Expectation–Maximization Formulations for SFT and RL Fine-Tuning of Transformers

February 18, 2026

Contents

1 Introduction

Transformer language models are commonly adapted to downstream tasks via supervised fine-tuning (SFT), and further improved via RL fine-tuning against a learned or human preference reward. While these procedures are usually presented as distinct (cross-entropy training versus policy optimization), both can be interpreted as alternating between constructing a training target distribution and then fitting the model to that target.

2 Research Questions

1. Can supervised fine-tuning and KL-regularized RL fine-tuning be expressed under a common EM/MAP formulation with a shared E-step/M-step interpretation?
2. In what precise sense does V-MPO correspond to regularized policy iteration, and how does that differ from direct policy-gradient optimization (e.g. PPO)?
3. Does adaptive temperature optimization in the E-step provide a practical stability advantage over fixed-temperature weighting at LLM scale?
4. How does the DAR closed-form target relate to the V-MPO target, and under which limits are they equivalent?
5. Do EM-style weighted-MLE updates provide practical benefits for transformer fine-tuning, including compatibility with dropout-style regularization?

3 Preliminaries

Maximum Likelihood and Maximum a Posteriori

Given observed data x and parameters θ, maximum likelihood (ML) estimation solves

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} \log p_\theta(x).$$

Maximum a posteriori (MAP) estimation adds a prior:

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta} (\log p_\theta(x) + \log p(\theta)).$$

In optimization form, $-\log p(\theta)$ acts as a regularizer. A flat prior recovers ML.

EM Lower Bound

Assume a latent variable model with latent z. For any auxiliary distribution $q(z)$:

$$\log p_\theta(x) = \underbrace{\mathbb{E}_{q(z)}[\log p_\theta(x, z) - \log q(z)]}_{\mathcal{L}(q,\theta)} + \mathrm{KL}(q(z) \,\|\, p_\theta(z \mid x)).$$

Since KL is nonnegative, $\mathcal{L}(q, \theta)$ is a lower bound on $\log p_\theta(x)$.
For MAP, the objective

$$F_{\mathrm{MAP}}(\theta) = \log p_\theta(x) + \log p(\theta)$$

admits the decomposition

$$F_{\mathrm{MAP}}(\theta) = \mathcal{L}_{\mathrm{MAP}}(q, \theta) + \mathrm{KL}(q(z) \,\|\, p_\theta(z \mid x)),$$

with

$$\mathcal{L}_{\mathrm{MAP}}(q, \theta) = \mathbb{E}_{q(z)}[\log p_\theta(x, z) - \log q(z)] + \log p(\theta).$$

EM Iterations

At iteration k, EM alternates:

$$\text{E-step:} \quad q_{k+1}(z) = p_{\theta_k}(z \mid x),$$
$$\text{M-step:} \quad \theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\text{MAP}}(q_{k+1}, \theta).$$

Equivalently, the MAP M-step maximises

$$Q_{\text{MAP}}(\theta, \theta_k) = \mathbb{E}_{z \sim p_{\theta_k}(z|x)}[\log p_\theta(x, z)] + \log p(\theta).$$

This gives the standard monotonic-improvement template: construct a target distribution in the E-step, then fit a parametric model to that target in the M-step.

EM-Style Policy Iteration vs Policy Gradients

Policy-gradient update (direct parameter-space step).

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)].$$

A policy-gradient method takes a local ascent step

$$\theta_{k+1} = \theta_k + \lambda \widehat{\nabla_\theta J(\theta_k)},$$

optionally with clipping or explicit KL penalties.

EM-style policy iteration (distribution-space then projection). EM-style methods introduce an auxiliary improved policy $q(a \mid s)$ and alternate:

$$\text{E-step:} \quad q_{k+1} = \arg\max_{q} \Big(\mathbb{E}_{s \sim d_{\pi_k},\, a \sim q}[A^{\pi_k}(s, a)] - \eta \, \text{KL}(q(\cdot \mid s) \,\|\, \pi_k(\cdot \mid s)) \Big),$$

whose solution has Boltzmann form

$$q_{k+1}(a \mid s) \propto \pi_k(a \mid s) \, \exp\Big(\frac{A^{\pi_k}(s, a)}{\eta} \Big).$$

Then the M-step projects back to the parametric family:

$$\theta_{k+1} = \arg\max_{\theta} \; \mathbb{E}_{s \sim d_{\pi_k},\, a \sim q_{k+1}}[\log \pi_\theta(a \mid s)],$$

often with an additional trust-region term on $\text{KL}(\pi_k \| \pi_\theta)$.

Policy gradients optimize parameters directly via noisy first-order steps. EM-style methods perform *policy improvement* in distribution space first, then do weighted maximum-likelihood fitting. This is why V-MPO is naturally viewed as regularized policy iteration in EM form rather than as a pure policy-gradient method.

4 PPO

Proximal Policy Optimisation (PPO) is an on-policy actor-critic algorithm that constrains each policy update to stay close to the behaviour policy via a clipped surrogate objective, avoiding the instability of unconstrained policy gradient steps.

A trajectory $\tau = (s_0, a_0, \ldots, s_{T-1}, a_{T-1})$ is collected under the current policy $\pi_{\theta_{\text{old}}}$. Per-step log-probabilities and their sum are

$$\ell_{\theta,t} = \log \pi_\theta(a_t \mid s_t), \qquad \ell_\theta(\tau) = \sum_{t=0}^{T-1} \ell_{\theta,t}.$$

The discounted reward-to-go from step t is

$$R_t = \sum_{k=t}^{T-1} \gamma^{k-t} r_k.$$

Policy Gradient Foundation

The policy gradient theorem gives the direction of steepest ascent for the expected return:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, A_t\right].$$

To reuse data collected under $\pi_{\theta_{\text{old}}}$, importance sampling introduces the per-step probability ratio

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} = \exp(\ell_{\theta,t} - \ell_{\text{old},t})\,.$$

The unclipped surrogate objective is then $L^{\text{PG}}(\theta) = \mathbb{E}_t[r_t(\theta)\, A_t]$, but without further constraint this can lead to destructively large updates.

Clipped Surrogate Objective

PPO resolves this by clipping the ratio to $[1-\epsilon,\, 1+\epsilon]$ and taking the pessimistic (minimum) of the clipped and unclipped terms:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min(r_t(\theta)\, A_t,\ \text{clip}(r_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\, A_t)\right].$$

When the advantage is positive the update is capped at a ratio of $1+\epsilon$; when negative it is capped at $1-\epsilon$. This prevents the policy from moving too far in either direction within a single update.

Value Function and Entropy

The critic is fitted by minimising a squared regression loss to the empirical returns:

$$L^{\text{VF}}(\phi) = \mathbb{E}_t\left[\left(V_\phi(s_t) - R_t\right)^2\right].$$

An entropy bonus encourages exploration by penalising premature policy collapse:

$$S(\pi_\theta(\cdot \mid s_t)) = -\sum_a \pi_\theta(a \mid s_t) \log \pi_\theta(a \mid s_t).$$

Full Objective

The three terms are combined into a single objective (to minimise):

$$\mathcal{L}(\theta,\phi) = -L^{\text{CLIP}}(\theta) + c_1\, L^{\text{VF}}(\phi) - c_2\, \mathbb{E}_t[S(\pi_\theta(\cdot \mid s_t))]\,,$$

where c_1 and c_2 are scalar coefficients balancing the three losses.

Generalised Advantage Estimation (GAE)

Rather than using raw Monte Carlo returns to estimate A_t, PPO typically uses GAE, which trades off bias and variance via a decay parameter $\lambda \in [0,1]$.

