NEAR TO FAR SELF DISTILLATION FOR OFFLINE GOAL-CONDITIONED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline goal-conditioned reinforcement learning (GCRL) aims to learn policies for reaching diverse goals from a fixed dataset. Existing methods use value estimates or subgoals to guide policy learning, but supervision for current decisions can remain weak when goals are distant. Inspired by the dense predictive supervision of on-policy distillation (OPD), we use teacher predictions as direct learning targets. This raises two challenges in offline GCRL: (1) the Guidance Selection Challenge, identifying privileged information that provides useful guidance toward the distant goal; and (2) the Information Gap Challenge, transferring such guidance to a student without access to the privileged information. To address these challenges, we propose NEFD, a Near-to-Far Self-Distillation method for offline GCRL. We observe that near-future states reveal local outcomes of recorded behavior and thus provide informative privileged references for current decisions. NEFD uses these states as teacher-only information and weights teacher guidance by estimated goal progress. To bridge the information gap, NEFD performs action and representation self-distillation within a shared actor: the former matches teacher action distributions, while the latter uses teacher decision representations as intermediate targets. Experiments on 29 OGBench datasets show that NEFD improves policy learning when distant goals provide weak supervision for current decisions.

1 INTRODUCTION

Offline goal-conditioned reinforcement learning (GCRL) aims to learn a universal policy for reaching diverse goals from a fixed dataset (Kaelbling, 1993; Schaul et al., 2015; Levine et al., 2020; Park et al., 2025a). Learning such policies is particularly challenging over long horizons, where supervision derived from distant goals can be weak or ambiguous for current decisions. Although the dataset records the actions taken, sparse feedback makes it difficult to determine which current actions should be reinforced to reach the desired goal (Park et al., 2023; Zhou & Kao, 2025).

Existing GCRL approaches can be categorized into two complementary directions: (1) Value-guided methods, which use goal-conditioned values, advantages, or temporal distance estimates to guide action learning (Yang et al., 2022; Ma et al., 2022; Eysenbach et al., 2022; Wang et al., 2023; Myers et al., 2025; Ke et al., 2025); and (2) Subgoal-based methods, which use subgoals to provide local guidance for policy learning (Park et al., 2023; Ahn et al., 2025; Zhou & Kao, 2025; Kim et al., 2023; Chane-Sane et al., 2021; Kang et al., 2026).

Despite these advances, providing effective supervision for current decisions remains challenging when the goal is distant. For value-guided methods, competing actions can have similar values when the goal is distant. Estimation errors can then obscure these small value differences, making it difficult to determine which actions should be reinforced (Park et al., 2023).

Subgoal-based methods provide direct local guidance. HIQL (Park et al., 2023) uses a high-level policy to generate subgoals for low-level control, but its effectiveness depends on the quality of the generated subgoals. SAW (Zhou & Kao, 2025) avoids subgoal generation by using a subgoal-conditioned policy to train a flat policy. However, learning to select actions from the current state and a distant goal can remain difficult for the flat policy, even with explicit action targets (Park et al., 2025b). Both methods show declining performance as goals become more distant. These limitations motivate stronger supervision for current decisions under distant goals.

We draw inspiration from the dense predictive supervision provided by on-policy distillation (OPD) (Song & Zheng, 2026). In language models, OPD uses the teacher's predictive distributions on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

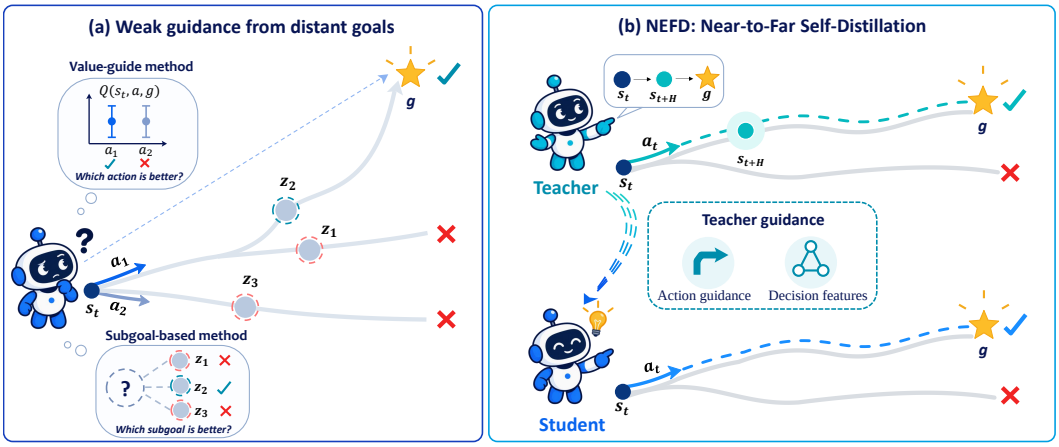


Figure 1: **Policy learning in offline goal-conditioned reinforcement learning.** (a) Under distant goals, value-guided methods struggle to determine which action is better, while subgoal-based methods struggle to identify reliable subgoals. (b) NEFD uses near-future states as privileged information alongside the distant goal to form teacher guidance. Action and representation self-distillation provide dense supervision for a student that makes decisions from its own history and the same distant goal.

sequences generated by the student as learning targets at each token. Some OPD methods (Zhao et al., 2026) additionally condition the teacher on privileged information, such as reference solutions available during training but withheld from the student. This motivates us to use a teacher with additional information to strengthen supervision for current decisions under distant goals.

However, exploiting privileged information for policy self-distillation in offline GCRL introduces two key challenges. First, the **Guidance Selection Challenge**: what privileged information can provide informative guidance for current decisions, and which resulting guidance should receive greater emphasis for reaching the distant goal? Offline trajectories contain rich future information, but different local outcomes can vary substantially in their relevance to the desired goal. Second, the **Information Gap Challenge**: how can such privileged teacher guidance be transferred to a student that does not have access to the same information? The student must learn the corresponding decisions from its own state–action history and distant goal alone, and this mapping can remain difficult over long horizons even when explicit action targets are available.

To address these challenges, we propose NEFD, a Near-to-Far Self-Distillation method for offline GCRL, as illustrated in Figure 1. NEFD constructs a privileged teacher and a deployable student within a shared causal Transformer actor (Chen et al., 2021). Both receive the same state–action history and distant goal, while only the teacher additionally observes privileged information from the offline trajectory. Our key observation is that the state reached a few steps later reveals the local outcome of the recorded behavior, providing a direct reference for the current decision beyond the distant goal. We therefore use this near-future state as the teacher's privileged information. However, not every observed local outcome contributes equally to reaching the desired goal. To further address the Guidance Selection Challenge, NEFD estimates the goal progress associated with each near-future state using a goal-conditioned value function parameterized by LAN (Kang et al., 2026), and uses this progress to modulate the strength of teacher supervision. Guidance associated with greater progress toward the distant goal thus receives greater emphasis during distillation.

For the Information Gap Challenge, NEFD transfers the resulting teacher guidance to the student at both the action and representation levels. Action self-distillation matches the teacher's predictive action distributions at each decision position, providing direct supervision for the student's current decisions. Yet action targets alone do not explain how the student should form the decision from its more limited inputs. We therefore further perform representation self-distillation, using the teacher's decision representations as intermediate targets to guide the student in encoding its own history and distant goal into decision-relevant features. Both distillation objectives are weighted by the estimated goal progress described above, so that the student preferentially learns from teacher guidance that is more useful for reaching the distant goal. In this way, NEFD transfers local guidance obtained from privileged near-future information into a policy that requires only the observed history and desired goal at deployment.

Extensive evaluations on 29 OGBench datasets (Park et al., 2025a) show that NEFD achieves state-of-the-art performance across a broad range of navigation and manipulation tasks, with improvements extending to both state and visual observations. The gains are particularly pronounced in challenging settings where distant goals provide weak supervision for current decisions, such as long-horizon navigation and multistage manipulation. Ablation studies further validate the effectiveness of the privileged teacher, action and representation self-distillation, and goal progress weighting.

2 Background

2.1 Decision Transformer

Sequence models use trajectory context to predict actions or future states (Janner et al., 2021; Zeng et al., 2023; Lei et al., 2026). Decision Transformer (Chen et al., 2021) conditions action prediction on the return-to-go $\widehat{R}_t = \sum_{t'=t}^{T} r_{t'}$ and represents a trajectory as

$$\tau = \left(\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_T, s_T, a_T\right). \tag{1}$$

The sequence is modeled by a causal Transformer (Vaswani et al., 2017), which predicts each action from the hidden representation associated with the corresponding state token. We adopt this causal sequence modeling architecture but condition the actor on a task goal rather than the return-to-go. The specific construction of the goal, state, and action sequence is introduced in Section 3.

2.2 On-Policy Distillation

On-policy distillation (OPD) (Agarwal et al., 2024; Gu et al., 2024; Song & Zheng, 2026) trains a language model using a teacher's token-level predictive distributions on sequences generated by the student, providing dense supervision at each generation step. Let π_{S} and π_{T} denote the student and teacher policies, respectively. Given a prompt x, the student generates a sequence $y \sim \pi_{\mathrm{S}}(\cdot \mid x)$. The distillation objective can be written as

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\mathrm{S}}(\cdot|x)}\left[\frac{1}{|y|}\sum_{t=1}^{|y|} D_{\mathrm{KL}}\left(\pi_{\mathrm{T}}(\cdot \mid x, y_{<t}) \,\middle\|\, \pi_{\mathrm{S}}(\cdot \mid x, y_{<t})\right)\right], \tag{2}$$

where $\mathcal{D}$ is the prompt dataset, $y_{<t}$ denotes the generated prefix before position t, and D_{KL} denotes the Kullback–Leibler divergence. The teacher predictions are treated as fixed targets during student updates.

