tabular logic, at each s_1 we use the optimal rollout distribution

$$r_{s_1}^\star = \mathop{\arg\max}_{r_{s_1} \in (\text{all policies})} \sum_\tau r_{s_1}(\tau) R(\tau) - \lambda \mathrm{KL}(r_{s_1}, p_\psi(\cdot|s_1)) \qquad \Rightarrow \qquad r_{s_1}^\star(\tau) \propto e^{R(\tau)/\lambda} p_\psi(\tau|s_1) \tag{30}$$

[7] The paper uses $\lambda_{\text{iter}} = \frac{1}{t_{\text{iter}}}$ where $t_{\text{iter}} = 1 - \frac{\text{iter}}{\text{iter}_{\max}}$, so the strength increases over policy iterations each of which corresponds to a mirror-descent step (28) and involves sampling batches from a particular rollout policy.

[8] Expanding $\mathrm{KL}(q_s, q_s^\star)$ with $q_s^\star(x) = e^{A_\psi(s,x)/\lambda} p_\psi(x|s)/Z_\psi(s)$ and rearranging, we have $\sum_x q_s(x) A_\psi(s, x) - \lambda \mathrm{KL}(q_s, p_\psi(\cdot|s)) = \lambda \log Z_\psi(s) - \lambda \mathrm{KL}(q_s, q_s^\star)$. The RHS is maximized exactly at $q_s = q_s^\star$ to $C_s^\star = \lambda \log Z_\psi(s)$.

This gives the characterization of an ideal policy. Taking the log and rearranging, it can be stated as $\lambda \log \frac{r^\star_{s_1}(\tau)}{p_\psi(\tau|s_1)} = R(\tau) - \lambda \log Z_\psi(s_1)$ where $Z_\psi(s_1) = \mathbf{E}_{\tau\sim p_\psi(\cdot|s_1)}[e^{R(\tau)/\lambda}]$. The paper proposes to promote this property by minimizing the regression loss

$$L_\psi^{\text{k1.5}}(\theta) = \mathop{\mathbf{E}}_{\substack{s_1\sim\mathbf{env}\\ \tau\sim p_\psi(\cdot|s_1)}} \left[\left(R(\tau) - \mathbf{E}_\psi[R|s_1] - \lambda \log \frac{p_\theta(\tau|s_1)}{p_\psi(\tau|s_1)} \right)^2 \right] \tag{31}$$

which samples from ψ for convenience (any full-support rollout distribution is valid) and uses the approximation $\lambda \log \mathbf{E}[e^{R/\lambda}] = \mathbf{E}[R] + \frac{\text{Var}(R)}{2\lambda} + O(\frac{1}{\lambda^2}) \approx \mathbf{E}[R]$ (justified for large λ).[9] To understand what it learns, we can write the gradient as

$$-\frac{1}{2\lambda}\nabla L_\psi^{\text{k1.5}}(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi} \left[\Big((R(\tau) - \mathbf{E}_\psi[R|s_1]) \nabla \log p_\theta(\tau|s_1) \Big) - \frac{\lambda}{2} \nabla \left(\log \frac{p_\theta(\tau|s_1)}{p_\psi(\tau|s_1)} \right)^2 \right]$$

The first part reduces to the REINFORCE gradient (3) (with constant baseline) when $\psi = \text{sg}(\theta)$. The second part is KL-like regularization, discouraging divergence.

7 Divergence Proximal Policy Optimization (DPPO)

DPPO (Qi *et al.*, 2026) introduces an adaptive clipping threshold $\epsilon_t = \frac{\epsilon}{p_\psi(x_t|s_t)}$ in the PPO objective (21):

$$J_\psi^{\text{dppo}}(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi} \left[\sum_{t=1}^{T} \min \left(\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t), \text{clip}_{1-\epsilon_t}^{1+\epsilon_t} \left(\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} \right) A_\psi(s_t, x_t) \right) \right] \tag{32}$$

Per-token clipping is still asymmetric (i.e., drop only in the correct direction), but now based on absolute probability change (contrast with (23–24)):

$$\Big(A_\psi(s_t, x_t) > 0 \;\bigwedge\; p_\theta(x_t|s_t) - p_\psi(x_t|s_t) > \epsilon \Big) \;\bigvee\; \Big(A_\psi(s_t, x_t) < 0 \;\bigwedge\; p_\psi(x_t|s_t) - p_\theta(x_t|s_t) > \epsilon \Big) \tag{33}$$

As a result, more low-confidence positive tokens are reinforced (exploration ↑ in the positive case) *and* more low-confidence negative tokens are suppressed (exploitation ↑ in the negative case).

7.1 TRPO Motivation

DPPO motivates absolute clipping $|P(x_t) - Q(x_t)| > \epsilon$ as a sample-dependent approximation of TRPO's original constraint $\text{TV}(P, Q) = \frac{1}{2}\sum_x |P(x) - Q(x)| \le \epsilon$. When P, Q are concentrated on x_t, the fake distributions P', Q' over $\{x_t, \varnothing\}$ with $P'(x_t) = P(x_t)$ and $Q'(x_t) = Q(x_t)$ are reasonable proxies. Then

$$\text{TV}(P', Q') = |P(x_t) - Q(x_t)| \le \text{TV}(P, Q)$$

While DPPO treats the LHS as an actual approximation of the RHS, note that it is technically a lower bound (by traingle inequality) so enforcing $\text{TV}(P', Q') \le \epsilon$ does not guarantee $\text{TV}(P, Q) \le \epsilon$.[10] Using this approximation to replace the state-wise TV constraint in TRPO (15) gives the step-wise surrogate constraint

$$\begin{aligned} \max_{\theta:\ \text{TV}(p_\psi(\cdot|s), p_\theta(\cdot|s)) \le \epsilon\ \forall s} \mathop{\mathbf{E}}_{\tau\sim p_\psi} \left[\sum_{t=1}^{T} \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t) \right] &\approx \max_{\theta:\ \text{TV}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t)) \le \epsilon\ \forall t} \sum_{t=1}^{T} \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t) \\ &\approx \max_{\theta:\ |p_\theta(x_t|s_t) - p_\psi(x_t|s_t)| \le \epsilon\ \forall t} \sum_{t=1}^{T} \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} A_\psi(s_t, x_t) \end{aligned}$$

[9] The expression can be verified using $e^x = 1 + x + \frac{x^2}{2} + O(x^3)$ and $\log(1+x) = x - \frac{x^2}{2} + O(x^3)$. But note that the general estimation problem of form $\log \mathbf{E}[e^{f(X)}]$ can be extremely sample-inefficient (McAllester and Stratos, 2020). Thus this approximation is more of a bias-variance tradeoff.