The TD residual at each step is

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t).$$

GAE accumulates these residuals with exponentially decaying weights:

$$A_t^{\text{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l\, \delta_{t+l},$$

which satisfies the efficient recurrence

$$A_t = \delta_t + \gamma\lambda\, A_{t+1}.$$

Advantages are then batch-normalised before use:

$$\hat{A}_t = \frac{A_t - \mu_A}{\sigma_A + \varepsilon}.$$

Sequence-Level PPO (LLM Case)

When the "action" is an entire generated sequence (as in LLM fine-tuning), the per-step ratios multiply into a sequence-level ratio:

$$r_{\text{seq}}(\theta) = \exp\left(\sum_{t=0}^{T-1}(\ell_{\theta,t} - \ell_{\text{old},t})\right).$$

A KL penalty between the updated and old policy can be estimated cheaply as

$$\widehat{\text{KL}} = \mathbb{E}_t[\ell_{\text{old},t} - \ell_{\theta,t}],$$

and is often added to the objective or used as an early-stopping criterion to keep the sequence-level ratio well-behaved.

5 V-MPO

V-MPO (Song et al. 2019) decomposes optimisation into EM phases.

The total objective is

$$\mathcal{L}(\phi, \theta, \eta, \alpha) = \mathcal{L}_V(\phi) + \mathcal{L}_{\text{V-MPO}}(\theta, \eta, \alpha),$$

where

$$\mathcal{L}_{\text{V-MPO}}(\theta, \eta, \alpha) = \mathcal{L}_\pi(\theta) + \mathcal{L}_\eta(\eta) + \mathcal{L}_\alpha(\theta, \alpha).$$

Implementation

github.com/nenuadrian/rl/tree/main/benchmarks

Policy Evaluation (Critic Update)

The value function is fitted via n-step bootstrapped regression:

$$\mathcal{L}_V(\phi) = \frac{1}{2|\mathcal{D}|}\sum_{s_t\sim\mathcal{D}}\left(V_\phi^\pi(s_t) - G_t^{(n)}\right)^2,$$

with

$$G_t^{(n)} = \sum_{k=t}^{t+n-1}\gamma^{k-t}r_k + \gamma^n V_\phi^\pi(s_{t+n}).$$

Advantages are defined as

$$A^\pi(s_t, a_t) = G_t^{(n)} - V_\phi^\pi(s_t).$$

Policy Improvement via EM

We formulate policy improvement as MAP estimation:

$$\theta^\star = \arg\max_\theta \log p_\theta(I = 1) + \log p(\theta),$$

where I denotes the improvement event.

Introduce a variational distribution $\psi(s, a)$:

$$\log p_\theta(I = 1) = \sum_{s,a}\psi(s, a)\log\frac{p_\theta(I = 1, s, a)}{\psi(s, a)} + \text{KL}(\psi(s, a) \parallel p_\theta(s, a \mid I = 1)).$$

We now alternate between E-step and M-step.

E-Step: Non-Parametric Policy Construction

The E-step solves

$$\begin{aligned}\psi^\star = \arg\max_{\psi} \; & \sum_{s,a} \psi(s,a) A^{\pi_{\theta_{\text{old}}}}(s,a) \\ \text{s.t.} \; & \sum_{s,a} \psi(s,a) \log \frac{\psi(s,a)}{p_{\theta_{\text{old}}}(s,a)} < \epsilon_\eta, \\ & \sum_{s,a} \psi(s,a) = 1.\end{aligned}$$

The Lagrangian is

$$\begin{aligned}J(\psi,\eta,\lambda) = & \sum_{s,a} \psi(s,a) A^{\pi_{\theta_{\text{old}}}}(s,a) \\ & + \eta\Big(\epsilon_\eta - \sum_{s,a} \psi(s,a) \log \frac{\psi(s,a)}{p_{\theta_{\text{old}}}(s,a)}\Big) \\ & + \lambda\Big(1 - \sum_{s,a} \psi(s,a)\Big).\end{aligned}$$

Stationarity gives

$$\psi(s,a) = \frac{p_{\theta_{\text{old}}}(s,a) \exp(A^{\pi_{\theta_{\text{old}}}}(s,a)/\eta)}{\sum_{s',a'} p_{\theta_{\text{old}}}(s',a') \exp(A^{\pi_{\theta_{\text{old}}}}(s',a')/\eta)}.$$

The temperature dual is

$$\mathcal{L}_\eta(\eta) = \eta\epsilon_\eta + \eta \log\Big(\sum_{s,a} p_{\theta_{\text{old}}}(s,a) \exp\left(A^{\pi_{\theta_{\text{old}}}}(s,a)/\eta\right)\Big).$$

M-Step: Parametric Projection with KL Constraint

The M-step minimises the negative lower bound:

$$\mathcal{L}_\pi(\theta) = -\sum_{s,a} \psi(s,a) \log \pi_\theta(a|s).$$

Subject to a KL trust-region constraint:

$$\mathbb{E}_{s\sim p(s)}[\mathrm{KL}(\pi_{\theta_{\text{old}}}(\cdot|s) \parallel \pi_\theta(\cdot|s))] < \epsilon_\alpha.$$

The Lagrangian form is

$$J(\theta,\alpha) = \mathcal{L}_\pi(\theta) + \alpha\Big(\epsilon_\alpha - \mathbb{E}_s \mathrm{KL}(\pi_{\theta_{\text{old}}} \| \, \pi_\theta)\Big). \tag{12}$$

In implementation, the loss becomes

$$\begin{aligned}\mathcal{L}_\alpha(\theta,\alpha) = & \alpha\Big(\epsilon_\alpha - \mathrm{sg}[\mathrm{KL}(\pi_{\theta_{\text{old}}} \| \, \pi_\theta)]\Big) \\ & + \mathrm{sg}[\alpha] \, \mathrm{KL}(\pi_{\theta_{\text{old}}} \| \, \pi_\theta).\end{aligned} \tag{1}$$

6 The Case for Dropout

Dropout (Srivastava et al. 2014) randomly zeros each hidden activation with probability p during training and scales the remaining activations by $1/(1-p)$:

$$\tilde{h}_j = \frac{m_j}{1-p} \, h_j, \qquad m_j \sim \text{Bernoulli}(1-p).$$

In a Transformer this is typically applied in two places: after the attention weights (attention dropout) and after each feed-forward sub-layer (residual dropout). If $\mathbf{h}^{(l)}$ is the hidden state at layer l, the residual-dropout forward pass through one sub-layer $f^{(l)}$ is

$$\mathbf{h}^{(l)} = \mathbf{h}^{(l-1)} + \text{Dropout}(f^{(l)}(\mathbf{h}^{(l-1)})).$$

During SFT, dropout serves its classical purpose: it regularises the model and reduces overfitting to the demonstration distribution. Most large-scale LLM pre-training runs (GPT-3, LLaMA, etc.) disable dropout entirely, relying instead on the sheer volume of data for regularisation. However, during fine-tuning the dataset is typically orders of magnitude smaller, and overfitting is a real concern, making dropout relevant again.

Why PPO Disables Dropout

Standard on-policy PPO collects a batch of trajectories under the current policy $\pi_{\theta_{\text{old}}}$ (in eval mode, no dropout), then performs several epochs of gradient updates on that batch. The clipped objective relies on the importance-sampling ratio

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}.$$

Problem 1: stochastic numerator. If dropout is active during the optimisation epochs, $\pi_\theta(a_t \mid s_t)$ becomes a random variable that changes on every forward pass even for fixed θ. The ratio $r_t(\theta)$ therefore fluctuates stochastically, injecting noise directly into the surrogate objective and its gradients. Because PPO's clip window $[1-\epsilon,\, 1+\epsilon]$ is deliberately narrow (typically $\epsilon = 0.2$), even moderate stochastic perturbation can push the ratio outside the clip region or mask genuine policy changes, degrading the signal.

