Expectation–Maximization Frameworks for LLM RL Fine-Tuning

February 19, 2026

Contents

1 Research Questions

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.

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?

2 Preliminaries

Maximum Likelihood Estimation and Maximum a Posteriori

MLE and MAP are methods for estimating some variable in the setting of probability distributions. They compute a single estimate, instead of a full distribution.

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

$$\theta_{\mathrm{MLE}} = \arg\max_{\theta} \log p_\theta(X) = \arg\max_{\theta} \sum_{i=1}^{N} \log p_\theta(x_i).$$

Maximum a posteriori (MAP) estimation adds a prior, coming from the Bayesian perspective:

$$p(\theta \mid X) = \frac{p(x \mid \theta)\, p(\theta)}{p(x)}$$

$$p(\theta \mid X) \propto p(X \mid \theta)\, p(\theta)$$

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta} \left[\sum_{i=1}^{N} \log p(x_i \mid \theta) + \log p(\theta) \right] = \arg\max_{\theta} (\log p_\theta(X) + \log p(\theta)).$$

EM Lower Bound – ELBO

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

$$\mathcal{L}_{\mathrm{MAP}}(\theta) = \log p_\theta(x) + \log p(\theta)$$

admits the decomposition

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

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 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 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.

3 V-MPO: Maximum a Posteriori Policy Optimization

V-MPO (Song et al. 2019) decomposes optimisation into EM phases. Its pairing with the Gated Transformer-XL architecture for RL is described in Appendix B.

The total objective is

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

where

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

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 $\psi(s, a)$:

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

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

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

Here, stationarity refers to the KKT first-order optimality condition with respect to the variational distribution ψ:

$$\frac{\partial J}{\partial \psi(s, a)} = A^{\pi_{\theta_{\text{old}}}}(s, a) - \eta \left(\log \frac{\psi(s, a)}{p_{\theta_{\text{old}}}(s, a)} + 1 \right) - \lambda = 0, \quad \forall (s, a).$$

Solving this stationarity condition for ψ 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

$$J(\theta,\alpha) = \mathcal{L}_\pi(\theta) + \alpha\Big(\epsilon_\alpha - \mathbb{E}_s \mathrm{KL}(\pi_{\theta_{\text{old}}} \| \, \pi_\theta)\Big). \tag{1}$$

In implementation, the loss becomes

$$\begin{aligned}\mathcal{L}_\alpha(\theta,\alpha) =& \alpha\Big(\epsilon_\alpha - \mathrm{sg}[\mathrm{KL}(\pi_{\theta_{\text{old}}} \| \, \pi_\theta)]\Big) \\ &+ \mathrm{sg}[\alpha]\, \mathrm{KL}(\pi_{\theta_{\text{old}}} \| \, \pi_\theta).\end{aligned} \tag{2}$$

4 PPO: Proximal Policy Optimization

PPO is a PG method as it directly estimates and follows the gradient of expected return in parameter space with 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

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

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.

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

When the advantage is positive the update is capped at a ratio of $1+\epsilon$; when negative it is capped at $1-\epsilon$.

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

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

An entropy bonus encourages exploration by penalising premature policy collapse:

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

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:

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

which satisfies the efficient recurrence

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

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

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

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

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:

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

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

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

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

5 DPO: Direct Preference Optimization

Direct Preference Optimization (DPO) (Rafailov et al. 2023) is a preference MLE method: it bypasses reward modelling and policy-gradient estimation entirely, reducing alignment to a single binary cross-entropy objective over preference pairs.

Unlike policy-gradient methods, DPO needs no reward model (the reward is implicit), no value function, no advantage estimation, no importance-sampling ratios, and no online rollouts. The entire training signal comes from offline preference pairs.

The standard RLHF objective is

$$\max_\pi \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi(\cdot|x)}[r(x, y)] - \beta \, D_{\mathrm{KL}}(\pi(\cdot|x) \,\|\, \pi_{\mathrm{ref}}(\cdot|x)), \tag{3}$$

where $\beta > 0$ controls regularisation toward the reference policy π_{ref}.

Closed-form optimal policy. The optimisation over π for each x is a KL-regularised linear problem with the same structure as the GEMPI E-step. The solution is

$$\pi^\star(y \mid x) = \frac{1}{Z(x)} \, \pi_{\mathrm{ref}}(y \mid x) \, \exp\Big(\frac{r(x, y)}{\beta}\Big), \tag{4}$$

where $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp(r(x, y)/\beta)$ is the partition function. This is a GEMPI E-step with $m = 1$, $\pi_1 = \pi_{\mathrm{ref}}$, $\lambda_1 = \beta$, $\Lambda = \beta$.

Reward reparameterisation. Rearranging (4) expresses the reward as a function of the optimal policy:

$$r(x, y) = \beta \log \frac{\pi^\star(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \tag{5}$$

Bradley–Terry preference model. The probability of preferring response y_w over y_l under the Bradley–Terry model is $p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$, where σ is the logistic function. Substituting (5):

$$p(y_w \succ y_l \mid x) = \sigma\bigg(\beta \log \frac{\pi^\star(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi^\star(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\bigg). \tag{6}$$

The partition function $Z(x)$ cancels in the reward difference—this is the key algebraic step that makes DPO tractable.

