IPD: Boosting Sequential Policy with Imaginary Planning Distillation in Offline Reinforcement Learning

Yihao Qin*
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yqin637@connect.hkust-gz.edu.cn

Yuanfei Wang*
Peking University
Beijing, China
yuanfei_wang@pku.edu.cn

Hang Zhou
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
hzhou269@connect.hkust-gz.edu.cn

Peiran Liu
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
pliu868@connect.hkust-gz.edu.cn

Hao Dong†
Peking University
Beijing, China
hao.dong@pku.edu.cn

Yiding Ji†
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
jiyiding@hkust-gz.edu.cn

ABSTRACT

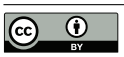
Decision transformer based sequential policies have emerged as a powerful paradigm in offline reinforcement learning (RL), yet their efficacy remains constrained by the quality of static datasets and inherent architectural limitations. Specifically, these models often struggle to effectively integrate suboptimal experiences and fail to explicitly plan for an optimal policy. To bridge this gap, we propose **Imaginary Planning Distillation (IPD)**, a novel framework that seamlessly incorporates offline planning into data generation, supervised training, and online inference. Our framework first learns a world model equipped with uncertainty measures and a quasi-optimal value function from the offline data. These components are utilized to identify suboptimal trajectories and augment them with reliable, imagined optimal rollouts generated via Model Predictive Control (MPC). A Transformer-based sequential policy is then trained on this enriched dataset, complemented by a value-guided objective that promotes the distillation of the optimal policy. By replacing the conventional, manually-tuned return-to-go with the learned quasi-optimal value function, IPD improves both decision-making stability and performance during inference. Empirical evaluations on the D4RL benchmark demonstrate that IPD significantly outperforms several state-of-the-art value-based and transformer-based offline RL methods across diverse tasks.

KEYWORDS

Offline reinforcement learning; decision transformer; sequential policy; world model; model predictive control

ACM Reference Format:
Yihao Qin*, Yuanfei Wang*, Hang Zhou, Peiran Liu, Hao Dong†, and Yiding Ji†. 2026. IPD: Boosting Sequential Policy with Imaginary Planning Distillation in Offline Reinforcement Learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/BELB5985

*Equal contribution; †Corresponding author.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/BELB5985

1 INTRODUCTION

Reinforcement Learning (RL) has achieved remarkable success across various applications, ranging from robotic control [3–6, 18, 20, 36, 42], autonomous driving [22] and chip design [29] to complex strategic games [28, 31, 38, 43]. However, the real-world deployment of online RL is often restrained by the high cost and safety risks associated with active exploration of under trained policies. As a safer alternative, offline RL enables policy training using fixed, pre-collected datasets without requiring environmental interactions [13]. Despite this advantage, static datasets present significant challenges, most notably the value overestimation caused by state-action distribution shifts [25]. To mitigate this issue, prior works have explored regularizing value function approximation or constraining policy deviation from the data collection policy.

Recently, a new class of offline RL algorithms, Decision Transformer [8] and its variants, have emerged and gained prominence in language [1, 44] and vision [9, 33] tasks. They leverage the Transformer's architecture, whose strong sequence modeling capabilities facilitate a reformulation of traditional Temporal Difference-based offline RL as a supervised conditional sequence generation problem. Although these models excel at modeling sequences, they rely on conditional sequence imitation and fundamentally lack the dynamic programming-based RL mechanisms, therefore struggling to stitch suboptimal trajectories into an optimal policy [7]. Several approaches have been developed to address this limitation, including elastic context selection [46], RL objective regularization [19, 48], and return-to-go relabeling [47]. However, these methods offer only marginal performance improvements, as they do not fully integrate the core principles of planning into the whole training and inference cycles of Transformer-based sequential policies.

In this work, we propose **Imaginary Planning Distillation (IPD)**, a novel framework that integrates implicit dynamic programming and explicit model predictive control into both training and inference of Transformer-based sequential policy, boosting optimal trajectory generation. We term this approach "imaginary planning" since both the MPC rollouts and value refinements are