3 Near-to-Far Self-Distillation

In offline GCRL over long horizons, supervision derived from distant goals can be weak or ambiguous for current decisions. To strengthen this supervision, we propose NEFD, a Near-to-Far Self-Distillation method with three components, as illustrated in Figure 2.

NEFD uses near-future states as privileged information alongside the distant goal to form teacher guidance (Section 3.1). Action and representation self-distillation transfer this guidance to the student (Section 3.2). Both objectives are weighted by estimated goal progress (Section 3.3).

3.1 Teacher and Student Policies

The teacher and student share a goal-conditioned causal Transformer actor π_θ (Chen et al., 2021) with a sequence length of K time steps. Both receive the same state–action history and desired goal g, with g held fixed for all K time steps in the sampled sequence. At each time step i, we use a shared condition encoder E to construct the condition tokens $e_i^{\mathrm{T}} = E(g, s_{i+H})$ for the teacher and $e_i^{\mathrm{S}} = E(g)$ for the student. Here, s_{i+H} denotes the state reached H steps later in the same offline trajectory and is provided only to the teacher during training as privileged information. For $v \in \{\mathrm{T}, \mathrm{S}\}$, the corresponding condition tokens are interleaved with state and action tokens to form the input sequence ending at time step t:

$$\mathcal{H}_t^v = \left(e_{t-K+1}^v, s_{t-K+1}, a_{t-K+1}, \ldots, e_{t-1}^v, s_{t-1}, a_{t-1}, e_t^v, s_t\right). \tag{3}$$

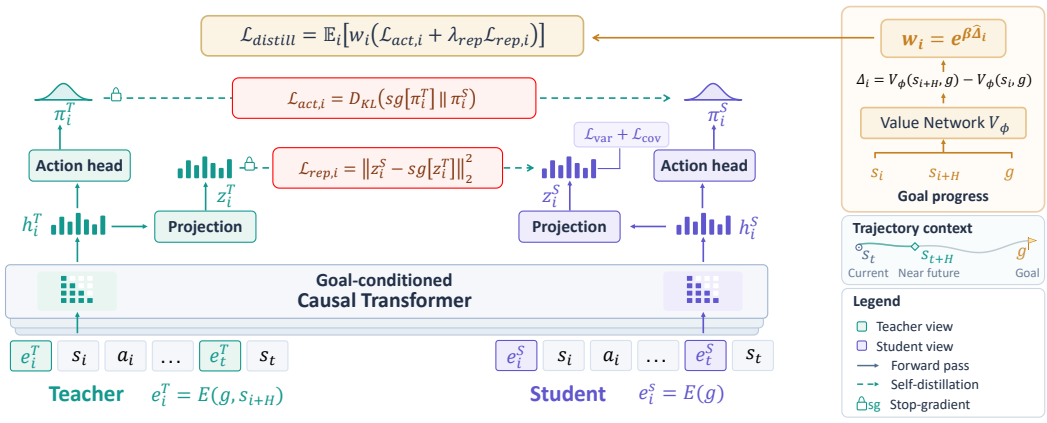


Figure 2: **NEFD Overview.** The teacher and student share a goal-conditioned causal Transformer, with the teacher additionally observing privileged near-future states. Teacher guidance is transferred to the student through action and representation self-distillation, both weighted by estimated progress toward the distant goal.

The actor predicts a_i from $\mathcal{H}^v_{t,\le i}$, the causal prefix of $\mathcal{H}^v_t$ ending at state token s_i.

Offline trajectories may contain suboptimal actions. We therefore train the shared actor under both conditions with advantage-weighted regression (Peng et al., 2019), giving greater weight to recorded actions with higher estimated advantages. We estimate advantages using a goal-conditioned value function V_ϕ (see Appendix A.2). For the teacher conditioned on both g and s_{i+H}, the observed near-future outcome provides a local reference for evaluating recorded actions. We therefore compute the teacher's action regression weights using the advantage $A_i^{\mathrm{T}} = V_\phi(s_{i+1}, s_{i+H}) - V_\phi(s_i, s_{i+H})$. For the student, we use $A_i^{\mathrm{S}} = V_\phi(s_{i+1}, g) - V_\phi(s_i, g)$ to retain direct supervision for reaching g. The resulting weighted behavior cloning objective is

$$\mathcal{L}_{\mathrm{beh}} = -\mathbb{E}\left[\frac{1}{K}\sum_{i=t-K+1}^{t}\sum_{v\in\{\mathrm{T},\mathrm{S}\}} e^{\alpha_v A_i^v}\log\pi_\theta\left(a_i \mid \mathcal{H}^v_{t,\le i}\right)\right], \tag{4}$$

where α_v is an inverse temperature controlling the strength of advantage weighting.

3.2 Near-to-Far Policy Distillation

The student must learn current decisions from its own history and distant goal, without the near-future states used to form the teacher's guidance. To transfer this guidance across the information gap, we perform self-distillation at both the action and representation levels.

Action self-distillation Supervision from distant goals can provide weak or ambiguous guidance for the student's current decisions. The teacher additionally observes the near-future state, giving it a more direct reference for predicting the current action. We therefore use the teacher's predictive action distribution as a supervision target for the student at each decision position i:

$$\mathcal{L}_i^{\mathrm{act}} = D_{\mathrm{KL}}\left(\mathrm{sg}\left[\pi_\theta\left(\cdot \mid \mathcal{H}^{\mathrm{T}}_{t,\le i}\right)\right] \,\|\, \pi_\theta\left(\cdot \mid \mathcal{H}^{\mathrm{S}}_{t,\le i}\right)\right), \tag{5}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. This objective provides direct action-level supervision from the teacher at each decision position.

Representation self-distillation Action self-distillation provides the student with a direct target for the current action at each decision position. However, the student must learn the internal representation needed to produce this prediction from its own sequence information and distant goal, without the privileged information available to the teacher. In long horizon tasks, mapping these inputs to the appropriate local action can remain difficult even when the action target is explicit (Park et al., 2025b).

The observed near-future state reveals the short term outcome of the recorded behavior, helping the teacher relate the current sequence to the local action. The teacher's representation is therefore learned with this local outcome information for the same action prediction task, making it a relevant intermediate target for the student. We distill this representation immediately before the action head to

guide the student's representation learning. This extends supervision from the final action prediction to the internal representation that supports it.

Let h_i^v denote the hidden representation at state token s_i before the action head, for $v \in \{\mathrm{T}, \mathrm{S}\}$. A shared projection head P maps these representations to $z_i^v = P(h_i^v)$. The student matches the teacher's projected representation by minimizing

$$\mathcal{L}_i^{\mathrm{rep}} = \left\| z_i^{\mathrm{S}} - \mathrm{sg}\left[z_i^{\mathrm{T}}\right] \right\|_2^2 . \tag{6}$$

To discourage collapse and reduce feature redundancy, we apply variance and covariance regularization to the student projections (Bardes et al., 2022). Implementation details are provided in Appendix A.3.

3.3 Goal Progress Weighting

The teacher uses observed near-future outcomes to guide current decisions, but these outcomes differ in their usefulness for reaching the desired goal. To focus the student's learning on guidance that supports goal achievement, we therefore adjust the strength of teacher supervision according to the estimated goal progress associated with each outcome:

$$\Delta_i = V_\phi(s_{i+H}, g) - V_\phi(s_i, g). \tag{7}$$

When the near-future state indicates greater estimated progress toward g from the current state, we give the corresponding teacher guidance a larger weight. Specifically, we standardize Δ_i within each minibatch to obtain $\widehat{\Delta}_i$ and apply exponential weighting, giving larger weights to teacher guidance associated with greater estimated progress. Since the teacher's action and representation targets both use the same near-future state, we apply the same weight to both distillation objectives:

$$\mathcal{L}_{\mathrm{distill}} = \mathbb{E}\left[\frac{1}{K}\sum_{i=t-K+1}^{t} e^{\beta\widehat{\Delta}_i}\left(\mathcal{L}_i^{\mathrm{act}} + \lambda_{\mathrm{rep}}\mathcal{L}_i^{\mathrm{rep}}\right)\right]. \tag{8}$$

Here, β controls the strength of goal progress weighting, while λ_{rep} balances the action and representation self-distillation.

4 Experiments

We evaluate NEFD against seven representative offline GCRL methods on 29 OGBench datasets. The results demonstrate strong performance in long-horizon navigation, trajectory stitching, and robotic manipulation, with improvements across both state and visual observations. Ablation studies examine the contributions of privileged teacher, action and representation self-distillation, and goal-progress weighting. Additional results and implementation details are provided in Appendix B.

4.1 Experimental Setup

Benchmarks. We evaluate NEFD on 29 datasets from OGBench (Park et al., 2025a). These datasets cover long-horizon maze navigation in `Pointmaze`, `Antmaze`, and `Humanoidmaze`, as well as robotic manipulation in `Cube` and `Scene`. The state-based maze tasks span `medium`, `large`, and `giant` layouts under the `navigate` and `stitch` settings, evaluating long-horizon goal reaching and trajectory stitching across different agent embodiments. The manipulation tasks require goal-conditioned object manipulation and sequential skill composition. Of the 29 datasets, 22 use state observations, while the remaining seven are variants of `Antmaze`, `Cube`, and `Scene` that use visual observations.

