dwDPO: Training Multi-Turn LLM Agents via Divergence-Weighted Direct Preference Optimization

Anonymous ACL submission

Abstract

Large language models deployed as multi-step agents are increasingly trained via Direct Preference Optimization (DPO) on trajectory-level comparisons. Yet standard DPO weights every step uniformly, penalizing correct early actions in a losing trajectory as severely as the causal error itself. We propose **Divergence-Weighted DPO (dwDPO)**, a drop-in replacement that *detects* the first divergence step between competing trajectories using implicit rewards already computed by DPO, then *concentrates* the loss around it via an asymmetric kernel with a closed-form scale factor that preserves the Bradley-Terry preference model. dwDPO recovers standard DPO as a special case and provably achieves higher gradient concentration at the critical step than both uniform and monotonic-discount baselines. Extensive experimental results across six deep-research benchmarks demonstrate the superiority of dwDPO over state-of-the-art open-source baselines of comparable scale at no additional inference cost.

1 Introduction

Language models are increasingly deployed as autonomous agents that carry out multi-step tasks: issuing search queries, calling APIs, reading retrieved documents, and synthesizing final answers (Yao et al., 2023; Schick et al., 2024; Xi et al., 2023). Preference optimization—most notably DPO (Rafailov et al., 2023)—has become a standard tool for aligning these agents with human intent. While effective for single-turn generation, DPO faces a structural challenge in multi-step settings: preference labels are assigned at the *trajectory level*, yet the quality of the final outcome is determined by a small number of *step-level* decisions buried within a long interaction history.

This mismatch gives rise to a credit assignment problem. In practice, two competing trajectories often share a long prefix of correct actions before diverging at a single critical step. Standard DPO, by treating all steps uniformly, assigns equal blame to every action in the losing trajectory—including the correct ones. The consequences are twofold. First, **catastrophic unlearning**: useful behaviors (e.g., effective search strategies) that happen to appear in a losing trajectory are suppressed, because the gradient pushes their probability down alongside the actual error. Second, **signal dilution**: the gradient from the truly critical step is drowned out among all other steps. As trajectories grow longer, the fraction of useful gradient shrinks rapidly while the harmful portion dominates (Appendix A). Both problems worsen as trajectories grow longer, making standard DPO progressively less effective precisely in the regime where capable agents are needed most.

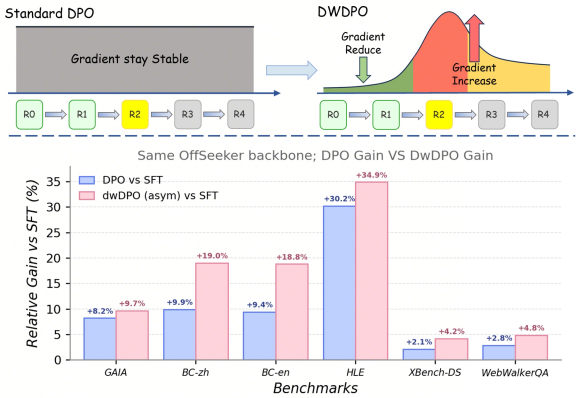

Figure 1: Comparison of standard DPO and dwDPO. dwDPO reallocates trajectory-level preference signals toward the critical divergence step and achieves stronger improvements over SFT.

Existing remedies either operate at the wrong granularity or introduce substantial overhead. Token-level DPO (Zeng et al., 2024) and Step-DPO (Lai et al., 2024) redistribute weights heuristically without modeling the causal structure of agent trajectories. CSO (Li et al., 2026) identifies critical steps via a process reward model, but re-

quires $O(K \times T)$ additional rollouts per trajectory—an impractical cost for long-horizon agent training. Monotonic discounting approaches such as DMPO (Shi et al., 2024) always favor early steps regardless of where the error occurs, which can *de-emphasize* the critical step when it falls late in the trajectory.

We propose **Divergence-Weighted DPO (dwDPO)**, a drop-in modification of the DPO loss that solves the trajectory credit assignment problem at zero additional inference cost. dwDPO operates in two stages:

- **Divergence detection.** We identify the step at which the winning and losing trajectories first make substantively different decisions, using the implicit per-step rewards already computed by DPO. This reuses the log-ratio computation that standard DPO already performs, adding negligible overhead (§4.1).
- **Distribution-based step weighting.** We center a parameterized kernel (e.g., an asymmetric Gaussian) at the detected divergence step and solve a closed-form constraint to set the global scale factor, provably preserving the Bradley-Terry preference structure (§4.2, §4.3).

The asymmetry of the kernel encodes a causal prior: errors at the divergence step propagate *forward* to corrupt subsequent actions, but should not retroactively penalize prior correct decisions. When no divergence is detected, the weights fall back to uniform and the loss reduces exactly to standard DPO, ensuring graceful degradation.

Contributions.

- We propose a zero-cost **divergence detection** procedure based on implicit rewards already computed by DPO, requiring no external reward model, rollouts, or annotations.
- We derive a family of **distribution-based step weighting** strategies with a closed-form scale factor that preserves the Bradley-Terry preference model. We prove that dwDPO recovers standard DPO as a special case and achieves strictly higher gradient concentration at the critical step than both uniform and monotonic-discount baselines.
- To validate the efficacy of dwDPO, we conduct extensive experiments on six deep-research benchmarks, and the results show that dwDPO achieves state-of-the-art performance among open-source baselines of comparable scale.

2 Related Work

DPO variants. DPO (Rafailov et al., 2023) reformulates RLHF as classification on preference pairs. Subsequent variants modify the loss or data: IPO (Azar et al., 2024), KTO (Ethayarajh et al., 2024), SimPO (Meng et al., 2024), ORPO (Hong et al., 2024), and HyPO (Yuan et al., 2026). HyPO reweights at the *pair* level (adjusting each pair's contribution); all retain uniform weighting *within* a trajectory. dwDPO is orthogonal and composable with any of them.

Fine-grained credit assignment in preference learning. Token-level DPO (Zeng et al., 2024) assigns per-token importance weights; Step-DPO (Lai et al., 2024) applies DPO per reasoning step; Chan et al. (2024) derive dense rewards from trajectory feedback; CSO (Li et al., 2026) identifies critical steps via $O(K \times T)$ rollouts plus a process reward model. These approaches either use heuristic weights that ignore causal structure, or require expensive external supervision. dwDPO detects the divergence step from implicit rewards already computed by DPO at zero additional cost.

Credit assignment in RL. GAE (Schulman et al., 2016), RUDDER (Arjona-Medina et al., 2019), and Hindsight Credit Assignment (Harutyunyan et al., 2019) all require explicit reward signals or value functions—precisely what DPO avoids. dwDPO performs credit assignment within the DPO framework using only log-probability ratios.

LLM agent training. Song et al. (2024) propose exploration-based trajectory optimization; Agent-FLAN (Chen et al., 2024) designs agent fine-tuning data; Agent Q (Putta et al., 2024) combines MCTS with DPO. Most relevant is DMPO (Shi et al., 2024), which uses a monotonically decreasing discount favoring early steps. When the critical error is late, DMPO de-emphasizes it ($\kappa<1$; Proposition 7). dwDPO adapts to the actual error location, achieving $\kappa>1$ regardless of position.

3 Preliminaries: Standard DPO for Agent Trajectories

We formalize multi-step agent trajectories as Markov decision processes (§3.1), derive the step-level implicit reward structure underlying DPO

(§3.2), and analyze why standard DPO's uniform weighting is ill-suited for these trajectories (§3.3).