Problem 2: inconsistent denominator. The denominator $\pi_{\theta_{\text{old}}}(a_t \mid s_t)$ is computed once at rollout time in eval mode (no dropout). If the training forward pass uses a different dropout mask, the numerator and denominator are computed under *different* effective networks. The ratio no longer measures how far the policy has moved; it conflates policy change with mask change:

$$r_t(\theta) = \frac{\pi_\theta^{(\text{mask}_1)}(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}^{(\text{no mask})}(a_t \mid s_t)}.$$

This breaks the theoretical guarantee that clipping r_t constrains the KL divergence between consecutive policies.

Problem 3: KL penalty corruption. Many PPO implementations add an auxiliary KL penalty $\widehat{\text{KL}} = \mathbb{E}_t[\ell_{\text{old},t} - \ell_{\theta,t}]$ or use it for early stopping. With dropout active, the estimated KL is biased upward (the mask-induced variance appears as divergence), causing premature termination of updates or an overly conservative step.

For these reasons, all mainstream PPO implementations (including those used for RLHF in InstructGPT, LLaMA, etc.) run the policy in eval mode during both rollout and optimisation, forgoing any regularisation benefit that dropout could provide.

Why V-MPO / EM-Style Methods Are Dropout-Compatible

The V-MPO update decomposes into an E-step that computes per-sample weights, followed by an M-step that is *pure weighted supervised learning*. Neither step requires a probability ratio between two forward passes under different modes.

E-step: weights from advantages, not ratios. The non-parametric target uses the same sequence-level exponential weighting as Eq. (11). Here $A^{(i)} = G^{(i)} - V_\phi(s^{(i)})$ depends on the value function and the observed return, not on a ratio of policy probabilities. The value function V_ϕ is a regression target; dropout can be disabled for the single forward pass that computes advantages at rollout time (just as PPO evaluates its value head in eval mode), or it can remain active since the advantage computation is a one-shot evaluation, not an iterative ratio.

M-step: standard cross-entropy. The M-step objective is the weighted teacher-forced cross-entropy in Eq. (13), which is structurally identical to SFT with per-sequence weights. Dropout is fully compatible here for the same reason it is compatible with any supervised cross-entropy objective: the loss is evaluated under a single stochastic forward pass, and the gradient is an unbiased estimator of the expected loss under the dropout distribution.

No ratio, no conflict. The key structural difference is summarised in the following table:

	PPO	**V-MPO / EM**
Gradient signal	$\nabla_\theta\, r_t(\theta)\, A_t$	$\nabla_\theta\, w_i \log \pi_\theta$
Requires ratio $\pi_\theta/\pi_{\theta_{\text{old}}}$?	Yes	No
Weights depend on current θ?	Yes (through r_t)	No (fixed from E-step)
Dropout in training pass	Corrupts ratio	Standard regularisation

Because the E-step weights w_i are *detached* from the current parameters, the M-step gradient with dropout active is

$$\nabla_\theta \mathcal{L}_\pi = -\sum_{i \in S} w_i \sum_t \nabla_\theta \log \pi_\theta^{(\text{mask})}(y_t^{(i)} \mid \cdot),$$

which is an unbiased estimate of the true weighted-MLE gradient under the dropout distribution, exactly as in supervised learning. No eval-mode forward pass needs to be compared against this quantity.

Practical Implications

In the LLM fine-tuning regime where data is limited and overfitting is a practical concern, V-MPO's EM structure therefore offers a regularisation advantage over PPO: dropout (and related stochastic regularisers such as DropPath or stochastic depth) can be enabled during the M-step without any algorithmic modification, providing the same generalisation benefits observed in supervised fine-tuning. PPO, by contrast, must rely on non-architectural regularisation (weight decay, gradient clipping, early stopping) because its core mechanism is incompatible with stochastic forward passes.

7 Direct Advantage Regression: Aligning LLMs with Online AI Reward

Direct Advantage Regression (DAR) (He et al. 2025) studies online alignment with AI-generated reward signals. The method preserves the regularised online RLHF objective while replacing iterative policy-gradient updates with a closed-form advantage-weighted target followed by supervised projection. This section summarises that objective and its relation to V-MPO.

RL Fine-tuning

Given a language model π_θ to be aligned, a prompt dataset $\mathcal{D}(x)$ and a reward model r, online RL fine-tuning aims to optimise:

$$J_{\text{RLHF}}(\pi_\theta; \pi_{\text{ref}}) = \max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}(x), y\sim\pi_\theta(y|x)} \left[r(x,y)\right] - \alpha D_{\text{KL}}\left(\pi_\theta(y|x) \parallel \pi_{\text{ref}}(y|x)\right).$$

Here $\alpha > 0$ controls KL regularization toward a fixed reference policy π_{ref}.

Advantage Weighted Regression

Advantage Weighted Regression (AWR) maximizes:

$$J_{\mathrm{AWR}}(\pi_\theta) = \max_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_\theta}(x), y\sim\pi_\theta(y|x)} \left[A(x,y)\right],$$

where

$$A(x,y) = r(x,y) - V^{\pi_t}(x).$$

To remove dependence on d_{π_θ}, we approximate using d_{π_t} and impose KL trust-region regularization:

$$J_{\mathrm{AWR}}(\pi_\theta;\pi_t) = \max_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x), y\sim\pi_\theta(y|x)} \left[A(x,y)\right] - \beta D_{\mathrm{KL}}\left(\pi_\theta(y|x) \parallel \pi_t(y|x)\right).$$

Direct Advantage Regression

Dual-Constrained Objective

DAR incorporates reference regularization:

$$J_{\mathrm{DAR}}(\pi_\theta;\pi_{\mathrm{ref}},\pi_t) = \max_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x), y\sim\pi_\theta(y|x)} \left[A(x,y)\right] - \alpha D_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}}) - \beta D_{\mathrm{KL}}(\pi_\theta\|\pi_t).$$

Theorem. Under mild assumptions, for the dual-constrained advantage (or reward) maximization objective above with strictly positive KL coefficients, the optimal policy is:

$$\pi^\star(y \mid x) = \frac{1}{Z(x)}\, \pi_{\mathrm{ref}}(y \mid x)^{\frac{\alpha}{\alpha+\beta}}\ \pi_t(y \mid x)^{\frac{\beta}{\alpha+\beta}}\ \exp\Big(\frac{A(x,y)}{\alpha+\beta}\Big),$$

where

$$Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x)^{\frac{\alpha}{\alpha+\beta}}\ \pi_t(y \mid x)^{\frac{\beta}{\alpha+\beta}}\ \exp\Big(\frac{A(x,y)}{\alpha+\beta}\Big),$$

is the partition function.