[10] It is easy to generalize this to include top-k tokens (i.e., support size in $\{k+2, k+1\}$) with a tighter bound, but the paper does not find it improve over just using x_t.

Promoting this constraint in the asymmetric style of PPO motivates the masked objective of form

$$\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T} M_t \times r_t A_t\right] \qquad M_t = \begin{cases} 0 & \text{if } (A_t > 0 \wedge p_{\theta,t} - p_{\psi,t} > \epsilon) \vee (A_t < 0 \wedge p_{\psi,t} - p_{\theta,t} > \epsilon) \\ 1 & \text{otherwise} \end{cases}$$

which is gradient-equivalent to the ratio-clipped objective (32).

7.2 Average Bound

The DPPO paper also gives an average bound on the TR surrogate objective's bias (Lemma C.6, Corollary C.7):

$$|\texttt{Bias}_\psi(\theta)| \le C \times \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T} \mathrm{TV}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t))\right] \le C' \times \sqrt{\mathrm{KL}_\psi(\theta)} \tag{34}$$

for some constants $C, C' > 0$. It retrospectively justifies the KL constraint in TRPO (17). It is not needed to motivate the per-step TV constraint in DPPO, and it is a weaker bound than TRPO's max bound (14) locally in the size of policy change (first-order, rather than second-order).[11]

8 Training Inference Mismatch (TIM)

Infra issues can cause a so-called (numerical) "TIM":

$$p_\psi^{\text{sampler}}(x|s) \neq p_\psi^{\text{trainer}}(x|s) \tag{35}$$

where the superscript marks if the rollout policy's token probability is computed by the sampler or the trainer. We delegate the causes of TIM to Section 8.1. This mismatch can create an extra layer of error. For instance, the basic score function identity breaks:

$$\mathop{\mathbf{E}}_{x\sim p_\psi^{\text{sampler}}(\cdot|s)}\left[\nabla \log p_\psi^{\text{trainer}}(x|s)\right] \neq 0$$

invalidating the justification for the baseline function in REINFORCE. A simple solution based on importance sampling:

$$\mathop{\mathbf{E}}_{x\sim p_\psi^{\text{sampler}}(\cdot|s)}\left[\frac{p_\psi^{\text{trainer}}(x|s)}{p_\psi^{\text{sampler}}(x|s)}\nabla \log p_\psi^{\text{trainer}}(x|s)\right] = \mathop{\mathbf{E}}_{x\sim p_\psi^{\text{trainer}}(\cdot|s)}\left[\nabla \log p_\psi^{\text{trainer}}(x|s)\right] = 0 \tag{36}$$

Note that this requires computing the rollout probabilities under both the sampler and the trainer.[12] This motivates applying an additional importance weight to per-token objectives, often with a guard function $\mathbf{G}$ to prevent extreme values from breaking optimization (e.g., different expert selections may yield a large ratio):

$$\mathbf{E}_\psi\left[\sum_{t=1}^{T} J_t\right] \longrightarrow \mathbf{E}_\psi\left[\sum_{t=1}^{T} \mathbf{G}\left(\frac{p_\psi^{\text{trainer}}(x|s)}{p_\psi^{\text{sampler}}(x|s)}\right) J_t\right]$$

Examples include

$$\mathbf{G}(r) = \begin{cases} \min(r, C) & \text{(truncated importance sampling (TIS), Yao } et\ al. \text{ (2025))} \\ [[C \le r \le C']]r & \text{(masked importance sampling (MIS), Team } et\ al. \text{ (2025b))} \end{cases}$$

Note that TIS caps importance weights, but MIS completely drops the token contribution outside a specified range.

[11] But the max bound is $O(T_{\max}^2\alpha^2)$ (Lemma C.5) whereas this bound is $O(T_{\max}\alpha)$ (Lemma C.6) where $\alpha = \max_s \mathrm{TV}(p_\theta(\cdot|s), p_\psi(\cdot|s))$, thus it may be numerically tighter if we account for the rollout length.

[12] The sampler probabilities are computed during generation anyway, so the controller can hand them off to the trainer along with the rollouts. The trainer probabilities can be computed by batched forward pass.

8.1 Causes of TIM

We list some known causes of TIM. Recall that addition and multiplication lose associativity in finite precision due to rounding errors, so any change in operation order can change the end result.

- **Kernel mismatch**: The sampler and trainer use different kernels (generation vs batched forward pass). Different kernels may differ in reduction order, choices of numerical approximation, or intermediate precision, even if they are implementing the same function.
- **Execution nondeterminism**: Some parallel operations (e.g., reduction with atomic updates) may execute in nondeterministic order and produce different results even with identical kernels, inputs, and hardware.
- **Batch dependency**: Some operations may depend on the given batch. This dependency can be mathematical (e.g., batchnorm, capacity-constrained expert routing) or numerical (e.g., different batch sizes may yield different matmul kernels).

Small numerical differences can also spike into more drastic changes. For instance, **expert mismatch** can happen from any of these issues, making the result very different for a mixture of experts.

9 Score Centering (SC)

SC (Marek and Ryabinin, 2026) presents an alternative to importance sampling for addressing policy staleness. The derivation assumes a practically motivated scenario where

1. We just sample from a wrong policy $\psi \neq \theta$ in the true policy gradient (3), and
2. We use a general advantage estimator $\widehat{A}_\psi(s_t, x_t) = R(\tau) - b(s_t)$ for $\tau \sim p_\psi$.

then tries to fix the resulting mismatch. That is,

$$\nabla J(\theta) = \sum_{t=1}^{T_{\max}} \mathop{\mathbf{E}}_{s\sim d_{\theta,t}} \left[\mathop{\mathbf{E}}_{x\sim p_\theta(\cdot|s)} \left[\widehat{A}_\theta(s,x) \nabla \log p_\theta(x|s) \right] \right] \approx \sum_{t=1}^{T_{\max}} \mathop{\mathbf{E}}_{s\sim d_{\psi,t}} \left[\mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)} \left[\widehat{A}_\psi(s,x) \nabla \log p_\theta(x|s) \right] \right] \tag{37}$$