3.1 Agent Trajectories as Markov Decision Processes

We model the LLM agent's environment interaction as a Markov decision process (MDP) over natural language states $\mathcal{S}$ and actions $\mathcal{A}$. Starting from an initial prompt $s_0 = x$, the agent samples an action $a_t \sim \pi_\theta(a_t \mid s_t)$ at step t. The environment returns an observation o_t, deterministically expanding the state history via concatenation:

$$s_{t+1} = [s_t;\, a_t;\, o_t]. \tag{1}$$

We denote the transition function as $P(s_{t+1} \mid s_t, a_t)$, which is equivalent to $P(o_t \mid s_t, a_t)$ under the deterministic concatenation in Eq. (1) and is governed by external factors outside the agent's control. A complete trajectory sequence $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$ factorizes cleanly into policy decisions and environment transitions:

$$\begin{aligned}\pi(\tau \mid x) &= \pi(a_0 \mid s_0) \cdot P(s_1 \mid s_0, a_0) \cdot \pi(a_1 \mid s_1) \\ &\quad \cdots P(s_T \mid s_{T-1}, a_{T-1}) \cdot \pi(a_T \mid s_T) \\ &= \pi(a_0 \mid s_0) \cdot \prod_{t=0}^{T-1} P(s_{t+1} \mid s_t, a_t)\, \pi(a_{t+1} \mid s_{t+1}).\end{aligned} \tag{2}$$

3.2 DPO and Step-Level Implicit Rewards

DPO (Rafailov et al., 2023) considers the KL-constrained reward maximization objective:

$$\max_{\pi}\ \mathbb{E}_{\tau\sim\pi(\cdot|x)}\big[R(\tau) - \beta \cdot D_{\mathrm{KL}}[\pi(\tau \mid x) \,\|\, \pi_{\mathrm{ref}}(\tau \mid x)]\big], \tag{3}$$

where $R(\tau)$ is a trajectory-level reward and $\beta > 0$ controls the KL penalty. The unique optimal solution to Eq. (3) is:

$$\pi^*(\tau \mid x) = \frac{1}{Z(x)} \pi_{\mathrm{ref}}(\tau \mid x) \exp\left(\frac{1}{\beta} R(\tau)\right), \tag{4}$$

where $Z(x) = \sum_\tau \pi_{\mathrm{ref}}(\tau \mid x) \exp\left(\frac{1}{\beta} R(\tau)\right)$ is the partition function. Solving for the reward:

$$R(\tau) = \beta \log \frac{\pi^*(\tau \mid x)}{\pi_{\mathrm{ref}}(\tau \mid x)} + \beta \log Z(x). \tag{5}$$

Under the Bradley-Terry model (Bradley and Terry, 1952), the probability that trajectory τ^w is preferred over τ^l given prompt x is:

$$P(\tau^w \succ \tau^l \mid x) = \sigma\big(R(\tau^w) - R(\tau^l)\big). \tag{6}$$

The key insight of DPO is to replace π^* with the learned policy π_θ in Eq. (5), yielding an implicit reward parameterized by θ. Substituting into Eq. (6), the partition function $Z(x)$ cancels (since both trajectories share the same prompt x), giving the standard DPO loss over a preference dataset $\mathcal{D}$ of triples (x, τ^w, τ^l):

$$\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E}_{(x,\tau^w,\tau^l)\sim\mathcal{D}}\left[\log \sigma\Big(r(\tau^w|x) - r(\tau^l|x)\Big)\right], \tag{7}$$

where $r(\cdot|x) = \beta \log \frac{\pi(\cdot|x)}{\pi_{\mathrm{ref}}(\cdot|x)}$ and β is the coefficient that controls π's deviation from π_{ref}.

Since both π and π_{ref} interact with the same environment, they share the same transition function $P(s_{t+1} \mid s_t, a_t)$, which depends only on the external system and not on the policy.

Lemma 1 (Trajectory KL Factorization)**.** *If policies π and π_{ref} share the same transition function $P(s_{t+1} \mid s_t, a_t)$, then:*

$$\log \frac{\pi(\tau \mid x)}{\pi_{\mathrm{ref}}(\tau \mid x)} = \sum_{t=0}^{T} \log \frac{\pi(a_t \mid s_t)}{\pi_{\mathrm{ref}}(a_t \mid s_t)}. \tag{8}$$

Proof. Expanding both trajectory distributions via Eq. (2), the transition factors $P(s_{t+1} \mid s_t, a_t)$ appear identically in numerator and denominator and cancel, leaving only the action probability ratios. □

Substituting the optimal policy (Eq. 5) into this factorization, each summand acquires a natural interpretation as a step-level reward signal, which we formalize below.

Definition 2 (Step-Level Implicit Reward)**.** *The* step-level implicit reward *(Cui et al., 2025) at step t is defined as:*

$$\hat{r}_t(\tau) := \beta \log \frac{\pi^*(a_t \mid s_t)}{\pi_{\mathrm{ref}}(a_t \mid s_t)}, \tag{9}$$

which measures the contribution of step t to the total trajectory reward under the optimal policy.

Combining Eq. (5) with Lemma 1, the trajectory reward decomposes into a sum of step-level contributions:

$$R(\tau) = \sum_{t=0}^{T} \underbrace{\beta \log \frac{\pi^*(a_t \mid s_t)}{\pi_{\mathrm{ref}}(a_t \mid s_t)}}_{\hat{r}_t(\tau)} + \underbrace{\beta \log Z(x)}_{C(x)}. \tag{10}$$

The remaining term $C(x) = \beta \log Z(x)$ is a prompt-dependent constant that does not depend on the trajectory content.

3.3 The Uniform Credit Problem

Applying standard DPO introduces a severe **uniform credit assignment problem** in multi-step agent settings, where correct early actions are actively penalized and the useful learning signal is drastically diluted. To resolve this structural deficiency, we introduce the Divergence-Weighted DPO (dwDPO) framework.

4 dwDPO: Divergence-Weighted DPO

Without requiring external reward models or additional rollouts, dwDPO first employs a **divergence step detection algorithm** that leverages inherent implicit rewards to identify where the losing trajectory first meaningfully deviates (§4.1). It then applies a **step-level weighted preference loss** using standard probability distributions centered at this critical divergence step (§4.2, §4.3). This redistributes the learning signal to concentrate optimization directly on the causal error while preserving full trajectory-level supervision.

4.1 Divergence Step Detection

The divergence step t^* identifies where two trajectories first make substantively different decisions. We consider two complementary approaches.

Approach 1: Implicit reward difference. We detect divergence using the step-level implicit rewards from Definition 2, which are already computed for the DPO loss.

Definition 3 (Divergence Step)**.** *Given a preference pair (τ^w, τ^l) and a threshold $\gamma > 0$:*

$$t^* = \Big(\min\Big\{t : |\hat{r}_t(\tau^w) - \hat{r}_t(\tau^l)| > \gamma\Big\}\Big) - 1. \tag{11}$$

If no step exceeds the threshold, we set $w_t = 1$ for all t in both trajectories, and the loss reduces to standard DPO for that pair.

The intuition is Markovian: in an MDP, s_{t+1} is determined by (s_t, a_t). If the implicit rewards first diverge at step t, it is the preceding action a_{t-1} that caused the state to diverge.

Approach 2: External annotation. As an alternative that avoids dependence on the policy being trained, an external model can annotate the divergence step. Given both trajectories, the annotator identifies which step in τ^l is the primary cause of failure, producing t^* labels entirely independent of π_θ (Li et al., 2026).

4.2 Step-Level Weighted DPO

We now introduce step-level weights that concentrate the learning signal on and around this critical point. By introducing weights $\{w_t\}$, we enable the model to pinpoint what went wrong in the losing trajectory, extracting a stronger training signal from each preference pair.

Weighted reward and the BT constraint. Starting from the reward decomposition in Eq. (10), we introduce step-level weights $w_t(\tau)$ on the implicit rewards and a scalar coefficient $c(\tau)$ on the prompt-dependent constant. We require that this reweighting preserves the original trajectory reward $R(\tau)$:

$$R(\tau) = \sum_{t=0}^{|\tau|-1} w_t(\tau)\hat{r}_t(\tau) + c(\tau)\beta \log Z(x), \tag{12}$$

where $w_t(\tau) \geq 0$ are step-level weights and $c(\tau)$ is the coefficient on the prompt-dependent constant $\beta \log Z(x)$.

