A Limitation

Although our method theoretically guarantees optimal policy invariance under value-based shaping [20], in practice, the shaping term can negatively affect training dynamics if the teacher's value function is severely misestimated or noisy. Although our method guarantees optimal policy invariance, it can be reach in suboptimal. Our method assumes that the teacher value function provides reliable signals. If the value is highly inaccurate or noisy, the shaping term may interfere with learning and lead to suboptimal policies. We do not test cases with weak or mismatched teachers, and the method's robustness in such settings is unclear.

B Mathematical Derivation

B.1 Mathematical Derivation of Equation 3

We begin with the value-guided shaping framework introduced in our main objective:

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\cdot|\mathbf{s}_t)} \left[\sum_{t=0}^{T-1} \left(r(\mathbf{s}_t, \mathbf{a}_t) + \alpha\, \psi_\phi(\mathbf{s}_t, \mathbf{a}_t) \right) \right], \tag{5}$$

where the shaping term is defined as:

$$\psi_\phi(s, a) := V_\phi(s') - V_\phi(s),$$

and s' is the next state resulting from action a in state s.

To connect this to the DPO setting, we follow the formulation in [25], which relates the reward $r(s, a)$ to a soft Q-function $Q^*(s, a)$. In our setting, the shaped Q-function takes the form:

$$Q^*(s_t, a_t) = r(s_t, a_t) + \psi_\phi(s_t, a_t) + \alpha\, V^*(s_{t+1}), \tag{6}$$

assuming s_{t+1} is not terminal. At terminal states, $V^*(s_T) = 0$.

Using this, we express the total cumulative reward across a trajectory $\tau = (s_0, a_0, \ldots, a_{T-1}, s_T)$ as:

$$\begin{aligned}
\sum_{t=0}^{T-1} r(s_t, a_t) &= \sum_{t=0}^{T-1} \left(Q^*(s_t, a_t) - V^*(s_{t+1}) - \alpha\, \psi_\phi(s_t, a_t) \right) \\
&= Q^*(s_0, a_0) + V^*(s_0) + \sum_{t=1}^{T-1} \left(Q^*(s_t, a_t) - V^*(s_t) - \alpha\, \psi_\phi(s_t, a_t) \right),
\end{aligned}$$

Applying the DPO interpretation of the soft Q-function as log-probabilities [25], we approximate: $Q^*(s_t, a_t) - V^*(s_{t+1}) = \beta \log \pi_\theta(a_t \mid s_t)$ and substitute $\psi_\phi(s_t, a_t) = V_\phi(s_{t+1}) - V_\phi(s_t)$ to obtain the shaped log-prob objective:

$$\sum_{t=0}^{T-1} r(s_t, a_t) = \sum_{t=0}^{T-1} \beta \log \frac{\pi_\theta(a_t \mid s_t)}{\exp\left(\frac{\alpha}{\beta} \psi_\phi(s_t, a_t)\right)} + V^*(s_0).$$

Thus, the TVKD loss over a pairwise preference dataset $\mathcal{D}$ becomes:

$$\begin{aligned}
\mathcal{L}_{\text{TVKD}}(\pi_\theta, \mathcal{D}; \pi_\phi) = -\mathbb{E}_{(\tau_w, \tau_l) \sim \mathcal{D}} \Bigg[\log \sigma \Bigg(& \sum_{t=0}^{|\tau_w|-1} \beta \log \frac{\pi_\theta(a_t^w \mid s_t^w)}{\exp\left(\frac{\alpha}{\beta} \psi_\phi(s_t^w, a_t^w)\right)} \\
& - \sum_{t=0}^{|\tau_l|-1} \beta \log \frac{\pi_\theta(a_t^l \mid s_t^l)}{\exp\left(\frac{\alpha}{\beta} \psi_\phi(s_t^l, a_t^l)\right)} \Bigg) \Bigg],
\end{aligned}$$

where $\psi_\phi(s, a) = V_\phi(s') - V_\phi(s)$.

Note that all conditions based on the initial state s_0 are shared between τ_w and τ_l and therefore are canceled out in the preference-based loss.