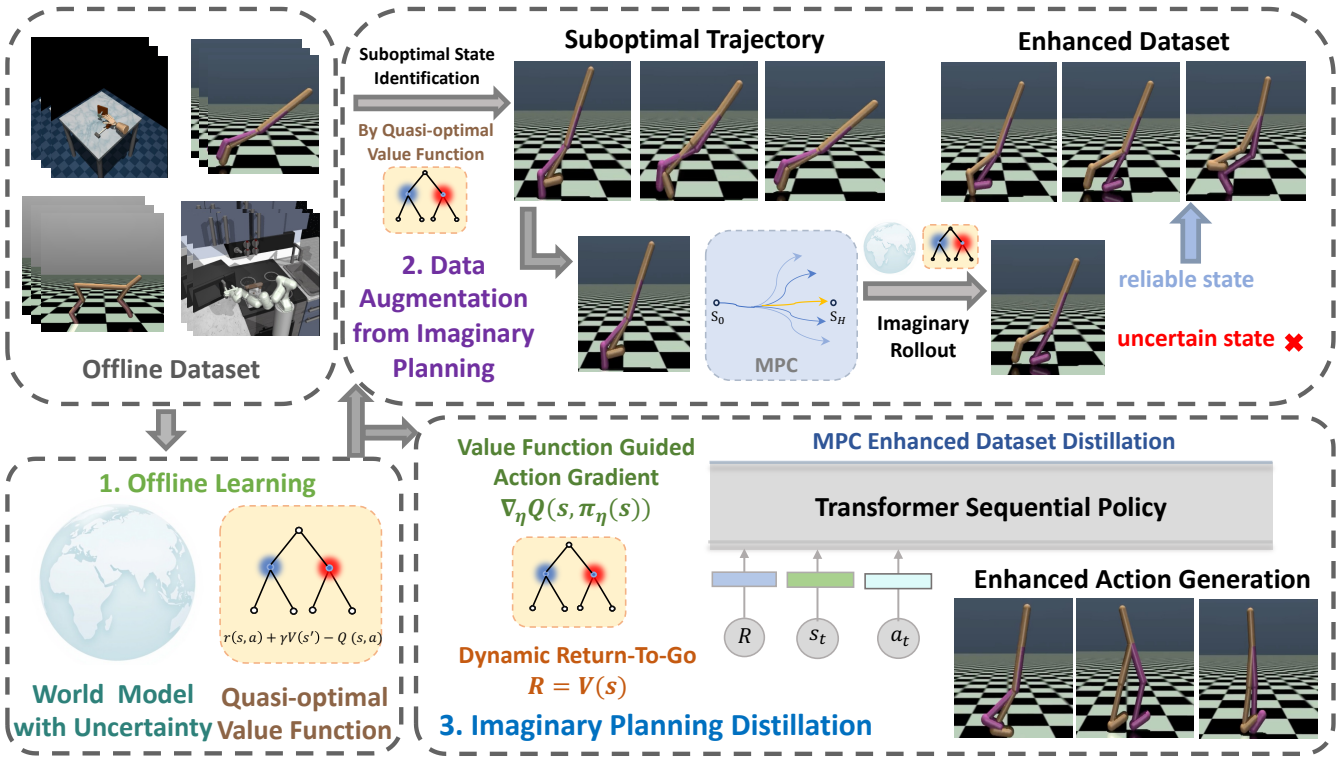


Figure 1: An overview of IPD. The process begins by learning a world model with uncertainty measure and a quasi-optimal value function from the original offline suboptimal dataset. Suboptimal states are identified using the value function, and their corresponding trajectories are replaced with imaginary rollouts generated via Model Predictive Control, using the learned world model and value function. Each generated trajectory is evaluated for uncertainty before incorporated into the enhanced dataset. Finally, a Transformer-based sequential policy is trained on this MPC-enhanced dataset, with additional supervision from the value function via action gradients and a dynamic return-to-go. By distilling the imaginary planning, which streamlines both MPC and dynamic programming, into the Transformer based policy, IPD enables the generation of superior actions.

performed entirely within a learned world model, requiring no direct interaction with the real world.

As illustrated in Figure 1, our method consists of the following key steps. First, we train a world model with well defined uncertainty measure and a quasi-optimal value function using the original offline dataset. Next, we leverage the learned world model, value function and the reliable set derived from uncertainty measure, for the purpose of identify suboptimal trajectories. They are then replaced by imagined rollouts in a reliable set generated from model predictive control, which effectively enhances the dataset's quality. Subsequently, we train the Transformer policy with an additional value objective, encouraging actions that yield higher Q-values. Thus, the results of dynamic programming are implicitly distilled from the learned value function. During inference, the conditioning is based on our proposed quasi-optimal value function rather than conventional return-to-go, which automatically predicts the optimal return and contributes to more effective decision-making.

We evaluate IPD on the D4RL benchmark [11] and results demonstrate consistent performance improvements over state-of-the-art value-based and Transformer-based offline RL methods. Ablation studies further validate key components of IPD, including MPC-driven data augmentation, value-guided action imitation, and return-to-go prediction. Additionally, we analyze how volume of imaginary data augmentation affects policy performance and uncover a scaling law, providing insights into the effectiveness of our framework.

In summary, our main contributions are threefold: 1) We introduce Imaginary Planning Distillation (IPD), a novel framework that seamlessly integrates supervised sequence modeling with imaginary planning. 2) IPD incorporates both implicit dynamic programming and explicit model predictive control into training and inference of Transformer-based policy, enhancing optimal trajectory generation. 3) We conduct extensive experiments and ablation studies on the D4RL benchmark, validating the superior performance of IPD over existing offline RL methods.

2 RELATED WORK

Offline Reinforcement Learning. Offline RL [13, 27] breaks away from the traditional paradigm of online reinforcement learning and exploration, enabling policy learning solely from pre-collected datasets. However, classic off-policy value-based or actor-critic algorithms [14, 30, 37, 40] suffer from out-of-distribution (OOD) issues,

where the value function tends to overestimate unseen state-action pairs [13, 25, 27]. To address this challenge of overestimation, the mainstream offline RL algorithms fall primarily into two categories: policy constraint and value regularization.

For policy constraint methods, they enforce an additional regularization term that measures the policy discrepancy between the learned policy and the behavior policy using different distance metrics, such as batch constraints[13], KL divergence [45], MMD distance [24], and MSE constraints [12]. For value regularization methods, they assign lower values to OOD state-action pairs for the value function, mitigating the overestimation problem [23, 25].

Transformer-based Sequential Policy. More recently, in contrast to the previous approaches, Decision Transformer (DT) [8] introduces a supervised sequence modeling paradigm by directly maximizing action likelihood. The Transformer's sequence modeling capabilities enable a reformulation of traditional Temporal Difference-based offline RL as a supervised conditional sequence generation problem. However, despite its strength in sequence modeling, the Transformer struggles to stitch suboptimal trajectories into an optimal policy[7], as it primarily relies on conditional sequence imitation rather than dynamic programming-based RL. To address this limitation, several techniques have been proposed, including elastic context selection[46], RL objective regularization[19, 48], and return-to-go relabeling[47]. However, these approaches yield only limited performance gains, as none fully incorporate planning principles throughout the training and inference process of Transformer-based sequential policies. In contrast, our proposed IPD manages to integrate a comprehensive planning distillation for the Transformer-based sequential policy.