Expanding the difference:

$$\begin{aligned} &R(\tau^w) - R(\tau^l) \\ &= \sum_{t=0}^{|\tau^w|-1} w_t(\tau^w)\hat{r}_t(\tau^w) - \sum_{t=0}^{|\tau^l|-1} w_t(\tau^l)\hat{r}_t(\tau^l) \\ &\quad + \big(c(\tau^w) - c(\tau^l)\big)\beta \log Z(x). \end{aligned} \tag{13}$$

The partition function $Z(x)$ is intractable. For the constant term to cancel in the preference difference, we require:

$$c(\tau^w) = c(\tau^l). \tag{14}$$

Recall that under the Bradley-Terry model, the preference probability is:

$$P(\tau^w \succ \tau^l \mid x) = \sigma\big(R(\tau^w) - R(\tau^l)\big). \tag{15}$$

When Eq. (14) holds, the weighted preference model reduces to:

$$\begin{aligned} &P(\tau^w \succ \tau^l \mid x) \\ &= \sigma\Big(\textstyle\sum_{t=0}^{|\tau^w|-1} w_t(\tau^w)\hat{r}_t(\tau^w) - \sum_{t=0}^{|\tau^l|-1} w_t(\tau^l)\hat{r}_t(\tau^l)\Big), \end{aligned} \tag{16}$$

which involves only the computable implicit rewards.

The reward-matching constraint. Combining Eq. (14) with the requirement that $\widetilde{R}(\tau) = R(\tau)$

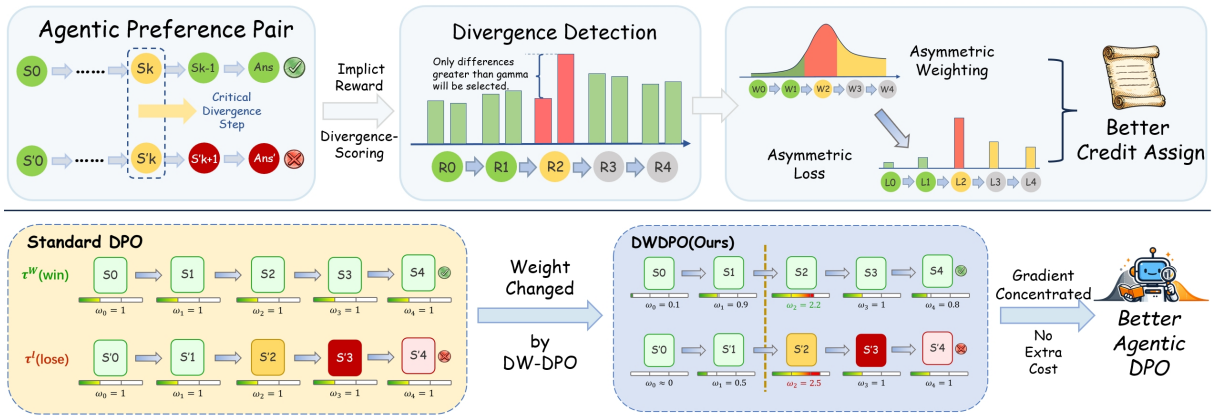

Figure 2: Overview of dwDPO. **Top:** Per-step implicit rewards identify the divergence step t^* (reward gap $> \gamma$); an asymmetric Gaussian kernel ($\sigma_L < \sigma_R$) centered at t^* concentrates the loss on the causal error and its downstream consequences. **Bottom:** Standard DPO applies uniform weights ($\omega_t = 1$); dwDPO redistributes them around t^*, improving credit assignment at no extra cost.

(the weighted reward must equal the original reward), we obtain a single scalar constraint linking the weights of both trajectories in a pair:

$$\sum_{t=0}^{|\tau^w|-1} w_t(\tau^w)\hat{r}_t(\tau^w) - \sum_{t=0}^{|\tau^l|-1} w_t(\tau^l)\hat{r}_t(\tau^l) = \sum_{t=0}^{|\tau^w|-1} \hat{r}_t(\tau^w) - \sum_{t=0}^{|\tau^l|-1} \hat{r}_t(\tau^l). \tag{17}$$

The constraint leaves substantial freedom in the weight design: any choice of weights satisfying Eq. (17) yields a valid preference model.

The dwDPO loss function. Combining the weighted BT model (Eq. 16) with the policy parameterization:

$$\mathcal{L}_{\text{dwDPO}} = -\mathbb{E}_{(x,\tau^w,\tau^l)\sim\mathcal{D}}\Big[\log\sigma\big(\beta\textstyle\sum_{t=0}^{|\tau^w|-1} w_t(\tau^w)\,\hat{r}_t(\tau^w) - \beta\textstyle\sum_{t=0}^{|\tau^l|-1} w_t(\tau^l)\,\hat{r}_t(\tau^l)\big)\Big]. \tag{18}$$

Note that the weights $\{w_t\}$ are not parameters to be optimized: they encode a credit assignment decision determined by the distribution family and the divergence step.

4.3 Distribution-Based Weight Assignment

Given the divergence step t^*, we parameterize the weight *shape* using a probability distribution ϕ (Appendix D describes 4 kinds of distributions we use) centered at t^*, and solve for a global scale factor λ to satisfy the reward-matching constraint (Eq. 17).

Normalized coordinate. For each trajectory τ of length $|\tau|$, we define a normalized coordinate centered at t^*:

$$u_t = \frac{t - t^*}{|\tau| - 1}, \qquad u_t \in [-1,\, 1] \tag{19}$$

so that t^* maps to $u_t = 0$. The actual range of u_t depends on the position of t^* within the trajectory.

Weight parameterization. The weights for both trajectories share a single scale factor λ:

$$w_t(\tau^w) = \lambda \cdot \phi(u_t^w), \qquad w_t(\tau^l) = \lambda \cdot \phi(u_t^l) \tag{20}$$

where ϕ is evaluated at the normalized coordinates of each trajectory. Since both trajectories use the *same* λ and the *same* distribution family, the constraint $c(\tau^w) = c(\tau^l)$ is automatically satisfied.

Closed-form solution for λ. Substituting Eq. (20) into the reward-matching constraint (Eq. 17) and solving:

$$\lambda = \frac{\sum_{t=0}^{|\tau^w|-1} \hat{r}_t(\tau^w) - \sum_{t=0}^{|\tau^l|-1} \hat{r}_t(\tau^l)}{\sum_{t=0}^{|\tau^w|-1} \phi(u_t^w)\cdot\hat{r}_t(\tau^w) - \sum_{t=0}^{|\tau^l|-1} \phi(u_t^l)\cdot\hat{r}_t(\tau^l)} \tag{21}$$

All quantities in both numerator and denominator are computable: the implicit rewards $\hat{r}_t(\tau) = \beta\log(\pi_\theta(a_t \mid s_t)/\pi_{\text{ref}}(a_t \mid s_t))$ come from the current and reference policies, and $\phi(u_t)$ comes from the chosen distribution evaluated at the normalized coordinates.

Table 1: Main results on deep research benchmarks. dwDPO(asym) uses an asymmetric Gaussian kernel (σ_L=0.3, σ_R=0.7); dwDPO(sym) uses a symmetric Gaussian (σ=0.5). BC-zh and BC-en denote BrowseComp-ZH and BrowseComp-EN. All scores are LLM-judge accuracy (%) averaged over three runs. **Bold**: best score among DPO-based agents; underline: second best. Ties broken by the full (unrounded) score. †WebExplorer-8B uses RL training and is shown for reference only.

Method	GAIA	BC-zh	BC-en	HLE	XBench-DS	WebWalkerQA
Large Foundation Models (>100B, no fine-tuning)						
OpenAI-o3	70.5	58.1	20.2	49.7	66.7	71.7
Claude-4-Sonnet	68.3	29.1	12.2	20.3	64.6	61.7
DeepSeek-V3.1	63.1	49.2	30.0	29.8	71.0	61.2
DeepSeek-V3.2	63.5	57.7	47.9	40.1	71.0	63.0
Kimi-K2	28.8	27.2	18.1	14.1	50.0	—
Small Deep Research Agents (<10B)						
MiroThinker-8B-DPO-v0.1	46.6	13.6	8.7	—	45.7	—
WebSailor-7B	33.0	14.2	6.7	—	34.3	—
DeepDive-9B (SFT)	—	15.7	5.6	—	35.0	—
DeepDive-9B (RL)	—	15.1	6.3	—	38.0	—
WebExplorer-8B (RL)†	47.6	31.8	14.6	12.4	53.0	58.0
DPO-Based Agents (OffSeeker-8B backbone)						
OffSeeker (SFT)	47.6	24.2	11.7	10.6	48.0	60.0
OffSeeker + DPO	51.5	26.6	12.8	13.8	49.0	61.7
OffSeeker + dwDPO(sym)	**52.2**	**29.6**	11.7	13.9	49.0	62.8
OffSeeker + dwDPO(asym)	**52.2**	28.8	**13.9**	**14.3**	**50.0**	**62.9**

5 Experiments

5.1 Experimental Setup

Tasks and evaluation. We evaluate dwDPO on six multi-step deep research benchmarks requiring long-horizon tool use and compound reasoning—GAIA (Mialon et al., 2023), BrowseComp-zh/en (Wei et al., 2025), HLE (Phan et al., 2025), XBench-DeepSearch (Liu et al., 2025), and WebWalkerQA (Luo et al., 2025). All benchmarks are evaluated using LLM-as-judge accuracy (Zheng et al., 2023) using GPT-4o as judger. Full details are provided in the Appendix F.