B.2 Mathematical Derivation of the Equation 4

We begin by defining the *shaped reward* at each timestep t using the teacher's potential-based shaping function ψ_ϕ, which is independent of the student parameters θ:

$$r_{\text{TVKD}}(a_t, s_t) = \beta \log \pi_\theta(a_t \mid s_t) - \alpha \psi_\phi(s_t, a_t).$$

For a pair of trajectories—one preferred (w) and one less preferred (l)—we define the total shaped reward as:

$$R^w := \sum_{t=0}^{|\tau_w|-1} r_{\text{TVKD}}(a_t^w, s_t^w), \quad R^l := \sum_{t=0}^{|\tau_l|-1} r_{\text{TVKD}}(a_t^l, s_t^l).$$

Then, the preference loss is given by the negative log-sigmoid of the total reward margin:

$$\mathcal{L}_{\text{TVKD}} = -\log \sigma(R^w - R^l),$$

where the scalar margin is:

$$M := R^w - R^l = \sum_{t=0}^{|\tau_w|-1} r_{\text{TVKD}}(a_t^w, s_t^w) - \sum_{t=0}^{|\tau_l|-1} r_{\text{TVKD}}(a_t^l, s_t^l).$$

Since ψ_ϕ is computed only from the teacher and does not depend on θ, its gradient vanishes:

$$\nabla_\theta \psi_\phi(s_t, a_t) = 0.$$

Therefore, the gradient of the margin M becomes:

$$\nabla_\theta M = \beta \left(\sum_{t=0}^{T^w-1} \nabla_\theta \log \pi_\theta(a_t^w \mid s_t^w) - \sum_{t=0}^{T^l-1} \nabla_\theta \log \pi_\theta(a_t^l \mid s_t^l) \right).$$

Applying the identity $\nabla_x \log \sigma(x) = \sigma(-x)$, the gradient of the loss becomes:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{TVKD}} &= -\nabla_\theta \log \sigma(M) = \sigma(-M) \cdot \nabla_\theta M \\ &= \beta \cdot \sigma(-M) \cdot \left(\sum_{t=0}^{T^w-1} \nabla_\theta \log \pi_\theta(a_t^w \mid s_t^w) - \sum_{t=0}^{T^l-1} \nabla_\theta \log \pi_\theta(a_t^l \mid s_t^l) \right). \end{aligned}$$

This formulation generalizes to trajectories of unequal lengths and cleanly separates the teacher shaping from the student optimization path. The teacher's value function contributes to the reward margin, while the gradient flows solely through the student model's log-probabilities.

B.3 Mathematical Derivation of Equation 3 in MaxEnt RL setting

We extend our value-guided objective to the entropy-regularized reinforcement learning (MaxEnt RL) framework. The resulting objective is:

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\cdot|\mathbf{s}_t)} \left[\sum_{t=0}^{T} \left(r(\mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\text{KL}} \left[\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \| \pi_{\text{ref}}(\mathbf{a}_t|\mathbf{s}_t) \right] + \alpha \, \psi_\phi(\mathbf{s}_t, \mathbf{a}_t) \right) \right], \tag{7}$$

where the shaping term is defined as $\psi_\phi(s, a) := V_\phi(s') - V_\phi(s)$, consistent with the formulation in our main objective.

We follow [25], which describes the recursive soft Q-function in token-level DPO as:

$$Q^*(s_t, a_t) = \begin{cases} r(s_t, a_t) + \beta \log \pi_{\text{ref}}(a_t|s_t) + V^*(s_{t+1}) + \alpha \, \psi_\phi(s_t, a_t), & \text{if } s_{t+1} \text{ is not terminal,} \\ r(s_t, a_t) + \beta \log \pi_{\text{ref}}(a_t|s_t) + V_\phi(s_t), & \text{if } s_{t+1} \text{ is terminal.} \end{cases} \tag{8}$$

Summing over a trajectory $\tau = (s_0, a_0, \ldots, a_{T-1}, s_T)$, the total reward becomes:

$$\sum_{t=0}^{T-1} r(s_t, a_t) = \sum_{t=0}^{T-1} [Q^*(s_t, a_t) - \beta \log \pi_{\text{ref}}(a_t|s_t) - V^*(s_{t+1}) - \alpha\, \psi_\phi(s_t, a_t)] \tag{9}$$

$$= \sum_{t=0}^{T-1} [\beta \log \pi_\theta(a_t|s_t) - \beta \log \pi_{\text{ref}}(a_t|s_t) - \alpha\, \psi_\phi(s_t, a_t)] \tag{10}$$

$$= \sum_{t=0}^{T-1} \beta \log \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)} - \sum_{t=0}^{T-1} \alpha\, \psi_\phi(s_t, a_t). \tag{11}$$