Model-based reinforcement learning. Model-based RL usually improves sample efficiency and generalization capacity by planning with learned dynamics models [15, 16, 21] or augmenting data using model-free methods [26, 34]. Recent efforts have extended these benefits to offline settings.

ROMI [41] uses a reverse dynamics model to generate rollouts reaching target goals, while MOCODA [34] introduces counterfactual transitions through locally factored models to address out-of-distribution generalization. MOBILE [39] incorporates uncertainty quantification for conservative model usage, and SUMO [35] estimates uncertainty via search-based cross-entropy alignment with in-distribution data. TD-MPC [15, 16] applies latent dynamics modeling and trajectory optimization for online control.

In contrast, IPD integrates dynamic programming-based value learning with MPC planning to synthesize stitched imaginary trajectories from offline data. These trajectories are filtered using uncertainty-aware checks and used to enhance Transformer-based sequential policy learning. To our knowledge, IPD is the first framework to combine model-based MPC and dynamic programming for implicit trajectory stitching, enabling Transformer policies to exceed the limitations of the original offline dataset.

3 METHOD

This section proposes Imaginary Planning Distillation (IPD), an offline reinforcement learning framework to improve Transformer based sequential policy learning by distilling imaginary planning through uncertainty-aware data augmentation and value based guidance. The process is structured into four distinct phases: learning a quasi-optimal foundation, equipping the word model with an uncertainty measure, augmenting dataset via MPC, and distilling imaginary planning knowledge into the final Transformer policy.

We begin by introducing a quasi-optimal value function, which is learned from the original offline dataset using offline Q learning, and grounded in the principles of dynamic programming. In parallel, we train a reliable world model with uncertainty estimation, which is employed in the subsequent data augmentation stage. Next is the data augmentation phase. The learned value function is used to identify suboptimal trajectories, which are then replaced with high-quality trajectories generated using model predictive control (MPC). This process is guided by both the world model and the value function. Each generated trajectory is evaluated using the uncertainty estimate from the world model before being added to the enhanced dataset. Finally, the Transformer based sequential policy is distilled from imaginary planning, consisting of three core components: the MPC enhanced dataset, action gradients guided by the value function, and a dynamic return to go. The "imagined" MPC rollouts are generated through the learned world model and the value function is shaped by offline Q learning, which implicitly incorporates the principle of dynamic programming. Distilling these components into the final policy mitigates the influence of suboptimal supervision and leads to improved policy performance.

3.1 Offline Quasi-Optimal Value Function Learning

IPD initiates by establishing a robust value function to mitigate the overestimation on out-of-distribution state-action pairs common in offline Q learning. We follow the principle of Implicit Q Learning (IQL)[23], which restricts the Bellman update within the support of the dataset distribution. Specifically, we replace the original expectile regression with a Huber-expectile regression in Eq. 1, which provides asymmetric weighting for optimal value estimation and is more resilient to dataset outliers [17]. Here, δ is a hyperparameter that controls the transition between quadratic and linear loss.

Given the original offline dataset D, we derive a quasi-optimal value function $V_\psi(s)$ and Q function $Q_\theta(s,a)$ by optimizing the objectives in Eq. (3) and Eq. (4), respectively. The target Q function is denoted as $Q_{\hat{\theta}}(s,a)$. The Huber loss can be represented as follows

$$\mathcal{L}^{\delta}_{\text{Huber}}(\mathbf{e}_V) = \begin{cases} \frac{1}{2}\mathbf{e}_V^2, & \text{if } |\mathbf{e}_V| \le \delta, \\ \delta\left(|\mathbf{e}_V| - \frac{1}{2}\delta\right), & \text{otherwise,} \end{cases} \tag{1}$$

which reduces the influence of large errors by transitioning from a quadratic to a linear penalty beyond a threshold δ. Specifically, the Huber loss above is expressed using expectile regression as follows:

$$\mathcal{L}_2^{\tau,\text{Huber}}(\mathbf{e}_V) = |\tau - \mathbb{I}(\mathbf{e}_V < 0)| \cdot \mathcal{L}^{\delta}_{\text{Huber}}(\mathbf{e}_V). \tag{2}$$

This formulation preserves the asymmetric weighting of expectile regression while improving robustness to outliers, where the parameter τ controls the degree to which the agent is encouraged to learn the optimal action-value function

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\mathcal{L}^{\delta}_{\text{Huber}}\left(Q_{\hat{\theta}}(s,a) - V_\psi(s)\right)\right] \tag{3}$$

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a)\right)^2\right] \quad (4)$$

Using $Q_\theta(s,a)$ and $V_\psi(s)$, we derive a quasi-optimal policy $\pi_\omega^{\mathbf{QOP}}$ modeled as a Gaussian distribution, via advantage-weighted regression.The policy is optimized using Eq. (5), where $\beta \in [0,\infty)$ denotes the inverse temperature. As β increases, $\pi_\omega^{\mathbf{QOP}}$ becomes increasingly aligned with the action that maximizes the Q function.

$$\mathcal{L}_\pi^{\mathbf{QOP}}(\omega) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp\left(\beta\left(Q_{\hat{\theta}}(s,a) - V_\psi(s)\right)\right) * \log \pi_\omega^{\mathbf{QOP}}(a \mid s)\right] \quad (5)$$

With $\beta \to \infty$, we extract a policy that maximizes the Q-values [23].

$$\pi_\omega^{\mathbf{QOP}}(\arg\max_{a'} Q_\theta(s,a') \mid s) \approx 1. \quad (6)$$

3.2 World Model with Uncertainty Measure

To facilitate effective data augmentation through model predictive contro (MPC), a world model is required to support the generation of imaginary rollouts. In our framework, it simultaneously learns a dynamic model $\mathcal{F}_\phi(\hat{s}_{t+1} \mid s_t, a_t)$ and a reward model $\mathcal{R}_\phi(\hat{r}_{t+1} \mid s_t, a_t)$.