The two sources of mismatch are: (i) wrong state distribution, (ii) wrong token distribution. The former is exactly the approximation in the TR surrogate; (37) reduces to $\nabla J_\psi(\theta)$ if we multiply each term by $\frac{p_\theta(x|s)}{p_\psi(x|s)}$. Instead, the paper proposes a different correction by comparing the true and false gradient side by side:

$$\mathop{\mathbf{E}}_{x\sim p_\theta(\cdot|s)} \left[\widehat{A}_\theta(s,x) \nabla \log p_\theta(x|s) \right] = \mathop{\mathrm{Cov}}_{x\sim p_\theta(\cdot|s)} \left(\widehat{A}_\theta(s,x), \nabla \log p_\theta(x|s) \right) \tag{38}$$

$$\mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)} \left[\widehat{A}_\psi(s,x) \nabla \log p_\theta(x|s) \right] = \mathop{\mathrm{Cov}}_{x\sim p_\psi(\cdot|s)} \left(\widehat{A}_\psi(s,x), \nabla \log p_\theta(x|s) \right) + \underbrace{\mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)} \left[\widehat{A}_\psi(s,x) \right] \mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)} \left[\nabla \log p_\theta(x|s) \right]}_{\text{“drift”}}$$

In the true gradient, the drift term vanishes since $\mathbf{E}_\psi[\nabla \log p_\theta(x|s)] = -\nabla \mathrm{KL}(p_\psi(\cdot|s), p_\theta(\cdot|s))$ becomes zero. But it *already* vanishes if $\widehat{A}_\psi(s,x) = A_\psi(s,x)$, thus the motivation only applies to non-exact advantage estimate (assumption 2). We can remove the drift term even when $\psi \neq \theta$ by centering the score function:[13]

$$\mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)} \left[\widehat{A}_\psi(s,x) \left(\nabla \log p_\theta(x|s) - \mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)} \left[\nabla \log p_\theta(x|s) \right] \right) \right] = \mathop{\mathrm{Cov}}_{x\sim p_\psi(\cdot|s)} \left(\widehat{A}_\psi(s,x), \nabla \log p_\theta(x|s) \right) \tag{39}$$

Note that this is still not equal to (38); SC only removes the drift and does not correct the remaining distribution mismatch. The gradient (39) can be computed with the surrogate objective:

$$J^{\mathrm{SC}}_\psi(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi} \left[\sum_{t=1}^{T} \widehat{A}_\psi(s_t, x_t) \left(\log p_\theta(x_t|s_t) - \mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s_t)} \left[\log p_\theta(x|s_t) \right] \right) \right] \tag{40}$$

[13] To motivate this, the paper considers the case $\widehat{A}_\psi(s,x) = C \neq 0$ is constant. Then the true gradient is zero but the new estimator is nonzero (since $C\mathbf{E}_\psi[\nabla \log p_\theta(x|s)] \neq 0$). SC can be seen as exercising an inductive bias “there should be nothing to learn here”.

9.1 Top-k Approximation

Given a rollout $x_t \sim p_\psi(\cdot|s_t)$ from the sampler, SC requires estimating the center (negative cross entropy)

$$C_{s_t}(\theta) = \sum_{x\in\mathcal{V}} p_\psi(x|s_t) \log p_\theta(x|s_t)$$

for each $t = 1 \dots T$. Naively this requires the sampler to send full-vocab logits for every token in the entire rollout batch. The paper proposes a top-k (e.g., 128) approximation that uses p_θ (with stop-gradient) to sketch the tail distribution since the latter is already available in a given minibatch. Letting $\mathcal{V} = \mathcal{H} \cup \mathcal{T}$ denote the partition,

$$\widehat{C}_{s_t}(\theta) = \sum_{x\in\mathcal{V}} q(x|s_t) \log p_\theta(x|s_t) \qquad\qquad q(x|s_t) = \begin{cases} p_\psi(x|s_t) & \text{if } x \in \mathcal{H} \\ \rho \text{sg}(p_\theta(x|s_t)) & \text{if } x \in \mathcal{T} \end{cases} \tag{41}$$

Solving for ρ to satisfy $\sum_{x\in\mathcal{V}} q(x|s_t) = 1$, we have $\rho = \frac{1-\sum_{x\in\mathcal{H}} p_\psi(x|s_t)}{1-\sum_{x\in\mathcal{H}} \text{sg}(p_\theta(x|s_t))}$ which can be easily computed per minibatch assuming the availability of the sampler's top-k probabilities. The paper goes a step further and note that the *gradient* of $\widehat{C}_{s_t}(\theta)$ only involves a top-k sum:

$$\begin{aligned} \nabla\widehat{C}_{s_t}(\theta) &= \sum_{x\in\mathcal{H}} p_\psi(x|s_t)\nabla \log p_\theta(x|s_t) + \rho \sum_{x\in\mathcal{T}} p_\theta(x|s_t)\nabla \log p_\theta(x|s_t) \\ &= \sum_{x\in\mathcal{H}} p_\psi(x|s_t)\nabla \log p_\theta(x|s_t) - \rho \sum_{x\in\mathcal{H}} p_\theta(x|s_t)\nabla \log p_\theta(x|s_t) \qquad (42) \\ &= \sum_{x\in\mathcal{H}} (p_\psi(x|s_t) - \rho p_\theta(x|s_t))\nabla \log p_\theta(x|s_t) \end{aligned}$$

where (42) uses the score function identity $\mathbf{E}_{x\sim p_\theta}[\nabla \log p_\theta(x)] = 0$. This yields the gradient-equivalent surrogate center

$$\widetilde{C}_{s_t}(\theta) = \sum_{x\in\mathcal{H}} \text{sg}\left(p_\psi(x|s_t) - \rho p_\theta(x|s_t)\right) \log p_\theta(x)$$

which only involves a sum over $\mathcal{H}$ and can be plugged into (40) to approximate $\mathbf{E}_{x\sim p_\psi(\cdot|s)}[\log p_\theta(x|s)]$.