Implementation Details. We conduct evaluations on two model groups, large foundation models >100B parameters and small deep research agents <10B. We use the OffSeeker-8B agent (Anonymous, 2026) as the base SFT reference policy, and their DPO pairs as our train dataset. We compare dwDPO against Standard DPO (Rafailov et al., 2023), SP-DPO, Step-DPO (Lai et al., 2024), Turn-DPO ablated baselines from the same checkpoint with identical β, with further detailed settings provided in the Appendix F.

5.2 Main Results

Table 1 end-to-end task performance. All gains reported below are averages over three runs.

State-of-the-Art Among DPO-Based Small Agents. Our dwDPO variants establish a new state of the art among DPO-based sub-10B agents across all six benchmarks. OffSeeker + dwDPO(asym) leads on four benchmarks — BC-en (13.9%), HLE (14.3%), XBench-DS (50.0%), and WebWalkerQA (62.9%) — and ties for first on GAIA (52.2%), while dwDPO(sym) tops BC-zh (29.6%). Compared to the SFT backbone, dwDPO(asym) yields gains of up to +4.6% on GAIA and BC-zh, and consistently outperforms standard DPO across every benchmark. Remarkably, despite relying solely on offline preference optimization, dwDPO(asym) surpasses the RL-trained WebExplorer-8B on three of five shared benchmarks, underscoring the effectiveness and practicality of our approach.

Table 2: Ablation results for step weighting strategies. All methods use the same OffSeeker-8B backbone and β. Scores are LLM-judge accuracy (%) averaged over three runs. SP-DPO applies the loss only at t^*. Step-DPO and Turn-DPO apply uniform weights within a step or turn boundary, respectively. **Bold**: best per column; underline: second best. Ties broken by unrounded score.

Method	GAIA	BC-zh	BC-en	HLE	XBench-DS	WebWalkerQA	AVG
DPO	51.5	26.6	12.8	13.8	49.0	61.7	35.9
SP-DPO	50.3	28.7	12.5	13.8	46.0	62.6	35.7
Step-DPO (uniform)	49.7	29.4	12.6	14.4	46.0	**64.7**	36.1
Turn-DPO (uniform)	50.5	28.4	11.6	**16.8**	47.8	61.9	36.2
dwDPO(sym) *(ours)*	**52.2**	**29.6**	11.7	13.9	49.0	62.8	36.5
dwDPO(asym) *(ours)*	**52.2**	28.8	13.9	14.3	**50.0**	62.9	**37.0**

Closing the Gap to Frontier Models. Despite being over 10× smaller, our dwDPO agents close a meaningful portion of the gap to frontier systems on several benchmarks. On WebWalkerQA, dwDPO(asym) (62.9%) actually surpasses Claude-4-Sonnet (61.7%), and on XBench-DS it reaches parity with Kimi-K2 while approaching DeepSeek-V3.x. The largest remaining gaps appear on knowledge-intensive tasks such as HLE and BrowseComp, pointing to world knowledge coverage and multi-hop reasoning depth as the primary scaling bottlenecks.

Symmetric vs. Asymmetric Weighting. The asymmetric kernel (σ_L=0.3, σ_R=0.7) generally outperforms the symmetric one (σ=0.5), winning on four of six benchmarks. This suggests that its design—penalizing ambiguous near-tied pairs more heavily while tolerating larger winning margins—better suits the noisy preference signals inherent in deep research tasks.

5.3 Ablation Studies

Table 2 compares dwDPO against four competing step-weighting baselines on the same backbone. While uniform step- and turn-level weighting (Step-DPO, Turn-DPO) do offer modest improvements over standard DPO on individual benchmarks — Step-DPO peaks on WebWalkerQA (64.7%) and Turn-DPO on HLE (16.8%) — these gains are inconsistent across benchmarks and neither method achieves a clear overall advantage, with average scores of 36.1% and 36.2% respectively. SP-DPO, which concentrates the loss exclusively at t^*, even regresses slightly below DPO on average (35.7%), suggesting that discarding all non-critical steps is too aggressive. In contrast, dwDPO(asym) achieves the best average score of 37.0%, outperforming all baselines on GAIA, XBench-DS, and BC-en simultaneously, demonstrating that soft distance-based weighting centered on t^* provides a more principled and consistent signal than hard uniform or single-step alternatives.

Per-benchmark results reveal task-specific sensitivity. On web-navigation tasks (WebWalkerQA, XBench-DS), where errors compound across steps, dwDPO(asym) matches or exceeds all baselines, confirming that proximity-to-divergence is especially informative for sequential decision-making. On knowledge-intensive HLE, Turn-DPO's peak (16.8%) suggests explicit turn boundaries capture complementary structure not fully encoded by distance weighting. These patterns indicate that while dwDPO(asym) is the most robust default, task-adaptive bandwidth selection remains a promising direction.

Comparing the two dwDPO variants, dwDPO (asym) leads on four of six benchmarks and on average (+0.5%), while dwDPO(sym) already surpasses all non-dwDPO baselines (36.5%), confirming that Gaussian weighting itself is the primary driver of improvement. The additional gain from asymmetry (σ_L<σ_R) reflects the benefit of penalizing ambiguous near-tied pairs more tightly while tolerating larger winning margins, a finding further corroborated by the hyperparameter ablation in Figure 3.

Hyperparameter Ablation over (σ_L, σ_R). Table 3 ablates the kernel bandwidths (σ_L, σ_R) over a 4×4 grid. Two observations stand out. First, configurations with $\sigma_L < \sigma_R$ (upper-right triangle) consistently outperform their reverse-asymmetric counterparts, confirming that penalizing near-tied ambiguous pairs more tightly on the left while remaining tolerant of larger winning margins on the

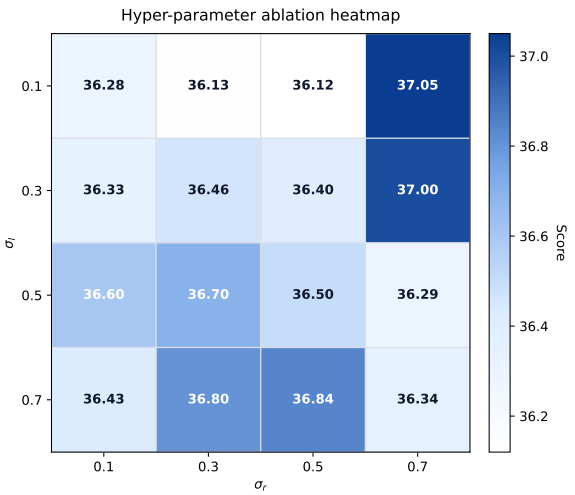

Figure 3: **Hyperparameter Ablation over** (σ_L, σ_R)**.** Average benchmark score (%) across a grid search of left and right bandwidth parameters. **Bold**: sym dwDPO (σ_L=σ_R=0.5, score 36.50) and asym dwDPO (σ_L=0.3, σ_R=0.7, score 37.00, our chosen configuration). Cells with $\sigma_L > \sigma_R$ (lower-left triangle) correspond to reverse-asymmetric kernels.

right is the correct inductive bias. Second, the best symmetric configuration (σ_L=σ_R=0.5, 36.50%) is itself competitive, yet the optimal asymmetric setting (σ_L=0.3, , σ_R=0.7, 37.00%) yields a clear +0.5% gain, justifying the added degree of freedom introduced by asymmetric kernel design.