In the offline setting, a standard dynamics model may fail to accurately capture the true system behavior due to limited coverage of the state-action space and inherent stochasticity in the transitions. Such errors will accumulate and degrade the quality of the generated trajectories during imaginary planning. Using a probabilistic ensemble (PE) world model to mitigate this issue, IPD explicitly models both aleatoric uncertainty arising from environmental randomness, and epistemic uncertainty stemming from limited knowledge or training data. Specifically, the next-state world model is implemented as an ensemble of Gaussian mixture models:

$$\begin{aligned}\tilde{\mathcal{F}}_{\mathbf{PE}}(\hat{s}_{t+1} \mid s_t, a_t) &= \frac{1}{E}\sum_{e=1}^{E}\mathcal{F}_{\phi_e}(\hat{s}_{t+1} \mid s_t, a_t);\\ \mathcal{F}_{\phi_e}(\hat{s}_{t+1} \mid s_t, a_t) &= \mathcal{N}\left(\mu_{\phi_e}(s_t,a_t), \Sigma_{\phi_e}(s_t,a_t)\right)\end{aligned} \quad (7)$$

where $e \in 1,\ldots,E$ indexes the ensemble members, and $\Sigma\phi_e(s_t,a_t)$ captures the aleatoric uncertainty. In addition to this, epistemic uncertainty is quantified by measuring the disagreement among ensemble members. Normally, this is computed as the Kullback-Leibler (KL) divergence between the ensemble's Gaussian mixture model $\tilde{\mathcal{F}}_{\mathbf{PE}}$ and each individual model distribution $\mathcal{F}_{\phi_e}$.

Unfortunately, the KL divergence between the Gaussian mixture models does not have a closed-form solution. To reinstate computational tractability during imaginary rollouts with MPC, we introduce a geometric Jensen-Shannon (GJS) divergence [10, 32] based uncertainty measure $\mathcal{U}$, which provides a tractable form for model disagreement. The measure decomposes the conventional KL divergence between the ensemble and individual models into the following iterative computations:

$$\begin{aligned}\mathcal{U}(s,a) &= \frac{1}{E(E-1)}\sum_{(i,j)\in\mathcal{P}}\mathcal{J}_{\mathrm{GJS}}(\mathcal{N}_i\|\mathcal{N}_j),\\ &\text{where } \mathcal{P} \triangleq \{(i,j) \mid 1 \le i < j \le E\}\end{aligned} \quad (8)$$

$$\mathcal{J}_{\mathrm{GJS}}(\mathcal{N}_i|\mathcal{N}_j) = \frac{1}{2}\left[\mathcal{J}_{\mathrm{KL}}(\mathcal{N}_i|\mathcal{N}_{ij}) + \mathcal{J}_{\mathrm{KL}}(\mathcal{N}_j|\mathcal{N}_{ij})\right] \quad (9)$$

$$\mathcal{N}_{ij} \sim (\mu_{ij}, \Sigma_{ij}), \quad \Sigma_{ij} = \left(\tfrac{1}{2}\Sigma_{\phi_i}^{-1} + \tfrac{1}{2}\Sigma_{\phi_j}^{-1}\right)^{-1} \quad (10)$$

$$\mu_{ij} = \Sigma_{ij}\left(\frac{1}{2}\left(\Sigma_{\phi_i}\right)^{-1}\mu_{\phi_i} + \frac{1}{2}\left(\Sigma_{\theta_j}\right)^{-1}\mu_{\theta_j}\right) \quad (11)$$

Based on the uncertainty measure $\mathcal{U}(s,a)$, we then define a threshold parameter κ to filter a reliable subset from a generated rollout set S, as defined in Eq. (12). This procedure evaluates the reliability of imagined rollouts, ensuring that only trajectories with acceptable uncertainty levels are included in the augmented dataset.

$$\mathcal{E} = \{(s,a) \in \mathcal{S} \mid \mathcal{U}(s,a) < \kappa\} \quad (12)$$

Although ensemble models with Gaussian outputs are effective at capturing uncertainty, they also suffer from training instability due to variance collapse or explosion. To mitigate this issue, we employ Gaussian reparameterization to compute the predicted next state of the ensemble model, denoted by $\mu_{\mathbf{PE}}$. Additionally, we introduce an exponential decay schedule to regularize the predicted covariance $\Sigma_{\mathbf{PE}}$ during training. These strategies are incorporated into the training loss to stabilize learning. The resulting state-transition consistency loss $\mathcal{L}_c$ is formulated as follows:

$$\begin{aligned}\Sigma_{\mathbf{PE}} &= \frac{1}{E}\sum_e \Sigma_{\phi_e}\\ \mu_{\mathbf{PE}} &= \frac{1}{E}\sum_e \mu_{\phi_e} + \Sigma_{\mathbf{PE}}^{\frac{1}{2}}\cdot\epsilon, \epsilon \sim \mathcal{N}(0,I)\end{aligned} \quad (13)$$

$$\begin{aligned}\mathcal{L}_c = \frac{1}{|\mathcal{D}|}\sum_{(s_t,a_t,s_{t+1})\in\mathcal{D}} &\|\mu_{\mathbf{PE}}(s_t,a_t) - s_{t+1}\|^2\\ &+ \left\|\gamma_{\exp}(\Sigma_{\mathbf{PE}}(s_t,a_t) - \Sigma_{\mathrm{reg}})\right\|^2\end{aligned} \quad (14)$$