9.2 Importance Sampling Hybrid

We first point out that applying IS in (37) recovers $\nabla J_\psi(\theta)$ (for $\widehat{A}_\psi \approx A_\psi$) and removes the need for score centering:

$$\mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)}\left[\frac{p_\theta(x|s)}{p_\psi(x|s)}\widehat{A}_\psi(s,x)\nabla \log p_\theta(x|s)\right] = \mathop{\mathbf{E}}_{x\sim p_\theta(\cdot|s)}\left[\widehat{A}_\psi(s,x)\nabla \log p_\theta(x|s)\right] = \mathop{\text{Cov}}_{x\sim p_\theta(\cdot|s)}\left(\widehat{A}_\psi(s,x), \nabla \log p_\theta(x|s)\right)$$

But it also creates the usual variance problem. The paper proposes truncating/masking the ratio $r \mapsto \mathbf{G}(r)$ to prevent extreme values (at the cost of bias), which *then* re-introduces the need for score centering because

$$\begin{aligned} &\mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)}\left[\mathbf{G}\left(\frac{p_\theta(x|s)}{p_\psi(x|s)}\right)\widehat{A}_\psi(s,x)\nabla \log p_\theta(x|s)\right] \qquad (43) \\ &= \mathop{\text{Cov}}_{x\sim p_\psi(\cdot|s)}\left(\widehat{A}_\psi(s,x), \mathbf{G}\left(\frac{p_\theta(x|s)}{p_\psi(x|s)}\right)\nabla \log p_\theta(x|s)\right) + \mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)}\left[\widehat{A}_\psi(s,x)\right] \mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s)}\left[\mathbf{G}\left(\frac{p_\theta(x|s)}{p_\psi(x|s)}\right)\nabla \log p_\theta(x|s)\right] \end{aligned}$$

Removing the drift from (43) yields the SC-IS gradient, which can be computed with the surrogate objective:

$$J_\psi^{\text{SC-IS}}(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T} \widehat{A}_\psi(s_t,x_t)\left(\text{sg}\left(\mathbf{G}\left(\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)}\right)\right)\log p_\theta(x_t|s_t) - \mathop{\mathbf{E}}_{x\sim p_\psi(\cdot|s_t)}\left[\text{sg}\left(\mathbf{G}\left(\frac{p_\theta(x|s_t)}{p_\psi(x|s_t)}\right)\right)\log p_\theta(x|s_t)\right]\right)\right] \tag{44}$$

The paper experiments with $\mathbf{G}(r) = \min(r, C)$ and $\mathbf{G}(r) = [[C \le r \le C']]r$ which are originally designed for TIM (TIS and MIS, Section 8). Experiments show that SC-IS improves over SC when the rollout policy is substantially stale (i.e., we make many updates within a rollout batch).

Top-k approximation. Again, an efficient center estimator can be obtained from the gradient. Using the top-k approximation in (41) (i.e., $q(x|s_t) = p_\psi(x|s_t)$ for $x \in \mathcal{H}$ and $q(x|s_t) = \rho \mathrm{sg}(p_\theta(x|s_t))$ for $x \in \mathcal{T}$), the gradient of the center term in (44) is

$$\begin{aligned}
&\sum_{x\in\mathcal{V}} q(x|s_t)\mathbf{G}\left(\frac{p_\theta(x|s_t)}{q(x|s_t)}\right)\nabla \log p_\theta(x|s_t) \\
&= \sum_{x\in\mathcal{H}} p_\psi(x|s_t)\mathbf{G}\left(\frac{p_\theta(x|s_t)}{p_\psi(x|s_t)}\right)\nabla \log p_\theta(x|s_t) + \sum_{x\in\mathcal{T}} \rho p_\theta(x|s_t)\mathbf{G}\left(\frac{p_\theta(x|s_t)}{\rho p_\theta(x|s_t)}\right)\nabla \log p_\theta(x|s_t) \\
&= \sum_{x\in\mathcal{H}} p_\psi(x|s_t)\mathbf{G}\left(\frac{p_\theta(x|s_t)}{p_\psi(x|s_t)}\right)\nabla \log p_\theta(x|s_t) - \rho\mathbf{G}\left(\frac{1}{\rho}\right)\sum_{x\in\mathcal{H}} p_\theta(x|s_t)\nabla \log p_\theta(x|s_t) \\
&= \sum_{x\in\mathcal{H}} \left(p_\psi(x|s_t)\mathbf{G}\left(\frac{p_\theta(x|s_t)}{p_\psi(x|s_t)}\right) - \rho\mathbf{G}\left(\frac{1}{\rho}\right)p_\theta(x|s_t)\right)\nabla \log p_\theta(x|s_t)
\end{aligned}$$

where we additionally use the fact that the guarded ratio becomes constant for the tail distribution. Thus the corresponding gradient-equivalent surrogate center is

$$\widetilde{C}^{\mathrm{IS}}_{s_t}(\theta) = \sum_{x\in\mathcal{H}} \mathrm{sg}\left(p_\psi(x|s_t)\mathbf{G}\left(\frac{p_\theta(x|s_t)}{p_\psi(x|s_t)}\right) - \rho\mathbf{G}\left(\frac{1}{\rho}\right)p_\theta(x|s_t)\right)\log p_\theta(x|s_t)$$

10 Advantage Estimation

We often need to estimate the advantage term

$$A_\theta(s_t, x_t) = \mathbf{E}_\theta[R|s_t, x_t] - \mathbf{E}_\theta[R|s_t] \tag{45}$$

for per-token weights in training objectives (e.g., (6)). This typically must be estimated from a single rollout (possibly truncated) with endpoint reward $R \in \mathbb{R}$ since it would be difficult to get another sample conditioned on these particular s_t and x_t.

10.1 Critic-Based

To amortize over samples, we can train a **critic** ϕ that predicts the value function of θ. Given a sample rollout $\tau_{1:K} \sim p_\theta$ ending in step $K \le T$ with endpoint reward $R(s_{K+1})$ (zero if truncated), the simplest training objective is to minimize the squared error for all states

$$\min_\phi \sum_{t=1}^{K} \left(V_\phi(s_t) - \widehat{\mathbf{E}}_\theta[R|s_t]\right)^2 \tag{46}$$

where $\widehat{\mathbf{E}}_\theta[R|s_t] \approx \mathbf{E}_\theta[R|s_t]$ is any estimator computable from this rollout. A straightforward choice is

$$\widehat{\mathbf{E}}_\theta[R|s_t] = R(s_{K+1}) + \nu(s_{K+1})V_{\mathrm{sg}(\phi)}(s_{K+1}) \tag{47}$$

where $\nu(s_{K+1}) = [[s_{K+1} \text{ is nonterminal}]]$ (i.e., we bootstrap the value at s_{K+1} if needed). Note that every $t = 1 \dots K$ gets the same estimate from the shared rollout endpoint. The critic ϕ is trained by (46) simultaneously with the main model θ (optionally with a warmup), during which it can provide a running estimate of (45) as

$$\widehat{A}_{\theta,t} = R(s_{K+1}) + \nu(s_{K+1})V_\phi(s_{K+1}) - V_\phi(s_t) \tag{48}$$

We view (47) and (48) as the default critic target and advantage estimator. This is equal to using GAE with $\lambda = 1$.