The DPO loss. Replacing the unknown $\pi^\star$ with a parametric policy π_θ and maximising the log-likelihood of observed preferences yields the DPO objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma\bigg(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\bigg) \right]. \tag{7}$$

DPO as Preference MLE The DPO loss (7) is binary cross-entropy over preference pairs—structurally, it is MLE on a classification task. The gradient has the form

$$\nabla_\theta \mathcal{L}_{\text{DPO}} = -\beta\, \mathbb{E}\Big[\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w))(\nabla_\theta \log \pi_\theta(y_w \mid x) - \nabla_\theta \log \pi_\theta(y_l \mid x))\Big],$$

where $\hat{r}_\theta(y) \coloneqq \beta \log(\pi_\theta(y \mid x)/\pi_{\text{ref}}(y \mid x))$ is the implicit reward. The sigmoid weight measures how wrong the current model is on this pair: it is large when the model incorrectly prefers y_l, and vanishes as the model learns the correct ranking. This is a weighted MLE update that increases the log-probability of the preferred response and decreases that of the dispreferred one.

DPO and the EM Framework The closed-form optimal policy (4) is identical to GEMPI with $m = 1$, $D_1 = \text{KL}$, $\pi_1 = \pi_{\text{ref}}$, $\Lambda = \beta$:

$$q^\star(y \mid x) \propto \pi_{\text{ref}}(y \mid x)\, \exp\Big(\frac{r(x, y)}{\beta}\Big).$$

In a standard EM method (V-MPO, DAR, AWR), one would now construct this target distribution and then project to π_θ via weighted MLE in the M-step.

The "collapsed EM" interpretation. DPO eliminates the explicit E-step entirely. Instead of constructing $q^\star$ and then fitting π_θ to it, DPO substitutes the functional form of the optimal policy directly into the Bradley–Terry preference likelihood. The intractable partition function $Z(x)$ cancels in the reward difference $r(x, y_w) - r(x, y_l)$, yielding a closed-form objective that implicitly solves both the E-step and M-step in a single stage.

What is lost. By collapsing the two stages, DPO gives up several degrees of freedom available to explicit EM methods:

- **Reward flexibility.** DPO's reward is defined as $\beta \log(\pi_\theta/\pi_{\text{ref}})$; it cannot incorporate richer reward signals (e.g. multi-aspect scores, process rewards) without redefining the objective.
- **Temperature adaptation.** The coefficient β is fixed; there is no dual mechanism to adapt it online as in V-MPO.
- **Multi-anchor regularisation.** DPO anchors to π_{ref} only. DAR's dual-anchor structure (π_{ref} and π_t) is not available within the DPO framework.
- **Online data.** DPO operates on a fixed offline preference dataset; the policy does not generate new rollouts during training.

Dropout compatibility. DPO's gradient involves only single forward passes through π_θ (computing $\log \pi_\theta(y_w \mid x)$ and $\log \pi_\theta(y_l \mid x)$), with no ratio between two forward passes under different modes. Dropout is therefore compatible, for the same structural reasons of SFT and EM-style M-steps.

6 AWR: Advantage-Weighted Regression

Consider an MDP with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, reward $r(s,a)$ and discount $\gamma \in [0,1)$. For a policy π define the return of a state–action pair under a sampling policy μ as

$$R^\mu_{s,a} \;=\; \sum_{t=0}^{\infty} \gamma^t r_t, \qquad V^\mu(s) \;=\; \mathbb{E}_{a\sim\mu(\cdot|s)}[R^\mu_{s,a}], \qquad A^\mu(s,a) \;=\; R^\mu_{s,a} - V^\mu(s).$$

Expected improvement objective. Define the expected improvement of a candidate policy π over μ as

$$\eta(\pi) \;=\; J(\pi) - J(\mu) \;=\; \mathbb{E}_{s\sim d_\pi}\mathbb{E}_{a\sim\pi(\cdot|s)}[A^\mu(s,a)],$$

where $d_\pi(s) = \sum_{t=0}^{\infty} \gamma^t p(s_t = s \mid \pi)$ is the unnormalised discounted state distribution. To avoid sampling from π during the optimisation we use the common first-order surrogate that uses the sampling state distribution d_μ:

$$\widehat{\eta}(\pi) \;=\; \mathbb{E}_{s\sim d_\mu}\mathbb{E}_{a\sim\pi(\cdot|s)}[A^\mu(s,a)].$$

Constrained policy search (primal). We formulate a constrained optimisation that maximises the surrogate expected improvement while keeping π close to μ in KL averaged under d_μ:

$$\begin{aligned} &\max_{\pi} \quad \mathbb{E}_{s\sim d_\mu}\mathbb{E}_{a\sim\pi(\cdot|s)}[R^\mu_{s,a} - V^\mu(s)], \\ &\text{s.t.} \quad \mathbb{E}_{s\sim d_\mu}[D_{\mathrm{KL}}(\pi(\cdot|s) \,\|\, \mu(\cdot|s))] \le \varepsilon, \qquad \forall s: \; \int_a \pi(a|s)\, da = 1. \end{aligned}$$