The above covariance regularization is introduced mainly to stabilize training. The predicted covariance $\Sigma_{\mathbf{PE}}$ is guided towards a reasonable regularization target Σ_{reg} during the initial training stage, under the control of a designed exponentially decayed regularization weight $\gamma_{\exp}$. The decay factor $\gamma_{\exp}(k)$ is defined as:

$$\gamma_{\exp}(k) = \gamma_0 \cdot \exp\left(-\frac{k}{T}\right), \quad (15)$$

where k is the current training iteration step, γ_0 is the initial regularization scale, T is the decay rate controlling how quickly the regularization diminishes steps. In addition, the loss of the reward model and the total loss of the world model can be calculated as

$$\mathcal{L}_r = \frac{1}{|\mathcal{D}|}\sum_{(s_t,a_t,r_{t+1})\in\mathcal{D}}\left|\mathcal{R}_\phi(s_t,a_t) - r_{t+1}\right|^2, \mathcal{L}_{\mathrm{world}} = \alpha_c\mathcal{L}_c + \alpha_r\mathcal{L}_r \quad (16)$$

For more details regarding the implementation for $\gamma_{\exp}$, Σ_{reg}, α_c and α_r, please refer to Experiments section.

3.3 Data Augmentation with Imaginary Planning

Building upon the quasi-optimal value function and the world model with uncertainty estimation, the next phase of IPD leverages these learned components to generate reliable data through MPC-based imaginary planning, which identifies and subsequently replaces suboptimal segments within the dataset.

The imaginary planning procedure consists of two key stages. The first stage, referred to as **Suboptimal State Identification**, aims to identify state-action pairs in the dataset that are likely to benefit most from enhancement. These are typically suboptimal samples where better policies yield improved returns. The identification is based on evaluating the discrepancy between the real return observed in the dataset and the return that could be obtained through imaginary rollouts. Specifically, let H_I denote the imaginary planning horizon. For each state in a trajectory, the imaginary return R_{Imagine} and the real return R_{Real} are computed as follows:

$$R_{\text{Imagine}}(s_t) = \sum_{k=0}^{H_I-1} \gamma^k \cdot \mathcal{R}_\phi(\hat{s}_{t+k}, \pi_\omega^{\mathbf{QOP}}(\hat{s}_{t+k})) + V_\psi(\hat{s}_{H_I}) \tag{17}$$

$$R_{\text{Real}}(s_t) = \sum_{k=0}^{H_I-1} \gamma^k \cdot r(s_{t+k}, a_{t+k}) + V_\psi(s_{H_I}) \tag{18}$$

where R_{Imagine} is computed by rolling out the quasi-optimal policy $\pi_\omega^{\mathbf{QOP}}$ for H_I steps using the learned world model to simulate future states $\hat{s}$. In contrast, R_{Real} is derived from the original trajectory stored in the offline dataset. By comparing these two returns, as defined in Eq. (17) and Eq. (18), IPD is able to identify candidate states within trajectories that significantly underperform relative to the imagined quasi-optimal rollouts. These states are selected for enhancement in the next phase.

The core principle of this stage is to select the top-K states ranked by their potential improvement, measured as the difference $R_{\text{Imagine}} - R_{\text{Real}}$. The resulting set of selected states, denoted as S_e, represents those with the highest potential for enhancement and are thus prioritized for data augmentation.

Fundamentally, the main principle for the **Suboptimal State Identification** stage is to select states with the greatest value differences $R_{\text{Imagine}}(s_t) - R_{\text{Real}}(s_t)$. To ensure sufficient context for the transformer policy while reserving space for imaginary rollout, we process trajectories with asymmetric windowing: each candidate state s_t maintains H_{con} historical states for conditioning while preserving H_I future steps for augmentation, forming segments:

$$\tau_{\text{window}} = \{s_{t-H_{\text{con}}}, ..., s_t, ..., s_{t+H_I}\}. \tag{19}$$

where the left H_{con} states provide conditioning and the right H_I states accommodate MPC-generated augmentation.

Given an original trajectory τ_o from dataset D, we segment it using a sliding window with step size 1, as defined in Eq. (19). The window ensures that the beginning of the trajectory contains H_{con} historical states and the end contains H_I future steps. The state at the center of each segment, i.e., the current state s_t, is added to the candidate state set S_o. We then compute the value difference for each candidate state and rank them in descending order as follows:

$$\mathcal{S}_{\text{sorted}} = \text{argsort}_{s_t \in \mathcal{S}_o} \left[R_{\text{Imagine}}(s_t) - R_{\text{Real}}(s_t) \right]. \tag{20}$$

We then select the top K states for augmentation, where K is determined by the augmentation ratio N_{aug}, as shown in Eq. (21):

$$K = \min\left(\left\lfloor \frac{N_{\text{aug}} \cdot |\tau_o|}{H_I} \right\rfloor, |\mathcal{S}_{\text{sorted}}| \right), \tag{21}$$

The final set of selected suboptimal states for augmentation is:

$$\mathcal{S}_{\text{aug}} = \mathcal{S}_{\text{sorted}}[1:K] \tag{22}$$

For each selected state $s_t^{(i)} \in \mathcal{S}_{\text{aug}}$, we generate an augmented trajectory segment of length H_I with MPC and world model:

$$\tau_{\text{aug}}^{(i)} = \left\{ s_t^{(i)}, \hat{s}_{t+1}^{(i)}, ..., \hat{s}_{t+H_I-1}^{(i)} \right\}, \tag{23}$$

where $\hat{s}_{t+k}$ refers to the generated imaginary rollout state. The rollout action $a_{t+k}^{(i)}$ is obtained through MPC planning:

$$a_{t+k}^{(i)} = \pi_{\text{mpc}}^*(s_{t+k}^{(i)}) \quad \text{for} \quad k = 0, ..., H_I - 2. \tag{24}$$

All generated state-action pairs in the MPC planning stage must satisfy the uncertainty constraint.