Baselines. We compare NEFD with seven representative offline GCRL baselines covering two complementary directions: (1) Value-guided methods, including GCIVL (Park et al., 2023), GCIQL (Kostrikov et al., 2022), CRL (Eysenbach et al., 2022), and QRL (Wang et al., 2023); and (2) Subgoal-based methods, including HIQL (Park et al., 2023), LAVL (Kang et al., 2026), and SAW (Zhou & Kao, 2025). To ensure a fair comparison, we report baseline results from OGBench or the corresponding papers whenever available. Otherwise, we evaluate the methods using their official implementations. Result sources and reproduction details are provided in Appendix B.1.2.

Table 1: **Full benchmark table.** Following the standard OGBench protocol (Park et al., 2025a), we average binary success rates (%) over five test-time goals per dataset and seed. We report mean ± standard deviation across 8 seeds for state-based tasks and 4 for visual tasks. See Appendix B.1.3 for details.

Environment	Dataset Type	GCIVL	GCIQL	QRL	CRL	HIQL	SAW	LAVL	NEFD
Pointmaze	medium-navigate-v0	63 ±6	53 ±8	82 ±5	29 ±7	79 ±5	97 ±2	92 ±1	**100** ±0
	large-navigate-v0	45 ±5	34 ±3	86 ±9	39 ±7	58 ±5	85 ±10	93 ±3	**100** ±0
	giant-navigate-v0	0 ±0	0 ±0	68 ±7	27 ±10	46 ±9	68 ±8	91 ±4	**100** ±0
	medium-stitch-v0	70 ±14	21 ±9	80 ±12	0 ±1	74 ±6	87 ±4	90 ±2	**100** ±0
	large-stitch-v0	12 ±6	31 ±2	84 ±15	0 ±0	13 ±6	8 ±9	87 ±8	**100** ±0
	giant-stitch-v0	0 ±0	0 ±0	50 ±8	0 ±0	0 ±0	0 ±0	95 ±2	**100** ±0
Antmaze	medium-navigate-v0	72 ±8	71 ±4	88 ±3	95 ±1	96 ±1	97 ±1	97 ±1	**98** ±1
	large-navigate-v0	16 ±5	34 ±4	75 ±6	83 ±4	91 ±2	90 ±3	93 ±1	**94** ±1
	giant-navigate-v0	0 ±0	0 ±0	14 ±3	16 ±3	65 ±5	73 ±4	85 ±2	**88** ±2
	medium-stitch-v0	44 ±6	29 ±6	59 ±7	53 ±6	94 ±1	96 ±1	97 ±1	**98** ±1
	large-stitch-v0	18 ±2	7 ±2	18 ±2	11 ±2	67 ±5	66 ±9	92 ±1	**94** ±1
	giant-stitch-v0	0 ±0	0 ±0	0 ±0	0 ±0	2 ±2	1 ±1	82 ±2	**83** ±3
Humanoidmaze	medium-navigate-v0	24 ±2	27 ±2	21 ±8	60 ±4	89 ±2	88 ±3	94 ±1	**97** ±1
	large-navigate-v0	2 ±1	2 ±1	5 ±1	24 ±4	49 ±4	46 ±4	74 ±3	**80** ±3
	giant-navigate-v0	0 ±0	0 ±0	1 ±0	3 ±2	12 ±4	35 ±4	83 ±2	**87** ±2
	medium-stitch-v0	12 ±2	12 ±3	18 ±2	36 ±2	88 ±2	75 ±5	93 ±2	**95** ±1
	large-stitch-v0	1 ±1	0 ±0	3 ±1	4 ±1	28 ±3	25 ±3	72 ±3	**82** ±2
	giant-stitch-v0	0 ±0	0 ±0	0 ±0	0 ±0	3 ±2	4 ±3	80 ±1	**83** ±2
Cube	single-play-v0	53 ±4	68 ±6	5 ±1	19 ±2	15 ±3	72 ±5	83 ±4	**96** ±3
	double-play-v0	36 ±3	40 ±5	1 ±0	10 ±2	6 ±2	40 ±7	42 ±8	**75** ±2
	triple-play-v0	1 ±0	3 ±1	0 ±0	4 ±1	3 ±1	4 ±2	11 ±2	**62** ±9
Scene	play-v0	42 ±4	51 ±4	5 ±1	19 ±2	38 ±3	63 ±6	88 ±4	**95** ±2
Visual-Antmaze	medium-navigate-v0	22 ±2	11 ±1	0 ±0	94 ±1	93 ±4	**95** ±0	92 ±1	**95** ±1
	large-navigate-v0	5 ±1	4 ±1	0 ±0	84 ±1	53 ±9	82 ±4	67 ±6	**86** ±2
	giant-navigate-v0	1 ±1	0 ±0	0 ±0	47 ±2	6 ±4	10 ±2	7 ±2	**51** ±2
Visual-Cube	single-play-v0	60 ±5	30 ±5	41 ±15	31 ±15	**89** ±0	88 ±3	83 ±2	87 ±2
	double-play-v0	10 ±2	1 ±1	5 ±0	2 ±1	39 ±2	40 ±3	16 ±4	**46** ±4
	triple-play-v0	14 ±2	15 ±1	16 ±1	17 ±2	21 ±0	20 ±1	8 ±5	**22** ±1
Visual-Scene	play-v0	25 ±3	12 ±2	10 ±1	11 ±2	49 ±4	47 ±6	17 ±3	**53** ±3

4.2 Comparison Study

Table 1 summarizes the benchmark results, demonstrating NEFD's ability to learn to make effective decisions toward distant goals under sparse feedback.

4.2.1 Maze environments

Maze environments require successive local decisions to lead to a distant destination across different agent embodiments (Park et al., 2025a). NEFD achieves the highest success rate across `Pointmaze`, `Antmaze`, and `Humanoidmaze` under both the `navigate` and `stitch` settings.

Navigate datasets. In the giant `Antmaze` and `Humanoidmaze` layouts, value-guided methods GCIVL, GCIQL, CRL, and QRL struggle to reach the goal. This is consistent with estimation errors obscuring small differences in local action values under distant goals (Park et al., 2023). HIQL and SAW provide local guidance through subgoals and perform better, but still lag substantially behind NEFD. NEFD also outperforms LAVL, the strongest baseline.

Stitch datasets. These datasets additionally require combining behaviors from short recorded trajectories to reach distant goals under sparse feedback (Park et al., 2025a). NEFD outperforms LAVL across all stitch datasets and maintains strong performance even in the giant layouts, where HIQL and SAW achieve near-zero success. These results demonstrate NEFD's effectiveness when reaching distant goals requires trajectory stitching.

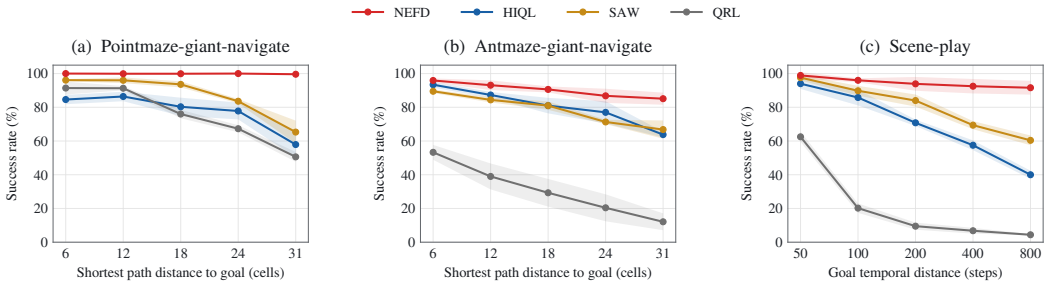


Figure 3: **Success rates across goal distances.** NEFD maintains high success rates as goals become more distant, while the baselines show sharper performance declines.

4.2.2 Manipulation environments

`Cube` and `Scene` require composing manipulation operations to reach a desired final configuration (Park et al., 2025a). In `Cube`, manipulating more blocks requires more picking and placing operations. NEFD achieves the highest mean success rate on all state-based manipulation datasets. HIQL and SAW struggle on `Cube-triple`, where LAVL, the strongest baseline, shows a sharp drop. NEFD's margin over LAVL widens from `Cube-single` to `Cube-double` and `Cube-triple`. The widening gap suggests that NEFD's dense teacher supervision for current decisions is particularly beneficial when reaching a distant final goal requires more manipulation operations.

4.2.3 Visual environments

Visual environments introduce the additional challenge of learning representations for control from images (Park et al., 2025a). NEFD matches or exceeds the strongest baseline in mean success rate on six of the seven visual datasets. For visual manipulation, the largest gain appears on `Visual-Cube-double`, followed by `Visual-Scene`, while the gain on `Visual-Cube-triple` is modest. NEFD remains competitive on `Visual-Cube-single` but falls slightly below HIQL and SAW. These results show that NEFD remains effective with visual observations, extending its benefits from state-based tasks to visual navigation and manipulation.

4.2.4 Performance across Goal Distances

To examine how performance changes as goals become farther away within a fixed environment, we compare NEFD with QRL, HIQL, and SAW on `Pointmaze-giant-navigate`, `Antmaze-giant-navigate`, and `Scene-play`. For the maze tasks, we randomly sample state–goal pairs and group them by shortest path distance on the maze grid. For `Scene-play`, we group pairs by goal temporal distance, defined as the number of recorded steps until the goal is first reached. Each distance bin contains 200 distinct pairs, and success rates are averaged over eight seeds. Further details on the distance bins and sampling procedure are provided in Appendix B.2.1.