Lagrangian and stationarity. Introduce Lagrange multiplier $\beta > 0$ for the KL constraint and pointwise normaliser α_s. The (softened) Lagrangian is

$$\mathcal{L}(\pi,\beta,\alpha) = \mathbb{E}_{s\sim d_\mu}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}[R^\mu_{s,a} - V^\mu(s)]\Big] + \beta\Big(\varepsilon - \mathbb{E}_{s\sim d_\mu}[D_{\mathrm{KL}}(\pi(\cdot|s)\|\mu(\cdot|s))]\Big) + \mathbb{E}_{s\sim d_\mu}[\alpha_s(1 - \int_a \pi(a|s)\, da)].$$

Differentiate $\mathcal{L}$ w.r.t. $\pi(a|s)$, set derivative to zero and solve for π. Rearranging yields the Boltzmann-tilted closed form for the optimal (per-state) conditional distribution:

$$\pi^\star(a|s) \;=\; \frac{1}{Z(s)}\, \mu(a|s)\, \exp\Big(\frac{1}{\beta}(R^\mu_{s,a} - V^\mu(s))\Big), \qquad Z(s) \;=\; \int \mu(a'|s) \exp\Big(\frac{1}{\beta}(R^\mu_{s,a'} - V^\mu(s))\Big)\, da'.$$

This formula is obtained by solving the stationarity condition $\partial\mathcal{L}/\partial\pi(a|s) = 0$ and enforcing normalisation; it is the soft, advantage-weighted reweighting of the behaviour distribution $\mu(a|s)$.

Projection to parameterised policy (regression step). If π is parameterised (e.g. neural network) we cannot set it equal to $\pi^\star$ pointwise. Instead project $\pi^\star$ onto the parameterised family by minimising expected KL:

$$\pi_{k+1} \;=\; \arg\min_{\pi\in\Pi} \; \mathbb{E}_{s\sim d_\mu}[D_{\mathrm{KL}}(\pi^\star(\cdot|s) \,\|\, \pi(\cdot|s))].$$

Expanding the KL and substituting $\pi^\star$ (6) yields an equivalent supervised regression objective:

$$\pi_{k+1} \;=\; \arg\max_{\pi\in\Pi} \; \mathbb{E}_{s\sim d_\mu}\mathbb{E}_{a\sim\mu(\cdot|s)}\Big[\log \pi(a|s)\; \exp\Big(\frac{1}{\beta}(R^\mu_{s,a} - V^\mu(s))\Big)\Big].$$

Fit π by maximum likelihood on behaviour data weighted by exponentiated advantages.

Value update. AWR alternates the policy update with a value regression that fits V to returns in $\mathcal{D}$, typically by minimising a squared TD(λ) or Monte-Carlo return loss:

$$V^{\mathcal{D}} \;=\; \arg\min_{V}\; \mathbb{E}_{(s,a)\sim\mathcal{D}}[\,(R^{\mathcal{D}}_{s,a} - V(s))^2].$$

7 DAR: 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.

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:

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

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

Advantage Weighted Regression Advantage Weighted Regression (AWR) maximizes:

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

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

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

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:

$$\phi(x,y) \;\coloneqq\; \pi_{\mathrm{ref}}(y \mid x)^{\frac{\alpha}{\alpha+\beta}}\; \pi_t(y \mid x)^{\frac{\beta}{\alpha+\beta}}\; \exp\!\left(\frac{A(x,y)}{\alpha+\beta}\right), \pi^\star(y \mid x) = \frac{1}{Z(x)}\,\phi(x,y),$$

where $Z(x) = \sum_y \phi(x,y)$, 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:

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

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

$$\pi^\star(y|x) \;=\; \frac{1}{Z(x)}\,\pi_{\text{ref}}(y|x)^{\frac{\alpha}{\alpha+\beta}}\pi_t(y|x)^{\frac{\beta}{\alpha+\beta}}\exp\Big(\frac{1}{\alpha+\beta}A(x,y)\Big). \tag{8}$$

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

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

Substituting $\pi^\star$ gives:

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

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

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

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

$$\pi^\star(y\,|\,x) \;=\; \frac{1}{Z(x)}\;\pi_{\text{ref}}(y\,|\,x)^{\frac{\alpha}{\alpha+\beta}}\;\pi_t(y\,|\,x)^{\frac{\beta}{\alpha+\beta}}\;\exp\Big(\frac{A(x,y)}{\alpha+\beta}\Big), \tag{9}$$

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 (9) produces the additive form used below:

$$\log\pi^\star(y\,|\,x) = -\log Z(x) + \frac{\alpha}{t}\log\pi_{\text{ref}}(y\,|\,x) + \frac{\beta}{t}\log\pi_t(y\,|\,x) + \frac{1}{t}A(x,y). \tag{10}$$

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:

$$\psi_{\text{V–MPO}}(y\,|\,x) \propto \pi_t(y\,|\,x)\ \exp\Big(\frac{A(x,y)}{\eta}\Big), \tag{11}$$

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 (9). Then the multiplicative powers collapse:

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

Substituting into (9) gives

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

Comparing with (11) 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 (10),

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

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

again matching (11) 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.