Divergence Step Detection Validation. Before evaluating end-to-end policy improvements, we validate dwDPO's zero-shot divergence detection against human annotation and empirical rollout verification: our method achieves 84.5% agreement with human judges on critical failure steps (comparable to human inter-annotator agreement). Figure 4 further compares the distribution of selected t^* across three strategies. dwDPO concentrates sharply at t=1–2 (40.4% and 51.5%), confirming that reward-differentiating divergences emerge early in most trajectories. LLM-choose turn follows a similar pattern but assigns notably more mass to t=0 (19.3%), suggesting the LLM judge more aggressively identifies first-step failures. CSO (Li et al., 2026) best turn, despite its small sample (N=81), exhibits a flatter and right-shifted distribution with non-trivial probability beyond t=3, indicating that exhaustive oracle search recovers late-turn critical steps that greedy heuristics systematically miss.

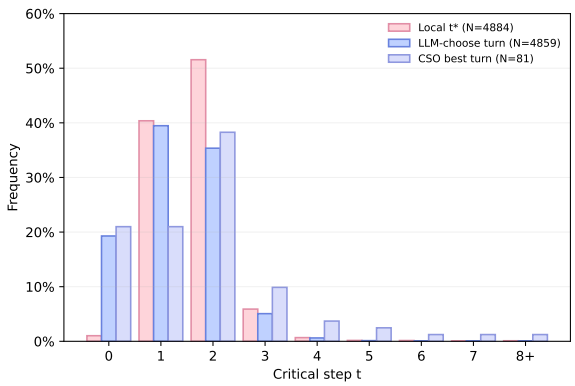

Figure 4: **Distribution of Divergence Step** t^* **Across Selection Strategies.** Normalized frequency histograms for dwDPO (N=4884), LLM-choose turn (N=4860), and CSO (Li et al., 2026) best turn (N=81). All three strategies concentrate mass at early turns ($t \leq 2$), though CSO (Li et al., 2026) best turn exhibits a flatter, right-shifted distribution, revealing late-turn critical steps that greedy methods tend to miss.

6 Conclusion

We presented dwDPO, a principled extension of Direct Preference Optimization for multi-step agent trajectories that addresses the credit assignment problem through two tightly coupled mechanisms: automatic divergence step detection via implicit rewards already available within the DPO objective, and an asymmetric Gaussian kernel that concentrates the learning signal at the causal error while shielding correct early steps from undue penalization.

The two mechanisms are theoretically grounded and mutually reinforcing. dwDPO preserves the Bradley-Terry complementarity axiom (Proposition 4), recovers standard DPO under uniform weights (Proposition 5), and achieves strictly higher gradient concentration at the critical step than both DPO and monotonic discounting baselines (Proposition 7). The asymmetry $\sigma_R > \sigma_L$ has a clear causal interpretation: because errors propagate forward in a Markov decision process, the kernel is deliberately right-permissive to reflect this directionality.

Empirically, dwDPO outperforms standard DPO across all six deep research benchmarks (mean +1.1,pp, up to +2.2,pp on BC-zh). Ablations confirm both components are necessary: SP-DPO underperforms even standard DPO, and symmetric kernels consistently trail their asymmetric counterparts, validating the full dwDPO design.

Limitations and future work. dwDPO detects only the first divergence step; extending to multi-point detection via a mixture of kernels is a natural direction. The weight hyperparameters are currently selected via grid search on a validation split; making the weight shape jointly learnable with the policy would improve flexibility and reduce the need for offline tuning. Online recomputation of the divergence step as the policy evolves may yield further gains. A direct empirical comparison with CSO (Li et al., 2026)—the most closely related step-level method—remains an important open item; CSO's $O(K \times T)$ rollout requirement makes it impractical to compare under identical compute budgets, but a controlled comparison at matched cost is a priority for future work. Finally, all experiments use a single backbone (OffSeeker-8B); evaluating generalization across architectures and SFT checkpoints is an important next step.

References

Anonymous. 2026. OffSeeker-8B: A deep research agent. Anonymous submission.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. RUDDER: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2024. A general theoretical paradigm to understand learning from human feedback. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Alex J. Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. 2024. Dense reward for free in reinforcement learning from human feedback. *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Zehui Chen, Kuikun Du, Wenqiang Chen, Zefan Liu, Yiwen Liu, Cong Zhao, Zhihao Wu, Yinghui Gu, Dongyan Wang, Shuang Li, and 1 others. 2024. AgentFLAN: Designing data and methods of effective agent tuning for large language models. *arXiv preprint arXiv:2403.12881*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, and 4 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P. van Hasselt, Greg Wayne, Satinder Singh, Doina Precup, and Rémi Munos. 2019. Hindsight credit assignment. *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-DPO: Step-wise preference optimization for long-chain reasoning of LLMs. *arXiv preprint arXiv:2406.18629*.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. 2026. Verified critical step optimization for LLM agents. *arXiv preprint arXiv:2602.03412*.

Zhen Liu, Zhiyuan Zhang, Liang Chen, and 1 others. 2025. XBench: Evaluating and benchmarking large language models on extremely hard tasks. *arXiv preprint arXiv:2503.10971*.

Jialong Luo, Wanjun Hua, Runnan Zhang, Zhuyu Yang, Zirui Chen, Wei Bi, Yankai Lin, and Zhifang Liu. 2025. WebWalker: Benchmarking LLMs in web traversal. *arXiv preprint arXiv:2501.07572*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems (NeurIPS)*, 37.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*.

Long Phan, Alice Gatti, Ziwen Han, Sean Li, Nathaniel Hu, Spencer Zhang, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent Q: Advanced reasoning and learning for autonomous AI agents. *arXiv preprint arXiv:2408.07199*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Francisco J. Rubio and Mark F. J. Steel. 2014. Inference in two-piece location-scale models with Jeffreys priors. *Bayesian Analysis*, 9(1):1–22.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. 2024. Direct multi-turn preference optimization for language agents. *arXiv preprint arXiv:2406.14868*.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for LLM agents. *arXiv preprint arXiv:2403.02502*.

Kenneth F. Wallis. 2014. The two-piece normal, binormal, or double Gaussian distribution: Its origin and rediscoveries. *Statistical Science*, 29(1):106–112.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations (ICLR)*.

Suqin Yuan, Xingrui Yu, Jiyang Zheng, Lei Feng, Dadong Wang, Ivor Tsang, and Tongliang Liu. 2026. Mitigating mismatch within reference-based preference optimization. In *International Conference on Learning Representations (ICLR)*.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

A Three Structural Pathologies of Uniform DPO on Agent Trajectories

Standard DPO assigns uniform weight $w_t = 1$ to every step, which creates three concrete failure modes that scale with trajectory length.

Pathology 1: Catastrophic unlearning of correct behaviors. Consider a preference pair where τ^w and τ^l share the same correct actions at steps $t \in \mathcal{C} = \{0, \ldots, t^* - 1\}$ (i.e., $a_t^w = a_t^l$). The DPO gradient simultaneously *increases* $\log \pi_\theta(a_t^w \mid s_t^w)$ for the winner and *decreases* $\log \pi_\theta(a_t^l \mid s_t^l)$ for the loser. When both trajectories share the same correct action a_t at step t, these gradients have opposite sign but similar magnitude, causing partial cancellation and accumulating noise rather than making progress on the correct behavior.

Pathology 2: Diluted learning signal. The DPO gradient decomposes into two components with opposing roles:

$$\nabla_\theta \mathcal{L}_{\text{DPO}} \propto \underbrace{\sigma' \sum_{t \in \mathcal{C}} \nabla_\theta \log \frac{\pi_\theta(a_t^l|s_t^l)}{\pi_{\text{ref}}(a_t^l|s_t^l)}}_{\text{harmful: penalizes correct steps}} + \underbrace{\sigma' \sum_{t \in \mathcal{E}} \nabla_\theta \log \frac{\pi_\theta(a_t^l|s_t^l)}{\pi_{\text{ref}}(a_t^l|s_t^l)}}_{\text{useful: penalizes erroneous steps}}, \quad (22)$$

where $\mathcal{E} = \{t^*, \ldots, |\tau^l| - 1\}$ are the erroneous steps and $\sigma' = 1 - \sigma(\Delta)$. The useful signal constitutes only a fraction $|\mathcal{E}|/|\tau^l|$ of the total gradient magnitude. For a trajectory of length 8 with a single error step, the useful signal is only 12.5%; the remaining 87.5% actively harms training.