Note that H_I (data augmentation horizon) differs from the MPC planning horizon H_m. H_I represents synthetic rollout length while H_m determines optimal action selection depth.

After identifying the states that require enhancement, IPD proceeds to the next stage by performing reliable optimal planning via MPC for these selected candidate states. This step leverages both the previously learned world model with uncertainty estimation and the quasi-optimal value function and policy.

To ensure reliability, IPD constrains the MPC rollouts to remain within the uncertainty set $\mathcal{E}$ defined in Eq. (12). This constraint guarantees that optimal actions are selected only from model-confident regions of the state space, thereby reducing the risk of compounding model errors during imaginary data generation.

For optimal planning, IPD samples N_{mpc} candidate trajectories of horizon H_m using the learned world model. Among these, the action of the first step of the trajectory with the highest cumulative discounted return is selected:

$$\begin{aligned} \pi_{mpc}^*(a_t \mid s_t) = \arg\max_{\{a_t^{(i)}\}_{i=1}^{N_{\text{mpc}}}} & \sum_{k=0}^{H_m} \gamma^k \mathcal{R}_\phi\left(\hat{s}_{t+k}^{(i)}, \pi_\omega^{\mathbf{QOP}}(\hat{s}_{t+k}^{(i)})\right) \\ & + \gamma^{H_m} V_\psi(\hat{s}_{t+H_m}^{(i)}) \end{aligned} \tag{25}$$

where $\hat{s}_{t+k}^{(i)}$ denotes the k-th state in the i-th sampled trajectory, predicted by $\tilde{\mathcal{T}}_{\mathbf{PE}}$. The pseudocode for the imaginary data augmentation procedure is provided in Algorithm 1.

3.4 Imaginary Planning Distillation

Based on the previously learned quasi-optimal value function and the augmented dataset, we further distill the knowledge obtained from these imaginary planning into the Transformer based sequential policy. This distillation process of IPD streamlines three key components. First, the learned quasi-optimal Q function provides action gradient guidance in the form of $\nabla_\eta Q(s, \pi_\eta(s))$, serving as a regularization signal to refine the policy. Second, the quasi-optimal value function dynamically guides the Transformer's return-to-go estimation, allowing it to infer the potential of future rewards directly from state inputs, without the need to manually define fixed target values. Third, the augmented dataset incorporates high-quality trajectories generated by MPC planning, which improves the overall performance of action supervision.

The joint effect of these components is captured by the total IPD loss function, which integrates both sequence modeling and quasi-optimal Q function guided regularization:

$$\mathcal{L}_{\text{IPD}}(\eta) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}_{\text{aug}}}\Bigg[\underbrace{\left(a_t - \pi_\eta\left(\{s,a\}_{t-K:t-1}, V_\psi(s_t), s_t\right)\right)^2}_{\text{Sequence Modeling Term}} - \alpha\cdot\underbrace{Q_\theta\left(s_t, \pi_\eta\left(\{s,a\}_{t-K:t-1}, V_\psi(s_t), s_t\right)\right)}_{\text{Q-Value Regularization Term}}\Bigg]. \tag{26}$$

where $\mathcal{D}_{\text{aug}}$ is the augmented dataset constructed in Alg. 1 and contains trajectories τ_i. (s_t, a_t) is a state-action pair sampled from $\mathcal{D}_{\text{aug}}$. $\{s,a\}_{t-K:t-1}$ represents the context window of past K states and actions. $V_\psi(s_t)$ is the estimated state value from the quasi-optimal value function, providing a dynamic prompt for return-to-go estimation. $\pi_\eta(\cdot)$ denotes the Transformer policy parameterized by η. α is a weighting coefficient that balances the two loss components.

This composite loss function ensures that the policy not only replicates the high-quality actions from the augmented dataset but is also regularized by the Q-function. Thus, actions are produced and *outperform* those in the dataset, facilitating performance improvement and knowledge distillation from the imaginary planner.

Algorithm 1 Imaginary Data Augmentation

Require: Pretrained model: V_ψ, π_ω^{QOP}, $\tilde{\mathcal{F}}_{\text{PE}}$, $\mathcal{R}_\phi$
Require: States set: $\mathcal{E}$, S_e
Require: horizon H_I, MPC horizon H_m, MPC rollout number N_{mpc}

Initialize: $\mathcal{D}_{\text{aug}} \leftarrow \emptyset$
for each state $s_0 \in S_e$ **do**
  $s \leftarrow s_0$
  $\tau \leftarrow [\,]$
  for $t \leftarrow 1$ **to** H_I **do**
    if $s \notin \mathcal{E}$ **then**
      break // Uncertainty check
    end if
    MPC Planning:
    $a^* \leftarrow \text{MPC}(s, H_m, N_{mpc}, \tilde{\mathcal{F}}_{\text{PE}}, \mathcal{R}_\phi, V_\psi)$ Eq. (25)
    $\hat{r}, \hat{s}' \leftarrow \mathcal{R}_\phi(s, a^*), \tilde{\mathcal{F}}_{\text{PE}}(s, a^*)$
    $\tau \leftarrow \tau \cup \{(s, a^*, \hat{r}, \hat{s}')\}$
    $s \leftarrow \hat{s}'$
  end for
  if $\tau \neq [\,]$ **then**
    $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}_{\text{aug}} \cup \{\tau\}$
  end if
end for
return $\mathcal{D}_{\text{aug}}$

4 EXPERIMENTS

In this section, we present comprehensive experiments in D4RL benchmark [11], which are designed to evaluate the performance of the proposed IPD in comparison with several baseline approaches. Specifically, we benchmark IPD against traditional Q-learning based methods, as well as recent Transformer based sequential policy methods, along with their improved versions that aim to address the problem of suboptimal trajectories.