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 and explicit dual updates which require numerical handling when lifted to LLMs.

Bridge to the EM. Two views of the 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 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. (24). The per-sequence weights differ only by the base factor:

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

8 MaxMin-RLHF

MaxMin-RLHF (Chakraborty et al. 2024) can be interpreted as a nested Expectation–Maximisation procedure for alignment under heterogeneous preferences. An EM algorithm learns a mixture of subgroup-specific reward models from pairwise preference data by treating subgroup identity as a latent variable and alternating between estimating subgroup responsibilities (E-step) and maximising subgroup-specific reward likelihoods (M-step). Policy optimisation proceeds via a KL-regularised EM update, but with the reward model selected adversarially: at each iteration the subgroup whose expected utility is currently minimal under the policy is identified, and the policy is updated through an exponential-tilted target distribution derived from that subgroup's reward, followed by a projection back to the parametric policy family.

Method. Let $\mathcal{V}$ be the token vocabulary and $\mathcal{X}$ the space of prompts. A language model (policy) is $\pi_\theta(\cdot \mid x)$ parameterised by θ and, given a prompt $x \in \mathcal{X}$, produces a response sequence $y \sim \pi_\theta(\cdot \mid x)$.

Preference data are pairwise comparisons (x, y_1, y_2) where a human annotator prefers y_1 to y_2. Under the Bradley–Terry (BT) parametrisation the preference probability induced by a latent reward $r^\star(y, x)$ is

$$p^\star(y_1 \succ y_2 \mid x) \;=\; \sigma\big(r^\star(y_1, x) - r^\star(y_2, x)\big) \;=\; \frac{\exp(r^\star(y_1, x))}{\exp(r^\star(y_1, x)) + \exp(r^\star(y_2, x))}, \tag{12}$$

where σ is the logistic function. A parametric reward model is $r_\phi(y, x)$, trained by binary cross-entropy (negative log-likelihood)

$$\mathcal{L}_R(\phi; \mathcal{D}) = -\mathbb{E}_{(x, y_1, y_2)\sim\mathcal{D}}\Big[\log\sigma\big(r_\phi(y_1, x) - r_\phi(y_2, x)\big)\Big].$$

The KL-regularised reinforcement objective used for policy fine-tuning is

$$\mathcal{J}(\pi; r) = \mathbb{E}_{x\sim\mathcal{P},\, y\sim\pi(\cdot|x)}[r(y, x)] - \beta\,\mathbb{E}_{x\sim\mathcal{P}}[D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))], \tag{13}$$

with regularisation weight $\beta > 0$ and reference policy π_{ref} (SFT checkpoint).

Diversity and an impossibility bound. Define a population composed of $|U|$ sub-populations $H = \bigcup_{u=1}^{|U|} H_u$. Let $p_u^\star$ denote the preference distribution for sub-population H_u. Diversity between two sub-populations is measured by total variation

$$\mathrm{Diversity}(i, j) \coloneqq \mathrm{TV}(p_i^\star, p_j^\star).$$

The paper proves (Theorem 1) that, under reasonable assumptions (linear reward parametrisation, bounded features, Lipschitz mapping from reward to policy) the alignment gap for single-reward RLHF admits a lower bound proportional to the population diversity and inversely proportional to model regularity constants. In one scalar form (adapted notation) the lower bound reads

$$\mathrm{Align-Gap} \;\gtrsim\; \frac{\lambda_\Psi}{\beta^2 L_\pi}\,\frac{\epsilon(1-\eta(u))}{D^2}, \tag{14}$$

where λ_Ψ is the minimum eigenvalue of the empirical feature matrix, L_π a Lipschitz constant relating policy and reward, β the KL weight, D a feature bound, $\eta(u)$ the mixing weight of subgroup u, and ϵ a gap derived from pairwise total-variation distances. The consequence is that high subpopulation diversity or vanishing subgroup mass $\eta(u)$ make accurate alignment via a single reward improbable. See the original proof and the supporting lemma for the precise constants and derivation.

To address this failure mode, we learn a mixture of $|U|$ reward models $\{r_{\phi_u}\}_{u=1}^{|U|}$ and simultaneously clusters annotators (or annotator-derived signals) so that each component models a coherent sub-population preference. Algorithmically this is solved with a standard hard/soft EM procedure. We reproduce the key steps and the small algebraic derivation that yields the E-step responsibilities.