Pathology 3: Severity scales with trajectory length. The fraction of harmful gradient is $|\mathcal{C}|/|\tau^l|$. As agent trajectories grow longer—driven by increasingly complex tasks requiring more search and reasoning steps—the harmful fraction increases, making uniform DPO progressively less effective. For a 4-step trajectory with 1 error, 75% of the gradient is harmful; for a 16-step trajectory with 1 error, 93.75% is harmful. This implies that uniform DPO becomes strictly harder to use as tasks become more complex, which is precisely the regime where capable agents are needed most.

B dwDPO Training Step

Algorithm 1 summarizes a single dwDPO training step, tying together the components described in §4. Given a preference pair (τ^w, τ^l), we first compute per-step implicit rewards $\hat{r}_t$ from the policy and reference log-ratios. We then locate the divergence step t^* as the last step before the implicit rewards of the two trajectories separate by more than γ. When no such divergence exists, the procedure gracefully falls back to standard DPO with uniform weights $w_t = 1$. Otherwise, we assign step weights $w_t = \lambda \cdot \phi(u_t)$ that concentrate learning signal around t^*, where λ is the closed-form scale factor that preserves the total gradient magnitude. The weighted preference loss $\mathcal{L}_{\text{dwDPO}}$ is then minimized via a single gradient step.

Algorithm 1 dwDPO Training Step

Require: Preference pair (τ^w, τ^l) for prompt x, threshold γ, reference policy π_{ref}, learning rate η

1. Implicit Reward Computation:
for $t = 0$ **to** $\max(|\tau^w|, |\tau^l|) - 1$ **do**
  Compute $\hat{r}_t(\tau) = \beta \log \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)}$ for both trajectories
end for
2. Divergence Step Detection (§4.1):
$t^* \leftarrow (\min\{t : |\hat{r}_t(\tau^w) - \hat{r}_t(\tau^l)| > \gamma\}) - 1$
if no divergence detected **then**
  $w_t \leftarrow 1$ for all t {Fallback to standard DPO}
else
  3. Weight Assignment (§4.2 & §4.3):
  Compute normalized coordinates u_t centered at t^* (Eq. 19)
  Evaluate distribution $\phi(u_t)$ for both trajectories
  Compute closed-form scale factor λ (Eq. 21)
  Set $w_t = \lambda \cdot \phi(u_t)$ for all steps in τ^w, τ^l
end if
4. Policy Update:
Compute weighted preference loss $\mathcal{L}_{\text{dwDPO}}$ using w_t and $\hat{r}_t$ (Eq. 18)
Update policy parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{dwDPO}}$

C Why implicit reward difference works

For steps in the shared prefix, both trajectories visit similar states and take similar actions, so their implicit rewards are close—shared steps are naturally filtered without any alignment heuristic or external model. Divergence is detected only when the actions meaningfully differ. Expanding the condition $|\hat{r}_t(\tau^w) - \hat{r}_t(\tau^l)| > \gamma$:

$$\left|\beta \log \frac{\pi^*(a_t^w \mid s_t^w)}{\pi_{\text{ref}}(a_t^w \mid s_t^w)} - \beta \log \frac{\pi^*(a_t^l \mid s_t^l)}{\pi_{\text{ref}}(a_t^l \mid s_t^l)}\right| > \gamma$$

$$\iff \quad \left|\log \frac{\rho_t^w}{\rho_t^l}\right| > \tfrac{\gamma}{\beta}$$

$$\iff \quad \frac{\rho_t^w}{\rho_t^l} > e^{\gamma/\beta} \text{ or } \frac{\rho_t^w}{\rho_t^l} < e^{-\gamma/\beta}, \tag{23}$$

where $\rho_t := \pi^*(a_t \mid s_t) \,/\, \pi_{\text{ref}}(a_t \mid s_t)$ is the *adjustment factor* at step t: how much the optimal policy has scaled the probability of an action relative to the reference policy. By dividing out the base probability, ρ isolates the learning signal. The ratio ρ_t^w/ρ_t^l measures whether the optimal policy has adjusted the two trajectories' actions by the same factor or by very different factors. When both adjustments are similar ($\rho_t^w/\rho_t^l \approx 1$), the step is non-critical. When one action is boosted while the other is suppressed ($\rho_t^w/\rho_t^l \gg 1$ or $\ll 1$), the policy has strongly differentiated, marking a critical step. Table 3 illustrates the contrast with arithmetic examples.

Table 3: Divergence criterion examples ($\beta = 0.5$, $\gamma = 0.15$; fires when $\rho^w/\rho^l > 1.35$ or < 0.74). $\rho_t = \pi^*(a_t \mid s_t)/\pi_{\text{ref}}(a_t \mid s_t)$ is the adjustment factor.

	$\pi^*/\pi_{\text{ref}}(a^w)$	$\pi^*/\pi_{\text{ref}}(a^l)$	ρ^w	ρ^l	ρ^w/ρ^l
Critical	**0.70** / 0.40	0.20 / 0.22	1.75	0.91	**1.92**
Non-crit.	0.45 / 0.40	0.20 / 0.22	1.13	0.91	1.24
Critical	0.45 / 0.50	**0.40** / 0.22	0.90	1.82	**0.49**
Non-crit.	0.45 / 0.50	0.25 / 0.22	0.90	1.14	0.79

D Candidate distribution families

We consider four distributions varying in tail behavior and symmetry, each evaluated as a proper probability density so that $\phi(u_t)$ is well-defined. Since λ absorbs any global scale, only the *shape* of ϕ matters—the normalization constant of the density cancels when computing λ.

Strategy 1: Symmetric Gaussian. A Gaussian density with bandwidth σ:

$$\phi(u) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \tag{24}$$

This provides smooth concentration around t^* but treats past and future steps identically.

Strategy 2: Split normal (asymmetric Gaussian). The split normal distribution (Wallis, 2014) joins two half-Gaussians at the mode with different scale parameters:

$$\phi(u) = \frac{2}{\sqrt{2\pi}(\sigma_L+\sigma_R)} \begin{cases} \exp\!\left(-\frac{u^2}{2\sigma_L^2}\right) & \text{if } u \le 0 \\ \exp\!\left(-\frac{u^2}{2\sigma_R^2}\right) & \text{if } u > 0 \end{cases} \tag{25}$$

The separate bandwidths σ_L and σ_R allow different rates of decay for steps before and after t^*. Whether the optimal configuration favors $\sigma_R > \sigma_L$ (forward propagation of the critical decision) or $\sigma_L > \sigma_R$ (backward importance of context-building steps) is an empirical question that may depend on the task domain. When $\sigma_L = \sigma_R$, this reduces to the symmetric Gaussian (Strategy 1).

Strategy 3: Student-t (heavy-tailed). The Student-t distribution with scale σ and ν degrees of freedom:

$$\phi(u) = \frac{\Gamma\!\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\sigma\,\Gamma\!\left(\frac{\nu}{2}\right)} \left(1 + \frac{u^2}{\nu\sigma^2}\right)^{-(\nu+1)/2} \tag{26}$$

Compared to the Gaussian, the Student-t has polynomial tail decay (rather than exponential), so steps far from t^* retain more weight. This is appropriate when the critical decision at t^* has long-range effects. As $\nu \to \infty$, the Student-t converges to the Gaussian; at $\nu = 1$ it becomes the Cauchy distribution with extremely heavy tails.

Strategy 4: Two-piece Student-t (asymmetric heavy-tailed). The two-piece Student-t (Rubio and Steel, 2014) combines asymmetry with heavy tails:

$$\phi(u) = \frac{2\,\Gamma\!\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,(\sigma_L+\sigma_R)\,\Gamma\!\left(\frac{\nu}{2}\right)} \cdot \begin{cases} \left(1 + \frac{u^2}{\nu\,\sigma_L^2}\right)^{-\frac{\nu+1}{2}} & u \le 0 \\ \left(1 + \frac{u^2}{\nu\,\sigma_R^2}\right)^{-\frac{\nu+1}{2}} & u > 0 \end{cases} \tag{27}$$

When $\sigma_L = \sigma_R$, this reduces to the symmetric Student-t (Strategy 3).