Proof.

$$\begin{aligned}
&\max_\pi \mathbb{E}_{x,y\sim\pi}[A(x,y)] \ - \ \alpha D_{\mathrm{KL}}[\pi(y|x) \,\|\, \pi_{\mathrm{ref}}(y|x)] \ - \ \beta D_{\mathrm{KL}}[\pi(y|x) \,\|\, \pi_t(y|x)] \\
&= \min_\pi \mathbb{E}_{x,y\sim\pi}\Big[\alpha \log \frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)} + \beta \log \frac{\pi(y|x)}{\pi_t(y|x)} - A(x,y)\Big] \\
&= \min_\pi \mathbb{E}_{x,y\sim\pi}\Big[(\alpha+\beta)\log\pi(y|x) - \alpha\log\pi_{\mathrm{ref}}(y|x) - \beta\log\pi_t(y|x) - A(x,y)\Big] \\
&= \min_\pi \mathbb{E}_{x,y\sim\pi}\Big[\log\pi(y|x) - \log\pi_{\mathrm{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}} - \log\pi_t(y|x)^{\frac{\beta}{\alpha+\beta}} - \frac{1}{\alpha+\beta}A(x,y)\Big] \\
&= \min_\pi \mathbb{E}_{x,y\sim\pi}\Big[\log \frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}}\ \pi_t(y|x)^{\frac{\beta}{\alpha+\beta}}} - \frac{1}{\alpha+\beta}A(x,y)\Big] \\
&= \min_\pi \mathbb{E}_{x,y\sim\pi}\Big[\log \frac{\pi(y|x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y|x)^{\frac{\beta}{\alpha+\beta}}\exp(\frac{1}{\alpha+\beta}A(x,y))} - \log Z(x)\Big].
\end{aligned}$$

Because the partition function does not depend on π, $\log Z(x)$ is constant with respect to the optimisation variable and can be dropped:

$$\begin{aligned}
&\min_\pi \mathbb{E}_{x,y\sim\pi}\Big[\log \frac{\pi(y|x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y|x)^{\frac{\beta}{\alpha+\beta}}\exp(\frac{1}{\alpha+\beta}A(x,y))}\Big] \\
&= \min_\pi \mathbb{E}_x\ D_{\mathrm{KL}}\Bigg[\pi(y|x) \,\Bigg\|\, \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y|x)^{\frac{\beta}{\alpha+\beta}}\exp(\frac{1}{\alpha+\beta}A(x,y))\Bigg].
\end{aligned}$$

By Gibbs' inequality, the KL term is minimised when the two distributions are identical:

$$\pi^\star(y|x) \;=\; \frac{1}{Z(x)}\,\pi_{\text{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}}\,\pi_t(y|x)^{\frac{\beta}{\alpha+\beta}}\,\exp\Big(\frac{1}{\alpha+\beta}A(x,y)\Big). \tag{2}$$

The improved parametric policy is then obtained by minimising the KL divergence to the target distribution above:

$$\min_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x)} D_{\text{KL}}[\pi^\star(\cdot \mid x) \parallel \pi_\theta(\cdot \mid x)],$$

Substituting $\pi^\star$ gives:

$$\min_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x)} D_{\text{KL}}\left[\frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y \mid x)^{\frac{\beta}{\alpha+\beta}}\,\exp\Big(\frac{1}{\alpha+\beta}A(x,y)\Big) \;\Big\|\; \pi_\theta(\cdot \mid x)\right],$$

Expanding the KL divergence and dropping terms independent of π_θ yields:

$$\min_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x)}\left[-\sum_y \frac{1}{Z(x)}\,\pi_{\text{ref}}(y \mid x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y \mid x)^{\frac{\beta}{\alpha+\beta}}\,\exp\Big(\frac{1}{\alpha+\beta}A(x,y)\Big)\,\log \pi_\theta(y \mid x)\right],$$

Since $Z(x)$ is a positive constant with respect to π_θ, it can be removed without changing the optimum:

$$\min_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x)}\Big[-\sum_y \pi_{\text{ref}}(y \mid x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y \mid x)^{\frac{\beta}{\alpha+\beta}}\,\exp\Big(\frac{1}{\alpha+\beta}A(x,y)\Big)\,\log \pi_\theta(y \mid x)\Big],$$

Using π_t as the sampling policy gives the final practical objective:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim d_{\pi_t}(x)}\mathbb{E}_{y\sim\pi_t(y|x)}\left[\left(\frac{\pi_{\text{ref}}(y \mid x)}{\pi_t(y \mid x)}\right)^{\frac{\alpha}{\alpha+\beta}}\,\exp\Big(\frac{1}{\alpha+\beta}A(x,y)\Big)\,\log \pi_\theta(y \mid x)\right].$$

Mapping DAR to V-MPO

The closed-form optimal policy derived in Direct Advantage Regression (DAR) reduces to the V-MPO target under simple limits and special-case identifications. The derivation clarifies when DAR may be viewed as a sequence-level variant of the V-MPO/AWR family.

DAR closed-form target. DAR yields a closed-form optimal policy of the form

$$\pi^\star(y\,|\,x) \;=\; \frac{1}{Z(x)}\;\pi_{\text{ref}}(y\,|\,x)^{\frac{\alpha}{\alpha+\beta}}\;\pi_t(y\,|\,x)^{\frac{\beta}{\alpha+\beta}}\;\exp\Big(\frac{A(x,y)}{\alpha+\beta}\Big), \tag{3}$$

where $(\alpha,\beta) > 0$ are the two KL coefficients (reference and sampling policy), π_{ref} a chosen reference policy, π_t the sampling policy used to collect $\mathcal{D}_{\pi_t}$, and $A(x,y)$ the (sequence-level) advantage or score. Define the shorthand

$$t \coloneqq \alpha+\beta > 0.$$

Taking logarithms of (3) produces the additive form used below:

$$\log \pi^\star(y\,|\,x) = -\log Z(x) + \frac{\alpha}{t}\log \pi_{\text{ref}}(y\,|\,x) + \frac{\beta}{t}\log \pi_t(y\,|\,x) + \frac{1}{t}A(x,y). \tag{4}$$

V-MPO canonical target. V-MPO (and related advantage-weighted regression methods) uses a target proportional to the sampling policy multiplied by an exponential advantage factor. In sequence notation:

$$\psi_{\text{V–MPO}}(y\,|\,x) \propto \pi_t(y\,|\,x)\;\exp\Big(\frac{A(x,y)}{\eta}\Big), \tag{5}$$

where $\eta > 0$ is the temperature (V-MPO treats η as a dual variable and may optimise it by solving a convex dual).

Special case 1: identical reference and sampling policies. Set $\pi_{\text{ref}} = \pi_t$ in (3). Then the multiplicative powers collapse:

$$\pi_{\text{ref}}^{\alpha/t}\pi_t^{\beta/t} = \pi_t^{(\alpha+\beta)/t} = \pi_t.$$

Substituting into (3) gives

$$\pi^\star(y\,|\,x) \propto \pi_t(y\,|\,x)\,\exp\Big(\frac{A(x,y)}{t}\Big).$$

Comparing with (5) we identify the V-MPO temperature as

$$\eta = t = \alpha + \beta.$$

Thus DAR reduces to the V-MPO structural form when the reference policy equals the sampling policy; the DAR temperature is the sum of the two KL coefficients.

Special case 2: $\alpha \to 0$ (reference KL vanishes). Examine the limit $\alpha \to 0$ while holding $t = \alpha + \beta$ finite (equivalently, let $\beta \to t$). From (4),

$$\lim_{\alpha\to 0} \log \pi^\star(y\,|\,x) = -\log Z(x) + \underbrace{\frac{\alpha}{t}}_{\to 0}\log \pi_{\text{ref}} + \underbrace{\frac{\beta}{t}}_{\to 1}\log \pi_t + \frac{1}{t}A(x,y),$$

so

$$\pi^\star(y\,|\,x) \propto \pi_t(y\,|\,x)\,\exp\Big(\frac{A(x,y)}{t}\Big),$$

again matching (5) with $\eta = t$. The algebraic limit is well-defined because the coefficients α/t and β/t converge to $(0, 1)$ respectively.