Figure 3 shows that, across both maze tasks and `Scene-play`, the success rates of QRL, HIQL and SAW decline more sharply than those of NEFD as goals become more distant. NEFD's advantage therefore becomes more pronounced at greater goal distances.

4.3 Ablation Studies

To examine the effectiveness of NEFD's key components, we conduct ablation studies of teacher construction, action and representation self-distillation, and goal progress weighting on `Pointmaze-giant-navigate`, `Antmaze-giant-navigate`, and `Scene-play`. We further analyze the role of representation supervision under distant goals.

Role of Privileged Teacher To examine the role of privileged information in constructing an effective teacher, we compare three variants. No Distill removes teacher supervision, OPD-Large uses a larger teacher with the same state–action history and far goal as the student, and OPD-Privilege additionally provides the teacher with the corresponding near-future state. Both OPD variants use only uniformly weighted action distribution distillation.

Table 2: Effect of privileged information in OPD.

Dataset	No Distill	OPD-Large	OPD-Privilege	NEFD
Pointmaze-giant-navigate-v0	56 ±10	55 ±12	74 ±10	**100** ±0
Antmaze-giant-navigate-v0	6 ±3	6 ±2	62 ±1	**88** ±2
Scene-play-v0	50 ±6	54 ±3	70 ±3	**95** ±2

Table 3: Ablation results of action-level and representation-level distillation.

Dataset	QRL	HIQL	w/o $\mathcal{L}_{\text{act}}$	w/o $\mathcal{L}_{\text{rep}}$	NEFD
Pointmaze-giant-navigate-v0	68 ±7	46 ±9	94 ±2	88 ±3	**100** ±0
Antmaze-giant-navigate-v0	14 ±3	65 ±5	82 ±2	73 ±2	**88** ±2
Scene-play-v0	5 ±1	38 ±3	87 ±7	85 ±4	**95** ±2

Table 4: Effect of Goal progress weighting.

Dataset	QRL	HIQL	w/o act weight	w/o rep weight	NEFD
Pointmaze-giant-navigate-v0	68 ±7	46 ±9	97 ±2	87 ±3	**100** ±0
Antmaze-giant-navigate-v0	14 ±3	65 ±5	83 ±1	76 ±2	**88** ±2
Scene-play-v0	5 ±1	38 ±3	90 ±3	81 ±1	**95** ±2

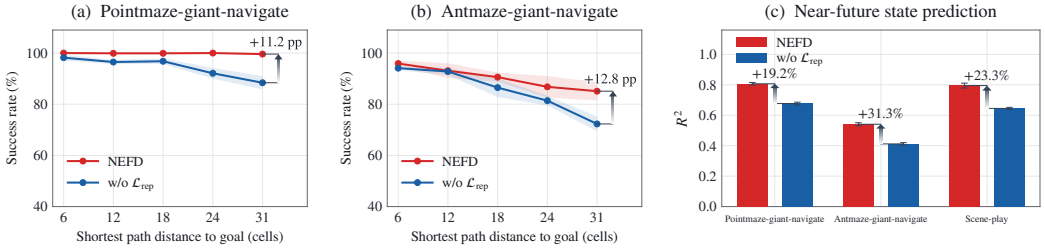


Figure 4: **Analysis of Representation Self-Distillation.** (a) Performance on Pointmaze-giant-navigate across goal distances. (b) Performance on Antmaze-giant-navigate across goal distances. (c) R^2 of near-future state prediction from student representations.

The results in Table 2 show that OPD-Large performs similarly to No Distill, whereas OPD-Privilege improves performance across all three tasks. This contrast indicates that simply introducing a larger teacher is insufficient, while privileged information is critical for effective teacher supervision. These results support our privileged teacher construction based on near-future states.

Contribution of Action and Representation Self-Distillation. To quantify the contribution of the two distillation objectives, we remove either $\mathcal{L}_{\text{act}}$ or $\mathcal{L}_{\text{rep}}$ while keeping all other components unchanged. Table 3 shows that removing either objective consistently reduces performance. Removing $\mathcal{L}_{\text{rep}}$ causes a larger drop even when action self-distillation is retained, indicating that representation supervision provides additional guidance beyond action predictions. Using both objectives achieves the best performance, demonstrating their complementary roles.

Role of Representation Supervision under Distant Goals. We examine whether intermediate representation targets help the student learn to make current decisions from its own sequence and distant goal. We compare NEFD with its variant without $\mathcal{L}_{\text{rep}}$ at different goal distances within each maze layout, retaining action self-distillation in both. Figure 4(a,b) shows that removing representation supervision causes larger losses in success rate for more distant goals on both maze tasks. Thus, even with action targets available, directly guiding the student's representation learning provides greater benefits when the desired goal is farther away.

To understand the benefit of representation supervision under distant goals, we evaluate how well the student representations predict the recorded near-future state s_{t+H}. Prediction quality is measured by R^2, with higher values indicating better prediction. Figure 4(c) shows consistently higher R^2

for NEFD, suggesting that representation self-distillation helps the student learn features that better capture these outcomes. Experimental details for both analyses are provided in Appendix B.3.7.

Importance of Goal Progress Weighting. To assess goal-progress weighting, we set the exponential weight $e^{\beta\widehat{\Delta}_i}$ to 1 for either action self-distillation (w/o act weight) or representation self-distillation (w/o rep weight), while keeping all other components unchanged. The results in Table 4 indicate that uniform weighting at either level consistently reduces performance across all three tasks. The larger degradation from removing w^{rep} suggests that estimated goal progress is particularly important for modulating the strength of representation supervision. Applying goal-progress weighting to both objectives yields the strongest performance, indicating that teacher guidance associated with greater estimated progress should receive stronger supervision.

5 Related Work

5.1 Offline Goal-Conditioned Reinforcement Learning

Offline GCRL learns policies for reaching diverse goals from fixed datasets (Levine et al., 2020; Park et al., 2025a; Andrychowicz et al., 2017; Ghosh et al., 2021). Value-guided methods derive policy supervision from goal-conditioned value estimates (Yang et al., 2022; Ma et al., 2022; Hejna et al., 2023), using expectile regression (Park et al., 2023; Kostrikov et al., 2022), contrastive learning (Eysenbach et al., 2022), or quasimetric models (Wang et al., 2023; Myers et al., 2025). Other work uses goal embeddings from learned value functions as policy inputs (Park et al., 2026). However, estimation errors can obscure small value differences between actions under distant goals, weakening supervision for current decisions (Park et al., 2023; Zhou & Kao, 2025).

Subgoal-based methods provide local guidance for policy learning. HIQL (Park et al., 2023) uses a high-level policy to select subgoals for low-level control, while OTA (Ahn et al., 2025) and LAVL (Kang et al., 2026) improve value learning within this hierarchical framework. However, the effectiveness of these methods depends on the quality of the generated subgoals. Other approaches use subgoal-conditioned policies to provide action targets for distant goals, including RIS (Chane-Sane et al., 2021) and PIG (Kim et al., 2023) in online GCRL and SAW (Zhou & Kao, 2025) in offline GCRL. Although these targets provide explicit action guidance, learning to predict actions from the current state and distant goal can remain difficult over long horizons (Park et al., 2025b).

NEFD instead retains the same distant goal for teacher and student, while providing the teacher with the near-future state as additional privileged information. Within a shared actor, action and representation self-distillation transfer the resulting guidance to the student at each decision position. Estimated goal progress determines the strength of both forms of supervision.

5.2 On-Policy Distillation

On-policy distillation provides dense supervision through teacher predictions on student-generated sequences (Agarwal et al., 2024; Gu et al., 2024), with subsequent work refining sampling and optimization (Xu et al., 2025; Jin et al., 2026; Cideron et al., 2025; Wan et al., 2026; Yoon et al., 2026). Related context and self-distillation methods form teacher guidance using information unavailable to the student (Lopez-Paz et al., 2016; Ye et al., 2026; Zhao et al., 2026; Penaloza et al., 2026; Hübotter et al., 2026; Yang et al., 2026), sometimes using a shared model under different input conditions (Zhao et al., 2026; Penaloza et al., 2026; Liu et al., 2026).

6 Conclusion

We propose NEFD, a Near-to-Far Self-Distillation method that provides dense teacher supervision for current decisions under distant goals in offline GCRL. The teacher and student share an actor and the same distant goal, with near future states provided only to the teacher as privileged information. Action and representation self-distillation transfer the teacher's guidance to the student, with both objectives weighted by estimated goal progress. Experiments on 29 OGBench datasets demonstrate NEFD's effectiveness across state-based and visual environments. The results show that NEFD strengthens supervision for current decisions, enabling the student to act effectively toward distant goals in tasks with sparse feedback.

AI USE STATEMENT

In this work, we used generative AI tools to polish the language of the manuscript and assist with code development. All AI-assisted text and code were reviewed and verified by the authors, who take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the NEFD training procedure in Appendix A. All experiments are conducted on the publicly available OGBench benchmark (Park et al., 2025a). Detailed evaluation protocols, training configurations, hyperparameter settings, and baseline implementation details are provided in Appendix B.