Complementary to those experiments, we conduct a series of ablation studies to assess the efficacy of different components of IPD. In particular, we explore how the superior planning quality of MPC compared to conventional greedy Q-learning strategies contributes to enhanced data generation, which highlight the advantages of MPC in handling complex decision spaces. Furthermore, we investigate how the scale of enhanced data generation influences the agent's ability to address complex tasks, with a focus on exploring potential scaling law in offline RL. Additionally, we examine the influence of the quasi-optimal value function as a guiding mechanism, comparing it with manually engineered return-to-go functions.

Our experiments are structured to investigate several critical research questions, each of which is explored in dedicated sections:

- Does IPD outperform Q-learning-based methods and conventional Transformer based sequential policy methods?
- Does IPD surpass state-of-the-art advancements that overcome the suboptimal trajectory limitations of Transformer based sequential policy methods?
- Does MPC enhance the quality of data generation compared to vanilla greedy Q-learning methods?
- What is the relation between the volume of data generation and the final performance of IPD?
- Does the quasi-optimal value function, as a guidance mechanism, mitigate the instability caused by manually engineered return-to-go and improve robustness?

4.1 Baseline Methods

We conduct a comprehensive comparison of IPD with several baseline methods, including traditional Q-learning-based approaches, such as Conservative Q-Learning (CQL) [25] and Implicit Q-Learning (IQL) [23], as well as vanilla Transformer-based sequential policy methods, notably the Decision Transformer (DT) [8]. Furthermore, we evaluate IPD against enhanced variants of DT. These methods either target Diffusion-Based Sequence Modeling or are specifically designed to overcome the challenge of suboptimal trajectory stitching, such as Decision Diffuser (DD) [2], Elastic Decision Transformer (EDT) [46], Q learning decision Transformer (QDT) [47], Q-value regularized Transformer (QT) [19] and Reinformer [48].

4.2 Main Results

This subsection summarizes the main results of our experiments carried out in three distinct domains: Gym tasks, Kitchen tasks, and Adroit tasks. These experiments cover ten tasks, as shown in Table 1, we performed 10 evaluation episodes for each task. To ensure a fair comparison and meaningful interpretation of the results, all scores have been normalized with respect to the D4RL benchmarks.

As illustrated in Table 1, IPD demonstrates superior performance compared to most offline Q-learning-based methods and transformer-based approaches. This highlights IPD's ability to leverage the sequence modeling strengths of transformers while simultaneously enhancing its performance through the integration of dynamic programming techniques.

For Gym Tasks, in the context of these medium datasets, IPD excels by effectively harnessing MPC infused with value function to generate more high-quality data. Meanwhile, for Kitchen tasks that require effectively generalizing to novel states and long-horizon

Gym Task	IQL	CQL	DT	QDT	DD	EDT	QT	Reinformer	IPD
walker-medium	78.3	83.0	74.0	67.1	82.5	72.8	87.6	80.5	**89.5 ± 3.7**
walker-medium-replay	73.9	77.2	79.4	58.2	68.9	74.8	94.2	72.9	**96.2 ± 2.1**
hopper-medium	66.3	69.4	67.6	66.5	79.3	64.5	78.0	63.5	**81.6 ± 4.8**
hopper-medium-replay	94.7	95.0	82.7	52.1	100.0	89.0	102.1	83.3	**103.2 ± 3.7**
halfcheetah-medium	47.4	49.2	42.6	39.3	49.1	42.5	49.1	42.9	**51.2±2.6**
halfcheetah-medium-replay	44.2	45.5	36.6	35.6	39.3	37.8	48.9	39.0	**49.9 ± 1.6**
Kitchen Tasks									
kitchen-complete	62.5	43.8	50.8	56.9	65.0	40.8	75.0	50.9	**78.4 ± 4.6**
kitchen-partial	46.3	49.8	57.9	58.3	57.0	10.0	73.2	73.1	**74.3 ± 2.8**
Adroit Tasks									
hammer-human-v1	1.4	4.4	0.2	6.8	1.9	14.2	**24.8**	17.2	22.9 ± 2.3
pen-cloned-v1	37.3	39.2	75.8	64.2	42.8	48.4	90.1	82.4	**92.8 ± 3.7**

Table 1: Evaluating IPD and State-of-the-Art Q-Learning and Transformer-Based Sequential Policy Methods on D4RL Benchmarks: Performance Analysis Through Tenfold Episode Evaluations

optimization and Adroit tasks that struggle in sparse human demonstration, IPD also demonstrates its advancements efficacy. By incorporating data generation and quasi-optimal value function guidance, IPD leverages the implicit dynamic programming to guide the agent in producing higher-value actions, which results in a notable boost in effectiveness, enabling IPD to deliver outstanding performance and setting it apart from other methods that may struggle in scenarios with limited high-quality trajectories.

4.3 Ablation Study

This subsection presents a comprehensive ablation study on key components of IPD, emphasizing its data generation and quasi-value guidance. We analyze their contributions and impacts.