Algebraic equivalence (log-space explanation). Write the DAR unnormalised log-weight for a sample (x, y) as

$$\ell_{\text{DAR}}(x,y) \;=\; \frac{\alpha}{t}\log \pi_{\text{ref}}(y\,|\,x) + \frac{\beta}{t}\log \pi_t(y\,|\,x) + \frac{1}{t}A(x,y).$$

If either $\pi_{\text{ref}} = \pi_t$ or $\alpha/t \to 0$, the first two terms reduce to $\log \pi_t(y\,|\,x)$. Hence

$$\ell_{\text{DAR}}(x,y) \to \log \pi_t(y\,|\,x) + \frac{1}{t}A(x,y),$$

and exponentiation recovers the V-MPO unnormalised weight $\exp(\log \pi_t + A/t) = \pi_t \exp(A/t)$.

Practical consequences and distinctions. Although DAR and V-MPO share the same advantage-exponential device, they differ in:

1. **Dual handling.** V-MPO treats η (and an average-KL dual α) as optimisation variables updated by convex-dual objectives; DAR typically treats α, β as explicit regularisation coefficients (or tunes them as hyperparameters), and derives a closed-form mixture. Consequently, V-MPO emphasises adaptive temperature control, while DAR emphasises calibrated mixture regularisation for LLM-scale training.

2. **Factorisation and tractability.** V-MPO is usually applied to low-dimensional action spaces (or per-step discrete actions); LLM-scale training uses sequence-level weights on teacher-forced token log-probabilities (the DAR practical objective).

3. **Engineered stabilisers.** DAR introduces practical stabilisers for large-vocabulary sequence training (weight clipping, advantage normalisation, importance-weight corrections). V-MPO implementations use top-k selection and explicit dual updates which require careful numerical handling when lifted to LLMs.

Bridge to the EM. The sequence-level EM procedure in Section 9, Eq. (11) through Eq. (13), and the DAR formulation are two views of the same update pattern: an E-step that constructs exponential advantage weights followed by a weighted teacher-forced M-step. In the LLM-scale V-MPO view, η is adapted online via a dual objective and the E-step may be restricted to top-k samples for stability; in the DAR view, the closed-form target introduces an additional reference/sampling mixture term and uses fixed coefficients (α, β), which recover the V-MPO-style form in the limits derived above.

Sequence-level weighted loss (common form). Both approaches use the same weighted teacher-forced M-step objective as in Eq. (13). The per-sequence weights differ only by the base factor:

$$w_i^{\mathrm{DAR}} \propto \exp\Big(\frac{\alpha}{t}\log \pi_{\mathrm{ref}}^{(i)} + \frac{\beta}{t}\log \pi_t^{(i)} + \frac{1}{t}A^{(i)}\Big), \qquad w_i^{\text{V–MPO}} \propto \exp\Big(\log \pi_t^{(i)} + \frac{1}{\eta}A^{(i)}\Big).$$

8 Generalized EM Policy Improvement (GEMPI)

The GEMPI tuple. A Generalized EM Policy Improvement method is specified by

$$\mathcal{G} = (\{D_j\}_{j=1}^m,\ \{\pi_j\}_{j=1}^m,\ \{\lambda_j\}_{j=1}^m,\ \mathcal{T},\ \mathcal{F},\ D_M,\ \varepsilon_M),$$

where $\{D_j\}$ are divergence functionals used in the E-step, $\{\pi_j\}$ are anchor policies, $\{\lambda_j > 0\}$ are regularization coefficients, $\mathcal{T}$ is the temperature mechanism (fixed, dual-adaptive, or closed-form), $\mathcal{F}$ is a filtering mechanism (identity or top-k), and D_M with budget ε_M is an optional M-step trust-region divergence.

General Regularized E-Step

The E-step constructs a non-parametric target distribution $q^\star(\cdot \mid x)$ by solving a regularized advantage maximization over m anchor policies:

$$q^\star = \arg\max_q \Big\{\mathbb{E}_{y\sim q}[A(x,y)] - \sum_{j=1}^m \lambda_j\, D_j(q(\cdot \mid x)\,\|\,\pi_j(\cdot \mid x))\Big\}, \tag{6}$$

subject to q being a valid distribution. The advantage $A(x, y)$ may be sequence-level (reward minus baseline) or token-level, and the divergences D_j penalize deviation from each anchor π_j.

Theorem (Multi-KL Closed Form). When all divergences are KL, i.e. $D_j = \mathrm{KL}$ for $j = 1, \ldots, m$, the solution to (6) is

$$q^\star(y \mid x) = \frac{1}{Z(x)} \prod_{j=1}^m \pi_j(y \mid x)^{\lambda_j/\Lambda}\ \exp\Big(\frac{A(x,y)}{\Lambda}\Big), \tag{7}$$

where $\Lambda := \sum_{j=1}^m \lambda_j$ is the *effective temperature* and $Z(x)$ is the partition function ensuring normalization.

Proof. Write the Lagrangian with multiplier μ for the normalization constraint:

$$\mathcal{J}(q,\mu) = \sum_y q(y \mid x)\, A(x,y) - \sum_{j=1}^m \lambda_j \sum_y q(y \mid x)\log\frac{q(y \mid x)}{\pi_j(y \mid x)} + \mu\Big(1 - \sum_y q(y \mid x)\Big).$$

Expanding the KL terms and grouping:

$$\mathcal{J} = \sum_y q(y \mid x)\Big[A(x,y) - \Lambda \log q(y \mid x) + \sum_{j=1}^m \lambda_j \log \pi_j(y \mid x)\Big] + \mu\Big(1 - \sum_y q(y \mid x)\Big).$$

Taking the functional derivative with respect to $q(y \mid x)$ and setting to zero:

$$A(x,y) - \Lambda(\log q(y \mid x) + 1) + \sum_{j=1}^m \lambda_j \log \pi_j(y \mid x) - \mu = 0.$$

Solving for $\log q(y \mid x)$:

$$\log q(y \mid x) = \frac{1}{\Lambda} A(x, y) + \sum_{j=1}^{m} \frac{\lambda_j}{\Lambda} \log \pi_j(y \mid x) + \text{const.}$$

Exponentiating and normalizing yields (7). □

Log-space form. The unnormalized log-weight for sample (x, y) is

$$\ell(x, y) = \sum_{j=1}^{m} \frac{\lambda_j}{\Lambda} \log \pi_j(y \mid x) + \frac{1}{\Lambda} A(x, y). \tag{8}$$

The coefficients λ_j/Λ sum to one, so the first term is a convex combination of the anchor log-policies—a *geometric mixture*—shifted by the scaled advantage. This reveals the E-step target as a geometry-aware interpolation between the anchors, tilted toward high-advantage sequences.

Multi-temperature dual. When the E-step is posed as a constrained problem—maximize expected advantage subject to $\mathrm{KL}(q\|\pi_j) < \varepsilon_j$ for each anchor—the Lagrangian yields the same functional form with each λ_j replaced by an optimal dual variable $\lambda_j^\star$. The dual objective generalizes V-MPO's scalar temperature dual:

$$L_{\text{dual}}(\lambda_1, \ldots, \lambda_m) = \sum_{j=1}^{m} \lambda_j \, \varepsilon_j + \Lambda \, \log Z(x; \lambda_1, \ldots, \lambda_m). \tag{9}$$

This is jointly convex in $(\lambda_1, \ldots, \lambda_m)$ and can be minimized by gradient descent, adapting all temperatures simultaneously.