ETHICS STATEMENT

This work focuses on developing and evaluating offline goal-conditioned reinforcement learning methods in simulated environments. Our experiments use publicly available benchmark datasets and do not involve human subjects, personally identifiable information, or sensitive personal data. The proposed method is evaluated solely on simulated navigation and robotic manipulation tasks. We do not identify specific ethical concerns related to privacy, fairness, or human-subject research arising from the experiments conducted in this work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024.

Hongjoon Ahn, Heewoong Choi, Jisu Han, and Taesup Moon. Option-aware temporally abstracted value for offline goal-conditioned reinforcement learning. In *NeurIPS*, 2025.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NeurIPS*, 2017.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *ICML*, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *NeurIPS*, 2021.

Geoffrey Cideron, Andrea Agostinelli, Johan Ferret, Sertan Girgin, Romuald Elie, Olivier Bachem, Sarah Perrin, and Alexandre Ramé. Diversity-rewarded cfg distillation. In *ICLR*, 2025.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *NeurIPS*, 2022.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *ICLR*, 2021.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *ICLR*, 2024.

Joey Hejna, Jensen Gao, and Dorsa Sadigh. Distance weighted supervised learning for offline interaction data. In *ICML*, 2023.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *ICML*, 2026.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *NeurIPS*, 2021.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *ICML*, 2026.

Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, 1993.

Hyungkyu Kang, Byeongchan Kim, and Min hwan Oh. Latent representation alignment for offline goal-conditioned reinforcement learning. In *ICML*, 2026.

Kaiqiang Ke, Qian Lin, Zongkai Liu, Shenghong He, and Chao Yu. Conservative offline goal-conditioned implicit v-learning. In *ICML*, 2025.

Junsu Kim, Younggyo Seo, Sungsoo Ahn, Kyunghwan Son, and Jinwoo Shin. Imitating graph-based planning with goal-conditioned policies. In *ICLR*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *ICLR*, 2022.

Xing Lei, Jincheng Wang, Xuetao Zhang, and Donglin Wang. QHyer: Q-conditioned hybrid attention-mamba transformer for offline goal-conditioned RL. In *ICML*, 2026.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. In *arXiv*, 2020.

Haoran Liu, Yuwei Zhang, Xiyao Li, Bohan Lyu, and Jingbo Shang. Hero: Hindsight-enhanced reflection from environment observations for agentic self-distillation. In *arXiv*, 2026.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Naumovich Vapnik. Unifying distillation and privileged information. In *ICLR*, 2016.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. Offline goal-conditioned reinforcement learning via f-advantage regression. In *NeurIPS*, 2022.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. In *NeurIPS*, 2025.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *NeurIPS*, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *ICLR*, 2025a.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. In *NeurIPS*, 2025b.

Seohong Park, Deepinder Mann, and Sergey Levine. Dual goal representations. In *ICLR*, 2026.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *ICML*, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. In *arXiv*, 2019.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *ICML*, 2015.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. In *arXiv*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Zhenglin Wan, Jingxuan Wu, Xingrui Yu, Chubin Zhang, Mingcong Lei, Bo An, Ivor Tsang, and Yang You. Adversarial dual on-policy distillation from expressive teacher. In *ICML*, 2026.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *ICML*, 2023.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *ICLR*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-Distilled RLVR. In *arXiv*, 2026.

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline RL. In *ICLR*, 2022.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. In *arXiv*, 2026.

Hee Suk Yoon, Eunseop Yoon, Jaehyun Jang, SooHwan Eom, Ji Woo Hong, Mark A. Hasegawa-Johnson, Qi Dai, Chong Luo, and Chang D. Yoo. Decomposed on-policy distillation for vision-language reasoning: Steering gradients for visual grounding. In *ICML*, 2026.

Zilai Zeng, Ce Zhang, Shijie Wang, and Chen Sun. Goal-conditioned predictive coding for offline reinforcement learning. In *NeurIPS*, 2023.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *ICML*, 2026.

John Luoyu Zhou and Jonathan Kao. Flattening hierarchies with policy bootstrapping. In *NeurIPS*, 2025.

Appendix

A Algorithm

A.1 Goal Sampling

We use the goal sampling scheme of OGBench (Park et al., 2025a), which samples goals from a mixture of the following four conditional distributions:

- $p^{\mathcal{D}}_{\mathrm{cur}}(g \mid s)$ is the Dirac delta distribution at the current state s, i.e., $g = s$.
- $p^{\mathcal{D}}_{\mathrm{traj}}(g \mid s)$ is the uniform in-trajectory distribution. When $s = s_t$ in a trajectory $(s_0, s_1, \ldots, s_T)$, we sample an index j uniformly from the future states in the same trajectory and set $g = s_j$.
- $p^{\mathcal{D}}_{\mathrm{geom}}(g \mid s)$ is the geometric in-trajectory distribution. When $s = s_t$, we sample a positive temporal offset $k \sim \mathrm{Geom}(1-\gamma)$ and set $g = s_{\min(t+k,T)}$, where γ is the discount factor.
- $p^{\mathcal{D}}_{\mathrm{rand}}(g \mid s)$ is the uniform distribution over states in the dataset $\mathcal{D}$. This distribution is independent of the current state s.

We denote the mixtures used for actor learning and value TD learning by $p^{\mathcal{D}}_{\mathrm{actor}}(g \mid s)$ and $p^{\mathcal{D}}_{\mathrm{mixed}}(g \mid s)$, respectively. Both are mixtures of the four distributions defined above, but use different mixture coefficients. Actor and value goals are sampled separately from their respective distributions. The mixture coefficients are specified in the order $(p_{\mathrm{cur}}, p_{\mathrm{traj}}, p_{\mathrm{geom}}, p_{\mathrm{rand}})$ and reported in Table 6.

A.2 Goal-Conditioned Value Learning

We adopt the value parameterization of LAVL (Kang et al., 2026) and follow its official implementation for value learning. The learned values are used to compute the behavior advantages for policy learning and the goal progress for distillation weighting in Section 3.3. The Latent Alignment Network (LAN) parameterizes the goal-conditioned value function as

$$V_\phi(s, g) = -\left\|\varphi_S(s) - \varphi_G(g)\right\|_2, \tag{A.1}$$

where $\varphi_S : \mathcal{S} \to \mathbb{R}^d$ and $\varphi_G : \mathcal{G} \to \mathbb{R}^d$ are separate state and goal representation networks, with parameters collectively denoted by ϕ.

Following the formulation in LAVL, the value function is trained using the expectile temporal-difference (TD) objective

$$\mathcal{L}_{\mathrm{TD}}(V_\phi) = \mathbb{E}_{(s,s')\sim\mathcal{D}, g\sim p^{\mathcal{D}}_{\mathrm{mixed}}(\cdot|s)}\left[\ell^\kappa_2\left(r(s,g) + \gamma\widetilde{V}(s',g) - V_\phi(s,g)\right)\right], \tag{A.2}$$

where $\ell^\kappa_2(x) = |\kappa - \mathbf{1}\{x < 0\}|x^2$ is the expectile loss with parameter κ, and $\widetilde{V}$ denotes the target value function. Here, $r(s,g)$ is the goal-conditioned reward, γ is the discount factor, and $p^{\mathcal{D}}_{\mathrm{mixed}}(\cdot \mid s)$ is the value-goal mixture defined in Appendix A.1.

To regularize local value variation, LAVL augments the TD objective with

$$\mathcal{L}_{\mathrm{Reg}}(V_\phi) = \mathbb{E}_{(s,s')\sim\mathcal{D}, g\sim p^{\mathcal{D}}_{\mathrm{rand}}(\cdot|s)}\left[\left(\left(V_\phi(s,g) - V_\phi(s',g)\right)^2 - \delta^2\right)_+\right], \tag{A.3}$$

where $(x)_+ = \max(x, 0)$ and $p^{\mathcal{D}}_{\mathrm{rand}}(\cdot \mid s)$ is the random-goal distribution defined in Appendix A.1. Goals for this regularizer are sampled separately from the TD goals, with the same goal used for both states in each transition. This term penalizes value differences between consecutive states whose magnitude exceeds δ. In the original formulation, the threshold is $\delta = 1 + (1-\gamma)|\bar{V}|$, where $\bar{V}$ is the batch mean of the value estimates. The complete value objective is

$$\mathcal{L}_V = \mathcal{L}_{\mathrm{TD}}(V_\phi) + w_c\mathcal{L}_{\mathrm{Reg}}(V_\phi), \tag{A.4}$$

where $w_c \geq 0$ controls the continuity regularization. Following the official LAVL implementation, the expectile weight is determined by the sign of the TD residual computed with the target value network, while the squared regression error uses the online value estimate. For the continuity regularizer, we approximate $(x)_+$ by $0.1\ \mathrm{softplus}(10x)$. The value statistic $\bar{V}$ used in the threshold $\delta = 1 + (1-\gamma)|\bar{V}|$ is maintained as an exponential moving average of the minibatch mean values from TD learning, rather than being recomputed from the current minibatch.

A.3 Representation Regularization

We apply variance and covariance regularization (Bardes et al., 2022) to the student projections $z_i^{\mathrm{S}} = P(h_i^{\mathrm{S}})$ defined in Section 3.2, discouraging representation collapse and reducing redundancy across projection dimensions. We collect these projections over all valid decision positions in a minibatch, excluding padding, and let N denote the total number of valid positions. We reindex them as $i = 1, \ldots, N$, where $z_i^{\mathrm{S}} \in \mathbb{R}^{d_p}$ and d_p is the projection dimension.