10.1.1 Generalized advantage estimator (GAE)

Upon inspection, we can see that (48) is the lowest-bias and highest-variance instance of a series of estimators

$$\widehat{A}^{(k)}_{\theta,t} = R(s_{t+k}) + \nu(s_{t+k})V_\phi(s_{t+k}) - V_\phi(s_t)$$

More explicitly, at each t, we have $K-t+1$ estimators with decreasing bias (less dependent on ϕ) and increasing variance (further out):

$$\begin{aligned}
\widehat{A}^{(1)}_{\theta,t} &= V_\phi(s_{t+1}) - V_\phi(s_t) && \cdots && \widehat{A}^{(K-t)}_{\theta,t} = V_\phi(s_K) - V_\phi(s_t) \\
\widehat{A}^{(2)}_{\theta,t} &= V_\phi(s_{t+2}) - V_\phi(s_t) && \cdots && \widehat{A}^{(K-t+1)}_{\theta,t} = R(s_{K+1}) + \nu(s_{K+1})V_\phi(s_{K+1}) - V_\phi(s_t) = (48)
\end{aligned}$$

We could trade off bias and variance by picking different k. Instead, GAE (Schulman *et al.*, 2015a) proposes to introduce a knob $\lambda \in [0,1]$ and take the exponential average of all estimators:[14]

$$\widehat{A}^{\mathrm{GAE}}_{\theta,t}(\lambda) = (1-\lambda)\widehat{A}^{(1)}_{\theta,t} + (1-\lambda)\lambda\widehat{A}^{(2)}_{\theta,t} + \cdots + (1-\lambda)\lambda^{K-t-1}\widehat{A}^{(K-t)}_{\theta,t} + \lambda^{K-t}\widehat{A}^{(K-t+1)}_{\theta,t} \tag{49}$$

(check the extremes at $\lambda = 0$ and $\lambda = 1$). A naive implementation would take $O(K^2)$ time to obtain all estimates for the trajectory (i.e., for each t, compute $\widehat{A}^{(k)}_{\theta,t}$ for all k). GAE uses $O(K)$-time dynamic programming that exploits shared parts between consecutive estimators. Define the value deltas

$$\begin{aligned}
d_t &= V_\phi(s_{t+1}) - V_\phi(s_t) && \cdots && d_{K-1} = V_\phi(s_K) - V_\phi(s_{K-1}) \\
d_{t+1} &= V_\phi(s_{t+2}) - V_\phi(s_{t+1}) && \cdots && d_K = R(s_{K+1}) + \nu(s_{K+1})V_\phi(s_{K+1}) - V_\phi(s_K)
\end{aligned}$$

All deltas $d_1 \dots d_K$ can be precomputed in $O(K)$ time. We can verify that

$$\begin{aligned}
\widehat{A}^{\mathrm{GAE}}_{\theta,t}(\lambda) &= d_t + \lambda d_{t+1} + \cdots + \lambda^{K-t-1}d_{K-1} + \lambda^{K-t}d_K \\
&= d_t + \lambda\widehat{A}^{\mathrm{GAE}}_{\theta,t+1}(\lambda)
\end{aligned}$$

The last expression gives a natural $O(K)$ backward pass to compute all estimates starting from $\widehat{A}^{\mathrm{GAE}}_{\theta,K}(\lambda) = d_K$. GAE also suggests an alternative value estimator for the critic's training objective (46)

$$\widehat{\mathbf{E}}_\theta[R|s_t] = \widehat{A}^{\mathrm{GAE}}_{\theta,t}(\lambda) + V_{\mathrm{sg}(\phi)}(s_t) \tag{50}$$

where $\widehat{A}^{\mathrm{GAE}}_{\theta,t}(\lambda)$ is computed by (49) using the frozen $\mathrm{sg}(\phi)$. If $\lambda = 1$, it degenerates to (47) and therefore unbiased (assuming perfect bootstrapping). For $\lambda < 1$, it is generally biased but yields step-dependent estimates unlike (47), thus potentially lowering variance.

10.2 Critic-Free

Group Relative Policy Optimization (GRPO) (Shao *et al.*, 2024) popularized a critic-free approach. It assumes a "group" of G rollouts per prompt and approximates for the i-th rollout

$$A^{(i)}_\psi(s_t, x_t) = \mathbf{E}_\psi[R|s_t, x_t] - \mathbf{E}_\psi[R|s_t] \approx \frac{R^{(i)} - \mathbf{E}_\psi[R|s_1]}{\mathrm{stdev}_\psi(R|s_1)} \qquad \forall t = 1 \dots T^{(i)} \tag{51}$$

where the mean and standard deviation are estimated from the group. For instance, in $G = 16$ rollouts with binary rewards $R^{(i)} \in \{0,1\}$, GRPO would assign the following empirical estimate:

prompt difficulty	impossible	trivial	hard	easy	medium
success rate (out of 16)	0	16	1	15	8
reward stdev	0	0	0.25	0.25	0.516
successful rollout per-step advantage	—	0	3.75	0.25	0.97
failed rollout per-step advantage	0	—	−0.25	−3.75	−0.97

We see that it captures the advantage at a rollout-level (e.g., even if a rollout succeeds, there is little advantage if most rollouts do). But since it does not distinguish individual steps (i.e., it conditions on s_1 not s_t), it is not a correct estimator. Dr. GRPO (Liu *et al.*, 2025) simplifies it further by removing the scaling:

$$A^{(i)}_\psi(s_t, x_t) = \mathbf{E}_\psi[R|s_t, x_t] - \mathbf{E}_\psi[R|s_t] \approx R^{(i)} - \mathbf{E}_\psi[R|s_1] \qquad \forall t = 1 \dots T^{(i)} \tag{52}$$

The estimator is still invalid for advantage, but the expected policy gradient is at least correct since $\mathbf{E}_\psi[\sum_t (R - \mathbf{E}_\psi[R|s_1])\nabla \log p_{\psi,t}] = \nabla J(\psi)$.

[14] The generative process is "continue with probability λ". The mass sums to $(1-\lambda)\sum_{k=0}^{K-t-1}\lambda^k + \lambda^{K-t} = 1$.

References

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274.

Kakade, S. M. (2001). A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, pages 1531–1538.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. (2025). Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Marek, M. and Ryabinin, M. (2026). Score centering stabilizes off-policy reinforcement learning. *arXiv preprint arXiv:2609.20807*.

McAllester, D. and Stratos, K. (2020). Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR.

Qi, P., Zhou, X., Liu, Z., Pang, T., Du, C., Lin, M., and Lee, W. S. (2026). Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015a). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015b). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., *et al.* (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., *et al.* (2025a). Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Team, L., Shen, A., Li, B., Hu, B., Jing, B., Chen, C., Huang, C., Zhang, C., Yang, C., Lin, C., *et al.* (2025b). Every step evolves: Scaling reinforcement learning for trillion-scale thinking model. *arXiv preprint arXiv:2510.18855*.