The four strategies form a two-dimensional family, varying independently in tail behavior and symmetry:

$$\begin{array}{lcc} & \text{Symmetric} & \text{Asymmetric} \\ \text{Light-tailed:} & \underbrace{\text{Gaussian}}_{\sigma} \;\to & \underbrace{\text{Split normal}}_{\sigma_L,\,\sigma_R} \\ \text{Heavy-tailed:} & \underbrace{\text{Student-}t}_{\sigma,\,\nu} \;\to & \underbrace{\text{Two-piece } t}_{\sigma_L,\,\sigma_R,\,\nu} \end{array} \tag{28}$$

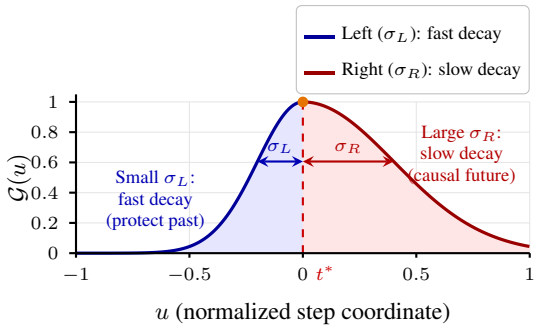

Figure 5: **The Asymmetric Gaussian Kernel $\mathcal{G}(u)$.** A small σ_L rapidly suppresses weights for steps before t^* (correct early decisions), while a larger σ_R maintains weights for steps after t^* (downstream causal effects).

Table 4: Hyperparameter search ranges for each distribution family.

Param.	Range	Families
σ	$\{0.1, 0.2, 0.3, 0.5, 0.7, 1.0\}$	Gauss., Stu.-t
σ_L, σ_R	$\{0.1, 0.2, 0.3, 0.5, 0.7, 1.0\}^2$	Split-N., 2P-t
ν	$\{1, 2, 3, 5, 10, 30\}$	Stu.-t, 2P-t

All four reduce to the uniform case (standard DPO) when $\phi(u_t)$ becomes constant across steps: for the symmetric families this occurs as σ grows large, and for the asymmetric families as $\sigma_L = \sigma_R$ grows large. The ablation study (§5.3) evaluates which axis—tail behavior or asymmetry—provides the greater empirical gain.

Hyperparameter selection. Each distribution family introduces hyperparameters (σ for the symmetric families; σ_L, σ_R for the asymmetric families; ν for the heavy-tailed families) that control the shape of the weight profile. We select these via grid search on a held-out validation set; sensitivity analysis and the effect of each axis are reported in the ablation study (§5.3).

E Theoretical Analysis

We establish two sets of formal guarantees for dwDPO: structural properties ensuring it is a well-behaved generalization of DPO (§E.1), and a gradient concentration result showing that dwDPO directs more learning signal to the critical step than both standard DPO and existing step-level methods (§E.2).

E.1 Structural Properties

We verify that dwDPO preserves two desirable properties: the Bradley-Terry complementarity axiom and exact recovery of standard DPO as a special case.

Proposition 4 (Complementarity)**.** *For any pair of trajectories (τ^a, τ^b), the weighted preference model (Eq. 16) satisfies:*

$$P(\tau^a \succ \tau^b \mid x) + P(\tau^b \succ \tau^a \mid x) = 1 \qquad (29)$$

provided that the weights $w_t(\cdot)$ depend only on the trajectory content, not on the winner/loser label.

Proof. The divergence step t^* is computed from $|\hat{r}_t(\tau^a) - \hat{r}_t(\tau^b)|$, which is invariant to swapping τ^a and τ^b. Since the weights depend only on the trajectory content, the score function $f(\tau^a, \tau^b) := \sum_t w_t(\tau^a) \cdot \hat{r}_t(\tau^a) - \sum_t w_t(\tau^b) \cdot \hat{r}_t(\tau^b)$ satisfies $f(\tau^b, \tau^a) = -f(\tau^a, \tau^b)$. The result follows from the sigmoid identity $\sigma(z) + \sigma(-z) = 1$. □

Proposition 5 (DPO Recovery)**.** *When $\phi(u_t) = c$ for some constant $c > 0$ (i.e., no step is preferred over another), the dwDPO loss (Eq. 18) reduces exactly to the standard DPO loss (Eq. 7).*

Proof. If $\phi(u_t) = c$ for all t, the denominator of Eq. (21) equals c times the numerator, giving $\lambda = 1/c$ and hence $w_t = \lambda \cdot c = 1$ for all t in both trajectories. Substituting $w_t = 1$ into the dwDPO loss and applying Lemma 1 yields the standard DPO loss. □

Together, these two properties ensure that dwDPO is a strict generalization of DPO: it inherits the well-studied BT preference structure and can always fall back to standard DPO by setting the distribution bandwidths large enough.

E.2 Gradient Concentration

We now formalize the structural deficiency of uniform weighting identified in §3.3. By Lemma 1, the DPO gradient decomposes into per-step terms that share a single scalar prefactor $\sigma(\Delta)$, which does not distinguish between steps. The dwDPO gradient isolates the divergence step:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{dwDPO}} = -\beta\, \mathbb{E}\Big[&\sigma(\Delta)\, w_{t^*} \big(\nabla_\theta \hat{r}_{t^*}(\tau^w) \\ &- \nabla_\theta \hat{r}_{t^*}(\tau^l)\big) \\ &+ \textstyle\sum_{t \neq t^*} w_t(\tau^w)\, \nabla_\theta \hat{r}_t(\tau^w) \\ &- \textstyle\sum_{t \neq t^*} w_t(\tau^l)\, \nabla_\theta \hat{r}_t(\tau^l)\Big], \qquad (30)\end{aligned}$$

where $\nabla_\theta \hat{r}_t(\tau) = \beta\, \nabla_\theta \log \pi_\theta(a_t \mid s_t)$ since π_{ref} does not depend on θ, and both trajectories share $w_{t^*} = \lambda \cdot \phi(0)$. Since $\sigma(\Delta)$ is identical to DPO

by the reward-matching constraint (Eq. 17), the first term directly contrasts the winning and losing actions at t^* with elevated weight; the remaining terms capture non-critical steps with suppressed weights $w_t < w_{t^*}$. To quantify the degree of concentration, we define:

Definition 6 (Concentration Ratio)**.** *The* concentration ratio *of a weighting scheme* $\{w_t\}$ *at the critical step* t^* *is:*

$$\kappa := \frac{w_{t^*}}{\frac{1}{|\tau^l|}\sum_{t=0}^{|\tau^l|-1} w_t(\tau^l)}. \tag{31}$$

$\kappa = 1$ *is uniform;* $\kappa > 1$ *concentrates at* t^**;* $\kappa < 1$ *de-emphasizes* t^**. We use* τ^l *for illustration; the same definition applies symmetrically to* τ^w*.*

Proposition 7 (Gradient Concentration)**.** *(i) DPO:* $\kappa = 1$*. (ii) dwDPO with unimodal* ϕ*:* $\kappa > 1$*. (iii) DMPO with discount* $\delta \in (0, 1)$*:* $\kappa < 1$ *whenever* $t^* > 0$*.*

Proof. (i) All weights equal 1, so $\kappa = 1$.
(ii) $\phi(u_t) < \phi(0)$ for $t \neq t^*$ by strict unimodality, so the average weight is less than w_{t^*}.
(iii) DMPO's discount is monotonically decreasing in t, so $w_t > w_{t^*}$ for all $t < t^*$, and the average exceeds w_{t^*}. □

For $|\tau^l| = 8$ with $t^* = 7$: DPO gives $\kappa = 1$; dwDPO with Gaussian $\sigma = 0.3$ gives $\kappa \approx 2.6$; DMPO with $\delta = 0.9$ gives $\kappa \approx 0.3$. dwDPO directs over twice the average gradient to the critical step, while DMPO directs less than a third—actively de-emphasizing the error.