General M-Step

The M-step projects the non-parametric target $q^\star$ back to the parametric policy family:

$$\theta_{k+1} = \arg\min_{\theta} \Big\{ -\mathbb{E}_{q^\star}[\log \pi_\theta(y \mid x)] + \gamma \, D_M(\pi_{\theta_k}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)) \Big\}, \tag{10}$$

where $\gamma \geq 0$ controls the M-step trust region. The three main variants are:

- **Unconstrained** ($\gamma = 0$): pure weighted maximum likelihood. Used by DAR and AWR.
- **Penalized** ($\gamma > 0$, $D_M = \mathrm{KL}$): soft KL penalty preventing the parametric policy from overshooting. The V-MPO M-step uses this with γ treated as a Lagrangian dual variable α optimized against a budget ε_α.
- **Hard-constrained**: replace the penalty with a constraint $D_M(\pi_{\theta_k}\|\pi_\theta) < \varepsilon_M$ and solve via the Lagrangian, yielding the stop-gradient decomposition used in V-MPO's implementation.

Importance-weighted practical form. When samples are drawn from the sampling policy π_t but $q^\star \neq \pi_t$, the M-step loss uses importance weights:

$$\mathcal{L}_\pi(\theta) = -\sum_{i \in S} w_i \sum_{t=1}^{T_i} \log \pi_\theta(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)}),$$

where $w_i \propto q^\star(y^{(i)} \mid x^{(i)})/\pi_t(y^{(i)} \mid x^{(i)})$. When π_t is one of the anchors, the corresponding factor in the geometric mixture cancels partially, simplifying the weights.

The closed-form target (7) is a population-level object: the partition function

$$Z(x) = \sum_{y} \prod_{j=1}^{m} \pi_j(y \mid x)^{\lambda_j/\Lambda} \exp\Big(\frac{A(x, y)}{\Lambda}\Big)$$

sums over the entire output space, which is intractable for large vocabulary sequence models. In practice, $Z(x)$ is estimated from a finite batch $\mathcal{B} = \{y^{(i)}\}_{i=1}^{N}$ drawn from the sampling policy π_t, giving

$$\hat{Z}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{\prod_j \pi_j(y^{(i)} \mid x)^{\lambda_j/\Lambda} \exp(\frac{1}{\Lambda} A(x, y^{(i)}))}{\pi_t(y^{(i)} \mid x)},$$

a self-normalised importance-weighted estimate. Three distinct approximation errors arise.

1. Partition function bias. $\hat{Z}(x)$ is a ratio estimator and is therefore *biased*. By the delta method,

$$\mathbb{E}[\hat{Z}(x)] = Z(x)\Big(1 + O(N^{-1})\Big),$$

so the bias is $O(N^{-1})$ and vanishes as the batch grows. The self-normalised importance weights

$$\hat{w}_i = \frac{\prod_j \pi_j(y^{(i)} \mid x)^{\lambda_j/\Lambda} \exp(\frac{1}{\Lambda} A(x, y^{(i)}))/\, \pi_t(y^{(i)} \mid x)}{\sum_{i'} \prod_j \pi_j(y^{(i')} \mid x)^{\lambda_j/\Lambda} \exp(\frac{1}{\Lambda} A(x, y^{(i')}))/\, \pi_t(y^{(i')} \mid x)}$$

converge to $q^\star(y^{(i)} \mid x)/\pi_t(y^{(i)} \mid x)$ almost surely as $N \to \infty$ by the strong law, provided $q^\star \ll \pi_t$ (absolute continuity).

2. Effective sample size and weight degeneracy. The quality of the finite-sample approximation is governed by the effective sample size

$$\mathrm{ESS} = \frac{(\sum_i \hat{w}_i)^2}{\sum_i \hat{w}_i^2},$$

which equals N when all weights are equal and degrades toward 1 when a single sample dominates. The ESS is controlled by the mismatch between $q^\star$ and π_t, quantified by their χ^2-divergence:

$$\mathrm{ESS} \approx \frac{N}{1 + \chi^2(q^\star \| \pi_t)}.$$

Within the GEMPI parameterisation, increasing Λ (i.e. raising regularisation coefficients or lowering the effective temperature) shrinks the advantage tilt, bringing $q^\star$ closer to the geometric anchor mixture and hence to π_t. This directly improves ESS, providing a formal justification for the practical observation that lower temperature produces more stable weighted updates.

3. M-step gradient bias under finite ESS. The M-step objective (10) evaluated with self-normalised weights is

$$\hat{\mathcal{L}}_\pi(\theta) = -\sum_{i \in S} \hat{w}_i \sum_{t=1}^{T_i} \log \pi_\theta(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)}).$$

The gradient $\nabla_\theta \hat{\mathcal{L}}_\pi$ is a biased estimator of $\nabla_\theta \mathbb{E}_{q^\star}[-\log \pi_\theta]$ because the self-normalised weights themselves depend on the sample. The bias is of order $O(N^{-1})$ and has been analysed in the self-normalised IS literature (Hesterberg 1995; Owen 2013); it does not affect consistency but does affect the finite-sample variance of the gradient.

Connection to GEMPI stabilisers. The three structural stabilisers discussed above can be understood as direct mitigations of the above errors.

- **Adaptive temperature** (dual optimisation of λ_j) minimises $\chi^2(q^\star \| \pi_t)$, maximising ESS and reducing partition function bias.
- **Top-k filtering** replaces soft self-normalised weighting over all N samples with hard selection of the k samples where $q^\star/\pi_t$ is largest. This acts as a variance-reduction strategy: by concentrating mass on the samples most likely under $q^\star$, it reduces the second moment of the importance weights at the cost of introducing a support truncation bias of order $O(\mathbb{E}_{q^\star}[q^\star/\pi_t \cdot \mathbf{1}_{i \notin S}])$, which is small when the top-k fraction ρ is chosen so that the omitted tail of $q^\star$ is negligible.

- **Log-sum-exp normalisation** addresses floating-point representation of $\hat{Z}$ rather than its statistical properties, but is a necessary precondition for the estimates above to be numerically meaningful at large $|A|/\Lambda$.

Remark on the population vs. sample limit. In the limit $N \to \infty$ with π_t having full support over the output space, $\hat{w}_i \to q^\star(y^{(i)} \mid x)/\pi_t(y^{(i)} \mid x)$ and the M-step recovers the population KL minimization $\min_\theta D_{\mathrm{KL}}(q^\star \| \pi_\theta)$ exactly. At finite N, the M-step minimizes the same KL subject to the support constraint imposed by the batch, which is a strictly easier problem and can lead to underfitting of the tails of $q^\star$. This finite-support effect is distinct from, and additive with, the self-normalisation bias above.

Recovering Existing Methods

The GEMPI framework subsumes the EM-style methods derived in the preceding sections. Each method corresponds to a specific instantiation of the tuple $\mathcal{G}$.

Method	m	**Anchors**	**Coefficients**	Λ	**Temperature**	**M-step** D_M	**Filtering**
SFT	0	—	—	—	—	None	—
AWR	1	π_t	β	β	Fixed	None	None
V-MPO	1	π_t	η	η	Adaptive dual	KL (α dual)	Top-k
DAR	2	$\pi_{\mathrm{ref}}, \pi_t$	α, β	$\alpha{+}\beta$	Fixed	None	None
RL-EM	1	π_{ref}	β	β	Fixed	Optional KL	None

V-MPO. Set $m = 1$, $D_1 = \mathrm{KL}$, $\pi_1 = \pi_t$. The closed form (7) becomes

$$q^\star(y \mid x) \propto \pi_t(y \mid x)\, \exp\Big(\frac{A(x,y)}{\eta}\Big),$$

with $\Lambda = \eta$. The temperature is treated as a dual variable optimised via the scalar dual (9) (which reduces to $\mathcal{L}_\eta = \eta\varepsilon_\eta + \eta \log(Z/k)$). Top-$k$ filtering restricts the support. The M-step adds a KL trust region with dual α. This recovers the V-MPO formulation derived earlier in this document.