Tomar, M., Shani, L., Efroni, Y., and Ghavamzadeh, M. (2020). Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, **8**(3-4), 229–256.

Yao, F., Liu, L., Zhang, D., Dong, C., Shang, J., and Gao, J. (2025). On the rollout-training mismatch in modern rl systems. In *OPT 2025: Optimization for Machine Learning*.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., *et al.* (2026). Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, **38**, 113222–113244.

A Markov Decision Process (MDP)

Let p_θ denote an LLM that generates a token $x \in \mathcal{V}$ with arbitrary conditioning. The model navigates an **environment** by state transition $(s, x) \mapsto s'$ (possibly stochastic) where a state $s \in \mathcal{S}$ is a full history of the interactions. To illustrate, a Unix environment could start with an initial state like

$s_1 =$ <system> You may use these tools: terminal(command: string): Execute a shell command. </system>
<user> How many files have write permissions in the current directory? </user>
<assistant>

The model may decide to use the terminal tool and generate, token by token,

$$(x_1 \dots x_t) = \boxed{\text{Let me look. <tool> \{“name”: “terminal”, “args”: \{“command”: “ls -l”\}\} </tool> </assistant>}}$$

with state transition $s_{i+1} = (s_i, x_i)$. Upon $x_t =$ **</assistant>**, the controller inspects the completed message to (i) print *Let me look.*, (ii) execute `ls -l`, and (iii) append the observation to the next state, e.g.,

$$s_{t+1} = \boxed{\begin{array}{l}\text{<system> You may use these tools: terminal(command: string): Execute a shell command. </system>}\\ \text{<user> How many files have write permissions in the current directory? </user>}\\ \text{<assistant> Let me look. <tool> \{“name”: “terminal”, “args”: \{“command”: “ls -l”\}\} </tool> </assistant>}\\ \textbf{<observation> -rw-r--r-- file1\textbackslash n-rwxr-xr-x file2 </observation> <assistant>}\end{array}}$$

The model goes on until it stops tool calling (e.g., just prints “There are 3 files with write permissions”). The user may engage and continue the session (“That’s wrong, check again”).

A.1 Training

For training purposes, the environment additionally supplies the knowledge of

- *Endpoint step* $K \geq 1$: Either s_{K+1} is a terminal state (**terminated**), or the rollout has reached a specified length/time limit (**truncated**).[15] Using T to denote the (random) termination step of the full trajectory, we have $K = T$ if terminated and $K < T$ if truncated.
- *Endpoint reward* $R(s_{K+1}) \in \mathbb{R}$: The reward is naturally binary in many verifiable tasks, but can also be more granular (e.g., rubric-based score by an LLM judge). We assume the reward is zero for nonterminal states, so $R(s_{K+1}) = 0$ if truncated.

B KL Estimation

From (16), it is sufficient to consider estimating the KL between two conditional token distributions

$$\mathrm{KL}(p, q) = \mathop{\mathbf{E}}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]$$

Define the random variable $r = \frac{q(x)}{p(x)}$ where $x \sim p$. An unbiased estimator is obviously

$$k_1 = -\log r \tag{53}$$

John Schulman popularized an alternative estimator by this blog:

$$k_3 = r - 1 - \log r \tag{54}$$

which remains unbiased simply by the virtue of $\mathbf{E}[r] = 1$. The idea is that $r - 1$ offsets the first-order effect of $\log r$ around $r = 1$. Write $r = 1 + u$ and Taylor expand both around $u = 0$:

$$k_1 = -u + \frac{u^2}{2} + O(u^3) = O(u) \qquad\qquad k_3 = \frac{u^2}{2} + O(u^3) = O(u^2)$$

So k_3 is much flatter than k_1 around $u = 0$, implying lower variance in the regime $p \approx q$. For completeness, Schulman considers one other (biased) estimator:[16]

$$k_2 = \frac{1}{2}(\log r)^2 \tag{55}$$

[15] A natural condition for termination is if $x_K =$ </assistant> and the response contains no tool call or simulated user followup. But it can also be domain-specific, e.g., generating an explicit `submit` action for coding.

[16] Although Schulman motivates (55) via f-divergence, there is no need. It can be directly seen as a quadratic approximation of k_3, since $k_3 = e^z - 1 - z \approx \frac{z^2}{2}$ where $z = \log r$. This is why k_2 has lower variance than k_1 around $p \approx q$ but is biased unlike k_3.

C Lemmas

Lemma C.1. The gradient of $J(\theta) = \mathbf{E}_\theta[R(s_{T+1})]$ can be written as $\mathbf{E}_\theta[\sum_{t=1}^T (R(s_{T+1}) - b(s_t))\nabla \log p_\theta(x_t|s_t)]$.

Proof. Let $J_{s_1}(\theta) = \mathbf{E}_{\tau\sim p_\theta(\cdot|s_1)}[R(s_{T+1})]$ for any fixed s_1. Then

$$\begin{aligned}
\nabla J_{s_1}(\theta) = \sum_\tau R(s_{T+1})\nabla p_\theta(\tau|s_1) &= \sum_\tau R(s_{T+1}) p_\theta(\tau|s_1)\nabla \log p_\theta(\tau|s_1) && \text{(log-derivative trick)} \quad (56)\\
&= \mathop{\mathbf{E}}_{\tau\sim p_\theta(\cdot|s_1)}[R(s_{T+1})\nabla \log p_\theta(\tau|s_1)] \\
&= \mathop{\mathbf{E}}_{\tau\sim p_\theta(\cdot|s_1)}\left[\sum_{t=1}^T R(s_{T+1})\nabla \log p_\theta(x_t|s_t)\right] \\
&= \mathop{\mathbf{E}}_{\tau\sim p_\theta(\cdot|s_1)}\left[\sum_{t=1}^T (R(s_{T+1}) - b(s_t))\nabla \log p_\theta(x_t|s_t)\right] && \text{(baseline function)}
\end{aligned}$$