Generative model for pairwise labels. For a fixed subpopulation index u, assume the pairwise preference likelihood under reward r_{ϕ_u} follows (12). For an observed labelled pair (x, y_1, y_2) where y_1 is preferred, the likelihood is

$$p(z = (y_1 \succ y_2) \mid u, \phi_u, x) \;=\; \frac{\exp(r_{\phi_u}(y_1, x))}{\exp(r_{\phi_u}(y_1, x)) + \exp(r_{\phi_u}(y_2, x))}.$$

This can be re-written as a normalised exponential,

$$p(z \mid u, \phi_u, x) = \frac{\exp(r_{\phi_u}(y_1, x))}{\exp(r_{\phi_u}(y_1, x)) + \exp(r_{\phi_u}(y_2, x))} \;=\; \frac{\exp(r_{\phi_u}(y_1, x))}{\sum_{j\in\{1,2\}} \exp(r_{\phi_u}(y_j, x))}.$$

E-step (responsibilities). Given a current parameter set $\{\phi_u\}$ and prior cluster weights, the posterior responsibility (or soft assignment) of cluster u for datum (x, y_1, y_2) is proportional to the product of the cluster prior and the pairwise likelihood. Under equal priors (or when priors are absorbed in normalisation) the local responsibility used in the paper's hard-assignment E-step reduces to the BT soft-likelihood:

$$w(\phi_u; x, y_1, y_2) \;=\; \frac{\exp(r_{\phi_u}(y_1, x))}{\exp(r_{\phi_u}(y_1, x)) + \exp(r_{\phi_u}(y_2, x))}. \tag{15}$$

This expression follows directly from the BT model and is the quantity used to assign users to clusters (Algorithm 8).

M-step (reward update). Given cluster assignments (hard or soft), update each reward parameter ϕ_u by minimising the negative log-likelihood restricted to the data assigned to cluster u, i.e.

$$\phi_u \leftarrow \arg\min_{\phi} \; - \sum_{(x,y_1,y_2)\in\mathcal{D}_u} \log \sigma(r_\phi(y_1, x) - r_\phi(y_2, x)),$$

where $\mathcal{D}_u$ are the pairs assigned to cluster u. In practice this is SGD on the binary cross-entropy loss. Convergence of the EM loop is monitored by cluster-stability or likelihood improvement.

Objective and policy iteration. Once a set of reward models $\{r_{\phi_u}\}$ is obtained, the alignment objective pursued is an Egalitarian (max–min) social-utility objective:

$$\pi^\star_{\mathrm{MM}} \;=\; \arg\max_{\pi} \; \min_{u\in U} \; \Big\{ F_{r_{\phi_u}}(\pi) - \beta\, \mathbb{E}_x[D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))] \Big\}, \tag{16}$$

with $F_r(\pi) := \mathbb{E}_{x\sim\mathcal{P},\, y\sim\pi(\cdot|x)}[r(y, x)]$ the expected return. Optimisation is performed by iterated policy improvement that (1) selects the worst-performing sub-population under the current policy and (2) updates the policy using a standard RL fine-tuner (e.g. PPO) toward improving that subgroup's objective. Concretely the paper proposes:

1. Compute $u_{\min} \leftarrow \arg\min_{u\in U} F_{r_{\phi_u}}(\pi_t)$.
2. Perform a policy update (PPO) to maximise $F_{r_{\phi_{u_{\min}}}}(\pi) - \beta\,\mathrm{KL}(\pi_t \| \pi_{\mathrm{ref}})$, producing π_{t+1}.
3. Repeat until convergence.

This iterated max–min policy iteration is summarised in Algorithm 8 below.

Algorithm 1: MaxMin-RLHF (policy-level)

1. **Input:** preference dataset $\mathcal{D}$, initial reward parameters $\{\phi_u^{(0)}\}_{u=1}^{|U|}$ (or pretraining), initial policy π_0, KL weight β, PPO (or other RL) trainer.
2. **Reward learning:** run Algorithm 8 to estimate $\{\phi_u\}$ (mixture learning via EM).
3. **For** $t = 0, \ldots, T-1$:
   (a) Evaluate subgroup utilities $F_{r_{\phi_u}}(\pi_t)$ for all $u \in U$ (via rollouts or importance-weighted evaluation).
   (b) Select $u_{\min} \leftarrow \arg\min_u F_{r_{\phi_u}}(\pi_t)$.
   (c) Update policy using PPO (or another KL-regularised optimiser) towards improving subgroup $u_{\min}$:
   $$\pi_{t+1} \leftarrow \mathrm{PPO\_step}\Big(\pi_t,\ r_{\phi_{u_{\min}}},\ \beta,\ \pi_{\mathrm{ref}}\Big).$$
4. **Output:** π_T aligned to the max-min objective.

Algorithm 2: Learning rewards with EM (clustered reward models)

1. **Input:** pairwise preference dataset $\mathcal{D}$, number of clusters $|U|$, initial $\{\phi_u\}$.
2. **Repeat until convergence:**
   (a) **E-step (hard or soft):** for each annotator (or datum) compute responsibilities $w(\phi_u; x, y_1, y_2)$ as in (15) and assign datum to the cluster with largest responsibility (hard assignment) or keep soft weights.
   (b) **M-step:** for each u update ϕ_u by minimising the binary cross-entropy over data assigned to u:
   $$\phi_u \leftarrow \arg\min_{\phi} - \sum_{(x,y_1,y_2)\in\mathcal{D}_u} \log \sigma(r_\phi(y_1, x) - r_\phi(y_2, x)).$$
3. **Return:** $\{\phi_u\}$ and cluster assignments.

9 Generalized EM Policy Improvement (GEMPI)

This is an attempt to describe a generalised EM policy improvement method that encompasses a wide range of algorithms as special cases. The key idea is to allow multiple anchor policies and multiple divergence functionals in the E-step, as well as flexible temperature and filtering mechanisms.