DAR. Set $m = 2$, $D_1 = D_2 = \mathrm{KL}$, $\pi_1 = \pi_{\mathrm{ref}}$ with coefficient α, $\pi_2 = \pi_t$ with coefficient β. The closed form (7) becomes

$$q^\star(y \mid x) \propto \pi_{\mathrm{ref}}(y \mid x)^{\alpha/(\alpha+\beta)}\, \pi_t(y \mid x)^{\beta/(\alpha+\beta)}\, \exp\Big(\frac{A(x,y)}{\alpha+\beta}\Big),$$

with $\Lambda = \alpha + \beta$. This is precisely the DAR optimal policy in Eq. (3).

AWR. Set $m = 1$, $D_1 = \mathrm{KL}$, $\pi_1 = \pi_t$, $\lambda_1 = \beta$ fixed. No dual optimization, no filtering, unconstrained M-step. The target is $q^\star \propto \pi_t \exp(A/\beta)$, recovering standard Advantage Weighted Regression (Peng et al. 2019).

SFT. The degenerate case with no E-step optimization: $q(\tau \mid x) = \delta(\tau = y)$ places all mass on demonstrated responses, and the M-step is pure maximum likelihood. This corresponds to $m = 0$ in the GEMPI tuple.

DAR→V-MPO as corollary. The reduction derived in the DAR-to-V-MPO mapping above is now a direct consequence of the GEMPI parameterisation: collapsing two anchors to one ($\pi_{\mathrm{ref}} = \pi_t$) merges the geometric mixture powers $\pi_{\mathrm{ref}}^{\alpha/\Lambda}\pi_t^{\beta/\Lambda} = \pi_t^1 = \pi_t$, recovering the single-anchor form with $\eta = \alpha + \beta$. Alternatively, sending $\alpha \to 0$ zeros out the π_{ref} contribution, yielding the same result.

The Role of Divergence Choice

The preceding derivations use KL divergence exclusively in the E-step. This is analogous to the most common PMD instantiation (negative entropy as mirror map). However, the GEMPI E-step (6) is defined for arbitrary divergences D_j. Replacing KL with other members of the f-divergence family changes the form of the advantage tilting: for KL the tilting is *exponential* in the advantage; for the χ^2-divergence ($f(u) = (u-1)^2$) the optimality condition yields *linear* tilting $q^\star \propto \pi_j(1 + A/(2\lambda_j))$; for Rényi divergences the tilting follows a power law. The choice of E-step divergence thus plays the same role in GEMPI that the choice of Bregman generating function plays in Bregman divergence theory, or the choice of mirror map in PMD.

PMD (policy gradients)	**GEMPI (EM alignment)**
Mirror map h	E-step divergence D_j
$h = -H$ (neg. entropy) $\to$ NPG/TRPO	$D_j = \mathrm{KL} \to$ V-MPO/DAR/AWR
$h = \frac{1}{2}\|\cdot\|^2 \to$ proj. gradient	$D_j = \chi^2 \to$ linear adv. weighting
Step size / temperature η	Effective temperature Λ
Single proximal step	Two-phase: E-step + M-step projection

Stability Properties as Structural Consequences

The GEMPI decomposition reveals that the practical stabilization mechanisms used across the various methods are not ad-hoc engineering choices but follow from specific structural decisions within the framework.

Adaptive temperature. Treating the coefficients λ_j as dual variables (Eq. (9)) rather than fixed hyperparameters gives automatic temperature adaptation. V-MPO exercises this with a single dual variable ($m = 1$); DAR currently uses fixed (α, β) but could, within GEMPI, optimise both as a two-temperature dual, gaining adaptive stability while retaining the dual-anchor structure.

Trust regions in the M-step. The M-step constraint $D_M(\pi_{\theta_k} \| \pi_\theta) < \varepsilon_M$ is an independent axis of variation: any E-step target can be paired with any M-step trust-region policy. V-MPO uses a KL trust region; DAR and AWR do not. The framework makes explicit that adding or removing M-step constraints is orthogonal to the E-step design.

Top-k filtering. Restricting the support of $q^\star$ to the top-k samples by advantage before computing the geometric mixture weights is a form of hard thresholding on the filter set $\mathcal{F}$. This can be applied to any GEMPI instantiation, not only V-MPO.

Dropout compatibility. The GEMPI template guarantees dropout compatibility whenever the M-step uses detached (stop-gradient) weights from the E-step. In this case the M-step gradient

$$\nabla_\theta \mathcal{L}_\pi = -\sum_{i \in S} w_i \sum_t \nabla_\theta \log \pi_\theta^{(\mathrm{mask})}(y_t^{(i)} \mid \cdot)$$

is an unbiased estimator of the weighted-MLE gradient under the dropout distribution, exactly as in supervised learning. This structural property generalises the later dropout discussion to a consequence of the GEMPI decomposition: any method instantiated within GEMPI inherits dropout compatibility in its M-step.

Novel Instantiations

The GEMPI framework, by making the axes of variation explicit, suggests several unexplored combinations.

Adaptive-Temperature DAR (AT-DAR). DAR with $m = 2$ but treating both α and β as dual variables with KL budgets ε_α and ε_β. The dual objective becomes

$$L_{\text{dual}}(\alpha, \beta) = \alpha\,\varepsilon_\alpha + \beta\,\varepsilon_\beta + (\alpha + \beta)\log Z(x; \alpha, \beta),$$

giving DAR the same adaptive stability as V-MPO while retaining the dual-anchor structure.

DAR with M-step Trust Region (DAR-TR). Adding a KL trust-region constraint to DAR's currently unconstrained M-step creates a method combining DAR's dual-anchor E-step with V-MPO's conservative parametric projection: the full GEMPI tuple with $m = 2$, $D_1 = D_2 = \text{KL}$, $D_M = \text{KL}$ with budget ε_M.

Top-k DAR. Applying V-MPO's top-k filtering to the DAR E-step. Currently DAR uses all samples; restricting to the top-k by advantage before computing the geometric mixture weights could improve sample efficiency by focusing the M-step on demonstrably high-quality sequences.

9 V-MPO for LLM RL within GEMPI

This section instantiates GEMPI for online RL alignment of autoregressive transformers. For prompts $x \sim \mathcal{D}_x$, the policy $\pi_\theta(y \mid x)$ generates full responses $y = (y_1, \ldots, y_T)$ and receives sequence-level rewards from a preference/reward model. We use sequence-level advantages

$$A^{(i)} = r^{(i)} - V_\phi(x^{(i)}),$$

and perform EM-style policy improvement with V-MPO structure.

GEMPI Instantiation for LLM V-MPO

The LLM variant corresponds to the GEMPI tuple

$$\mathcal{G}_{\text{LLM-V–MPO}} = (\{\text{KL}\}, \{\pi_t\}, \{\eta\}, \text{dual-adaptive}, \text{top-}k, \text{KL}, \varepsilon_\alpha),$$

that is, a single KL anchor π_t in the E-step, dual adaptation of η, top-k support filtering, and an optional KL trust region in the M-step.