The final TVKD loss under MaxEnt RL:

$$\mathcal{L}_{TVKD}(\pi_\theta, \mathcal{D}; \pi_{ref}, \pi_\phi) = -\mathbb{E}_{(\tau_w, \tau_l)\sim\mathcal{D}}\Bigg[\log \sigma\Bigg(\sum_{t=0}^{|\tau_w|-1} \beta \log \frac{\pi_\theta(a_t^w|s_t^w)}{\exp(\frac{\alpha}{\beta}\psi_\phi(s_t^w, a_t^w))\pi_{ref}(a_t^w|s_t^w)} \tag{12}$$

$$- \sum_{t=0}^{|\tau_l|-1} \beta \log \frac{\pi_\theta(a_t^l|s_t^l)}{\exp(\frac{\alpha}{\beta}\psi_\phi(s_t^l, a_t^l))\pi_{ref}(a_t^l|s_t^l)}\Bigg)\Bigg]$$

C Proof of Lemma

C.1 Proof of Lemma 1

We build on the interpretation introduced by Rafailov et al. [25], which characterizes DPO-trained policies as soft-optimal solutions in a deterministic token-level MDP. Specifically, they show that the token-level logits $Q_\phi(s, a)$ of a DPO-trained model correspond to the soft Q-function of an optimal Boltzmann policy under the maximum entropy reinforcement learning (MaxEnt RL) framework.

Assumption (from [25]). Let $\pi_\phi(a \mid s)$ be the DPO-trained policy defined as:

$$\pi_\phi(a \mid s) = \frac{\exp(Q_\phi(s, a)/\beta)}{\sum_{a'} \exp(Q_\phi(s, a')/\beta)},$$

for a fixed temperature $\beta > 0$. Then π_ϕ is a soft-optimal policy with Q-function $Q_\phi(s, a)$, and its corresponding soft value function is defined by:

$$V_\phi(s) := \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[Q_\phi(s, a)] + \beta\mathcal{H}[\pi_\phi(\cdot \mid s)],$$

where $\mathcal{H}[\pi] = -\sum_a \pi(a) \log \pi(a)$ denotes the entropy of the policy.

We now derive a closed-form expression for $V_\phi(s)$ based on the Boltzmann structure of π_ϕ.

Derivation. First, recall the identity:

$$\log \pi_\phi(a \mid s) = \frac{Q_\phi(s, a)}{\beta} - \log \sum_{a'} \exp(Q_\phi(s, a')/\beta).$$

Thus, the entropy term becomes:

$$\begin{aligned}
\mathcal{H}[\pi_\phi] &= -\sum_a \pi_\phi(a \mid s) \log \pi_\phi(a \mid s) \\
&= -\sum_a \pi_\phi(a \mid s) \left(\frac{Q_\phi(s, a)}{\beta} - \log \sum_{a'} \exp(Q_\phi(s, a')/\beta) \right) \\
&= -\frac{1}{\beta} \sum_a \pi_\phi(a \mid s) Q_\phi(s, a) + \log \sum_{a'} \exp(Q_\phi(s, a')/\beta).
\end{aligned}$$

Now substitute back into the soft value function definition:

$$\begin{aligned} V_\phi(s) &= \sum_a \pi_\phi(a \mid s) Q_\phi(s,a) + \beta \mathcal{H}[\pi_\phi] \\ &= \sum_a \pi_\phi(a \mid s) Q_\phi(s,a) + \beta \left(-\frac{1}{\beta} \sum_a \pi_\phi(a \mid s) Q_\phi(s,a) + \log \sum_{a'} \exp(Q_\phi(s,a')/\beta) \right) \\ &= \log \sum_{a \in \mathcal{V}} \exp(Q_\phi(s,a)/\beta). \end{aligned}$$

This completes the proof.

C.2 Proof of Lemma 2

In Section 3 we stated that augmenting the DPO reward with the Teacher Temporal Difference term $V_\phi(s_{t+1}) - V_\phi(s_t)$ does not change the optimal policy. Below we provide a formal proof.