F Experiment Setup

Base Models and Preference Data. We evaluate against two groups: large foundation models (>100B parameters, reported without fine-tuning) and small deep research agents (<10B, fine-tuned from the same SFT checkpoint). For preference optimization we use the OffSeeker-8B agent (Anonymous, 2026) as the base SFT reference policy (π_{ref}), and their DPO dataset as our training dataset.

Baselines. We compare dwDPO against the following trajectory-level and step-level baselines, all trained from the same π_{ref} checkpoint with the same β:

(a) **Standard DPO** (Rafailov et al., 2023): Uniform weighting across all trajectory steps; our primary baseline.

(b) **SP-DPO (Single-Point)**: DPO loss applied *only* at t^*, zeroing all other steps. Isolates step selection without distribution-based spreading.

(c) **Step-DPO (Uniform)** (Lai et al., 2024): Uniform DPO weight within a single selected reasoning step.

(d) **Turn-DPO (Uniform)**: Uniform weights within a single dialogue turn boundary.

Hyperparameters and Computational Cost. A key advantage of dwDPO is its minimal tuning overhead. The KL-penalty β is inherited directly from the DPO baseline. The divergence threshold γ is selected via grid search over candidate quantiles (40th, 50th, 60th percentiles) of the per-step implicit reward difference distribution on a held-out validation split; in all our experiments the 50th percentile (median) was selected, so we report this as the default. All main experiments use a single default asymmetric Gaussian kernel (σ_L=0.2, σ_R=0.5), selected once on the same held-out validation split and applied unchanged across all six benchmarks. dwDPO adds no additional forward passes: the implicit rewards $\hat{r}_t$ are a byproduct of the log-ratio computation already required by standard DPO.

Tasks and Benchmarks. We evaluate dwDPO on six multi-step deep research benchmarks that require long-horizon tool use and compound reasoning—precisely the regime where credit assignment is hardest.

- **GAIA (text-only)** (Mialon et al., 2023): A factual QA benchmark requiring agents to retrieve, read, and synthesize information across multiple web sources. Trajectories average 4–8 turns with a single final answer graded for correctness.
- **BrowseComp-zh / BrowseComp-en (BC-zh, BC-en)** (Wei et al., 2025): Bilingual browsing-based comprehension benchmarks in which agents must locate and integrate information scattered across retrieved web pages. BC-zh is notably harder due to noisier retrieval and longer context chains.
- **HLE (Humanity's Last Exam)** (Phan et al., 2025): An expert-level multi-domain reasoning benchmark covering mathematics, science, and humanities. Questions require multi-step reasoning chains that are particularly sensitive to early reasoning errors.

- **XBench-DeepSearch (XBench-DS)** (Liu et al., 2025): A multi-hop retrieval benchmark requiring iterative search-read-reason cycles, making correct credit assignment across search and reasoning steps especially important.
- **WebWalkerQA (WWQA)** (Luo et al., 2025): A web navigation benchmark in which agents traverse hyperlinks to answer questions; trajectory quality depends critically on early navigation decisions.

Evaluation Protocol GPT-4o is prompted to compare the agent's final answer against the gold answer and return a binary correctness judgment. We use the same judge prompt and model version across all benchmarks to ensure comparability. For each model we report the mean accuracy over three independent decoding runs; standard deviations are consistently below ± 0.6 pp, confirming that observed gains are not attributable to decoding variance. Table 5 reports the standard deviation of LLM-judge accuracy across three independent decoding runs for each method and benchmark. All standard deviations are below ± 0.6 pp, confirming that the observed gains are not attributable to decoding variance.

Table 5: Standard deviation (pp) of accuracy over three decoding runs.

Method	GAIA	BC-zh	BC-en	HLE	XBench-DS	WWQA
DPO	0.4	0.5	0.3	0.4	0.5	0.4
dwDPO(asym)	0.3	0.6	0.4	0.5	0.4	0.3
dwDPO(sym)	0.4	0.5	0.5	0.4	0.5	0.4

F.1 Analysis

Weight Profile Visualization. Figure 6 illustrates how dwDPO redistributes the learning signal for an 8-step trajectory where the causal divergence occurs at t^*=3. Standard DPO assigns a flat weight (w_t=1) across all steps; dwDPO(asym) concentrates a peak weight of approximately 1.5 at t^* with asymmetric decay. The left side ($t < t^*$) decays rapidly, suppressing the earliest step to near zero and protecting shared correct prefixes from penalization. The right side ($t > t^*$) decays more slowly, sustaining the signal across causal downstream consequences. This profile directly addresses both failure modes identified in §3.3: catastrophic unlearning of correct early behaviors (resolved by near-zero left weights) and diluted learning signal at the critical step (resolved by concentrated right weights).

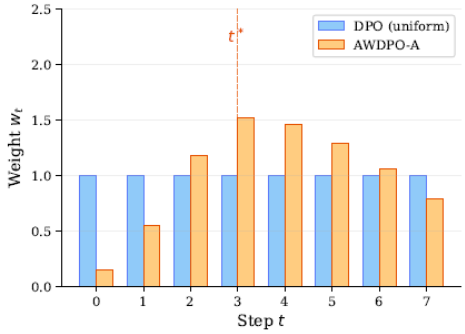

Figure 6: **Weight Profile Comparison.** For an 8-step trajectory diverging at t^*=3, dwDPO(asym) concentrates weight at the divergence step and sustains it across causal downstream steps, while decaying rapidly for shared correct prefix steps ($t < t^*$). Standard DPO penalizes all steps equally (w_t=1).

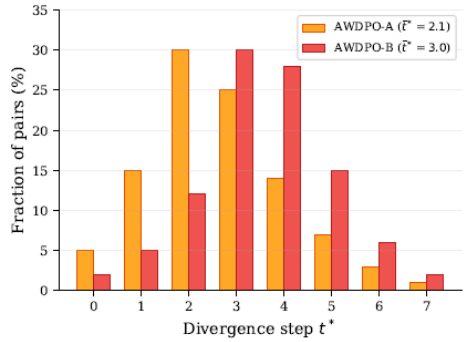

Figure 7: **Divergence Step Distribution.** Histogram of detected t^* values over all training preference pairs, broken down by benchmark. The distribution is skewed toward early turns on all benchmarks, justifying the asymmetric kernel ($\sigma_R > \sigma_L$). XBench-DS has a heavier right tail, motivating the wider default bandwidth σ_R=0.5.

Divergence Step Distribution Across Benchmarks. Figure 7 shows the empirical distribution of detected t^* values across the training preference pairs for each benchmark. On all benchmarks the distribution is skewed toward earlier turns (median $t^* \in [1, 3]$), consistent with the causal rollout findings in §5.3. GAIA and WWQA show the most concentrated distributions (early divergence), while XBench-DS exhibits a heavier tail (divergence spread over turns 2–5). This explains why the right bandwidth σ_R=0.5 rather than 0.3 yields better performance: the wider kernel must cover a more diffuse set of causal downstream steps.

Where dwDPO Underperforms. On HLE, dwDPO(sym) (11.7) underperforms DPO (13.8), while dwDPO(asym) (14.3) does not. We attribute this to HLE's expert-level questions requir-

ing dense multi-step deductive reasoning rather than search-and-retrieve patterns: symmetric kernels assign non-trivial weight to steps *before* t^*, which for deductive tasks encode prerequisite reasoning the model should not be discouraged from learning. The asymmetric kernel, which protects earlier steps more aggressively, avoids this pathology. This is also why Turn-DPO achieves the best single-benchmark HLE score (16.8): its turn-level granularity provides a coarser but less noisy signal on this benchmark. Future work may explore task-adaptive kernel selection.

G AI Assistance Statement

We used generative AI tools, including ChatGPT, as auxiliary support during manuscript preparation and experimental organization. Their use was limited to language polishing, grammar and clarity editing, formatting suggestions, code-command drafting, log summarization, and consistency checks for reported tables and figures. The tools did not independently generate research ideas, select data, run experiments, produce reported results, or determine the paper's scientific claims. All experimental outputs were obtained from author-executed code, and all analyses, numerical results, and final text were checked and approved by the authors, who take full responsibility for this paper.