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:

$$q^\star = \arg\max_q \left\{ \mathbb{E}_{y\sim q}[A(x,y)] - \sum_{j=1}^m \lambda_j\, D_j(q(\cdot \mid x) \,\|\, \pi_j(\cdot \mid x)) \right\}, \tag{17}$$

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 (17) is

$$q^\star(y \mid x) = \frac{1}{Z(x)} \prod_{j=1}^{m} \pi_j(y \mid x)^{\lambda_j/\Lambda} \, \exp\Big(\frac{A(x,y)}{\Lambda}\Big), \tag{18}$$

where $\Lambda \coloneqq \sum_{j=1}^{m} \lambda_j$ 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 (18).

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

$$\ell(x,y) = \sum_{j=1}^{m} \frac{\lambda_j}{\Lambda} \log \pi_j(y \mid x) + \frac{1}{\Lambda} A(x,y). \tag{19}$$

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:

$$L_{\text{dual}}(\lambda_1, \ldots, \lambda_m) = \sum_{j=1}^{m} \lambda_j \, \varepsilon_j + \Lambda \, \log Z(x; \lambda_1, \ldots, \lambda_m). \tag{20}$$

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:

$$\theta_{k+1} = \arg\min_\theta \Big\{ -\mathbb{E}_{q^\star}[\log \pi_\theta(y \mid x)] + \gamma \, D_M(\pi_{\theta_k}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)) \Big\}, \tag{21}$$

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 constraining the parametric policy. 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 (18) 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 (21) 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
DPO (collapsed)	1	π_{ref}	β	β	Fixed	None	—

V-MPO. Set $m = 1$, $D_1 = \mathrm{KL}$, $\pi_1 = \pi_t$. The closed form (18) 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 (20) (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.

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 (18) 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. (9).

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 AWR (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 ML. This corresponds to $m = 0$ in the GEMPI tuple.

DPO (collapsed). The KL-regularised RLHF objective (3) yields the same E-step closed form as GEMPI with $m = 1$, $\pi_1 = \pi_{\mathrm{ref}}$, $\Lambda = \beta$: namely $q^\star \propto \pi_{\mathrm{ref}} \exp(r/\beta)$. DPO does not construct this target explicitly. Instead, it substitutes the optimal policy form into the Bradley–Terry preference likelihood, and the partition function $Z(x)$ cancels in the reward difference. The result is a single-stage preference MLE (7) that implicitly solves both the E-step and M-step. DPO can thus be viewed as a degenerate GEMPI instantiation in which the E-step is analytically eliminated.

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.

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 (17) is defined for arbitrary divergences D_j. Replacing KL with another f-divergence 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) → NPG/TRPO	$D_j = \mathrm{KL}$ → V-MPO/DAR/AWR
$h = \frac{1}{2}\|\cdot\|^2$ → proj. gradient	$D_j = \chi^2$ → 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. (20)) 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

Dropout (Srivastava et al. 2014) randomly zeros each hidden activation with probability p during training, scaling survivors by $1/(1-p)$. Most large-scale pre-training runs disable it, but during fine-tuning the dataset is orders of magnitude smaller and overfitting is a real concern, making dropout relevant again. Yet all mainstream PPO implementations forgo dropout entirely. The GEMPI decomposition explains why ratio-based methods cannot use dropout while EM-style methods can.

Why ratio-based methods break. PPO's clipped objective depends on the importance-sampling ratio $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$. If dropout is active during optimisation, three problems arise: (i) the numerator π_θ becomes a random variable that fluctuates across forward passes even for fixed θ, injecting noise into the narrow clip window $[1-\epsilon, 1+\epsilon]$; (ii) the denominator is computed once at rollout time in eval mode (no dropout), so the ratio conflates policy change with mask change, breaking the KL-constraining guarantee of clipping; (iii) auxiliary KL penalties $\widehat{\text{KL}} = \mathbb{E}_t[\ell_{\text{old},t} - \ell_{\theta,t}]$ are biased upward by mask-induced variance, causing premature termination of updates.

Why GEMPI methods are compatible. The GEMPI template guarantees dropout compatibility whenever the M-step uses detached (stop-gradient) weights from the E-step. The E-step weights depend on advantages and anchor log-probabilities, not on a ratio of two forward passes under different modes. The M-step is weighted teacher-forced cross-entropy—structurally identical to SFT with per-sequence weights—so dropout is compatible for the same reason it is compatible with any supervised cross-entropy objective. 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 estimator of the weighted-MLE gradient under the dropout distribution, exactly as in supervised learning. No eval-mode forward pass needs to be compared against this quantity.

Summary. The key structural difference is:

	PPO	**GEMPI (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

All methods instantiated within GEMPI inherit this M-step dropout compatibility.

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.