Proof. Consider a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T)$. Let the original return be:

$$G(\tau) = \sum_{t=0}^{T-1} r(s_t, a_t).$$

Now, define a potential-based shaped reward:

$$\tilde{r}(s_t, a_t, s_{t+1}) = r(s_t, a_t) + \Phi(s_{t+1}) - \Phi(s_t),$$

and let the corresponding shaped return be:

$$\tilde{G}(\tau) = \sum_{t=0}^{T-1} \left[r(s_t, a_t) + \Phi(s_{t+1}) - \Phi(s_t) \right].$$

By telescoping,

$$\sum_{t=0}^{T-1} (\Phi(s_{t+1}) - \Phi(s_t)) = \Phi(s_T) - \Phi(s_0),$$

so the shaped return becomes:

$$\tilde{G}(\tau) = G(\tau) + \Phi(s_T) - \Phi(s_0).$$

This additive term depends only on the start and end states, not on the specific actions taken. Therefore, for any policy π, the expected shaped state-action value is:

$$\tilde{Q}^\pi(s,a) = Q^\pi(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)}[\Phi(s')] - \Phi(s).$$

Since the difference between shaped Q-values at any state s only depends on $\Phi(s)$ and the expected $\Phi(s')$, the ranking over actions remains unchanged:

$$\arg\max_a \tilde{Q}^*(s,a) = \arg\max_a Q^*(s,a).$$

Hence, the optimal policy is invariant under this form of reward shaping. □

C.3 Proof of Corollary 2.1

In Section 3 we claimed that action-based shaping terms such as $\psi(s,a) = \alpha \log \pi_\phi(a \mid s)$ or $\psi(s,a) = \alpha Q_\phi(s,a)$ violate the conditions required for potential-based shaping and may alter the optimal policy. We now provide a formal justification.

Proof. Suppose we augment the DPO reward with an action-dependent shaping term:

$$\tilde{r}(s_t, a_t) = r(s_t, a_t) + \psi(s_t, a_t), \quad \text{where} \quad \psi(s_t, a_t) \neq \Phi(s_t) - \Phi(s_{t+1}).$$

Then the shaped return for a trajectory $\tau = (s_0, a_0, s_1, \ldots, s_T)$ becomes:

$$\tilde{G}(\tau) = \sum_{t=0}^{T} \left[r(s_t, a_t) + \psi(s_t, a_t) \right].$$

Unlike the state-based shaping case in Lemma 2, there is no telescoping cancellation of the added shaping terms because $\psi(s_t, a_t)$ depends explicitly on the action at each step and not solely on the state. As a result, the shaped return $\tilde{G}(\tau)$ differs from the original return $G(\tau)$ by an amount that depends on the entire sequence of actions:

$$\tilde{G}(\tau) = G(\tau) + \sum_{t=0}^{T} \psi(s_t, a_t),$$

where $\sum_t \psi(s_t, a_t)$ varies with the policy and trajectory.

This action-dependent distortion implies that:

$$\tilde{Q}^{\pi}(s, a) = Q^{\pi}(s, a) + \mathbb{E}_{\tau \sim \pi | (s_0 = s, a_0 = a)} \left[\sum_{t=0}^{T} \psi(s_t, a_t) \right],$$

which can shift the relative ranking of actions at state s, thereby changing $\arg\max_a \tilde{Q}^*(s, a)$.

Hence, the optimal policy under $\tilde{r}$ may differ from the one under the original reward r, violating policy invariance. □

D Relation to Advantage Function in RL

Our method can be interpreted as a special form of value-based reward shaping, closely related to the advantage function in reinforcement learning. In standard RL, the advantage function is defined as:

$$A(s, a) = r(s, a) + V(s') - V(s),$$

which quantifies the relative benefit of taking action a at state s compared to the expected return from the state baseline $V(s)$.

In our formulation, we do not use an explicit reward $r(s, a)$, as it is typically unavailable in offline preference datasets. Instead, we define the reward shaping term using the teacher's value function:

$$\psi(s, a) := V_{\phi}(s') - V_{\phi}(s),$$

where V_{ϕ} is the value function learned by the DPO teacher model, and s' denotes the next state (i.e., the updated token-level context after applying action a).

This formulation implicitly captures a notion of advantage by measuring the change in future utility as estimated by the teacher. It provides directional feedback for the student model without relying on noisy or hard-to-estimate token-level rewards.

This connection offers a theoretical justification for the stability and effectiveness of our method, as value-based shaping of this form is known to preserve the optimal policy under standard conditions [20].

E Experiment Details

E.1 Training Details

Training Details The SFT for both student and teacher models is conducted over 3 epochs, using a learning rate of 2×10^{-5} and a batch size of 128. The DPO teacher is trained with $\beta = 0.05$, a learning rate of 5×10^{-7}, a batch size of 128, and for 2 epochs. For distillation methods, we employ context distillation [3] to improve efficiency. Specifically, we precompute the top 50 probabilities and save them to use for distillation following [9]. In TVKD, we experiment with $\alpha \in [0.1, 0.2, 0.5, 0.7, 1.0, 1.5]$ and $\beta \in [0.1, 0.2, 0.5, 1, 2, 5]$.