The mean and covariance matrix are

$$\mu = \frac{1}{N}\sum_{i=1}^{N} z_i^{\mathrm{S}}, \qquad C = \frac{1}{N}\sum_{i=1}^{N}(z_i^{\mathrm{S}} - \mu)(z_i^{\mathrm{S}} - \mu)^{\top}. \tag{A.5}$$

Using the diagonal entries of C as the per-dimension variances, we define

$$\begin{aligned} \mathcal{L}_{\mathrm{var}} &= \frac{1}{d_p}\sum_{j=1}^{d_p} \max\left(0,\, 1 - \sqrt{C_{jj} + \epsilon}\right), \\ \mathcal{L}_{\mathrm{cov}} &= \frac{1}{d_p}\sum_{j=1}^{d_p}\sum_{\substack{\ell=1 \\ \ell \neq j}}^{d_p} C_{j\ell}^2, \end{aligned} \tag{A.6}$$

where $\epsilon = 10^{-4}$ ensures numerical stability. The variance term penalizes per-dimension standard deviations below one, while the covariance term penalizes off-diagonal covariance. Both regularizers are applied only to the student projections and treat all valid decision positions uniformly, without goal-progress weighting.

A.4 Training Procedure

Algorithm 1 summarizes the complete training procedure of NEFD. The Teacher and Student share the same actor parameters and are evaluated on the same length-K state–action sequence with a goal g fixed across all valid decision positions. They differ only in their conditioning information: the Teacher additionally receives the corresponding near-future state s_{i+H}, while the Student is conditioned only on g. The shared actor is trained under both conditions with the behavior objective, after which the Teacher provides action and representation targets for self-distillation. Both distillation objectives are weighted according to the estimated goal progress associated with the near-future state.

For actor learning, the goal is sampled either as a future state $g = s_j$ from the same trajectory or as a random state from the dataset. For a future goal, the Teacher's privileged state at decision position i is $s_{\min(i+H,j)}$; for a random dataset goal, it is $s_{\min(i+H,T)}$, where T denotes the final state index of the trajectory. For notational consistency with the main text, we denote both cases by s_{i+H} below. The same privileged state is used to construct the Teacher condition, compute the Teacher behavior advantage, and estimate goal progress.

The goal-conditioned value function is optimized with $\mathcal{L}_V$ in Appendix A.2 and provides the value estimates used to compute the behavior advantages and goal progress. The student projections are additionally regularized by $\mathcal{L}_{\mathrm{var}}$ and $\mathcal{L}_{\mathrm{cov}}$ as described in Appendix A.3.

For goal-progress weighting, we standardize the progress estimates Δ_i jointly over all valid decision positions in each minibatch, excluding padded positions. The resulting normalized progress $\widehat{\Delta}_i$ is used to construct the exponential weight $e^{\beta\widehat{\Delta}_i}$. This normalization makes the weighting depend on the relative progress of the sampled outcomes within a minibatch, reducing sensitivity to the absolute scale of the learned value function. Since the action and representation targets at each decision position are constructed using the same privileged near-future state, the same progress weight is applied to both distillation objectives.

During actor optimization, the Teacher action distributions and projected representations are treated as fixed targets through stop-gradient, as defined in Section 3.2. The value estimates used to compute the behavior advantages and goal-progress weights are also treated as fixed quantities for the actor update; gradients from $\mathcal{L}_{\mathrm{actor}}$ are not propagated into the value function. Accordingly, the shared

Algorithm 1 Near-to-Far Self-Distillation (NEFD)

Require: Offline dataset $\mathcal{D}$, sequence length K, near future horizon H

Initialize shared actor π_θ (including E and P), value function V_ϕ, and target networks
for each training iteration **do**
`// Teacher and Student Policy Learning`
  Sample a length-K trajectory segment and a distant goal $g \sim p^{\mathcal{D}}_{\text{actor}}$
  Use s_{i+H} as the Teacher's privileged state at each decision position i
  $e^{\mathrm{T}}_i \leftarrow E(g, s_{i+H}), \quad e^{\mathrm{S}}_i \leftarrow E(g)$
  Form $\mathcal{H}^{\mathrm{T}}_t$ and $\mathcal{H}^{\mathrm{S}}_t$ using Eq. equation 3
  Teacher: $\{\pi_\theta(\cdot \mid \mathcal{H}^{\mathrm{T}}_{t,\le i}), h^{\mathrm{T}}_i\}_i \leftarrow \mathrm{Forward}_\theta(\mathcal{H}^{\mathrm{T}}_t)$
  Student: $\{\pi_\theta(\cdot \mid \mathcal{H}^{\mathrm{S}}_{t,\le i}), h^{\mathrm{S}}_i\}_i \leftarrow \mathrm{Forward}_\theta(\mathcal{H}^{\mathrm{S}}_t)$
  Evaluate $\mathcal{L}_{\text{beh}}$ using Eq. equation 4

`// Near-to-Far Self-Distillation`
  $\mathcal{L}^{\text{act}}_i \leftarrow D_{\mathrm{KL}}\left(\mathrm{sg}\left[\pi_\theta(\cdot \mid \mathcal{H}^{\mathrm{T}}_{t,\le i})\right] \,\|\, \pi_\theta(\cdot \mid \mathcal{H}^{\mathrm{S}}_{t,\le i})\right)$
  $z^v_i \leftarrow P(h^v_i), \quad v \in \{\mathrm{T}, \mathrm{S}\}$
  $\mathcal{L}^{\text{rep}}_i \leftarrow \left\| z^{\mathrm{S}}_i - \mathrm{sg}[z^{\mathrm{T}}_i] \right\|_2^2$
  Evaluate $\mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{cov}}$ as in Appendix A.3

`// Goal Progress Weighting`
  $\Delta_i \leftarrow V_\phi(s_{i+H}, g) - V_\phi(s_i, g)$
  $\widehat{\Delta}_i \leftarrow \mathrm{Standardize}(\Delta_i)$
  $\mathcal{L}_{\text{distill}} \leftarrow \frac{1}{K} \sum_{i=t-K+1}^{t} e^{\beta \widehat{\Delta}_i} \left(\mathcal{L}^{\text{act}}_i + \lambda_{\text{rep}} \mathcal{L}^{\text{rep}}_i\right)$

`// Value Learning and Update`
  Evaluate $\mathcal{L}_V$ according to Appendix A.2
  $\mathcal{L}_{\text{actor}} \leftarrow \mathcal{L}_{\text{beh}} + \mathcal{L}_{\text{distill}} + \lambda_{\text{var}} \mathcal{L}_{\text{var}} + \lambda_{\text{cov}} \mathcal{L}_{\text{cov}}$
  Update θ with $\mathcal{L}_{\text{actor}}$ and ϕ with $\mathcal{L}_V$
  Update the target value network
end for
return the Student policy conditioned only on g

actor and projection head are optimized with $\mathcal{L}_{\text{actor}}$, while the value function is optimized separately with $\mathcal{L}_V$. The target value network is updated after the value update.

At test time, the privileged near-future state is no longer available. We therefore discard the Teacher condition and execute the shared actor under the Student condition, using only the desired goal and the observed state–action history. No privileged state or additional subgoal prediction is required during deployment.

B Experiment Details

B.1 Experimental Setup

B.1.1 Environments and Datasets

We evaluate NEFD on 29 datasets from OGBench (Park et al., 2025a), including 22 datasets with state observations and seven with image observations. Following the organization in the main text, we group them into maze, manipulation, and visual environments. All policy learning uses the provided offline datasets.

Maze environments. The maze suite includes `Pointmaze`, `Antmaze`, and `Humanoidmaze`, which involve controlling a point mass moving in two dimensions, a quadrupedal robot, and a humanoid robot, respectively. The objective is to reach a specified location while navigating around maze walls. These environments require both navigation over long distances and local control appropriate to each agent's dynamics. We consider the `medium`, `large`, and `giant` layouts, with larger layouts requiring longer paths and more extended sequences of decisions.

For each agent and layout, we use both the `navigate` and `stitch` datasets, yielding 18 datasets in total. The `navigate` datasets are collected by noisy expert policies that repeatedly move toward randomly sampled goals. The `stitch` datasets contain short trajectories covering distances of at most four maze cell units. Reaching distant evaluation goals therefore requires combining behaviors from multiple recorded trajectories.

Manipulation environments. The manipulation suite includes `Cube-single`, `Cube-double`, `Cube-triple`, and `Scene`. The `Cube` environments require a robot arm to arrange one, two, or three cubes into target configurations by picking them up and placing them at desired locations. Depending on the task, the agent must relocate, stack, or exchange the positions of cubes. Manipulating multiple cubes requires coordinating successive grasping, transport, and placement operations.

`Scene` involves a cube, a drawer, a window, and buttons that control the locks of the drawer and window. Tasks require combining object interactions in a suitable order. For example, placing the cube inside a locked drawer requires unlocking and opening the drawer before inserting the cube and closing it again. We use the `play` datasets for all four environments. These datasets are collected by scripted policies that repeatedly manipulate randomly selected objects, providing diverse behaviors from which the agent must learn to achieve specified configurations.

Visual environments. The visual suite contains seven datasets: `Visual-Antmaze` with `medium`, `large`, and `giant` layouts under the `navigate` setting, together with `Visual-Cube-single`, `Visual-Cube-double`, `Visual-Cube-triple`, and `Visual-Scene` under the `play` setting. These environments retain the navigation and manipulation objectives described above, while providing RGB observations and goal images at a resolution of 64×64. The policy must infer information needed for control, such as the agent's location and object configurations, directly from images. These tasks therefore evaluate representation learning alongside navigation over long horizons and sequential manipulation.