Worked Instantiation: LLM V-MPO

This subsection 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 tuple. The LLM variant corresponds to

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

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

$$\psi(i) \propto \exp\Big(\frac{A^{(i)}}{\eta}\Big). \tag{22}$$

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

$$g(\eta) = m + \log\Big(\frac{1}{k}\sum_{i \in S} \exp(u_i - m)\Big), \qquad u_i = \frac{A^{(i)}}{\eta}, \quad m = \max_{i \in S} u_i. \tag{23}$$

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:

$$\mathcal{L}_\pi(\theta) = -\sum_{i \in S} w_i \sum_{t=1}^{T_i} \log \pi_\theta(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)}). \tag{24}$$

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 compatibility. Because the M-step weights are detached from current parameters, dropout applies directly as standard regularisation (Section 9).

10 References

Chakraborty, Souradip et al. (2024). *MaxMin-RLHF: Alignment with Diverse Human Preferences*. arXiv: 2402.08925 [cs.CL]. URL: https://arxiv.org/abs/2402.08925.

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.

Rafailov, Rafael et al. (2023). *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. arXiv: 2305.18290 [cs.LG]. URL: https://arxiv.org/abs/2305.18290.

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.

A Method Comparison

Criterion	PPO	MPO / V-MPO	AWR	DAR	DPO	GEMPI
Optimisation paradigm	Policy gradient (PMD, neg. entropy mirror map)	EM / regularised policy iteration	EM / weighted regression	EM / closed-form dual-KL target	Preference MLE (collapsed EM)	Generalised EM (subsumes all)
Core objective	Clipped IS surrogate $L^{\mathrm{CLIP}}(\theta)$	MAP improvement $\log p_\theta(I{=}1){+}\log p(\theta)$ (KL-constrained)	Advantage-weighted MLE	Dual-KL advantage maximisation J_{DAR}	Binary cross-entropy on pref. pairs $\mathcal{L}_{\mathrm{DPO}}$	$\max_q \mathbb{E}_q[A] - \sum_j \lambda_j D_j(q\|\pi_j)$, then project
Update phases	Single-phase gradient step	E-step (non-parametric target) + M-step (parametric projection)	E-step + M-step	E-step (closed form) + M-step	Single-phase; no explicit E-step	E-step + M-step (fully configurable)
E-step target $q^\star$	N/A	$\pi_t \exp(A/\eta)$	$\pi_t \exp(A/\beta)$	$\pi_{\mathrm{ref}}^{\alpha/\Lambda}\pi_t^{\beta/\Lambda}\exp(A/\Lambda)$	$\pi_{\mathrm{ref}}\exp(r/\beta)$ (implicit; never materialised)	$\prod_j \pi_j^{\lambda_j/\Lambda}\exp(A/\Lambda)$
E-step anchors (m)	N/A	1 (π_t)	1 (π_t)	2 ($\pi_{\mathrm{ref}}, \pi_t$)	1 (π_{ref}, implicit)	$m \geq 0$ (free)
KL anchor(s)	Implicit via clipping; optional KL penalty to π_{old}	π_t (E-step); π_t (M-step trust region)	π_t (E-step only)	π_{ref} and π_t (dual anchor)	π_{ref} (single, fixed)	Any set $\{\pi_j\}$
Temperature mechanism	N/A; clip threshold ϵ fixed	Adaptive dual: η optimised via convex $\mathcal{L}_\eta$	Fixed β	Fixed (α, β); AT-DAR variant makes both adaptive	Fixed β	Fixed or dual-adaptive (λ_j per anchor)
Gradient signal	$\nabla_\theta r_t(\theta)\, A_t$ (ratio-coupled to θ)	$w_i \nabla_\theta \log \pi_\theta$ (detached weights)	$w_i \nabla_\theta \log \pi_\theta$ (detached)	$w_i \nabla_\theta \log \pi_\theta$ (detached)	$\sigma(\cdot)(\nabla_\theta \log \pi_\theta(y_w) - \nabla_\theta \log \pi_\theta(y_l))$	$w_i \nabla_\theta \log \pi_\theta$ (detached)
IS ratio $\pi_\theta/\pi_{\mathrm{old}}$ required?	**Yes** — clipped to $[1\pm\epsilon]$	No	No	No (optional IS correction if off-policy)	No	No (IS weights optional for off-policy)
M-step loss	N/A (single-phase)	$-\sum_i w_i \log \pi_\theta(a_i\|s_i)$ + KL trust region	$-\sum_i w_i \log \pi_\theta(a_i\|s_i)$ (unconstrained)	$-\sum_i w_i \log \pi_\theta(y_i\|x_i)$ (unconstrained)	N/A (collapsed)	$-\mathbb{E}_{q^\star}[\log \pi_\theta] + \gamma D_M(\pi_{\theta_k}\|\pi_\theta)$
M-step trust region	N/A	**Yes** — KL with Lagrange dual α (1)	No	No; DAR-TR variant adds KL constraint	N/A	Optional (D_M, budget ε_M)
Top-k / sample filtering	No (full-batch IS)	**Yes** — top-k fraction ρ by advantage	No (all samples)	No; Top-k DAR variant available	No	Configurable ($\mathcal{F}$: identity or top-k)
Value function / baseline?	Yes — GAE with V_ϕ	Yes — n-step bootstrapped V_ϕ^π	Yes — $V^\mu(s)$ by TD/λ	Yes — $V^{\pi_t}(x)$ for advantage	No	Yes (when A uses a learned baseline)
Explicit reward model?	Yes	Yes	Yes	Yes	No — implicit $\hat{r}_\theta{=}\beta\log(\pi_\theta/\pi_{\mathrm{ref}})$	Yes (typical; not mandated by framework)
Online / offline?	Online (on-policy rollouts; mini-epochs with IS)	Online (on-policy rollouts)	Offline or near-off-policy	Online (rollouts from π_t)	Offline (fixed preference dataset)	Configurable
Training data format	State–action trajectories + scalar rewards	State–action trajectories + scalar rewards	State–action trajectories + returns	Prompt–response pairs + scalar rewards	Pairwise preference triples (x, y_w, y_l)	General (reward or preference signals)
Dropout compatible?	**No** — stochastic numerator corrupts IS ratio; inconsistent denominator	**Yes** — detached M-step weights; no ratio in objective	**Yes** — detached weights; standard cross-entropy	**Yes** — detached weights	**Yes** — single forward passes only; no ratio	**Yes** — guaranteed when M-step weights detached
EM / MAP derivation?	No — direct PG theorem	**Yes** — full MAP EM (Sections 3–4)	**Yes** — constrained policy search $\rightarrow$ Boltzmann $\pi^\star$	**Yes** — dual-KL optimisation $\rightarrow$ closed-form $\pi^\star$	Partial — optimal policy derived, then EM collapsed via BT cancellation	**Yes** — defines the general EM template
Adaptive temperature?	No	**Yes** (η via dual $\mathcal{L}_\eta$)	No (β fixed)	No (fixed α, β); AT-DAR variant adds it	No (β fixed)	Yes (dual-adaptive mode)
GEMPI tuple	N/A (policy-gradient paradigm)	$m{=}1$, $D_1{=}$KL, $\pi_1{=}\pi_t$, dual η, top-k, $D_M{=}$KL	$m{=}1$, $\pi_1{=}\pi_t$, $\lambda_1{=}\beta$ fixed, no D_M, no filter	$m{=}2$, $\pi_1{=}\pi_{\mathrm{ref}}$, $\pi_2{=}\pi_t$, (α, β) fixed, no D_M	$m{=}1$, $\pi_1{=}\pi_{\mathrm{ref}}$, $\Lambda{=}\beta$, E-step collapsed (no M-step)	Full tuple $\mathcal{G}$ — all axes free
Novel variants / extensions	Seq-level PPO (product IS ratio; KL early stop)	LLM V-MPO (seq-level E/M-step, Section 9)	—	AT-DAR; DAR-TR (M-step KL); Top-k DAR	IPO, β-DPO, rDPO	Any combo of m, D_j, $\mathcal{T}$, $\mathcal{F}$, D_M
Primary LLM use-case	Online RLHF (InstructGPT, LLaMA-RLHF)	Online RL fine-tuning with EM stability	Near-offline advantage-weighted SFT	Online alignment with dual regularisation	Offline preference alignment (no RM needed)	Unifying design framework