The last step uses a well-known property of the score function along with the fact that $b(s_t)$ is independent of x_t:

$$\begin{aligned}
\mathop{\mathbf{E}}_{\tau\sim p_\theta(\cdot|s_1)}\left[\sum_{t=1}^T b(s_t)\nabla \log p_\theta(x_t|s_t)\right] &= \mathop{\mathbf{E}}_{\tau\sim p_\theta(\cdot|s_1)}\left[\sum_{t=1}^T \mathop{\mathbf{E}}_{x_t\sim p_\theta(\cdot|s_t)}[b(s_t)\nabla \log p_\theta(x_t|s_t)]\right] \\
&= \mathop{\mathbf{E}}_{\tau\sim p_\theta(\cdot|s_1)}\left[\sum_{t=1}^T b(s_t)\underbrace{\mathop{\mathbf{E}}_{x_t\sim p_\theta(\cdot|s_t)}[\nabla \log p_\theta(x_t|s_t)]}_{0_d}\right] = 0_d
\end{aligned}$$

Taking $\nabla J(\theta) = \mathbf{E}_{s_1\sim\mathbf{env}}[\nabla J_{s_1}(\theta)]$ gives the result. □

Lemma C.2. Holding s_t fixed, we have

$$b^\star = \mathop{\arg\min}_{b\in\mathbb{R}} \operatorname{tr}\left(\operatorname{Cov}\left(\underbrace{(R(s_{T+1}) - b)\nabla \log p_\theta(x_t|s_t)}_{X}\right)\right) = \frac{\mathbf{E}[R(s_{T+1})\,||\nabla \log p_\theta(x_t|s_t)||_2^2]}{\mathbf{E}[||\nabla \log p_\theta(x_t|s_t)||_2^2]} \quad (57)$$

Proof. Recall the total variance of $X \in \mathbb{R}^d$ is computed as $\operatorname{tr}(\operatorname{Cov}(X)) = \operatorname{tr}\left(\mathbf{E}[XX^\top]\right) - \operatorname{tr}\left(\mathbf{E}[X]\mathbf{E}[X]^\top\right) = \mathbf{E}[\|X\|_2^2] - \|\mathbf{E}[X]\|_2^2$. Here we have the form $X = (R-b)g$ where $\mathbf{E}[X] = \mathbf{E}[Rg] - b\mathbf{E}[g] = \mathbf{E}[Rg]$ is independent of the optimization parameter, so the minimization problem (57) is equivalent to

$$b^\star = \mathop{\arg\min}_{b\in\mathbb{R}} \operatorname{tr}(\operatorname{Cov}((R-b)g)) = \mathop{\arg\min}_{b\in\mathbb{R}} \mathbf{E}[(R-b)^2\|g\|_2^2] = \frac{\mathbf{E}[R\|g\|_2^2]}{\mathbf{E}[\|g\|_2^2]}$$

□

Lemma C.3.

$$1 - \sum_x \min(P(x), Q(x)) = \min_{(X,Y):\ X\sim P,\ Y\sim Q} \Pr(X \neq Y) \quad (58)$$

Proof. Consider any (X, Y) with marginals $X \sim P$ and $Y \sim Q$. We have

$$\Pr(X = Y) = \sum_x \Pr(X = Y = x) \le \sum_x \min(P(x), Q(x)) \quad \Rightarrow \quad 1 - \sum_x \min(P(x), Q(x)) \le \Pr(X \neq Y) \quad (59)$$

Now we construct such (X, Y) that achieves equality, which shows (58). Let $m = \sum_x \min(P(x), Q(x))$.

- With probability m, draw $X = Y = x$ proportional to $\min(P(x), Q(x))$.
- With probability $1 - m$

– Draw $X = x$ proportional to $P(x) - \min(P(x), Q(x))$, and
– Draw $Y = y$ proportional to $Q(y) - \min(P(y), Q(y))$.

Crucially, note that $X \neq Y$ in this case.

By construction, this defines a valid joint distribution over (X, Y) with $\Pr(X \neq Y) = 1 - m$. Thus we only need to verify they have the correct marginals:

$$\Pr(X = x) = m \times \frac{\min(P(x), Q(x))}{\sum_x \min(P(x), Q(x))} + (1 - m) \times \frac{P(x) - \min(P(x), Q(x))}{\sum_x (P(x) - \min(P(x), Q(x)))} = P(x)$$

Similarly $\Pr(Y = x) = Q(x)$. □

Lemma C.4. $\mathrm{TV}(d_{\theta,t}, d_{\psi,t}) \leq (t-1)\alpha$ where $\alpha = \max_s \mathrm{TV}(p_\theta(\cdot|s), p_\psi(\cdot|s))$.

Proof. By (59)

$$\mathrm{TV}(d_{\theta,t}, d_{\psi,t}) \leq \Pr(s_{\theta,t} \neq s_{\psi,t})$$

where $s_{\theta,t} \sim d_{\theta,t}$ and $s_{\psi,t} \sim d_{\psi,t}$. By the union bound

$$\Pr(s_{\theta,t+1} \neq s_{\psi,t+1}) \leq \Pr(s_{\theta,t} \neq s_{\psi,t}) + \Pr(s_{\theta,t} = s_{\psi,t} \wedge x_{\theta,t} \neq x_{\psi,t})$$

For the last term, WLOG we may characterize $x_{\theta,t}, x_{\psi,t}$ by the maximum coupling in Lemma C.3 whenever $s_{\theta,t} = s_{\psi,t} = s$ since they have the correct conditional marginals $p_\theta(\cdot|s)$ and $p_\psi(\cdot|s)$. Then

$$\begin{aligned}\Pr(s_{\theta,t} = s_{\psi,t} \wedge x_{\theta,t} \neq x_{\psi,t}) &= \sum_s \Pr(s_{\theta,t} = s_{\psi,t} = s) \Pr(x_{\theta,s} \neq x_{\psi,s} | s_{\theta,t} = s_{\psi,t} = s) \\ &= \sum_s \Pr(s_{\theta,t} = s_{\psi,t} = s) \underbrace{\mathrm{TV}(p_\theta(\cdot|s), p_\psi(\cdot|s))}_{\leq \alpha} \leq \alpha\end{aligned}$$

Since $s_{\theta,1} = s_{\psi,1}$, it follows $\Pr(s_{\theta,t} \neq s_{\psi,t}) \leq (t-1)\alpha$. □

Lemma C.5. Let $\alpha = \max_s \mathrm{TV}(p_\theta(\cdot|s), p_\psi(\cdot|s))$ and assume $|R| \leq \xi$ for some constant $\xi \geq 0$. Then

$$|(J(\theta) - J(\psi)) - J_\psi(\theta)| \leq 4\xi T_{\max}(T_{\max} - 1)\alpha^2$$