Sequence-Level E-Step for Transformer Rollouts

Given a rollout batch

$$\mathcal{B} = \{(x^{(i)}, y^{(i)}, A^{(i)})\}_{i=1}^N,$$

and selected subset $S \subset \{1, \ldots, N\}$, the non-parametric E-step target is

$$\psi(i) \propto \exp\Big(\frac{A^{(i)}}{\eta}\Big). \tag{11}$$

After normalisation, these become per-sequence weights w_i used by the M-step.

The temperature is adapted through the dual objective

$$\mathcal{L}_\eta(\eta) = \eta\,\varepsilon_\eta + \eta \log\Big(\frac{1}{k}\sum_{i \in S} \exp\Big(\frac{A^{(i)}}{\eta}\Big)\Big),$$

with $k = |S|$. A numerically stable log-sum-exp form is

$$g(\eta) = m + \log\Big(\frac{1}{k}\sum_{i \in S} \exp(u_i - m)\Big), \qquad u_i = \frac{A^{(i)}}{\eta}, \quad m = \max_{i \in S} u_i. \tag{12}$$

In practice, S is chosen as top-k by detached scaled advantage:

$$u_i^{\text{det}} = \frac{A^{(i)}}{\eta_{\text{det}}}, \qquad k = \max(1, \lfloor \rho N \rfloor), \qquad S = \text{Top-}\,k\big(u_i^{\text{det}}\big)\,.$$

M-Step as Weighted Teacher-Forced Transformer Training

The parametric projection is weighted teacher-forced cross-entropy:

$$\mathcal{L}_\pi(\theta) = -\sum_{i \in S} w_i \sum_{t=1}^{T_i} \log \pi_\theta(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)}). \tag{13}$$

This keeps the update in standard transformer training form (teacher forcing over tokens) while importing RL information through detached sequence weights w_i. When desired, a KL trust-region term to π_t is added exactly as in the V-MPO M-step.

Transformer-Specific Practicalities

- **Sequence-level credit assignment.** Rewards are naturally sequence-level in LLM alignment; the EM weighting avoids unstable token-wise importance ratios.
- **Long-context batching.** Variable-length prompts and completions require masking and packed batches; the weighted M-step remains unchanged under standard attention masks.
- **Large-vocabulary stability.** Top-k filtering and log-sum-exp normalisation reduce weight degeneracy and floating-point overflow in mixed-precision training.
- **Distributed training compatibility.** The objective is a weighted cross-entropy, so it integrates with gradient accumulation, FSDP/ZeRO, and standard transformer training pipelines.

Dropout in the LLM V-MPO M-Step

Because E-step weights are detached from current parameters, the M-step with dropout mask sampling remains

$$\nabla_\theta \mathcal{L}_\pi = -\sum_{i \in S} w_i \sum_t \nabla_\theta \log \pi_\theta^{(\text{mask})}(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)}),$$

which is the same unbiased estimator structure as weighted supervised learning. In contrast to PPO, no ratio $\pi_\theta / \pi_{\theta_{\text{old}}}$ is required inside the objective, so stochastic forward passes do not corrupt the core optimisation signal. A detailed PPO-vs-EM comparison is given in Section 6.

10 Transformers - TrXL

The V-MPO authors (Song et al. 2019) reported strong empirical results on Transformers in Atari using TrXL (Dai et al. 2019), an architecture later refined for RL through GTrXL (Parisotto et al. 2019).

Implementation

- github.com/kimiyoung/transformer-xl
- github.com/nenuadrian/DI-engine/tree/main/benchmarks

GTrXL with V-MPO

The V-MPO paper (Song et al. 2019) replaces the LSTM core with a Transformer-XL (TrXL) for single-task Atari, motivated by the argument that in a fully observable environment recurrent architectures enable the agent to utilise more useful representations than are available in the immediate observation. However, standard TrXL fails in RL, performing at random-policy level on benchmarks such as DMLab-30. The Gated Transformer-XL (GTrXL) (Parisotto et al. 2019) addresses this with two targeted modifications.

Identity Map Reordering (TrXL-I)

Layer normalisation is moved to the *input* of each sub-layer rather than the output, creating a direct identity path from the first layer's input to the last layer's output. At initialisation this biases the network towards a Markovian (reactive) policy and provides a clear gradient path, making the training landscape far more amenable to policy-gradient methods.

Gating Layers

Residual connections are replaced with learnable gating layers. The best-performing variant uses GRU-type gates:

$$\begin{aligned} r &= \sigma(W_r y + U_r x), \\ z &= \sigma(W_z y + U_z x - b_g), \\ \hat{h} &= \tanh(W_g y + U_g(r \odot x)), \\ g(x, y) &= (1 - z) \odot x + z \odot \hat{h}, \end{aligned}$$

where x is the residual input and y is the sub-layer output. Initialising the bias $b_g = 2$ places each gate near the identity at the start of training, preserving the reactive-policy initialisation provided by the layer-norm reordering.

Why GTrXL Suits V-MPO

V-MPO is an on-policy algorithm that performs policy improvement via an EM procedure (E-step top-k advantage weighting; M-step weighted maximum-likelihood with a KL trust-region). Because it collects fresh trajectories each update, the memory architecture is critical: the agent must integrate information over long horizons to form useful state representations. GTrXL provides:

- **Long-range memory.** Relative position encodings inherited from TrXL let the network attend over a memory tensor spanning up to thousands of past time-steps, far beyond what an LSTM can retain.
- **Stable optimisation.** The GRU gating achieves a 0% divergence rate across hyperparameter sweeps, compared with 16% for the plain TrXL-I variant, making it compatible with V-MPO's fixed Adam learning rate and without requiring population-based hyperparameter search.
- **Robust performance on memory tasks.** On DMLab-30, GTrXL (GRU) reaches a human-normalised score of 117.6 ± 0.3 versus 99.3 ± 1.0 for LSTM, with the largest gains on memory-dependent levels and no regression on reactive ones.

Architecture and Training Configuration

For single-task Atari with TrXL, the shared policy-value network consists of a convolutional ResNet backbone feeding into the transformer core, with previous reward and action as additional inputs. Representative TrXL hyperparameters used in V-MPO experiments: embedding size 256, 8 layers, 4 attention heads, key/value size 32, MLP size 512, unroll length 63, batch size 128. The value and policy heads branch from the transformer output; parameters are updated jointly under the combined V-MPO objective $\mathcal{L}(\phi, \theta, \eta, \alpha)$ using Adam at a fixed learning rate of 10^{-4}.

11 References

Dai, Zihang et al. (2019). *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context.* arXiv: 1901.02860 [cs.LG]. URL: https://arxiv.org/abs/1901.02860.

He, Li et al. (2025). *Direct Advantage Regression: Aligning LLMs with Online AI Reward*. arXiv: 2504.14177 [cs.AI]. URL: https://arxiv.org/abs/2504.14177.

Hesterberg, Tim (1995). "Weighted average importance sampling and defensive mixture distributions". In: *Technometrics* 37.2, pp. 185–194. DOI: 10.2307/1269620.

Owen, Art B. (2013). *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/.

Parisotto, Emilio et al. (2019). *Stabilizing Transformers for Reinforcement Learning*. arXiv: 1910.06764 [cs.LG]. URL: https://arxiv.org/abs/1910.06764.

Peng, Xue Bin et al. (2019). *Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning*. arXiv: 1910.00177 [cs.LG]. URL: https://arxiv.org/abs/1910.00177.

Song, H. Francis et al. (2019). *V-MPO: On-Policy Maximum a Posteriori Policy Optimization for Discrete and Continuous Control*. arXiv: 1909.12238 [cs.AI]. URL: https://arxiv.org/abs/1909.12238.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.