Notes. *Detached weights*: the per-sample weights w_i entering the M-step are computed with `stop_gradient`, so gradients flow only through $\log \pi_\theta$, not through the weights. $\Lambda := \sum_j \lambda_j$ is the effective temperature in the GEMPI E-step. AT-DAR and DAR-TR are novel instantiations derived within GEMPI (Section 9). BT: Bradley–Terry preference model used in the DPO derivation.

B GTrXL: Gated Transformer-XL for RL

The V-MPO authors (Song et al. 2019) reported strong empirical results pairing V-MPO with a Transformer-XL (TrXL) backbone (Dai et al. 2019) on Atari, and later with the Gated Transformer-XL (GTrXL) (Parisotto et al. 2019) on DMLab-30. This appendix summarises the GTrXL modifications and why they complement V-MPO's EM structure. Reference implementations are available at github.com/kimiyoung/transformer-xl and github.com/nenuadrian/DI-engine.

Standard TrXL fails in RL, performing at random-policy level on benchmarks such as DMLab-30. GTrXL 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 clean gradient path, making the training landscape more amenable to policy optimisation.

GRU 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 $b_g = 2$ places each gate near the identity at the start of training, preserving the reactive-policy initialisation from the layer-norm reordering.

Why GTrXL suits V-MPO. V-MPO collects fresh on-policy trajectories each update, making long-horizon memory critical. GTrXL provides: (i) **long-range memory** via relative position encodings spanning thousands of past time-steps; (ii) **stable optimisation** — GRU gating achieves a 0% divergence rate across hyperparameter sweeps vs. 16% for plain TrXL-I, compatible with V-MPO's fixed Adam learning rate; (iii) **robust performance** — on DMLab-30, GTrXL (GRU) reaches 117.6 ± 0.3 human-normalised score vs. 99.3 ± 1.0 for LSTM.

Architecture and training configuration. For Atari, the shared policy-value network uses a convolutional ResNet backbone feeding the transformer core, with previous reward and action as additional inputs. Representative hyperparameters: embedding size 256, 8 layers, 4 attention heads, key/value size 32, MLP size 512, unroll length 63, batch size 128. Parameters are updated jointly under $\mathcal{L}(\phi, \theta, \eta, \alpha)$ using Adam at 10^{-4}.