Proof. The common factor between $J(\theta) - J(\psi)$ and $J_\psi(\theta)$ is the advantage $f(s) = \mathop{\mathbf{E}}_{x \sim p_\theta(\cdot|s)}[A_\psi(s, x)]$, where by (13)

$$|f(s)| = \left| \mathop{\mathbf{E}}_{x \sim p_\theta(\cdot|s)}[A_\psi(s,x)] - \mathop{\mathbf{E}}_{x \sim p_\psi(\cdot|s)}[A_\psi(s,x)] \right| \leq 2 \underbrace{||A_\psi(s,\cdot)||_\infty}_{\leq 2\xi} \underbrace{\mathrm{TV}(p_\theta(\cdot|s), p_\psi(\cdot|s))}_{\leq \alpha} \leq 4\xi\alpha$$

Expressing the difference in the common factor, again by (13)

$$|(J(\theta) - J(\psi)) - J_\psi(\theta)| \leq \sum_{t=1}^{T_{\max}} \left| \mathop{\mathbf{E}}_{s \sim d_{\theta,t}}[f(s)] - \mathop{\mathbf{E}}_{s \sim d_{\psi,t}}[f(s)] \right| \leq 2 \underbrace{||f||_\infty}_{\leq 4\xi\alpha} \sum_{t=1}^{T_{\max}} \underbrace{\mathrm{TV}(d_{\theta,t}, d_{\psi,t})}_{\leq (t-1)\alpha} \leq 4\xi T_{\max}(T_{\max} - 1)\alpha^2$$

where $\mathrm{TV}(d_{\theta,t}, d_{\psi,t}) \leq (t-1)\alpha$ by Lemma C.4. □

Lemma C.6. Let $\alpha = \max_s \mathrm{TV}(p_\theta(\cdot|s), p_\psi(\cdot|s))$ and assume $|R| \leq \xi$ for some constant $\xi \geq 0$. Then

$$|J_\psi(\theta) - (J(\theta) - J(\psi))| \leq 4\xi \mathop{\mathbf{E}}_{\tau \sim p_\psi} \left[\sum_{t=1}^{T} \mathrm{TV}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t)) \right] \leq 4\xi T_{\max}\alpha \tag{60}$$

Proof. Let $r_t = \frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)}$, which is deterministic given s_t and x_t but random given only s_t with the expectation $\mathbf{E}_\psi[r_t|s_t] = 1$. We can write

$$J(\theta) - J(\psi) = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\frac{p_\theta(\tau|s_1)}{p_\psi(\tau|s_1)}R - R\right] = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[R\left(\prod_{t=1}^{T} r_t - 1\right)\right]$$

and also

$$\begin{aligned} J_\psi(\theta) = \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T} r_t\Big(\mathbf{E}_\psi[R|s_t,x_t] - \mathbf{E}_\psi[R|s_t]\Big)\right] &= \sum_{t=1}^{T_{\max}} \mathbf{E}_\psi[\mathbf{E}_\psi[r_t R|s_t,x_t] - \mathbf{E}_\psi[r_t|s_t]\mathbf{E}_\psi[R|s_t]] \\ &= \sum_{t=1}^{T_{\max}} \mathbf{E}_\psi[\mathbf{E}_\psi[r_t R|s_t,x_t] - \mathbf{E}_\psi[R|s_t]] \\ &= \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[R\sum_{t=1}^{T}(r_t-1)\right] \end{aligned}$$

So they differ by prod-vs-sum and can be related through the telescoping product identity:

$$\prod_{t=1}^{T} r_t - 1 = \sum_{t=1}^{T}(r_t - 1)\prod_{j=t+1}^{T} r_j$$

Plugging this in, we have by triangle inequality

$$|J_\psi(\theta) - (J(\theta) - J(\psi))| = \left|\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[R\sum_{t=1}^{T}(r_t-1)\left(1 - \prod_{j=t+1}^{T} r_j\right)\right]\right| \le \xi \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\left|r_t - 1\right|\left|1 - \prod_{j=t+1}^{T} r_j\right|\right]$$

Here we can use the law of iterated expectations (LIE) with $|1-z| \le 1 + |z|$ to bound[17]

$$\mathbf{E}_\psi\left[\left|1 - \prod_{j=t+1}^{T} r_j\right|\middle| s_{t+1}\right] \le \mathbf{E}_\psi\left[1 + \prod_{j=t+1}^{T} r_j\middle| s_{t+1}\right] \le 2$$

resulting in

$$|J_\psi(\theta) - (J(\theta) - J(\psi))| \le 2\xi \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}|r_t - 1|\right] = 4\xi \mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\mathrm{TV}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t))\right]$$

where the last equality uses another round of LIE

$$\mathbf{E}_\psi[|r_t - 1|\,|s_t] = \sum_{x_t} p_\psi(x_t|s_t)\left|\frac{p_\theta(x_t|s_t)}{p_\psi(x_t|s_t)} - 1\right| = \sum_{x_t}|p_\theta(x_t|s_t) - p_\psi(x_t|s_t)| = 2\mathrm{TV}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t))$$

□

Corollary C.7. Assume $|R| \le \xi$ for some constant $\xi \ge 0$. Then

$$|J_\psi(\theta) - (J(\theta) - J(\psi))| \le 4\xi\sqrt{\frac{T_{\max}}{2}\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\mathrm{KL}(p_\psi(\cdot|s_t), p_\theta(\cdot|s_t))\right]} = 4\xi\sqrt{\frac{T_{\max}}{2}\mathrm{KL}_\psi(\theta)}$$

We may also switch the direction of KL inside the expectation without losing correctness.

[17] This is the loose part, since we bound this by 2 ignoring the ratios even when they are close to 1.

Proof. We can bound the expectation in (60) (Lemma C.6) as

$$\begin{aligned}
\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\mathrm{TV}(p_\psi(\cdot|s_t),p_\theta(\cdot|s_t))\right] &\le \sqrt{\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\left(\sum_{t=1}^{T}\mathrm{TV}(p_\psi(\cdot|s_t),p_\theta(\cdot|s_t))\right)^2\right]} && \text{(Jensen/Cauchy-Schwarz)}\\
&\le \sqrt{\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[T\left(\sum_{t=1}^{T}\mathrm{TV}(p_\psi(\cdot|s_t),p_\theta(\cdot|s_t))^2\right)\right]} && \text{(Cauchy-Schwarz)}\\
&\le \sqrt{\frac{T_{\max}}{2}\mathop{\mathbf{E}}_{\tau\sim p_\psi}\left[\sum_{t=1}^{T}\mathrm{KL}(p_\psi(\cdot|s_t),p_\theta(\cdot|s_t))\right]} && \text{(Pinsker)}
\end{aligned}$$

Note that Pinsker could choose the other KL direction per token, but the rollout comes from ψ. □