B.1.2 Baselines

We compare NEFD with seven representative offline GCRL methods. Following the main text, we organize these methods into value-guided policy learning and subgoal-based policy learning according to their main approach to improving policy supervision.

Value-guided policy learning. These methods use goal-conditioned values, advantages, or temporal distance estimates to guide policy learning. GCIVL and GCIQL build on implicit Q-learning, using expectile regression to learn value functions from offline data. GCIVL learns only a state value function, whereas GCIQL learns both state and action value functions. CRL and QRL learn goal-conditioned estimates through contrastive learning and quasimetric modeling, respectively.

Subgoal-based policy learning. These methods use intermediate goals to provide local guidance for reaching distant goals. HIQL and LAVL decompose decision making into high-level subgoal selection and low-level action control. HIQL learns a high-level policy to predict latent subgoals

and a low-level policy to select actions toward them. LAVL adopts a similar hierarchical policy structure while additionally using LAN and local continuity regularization to improve value learning. SAW instead uses policy bootstrapping, where action distributions predicted by a separately trained subgoal-conditioned policy supervise a flat policy conditioned on the desired goal. The resulting policy does not require subgoal generation at deployment.

Table 5 summarizes the baseline algorithms and provides links to their implementations.

Table 5: Baseline algorithms and their main mechanisms.

No.	Algorithm	Brief Description
1	GCIVL[1] (Kostrikov et al., 2022; Park et al., 2023)	Learns a goal-conditioned state value function and extracts a policy through advantage-weighted regression.
2	GCIQL[2] (Kostrikov et al., 2022)	Extends implicit Q-learning to learn goal-conditioned state and action values for policy optimization.
3	CRL[3] (Eysenbach et al., 2022)	Uses contrastive learning to estimate goal-conditioned values that guide policy learning.
4	QRL[4] (Wang et al., 2023)	Learns goal-reaching distances through a quasimetric representation and uses them to guide policy learning.
5	HIQL[5] (Park et al., 2023)	Uses a high-level policy to generate latent subgoals and a low-level policy to select actions toward them.
6	SAW[6] (Zhou & Kao, 2025)	Trains a flat policy through subgoal advantage-weighted policy bootstrapping from a separately trained subgoal-conditioned policy
7	LAVL[7] (Kang et al., 2026)	Uses hierarchical subgoal selection and low-level action control, together with LAN and local continuity regularization for improved value learning.

Sources of baseline results. Results for GCIVL, GCIQL, CRL, QRL, and HIQL are taken from OGBench (Park et al., 2025a). SAW results and LAVL results for environments with state observations are taken from their respective papers (Zhou & Kao, 2025; Kang et al., 2026). For the seven visual datasets, we reproduce LAVL using its official implementation in the comparison tables.

B.1.3 Evaluation Protocol

We follow the evaluation protocol of OGBench (Park et al., 2025a). We train NEFD for 1M gradient steps on datasets with state observations and 500K steps on datasets with image observations. Every 100K steps, we evaluate the policy on the five predefined evaluation tasks, using 50 rollouts per task. Performance is measured by the binary success rate, expressed as a percentage, using the success criteria and episode length limits provided by OGBench.

For each seed, we average success rates across the five tasks and the last three evaluation checkpoints: 800K, 900K, and 1M steps for state observations, and 300K, 400K, and 500K steps for image observations. Each resulting score therefore aggregates 750 rollouts. We report the mean and standard deviation across eight random seeds for state observations and four random seeds for image observations.

B.1.4 Training Configuration and Hyperparameters

NEFD adopts the Latent Alignment Network (LAN) and local continuity regularization from LAVL (Kang et al., 2026) for goal-conditioned value learning. For each environment, we follow the corresponding LAVL configuration for the value learning component, including the expectile κ, discount factor γ, and local continuity coefficient w_c. Unless otherwise specified, the remaining value network settings also follow LAVL.

[1]GCIVL implementation: https://github.com/seohongpark/HIQL.
[2]GCIQL implementation: https://github.com/seohongpark/ogbench.
[3]https://github.com/google-research/google-research/tree/master/contrastive_rl
[4]https://github.com/quasimetric-learning/quasimetric-rl
[5]https://github.com/seohongpark/HIQL
[6]https://github.com/johnlyzhou/saw
[7]https://github.com/oh-lab/LAVL

Table 6: Common architecture and training hyperparameters.

Hyperparameter	Value
optimizer	Adam
learning rate	0.0003
batch size	256
transformer blocks	3
hidden size	128
attention head	4
α_T	3
α_S	3
β	3
λ_{rep}	8
λ_{var}	1
λ_{cov}	1
projection dimension	32
visual encoder	IMPALA-small (visual only)
actor $(p_{\text{cur}}, p_{\text{traj}}, p_{\text{geom}}, p_{\text{rand}})$ ratio	(0,1,0,0)(default), (0,0.5,0,0.5)(scene and stitch)
value $(p_{\text{cur}}, p_{\text{traj}}, p_{\text{geom}}, p_{\text{rand}})$ ratio	(0.2,0,0.5,0.3)(default), (0.2,0.3,0,0.5)(stitch)

The Teacher and Student share a goal-conditioned causal Transformer actor (Chen et al., 2021) and are jointly optimized using the policy learning and self-distillation objectives described in the main text. Most architecture, optimization, and distillation hyperparameters are shared across all datasets and are summarized in Table 6. The environment-specific value learning and temporal settings are reported in Table 7. Here, K denotes the Transformer sequence length, and H denotes the near-future horizon.

B.1.5 Computing Resources

The experimental work was carried out on a high-performance computing cluster that includes several NVIDIA GeForce RTX 4090 GPUs to supply the required computing power. The cluster is also equipped with Intel(R) Xeon(R) Gold 6348 CPUs, each operating at a frequency of 2.60GHz.

B.2 Additional Comparison Study

B.2.1 Experimental Setup for Evaluation across Goal Distances

We further evaluate NEFD, QRL, HIQL, and SAW at different goal distances on `Pointmaze-giant-navigate`, `Antmaze-giant-navigate`, and `Scene-play`. Rather than changing the environment layout, we construct evaluation state–goal pairs with different distances within the same environment and report performance separately for each distance range.

For `Pointmaze-giant-navigate` and `Antmaze-giant-navigate`, we randomly sample candidate initial state–goal pairs and map their positions to the corresponding cells of the maze grid. We then compute the shortest traversable path between the two cells, so that the distance reflects the maze structure rather than their Euclidean separation. Candidate pairs are assigned to five ranges: 1–6, 7–12, 13–18, 19–24, and 25–31 cells. We continue sampling until each range contains 200 distinct state–goal pairs.

For `Scene-play`, spatial distance does not directly reflect the difficulty of the sequential manipulation task. We therefore use goal temporal distance. We sample an initial state and a future goal from the same recorded trajectory and define their distance as the number of transitions from the initial state until that goal is first reached in the trajectory. The sampled pairs are grouped into 1–50, 51–100, 101–200, 201–400, and 401–800 steps, with 200 distinct pairs in each range.

This procedure yields 1,000 evaluation pairs for each environment. We use the default OGBench episode limits and success criteria and evaluate each method separately on every distance range. Success rates are averaged over the 200 pairs in each range and eight random seeds, and the shaded

Table 7: Environment-specific hyperparameters of NEFD.

Environment	Dataset Type	Expectile κ	Discount γ	w_c	H	K
Pointmaze	medium-navigate-v0	0.9	0.999	10	25	20
	large-navigate-v0	0.9	0.999	10	25	20
	giant-navigate-v0	0.9	0.999	10	25	20
	medium-stitch-v0	0.9	0.999	10	25	20
	large-stitch-v0	0.9	0.999	10	25	20
	giant-stitch-v0	0.9	0.999	10	25	20
Antmaze	medium-navigate-v0	0.9	0.999	10	25	20
	large-navigate-v0	0.9	0.999	10	25	20
	giant-navigate-v0	0.9	0.999	10	25	20
	medium-stitch-v0	0.9	0.999	10	25	20
	large-stitch-v0	0.9	0.999	10	25	20
	giant-stitch-v0	0.9	0.999	10	25	20
Humanoidmaze	medium-navigate-v0	0.9	0.999	0	100	20
	large-navigate-v0	0.9	0.999	0	100	20
	giant-navigate-v0	0.9	0.999	0	100	20
	medium-stitch-v0	0.9	0.999	0	100	20
	large-stitch-v0	0.9	0.999	0	100	20
	giant-stitch-v0	0.9	0.999	0	100	20
Cube	single-play-v0	0.7	0.99	0	10	5
	double-play-v0	0.7	0.99	0	10	5
	triple-play-v0	0.7	0.99	0	10	5
Scene	play-v0	0.7	0.998	10	10	5
Visual-Antmaze	medium-navigate-v0	0.9	0.999	0	25	5
	large-navigate-v0	0.9	0.999	0	25	5
	giant-navigate-v0	0.9	0.999	0	25	5
Visual-Cube	single-play-v0	0.7	0.99	0	10	5
	double-play-v0	0.7	0.99	0	10	5
	triple-play-v0	0.7	0.99	0	10	5
Visual-Scene	play-v0	0.7	0.998	0	10	5

regions show standard deviation across seeds. The horizontal axis uses the upper bound of each range; for example, 31 corresponds to state–goal pairs with shortest path distances between 25 and 31 cells.