Table 6: Overall performance of Open LLM Leaderboard trained with DPOMIX datasets.

Method	Average	ARC-easy	GSM8K	Hellaswag	MMLU	TruthfulQA	Winogrande
DPO teacher	53.448	76.800	47.900	61.480	60.230	04.200	70.080
SFT teacher	53.152	76.200	47.300	60.844	60.070	04.500	70.000
SFT	40.176	59.040	05.500	49.570	37.950	29.488	59.510
DPO	35.475	59.000	06.300	49.760	37.840	00.600	59.350
SimPO	41.497	64.942	06.975	**50.129**	37.943	29.009	59.984
WPO	41.356	64.850	06.369	49.642	37.958	29.253	60.063
TDPO	40.950	64.310	06.431	49.991	37.560	27.907	59.747
VanillaKD	41.166	64.980	05.838	49.353	37.922	28.764	60.141
SeqKD	41.166	64.980	06.141	48.840	**37.820**	29.002	60.210
DDPO	41.225	65.000	06.141	49.492	37.840	29.131	59.747
DPKD	41.354	64.890	06.388	49.542	37.986	29.253	60.063
GKD	41.191	64.770	06.670	49.070	37.900	28.274	60.460
DCKD	41.328	**65.110**	05.990	49.340	37.850	29.540	60.140
ADPA	40.288	60.185	**06.670**	48.626	37.450	28.760	60.039
TVKD (Ours)	**41.605**	64.550	05.760	49.650	**38.540**	**29.880**	**61.250**

Detailed baseline setting For ADPA, following the original implementation [9], we first generated a single student rollout over the entire dataset. We then computed and saved the differences between the DPO teacher and the SFT teacher, which were used during training. For other online-based KD method(SeqKD,GKD), we use only single student or teacher rollout over the entire datasets following [9]. For online-based preference distillation method, we use ground-truth chosen and rejected answer instead of teacher and student rollouts following [9]. It can be different original implementation in theoretically. In the case of our method, we saved the DPO teacher's logits over the original dataset once and used them for training.

Computational Efficiency We conducted our experiments using four Nvidia RTX 4090 GPUs. The total wall time for TVKD was approximately 2.5 to 3 hours. In our method, we precompute and store the DPO teacher's logits over the entire dataset once, which reduces computational overhead during training at the cost of increased storage usage. In contrast, methods such as GKD, SeqKD, and ADPA require generating additional student or teacher rollouts across the entire dataset and subsequently computing and storing teacher logits for each of those rollouts. This results in a larger storage overhead and slightly higher computational cost compared to our approach. In particular, ADPA requires not only the DPO-trained teacher but also the SFT-trained model to compute the logit differences, further increasing computational resource demands.

F Additional Results

F.1 Open LLM Leaderboard

We show the overall score of the OLLs in the main table as shown below in Table 6, Table 7.

F.2 MT-bench

We represent the hexagons in Figure 3 that represent the performance of each task on MT-bench. Our method shows consistently high performance compared to other methods, except for coding.

F.3 Robustness to Distillation Strength

In Figure 4a, we conduct an ablation study on the hyperparameter α, which controls the influence of the teacher's value signal during training. We observe that TVKD achieves optimal performance on both the RM score and MT-bench when $\alpha = 0.7$, with performance dropping at both lower and higher values. This indicates that our method offers a tunable mechanism to balance teacher guidance and data-driven learning. A small α may underutilize the teacher's future preference signal, while an excessively large α can suppress the student's capacity to fit dataset-specific patterns. The ability

Table 7: Overall performance of Open LLM Leaderboard trained with Helpsteer2 datasets.