B.3 Additional Ablation Studies

B.3.1 Effect of the Value Function

Our method requires only scalar goal-conditioned value estimates to compute behavior and distillation weights, allowing different value parameterizations to be used. The quality of these weights depends on how reliably value differences reflect progress toward the corresponding goals. To examine the sensitivity to the choice of value function, we compare three variants using an MLP value function, a quasimetric value function based on QRL (Wang et al., 2023), and the default LAN value function (Kang et al., 2026). The actor architecture and distillation objectives are kept unchanged across all variants.

As shown in Table 8, LAN achieves the best performance, while the MLP and QRL variants exhibit only modest performance drops. These results suggest that our method is relatively insensitive to the choice of value function on the evaluated tasks.

Table 8: Ablation on the goal-conditioned value-function parameterization.

Dataset	QRL	HIQL	NEFD (MLP Value)	NEFD (QRL Value)	NEFD (LAN Value)
Pointmaze-giant-navigate-v0	68 ±7	46 ±9	98 ±2	97 ±1	**100** ±0
Antmaze-giant-navigate-v0	14 ±3	65 ±5	84 ±1	85 ±2	**88** ±2
Scene-play-v0	5 ±1	38 ±3	90 ±3	88 ±2	**95** ±2

B.3.2 Sensitivity to Privileged Information

To examine how the accuracy of privileged information affects student learning, we add Gaussian noise to the positional components of the near future states observed by the teacher during training. We evaluate three noise levels with standard deviations $\sigma = 0.05$, 0.10, and 0.20. All other training and evaluation settings remain unchanged.

Table 9 shows that NEFD retains performance close to the unperturbed setting under low and medium noise across all three tasks. High noise causes larger performance losses on Pointmaze and Scene-play, while Antmaze is less affected. Even at the highest noise level, NEFD outperforms QRL and HIQL on all three tasks. These results demonstrate that NEFD tolerates moderate inaccuracies in privileged information, while substantial corruption can reduce the benefit of the teacher's local reference.

Table 9: Sensitivity to Gaussian noise in privileged near future states.

Dataset	QRL	HIQL	$\sigma = 0.05$	$\sigma = 0.10$	$\sigma = 0.20$	NEFD
Pointmaze-giant-navigate-v0	68 ±7	46 ±9	98 ±4	99 ±1	78 ±1	**100** ±0
Antmaze-giant-navigate-v0	14 ±3	65 ±5	84 ±1	82 ±2	81 ±3	**88** ±2
Scene-play-v0	5 ±1	38 ±3	94 ±3	92 ±2	69 ±5	**95** ±2

B.3.3 Importance of the Privileged Information Horizon (H)

The horizon H determines how far into the future the privileged state provided to the Teacher is selected. Inspired by the temporal abstraction used for local subgoals in hierarchical GCRL methods (Park et al., 2023; Ahn et al., 2025), we use a common temporal scale within each task family: approximately 2.5 seconds for navigation tasks and 0.5 seconds for manipulation tasks requiring fine control. To examine the sensitivity to this choice, we additionally evaluate shorter and longer horizons, corresponding to $0.8H$ and $1.2H$, while keeping all other components unchanged. NEFD denotes the default horizon H.

As shown in Table 10, NEFD remains effective when the privileged information horizon is varied to $0.8H$ or $1.2H$, while the default horizon achieves the best or tied-best performance across all three tasks. These results indicate that NEFD is not highly sensitive to moderate changes in H and support using approximately 2.5 seconds for navigation and 0.5 seconds for manipulation tasks requiring fine control as shared temporal scales, without environment-specific horizon tuning.

Table 10: Effect of the privileged information horizon H.

Dataset	QRL	HIQL	$0.8H$	$1.2H$	NEFD
Pointmaze-giant-navigate-v0	68 ±7	46 ±9	99 ±1	100 ±0	**100** ±0
Antmaze-giant-navigate-v0	14 ±3	65 ±5	81 ±1	84 ±1	**88** ±2
Scene-play-v0	5 ±1	38 ±3	95 ±2	93 ±3	**95** ±2

B.3.4 Effect of Transformer Sequence Length

We examine the effect of Transformer sequence length on Pointmaze-giant-navigate-v0 and Antmaze-giant-navigate-v0. We compare $K = 1$, 10, and 30 with the default $K = 20$

used for both tasks. The teacher and student use the same sequence length, while all other model and training settings remain unchanged, including the temporal offset H of the teacher's privileged near-future state.

Table 11 shows that increasing K from 1 to 10 or 20 improves success rates on both tasks. In NEFD, the shared actor uses sequence information to form both action predictions and the internal representations supervised by self-distillation. Short sequences limit the recent states and actions available to this process, while longer sequences provide additional behavioral context for learning current decisions under distant goals. However, performance does not improve continuously with sequence length. Extending K from 20 to 30 provides no additional gain on `Pointmaze` and reduces the mean success rate on `Antmaze`. We therefore use $K = 20$ as the default sequence length for maze tasks. For manipulation and visual tasks, we use $K = 5$.

Table 11: Effect of Transformer sequence length K.

Dataset	**QRL**	**HIQL**	**1**	**10**	**20**	**30**
`Pointmaze-giant-navigate-v0`	68 ±7	46 ±9	95 ±5	99 ±1	100 ±0	100 ±0
`Antmaze-giant-navigate-v0`	14 ±3	65 ±5	79 ±4	86 ±3	88 ±2	84 ±3

B.3.5 Role of the Projection Head

The Teacher and Student form their internal representations from different conditioning information, since only the Teacher observes the privileged near-future state. Directly matching their full hidden representations may therefore impose an overly restrictive constraint on the Student. We use a projection head to instead align the two representations in a shared distillation space.

To examine the effect of this design, we compare NEFD with NEFD (w/o P), which removes the projection head and applies representation self-distillation directly to the hidden representations h_i^{T} and h_i^{S}. All other components and training settings are kept unchanged.

As shown in Table 12, removing the projection head reduces performance across all three tasks. The effect is small on Pointmaze, but becomes more pronounced on `Antmaze` and especially `Scene-play`. These results suggest that directly constraining the full Teacher and Student hidden representations can be overly restrictive under their different conditioning information. Aligning them in a learned projection space provides a more suitable representation-level supervision signal, while allowing the Student to retain flexibility in its own hidden representation.

Table 12: Effect of the projection head in representation self-distillation.

Dataset	**QRL**	**HIQL**	**NEFD (w/o P)**	**NEFD**
`Pointmaze-giant-navigate-v0`	68 ±7	46 ±9	99 ±1	**100** ±0
`Antmaze-giant-navigate-v0`	14 ±3	65 ±5	84 ±2	**88** ±2
`Scene-play-v0`	5 ±1	38 ±3	79 ±9	**95** ±2

B.3.6 Experimental Setup for the Privileged Teacher Ablation

We evaluate No Distill, OPD-Large, and OPD-Privilege on `Pointmaze-giant-navigate-v0`, `Antmaze-giant-navigate-v0`, and `Scene-play-v0`. No Distill retains NEFD's Transformer architecture and student inputs: the state–action sequence and distant goal. It is trained only with advantage-weighted regression (AWR) on recorded actions. The teacher branch conditioned on recorded near-future states is removed, together with action and representation self-distillation.

OPD-Large introduces a teacher with twice as many parameters as the student. The teacher and student receive the same recorded state–action sequence and distant goal. At each decision position in the offline sequences, the student learns to match the teacher's action distribution. We use uniformly weighted action distillation, with all weights fixed to one, and omit both NEFD's goal-progress weighting and representation distillation.

OPD-Privilege additionally provides its teacher with the corresponding recorded near-future state, while the student continues to use the sequence and distant goal alone. Teacher action distributions provide supervision at each decision position in the recorded offline sequences. As in OPD-Large, we use uniformly weighted action distillation and omit both goal-progress weighting and representation distillation.

B.3.7 Experimental Setup for Representation Analysis

Performance across goal distances. For Figure 4(a,b), we compare NEFD with its variant without $\mathcal{L}_{\text{rep}}$ on `Pointmaze-giant-navigate-v0` and `Antmaze-giant-navigate-v0`, retaining action self-distillation in both. Within each fixed maze layout, we randomly sample initial state–goal pairs and group them by shortest path distance on the maze grid. The five distance ranges are 1–6, 7–12, 13–18, 19–24, and 25–31 cells, with 200 distinct pairs per range and 1,000 pairs per environment. We evaluate both variants separately in each distance range using the default OGBench (Park et al., 2025a) episode limits and success criteria. Success rates are averaged over the 200 pairs in each range and eight random seeds, with shaded regions indicating the standard deviation across seeds. The horizontal axis shows the upper bound of each distance range; for example, 31 corresponds to shortest path distances between 25 and 31 cells.

Near-future outcome prediction. We keep each actor fixed and train a linear model using the student representation h_t^S before the action head (Figure 4(c)). The model predicts changes in the agent's XY position over the next 25 steps in navigation and changes in the cube's XYZ position and the drawer and window positions over the next 10 steps in `Scene-play`. We sample 500 episodes from each environment's OGBench offline training data and select four steps per episode. Each selected step is paired with four goals at different temporal distances, yielding 8,000 samples. Both variants use the same samples and prediction settings. We train each prediction model on 400 episodes and evaluate it on the remaining 100, repeating this procedure five times so that each episode is used once for evaluation. We combine predictions from all five evaluations to compute one R^2 score per seed over all 8,000 samples. Results are averaged over eight seeds, with error bars showing variation across seeds.