Method	Average	ARC-easy	GSM8K	Hellaswag	MMLU	TruthfulQA	Winogrande
DPO teacher	57.949	78.746	46.745	61.085	60.109	30.290	70.718
SFT teacher	53.152	76.200	47.300	60.844	60.070	04.500	70.000
SFT	41.064	64.983	06.369	48.676	37.787	29.376	59.195
DPO	41.243	64.680	05.980	**49.572**	37.972	29.253	60.000
SimPO	41.308	64.680	06.141	48.974	37.958	30.110	59.984
WPO	41.190	64.915	06.141	48.645	38.043	28.860	60.537
TDPO	41.130	64.140	07.126	49.114	36.767	28.886	60.805
VanillaKD	41.089	65.194	06.065	48.645	37.081	29.009	60.537
SeqKD	41.169	65.194	05.838	48.964	**38.200**	29.009	59.809
DDPO	41.234	65.109	06.293	49.522	37.958	29.009	59.511
DPKD	41.136	64.948	**06.358**	49.422	37.892	28.764	59.433
GKD	40.924	65.025	05.075	49.104	37.858	29.131	59.353
DCKD	41.092	65.125	06.520	48.516	37.253	28.519	**60.616**
ADPA	41.259	**65.446**	05.898	49.104	38.007	28.641	60.458
TVKD (Ours)	**41.352**	64.229	05.681	48.974	38.157	**30.772**	60.300

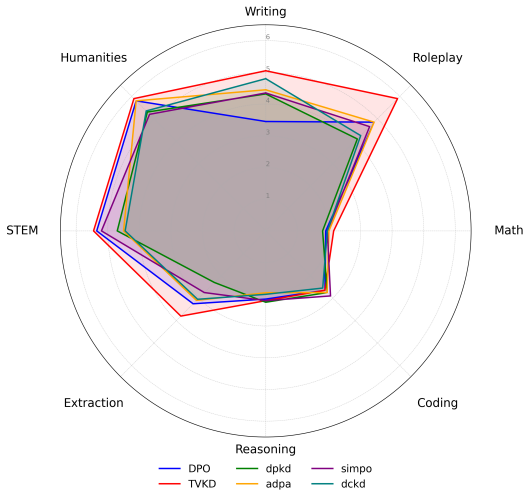


Figure 3: Visualization of token-wise preference

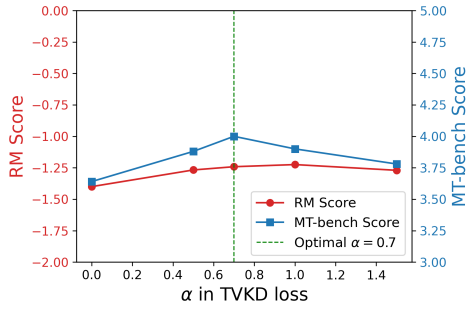
to adjust α provides practical flexibility, enabling users to calibrate the distillation strength to suit various datasets, teacher qualities, or deployment constraints.

F.4 Robustness to Temperature Parameter

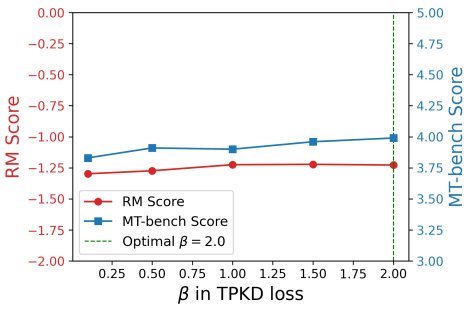
We present the robustness evaluation with respect to the temperature parameter β in Figure 4b. Our method achieves the highest performance at $\beta = 2$, while overall demonstrating robust behavior across a wide range of values. This suggests that although higher temperatures yield a sharper value function and thus increase discriminability, the influence of β remains limited since our method captures not the absolute value itself, but the difference in value before and after the current action.

F.5 Teacher Model Analysis

We show additional experiments in Table 8 and Table 9 to demonstrate that TPKD is robust distillation to teacher quality. We compare the disitllation performance on three different teachers: the default teacher model SFTed on Deita-10k-V0, a teacher model trained with Helpsteer2 as a DPO, and Instruct teacher, the instruct tuning version posted on Huggingface. Our method outperforms DCKD on all teacher settings. This shows that TPKD is relatively robust to the quality of the teachers. Note that Instruct teacher performs poorly on RM using helpsteer2's test set, but performs as well as DPO teacher on MT-bench, which is a comprehensive evaluation.



(a) Robustness to Distillation Strength



(b) Robustness to Temperature Parameter

Figure 4: Sensitivity analysis of distillation hyperparameters.

Table 8: RM (higher is better) comparison between DCKD and TPKD across teacher types.

Teacher Type	DPO	SFT	Instruct
DCKD	-1.46	-1.70	-1.68
TPKD	-1.21	-1.46	-1.56

Table 9: MT-bench (higher is better) comparison between DCKD and TPKD across teacher types.

Teacher Type	DPO	SFT	Instruct
DCKD	3.51	3.38	3.60
TPKD	3.98	3.73	3.96