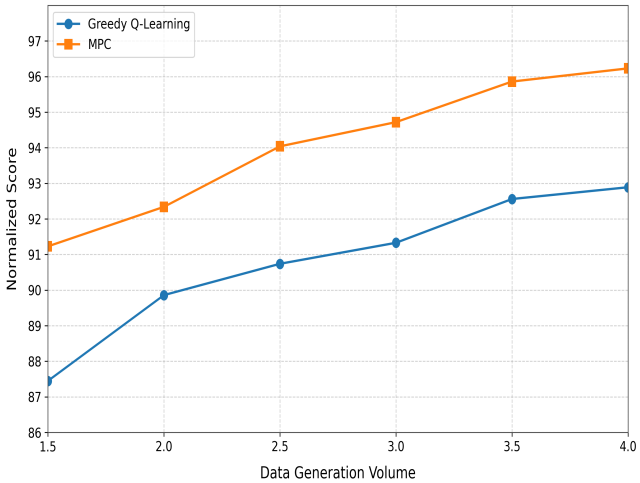


Figure 2: Comparison between MPC and Greedy Q-Learning data augmentation in Walker2d-medium-replay task.

Analysis for Data Generation. We compare our MPC-based data generation procedure with a greedy Q-learning procedure, which utilizes the quasi-optimal policy in Eq. 5 to directly generate action for rollouts. Fig. 2 shows that data generation empowered by MPC surpasses vanilla greedy Q-learning in performance. By utilizing pretrained world model and quasi-value function, MPC can sample multiple trajectories and select the optimal one for action selection. By choosing the most optimal actions based on these sampled trajectories, MPC achieves a higher performance level. This approach not only demonstrates the effectiveness of our trained world model but also highlights MPC's ability to make more informed and strategic decisions.

In addition, the quantity of data generation reveals that as more data is generated, the performance in IPD shows an approximately linear improvement. This trend further demonstrates the effectiveness of our approach, and the data scaling effect in offline data generation. By utilizing world model and MPC for planning, we can generate better trajectories for the transformer-based sequential policy. This approach enhances the quality of the training data, leading to improved policy performance.

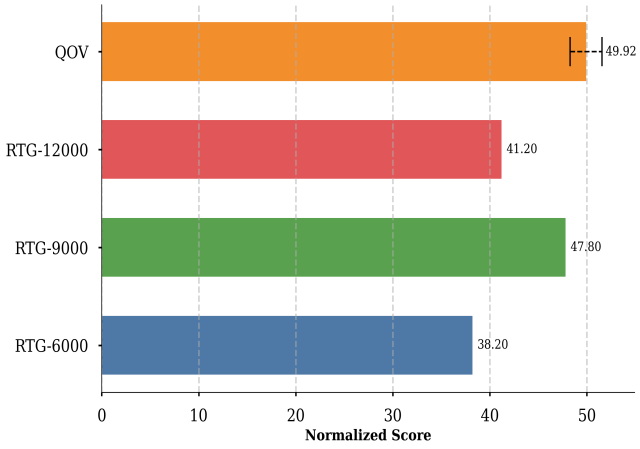


Figure 3: Performance Comparison between quasi-optimal value function and different setting of return-to-go in halfcheetah-medium-replay task.

Analysis for Quasi-Value Function Guidance. A critical challenge in Decision Transformers is the sensitivity to manually engineered Return-To-Go (RTG) values, which are computationally expensive during inference and result in instability of the algorithm's

performance. This is primarily due to the arbitrary RTG values, which may not accurately reflect the optimal trajectory and cause suboptimal decision-making. We replace arbitrary RTG values with the learned Quasi-Optimal Value (QOV), which streamlines the inference process, eliminating the need for costly manual tuning while enhancing the robustness and stability of the algorithm.

As evidenced in Table 1 and Fig. 3, our approach yields a significantly lower variance in tests, indicating consistent and stable outcomes across different trials. In contrast, varying RTG values results in noticeable performance degradation and instability during inference, as the agent struggles to adapt to inconsistent guidance.

4.4 Implementation Details

In this subsection, we detail the implementation, including neural network architecture and hyperparameter settings for IPD, all experiments were conducted on NVIDIA GeForce RTX 4090 GPU.

4.4.1 Quasi-optimal Value Function Learning . We use the Adam optimizer with an initial learning rate of 3×10^{-4}. The Q, V, and actor networks each consist of two fully connected layers with 256 hidden units and ReLU activation functions. For the actor network, we apply a cosine annealing schedule to adjust the learning rate during training. To enhance training stability for the critic, we adopt a double Q-learning strategy. The policy is modeled as a Gaussian distribution with a state-independent standard deviation. For updating the target Q network, we use a soft update mechanism with a smoothing coefficient $a = 0.005$. The threshold for the Huber loss is set to the δ-th percentile of the historically collected value error $\mathbf{e}_V$. Detailed hyperparameter settings are provided in Table 2.

Value Function	Value	World Model	Value
τ	0.7	Σ_{reg}	$e^{-4.5} \cdot I$
β	3	γ_0	1.0
δ	0.96	T	10^5
γ	0.99	α_c	0.5
Dropout	0.01	α_r	0.5

Table 2: Hyperparameters of Quasi-Optimal Value Function and World Model with Uncertainty Measure Learning.

4.4.2 World Model with Uncertainty Measure. We use an ensemble of three models for world model, each of it is trained using the Adam optimizer with a learning rate of 3×10^{-4}. The architecture consists of multiple fully connected layers with 400 units, Layer Normalization, and ReLU activation functions to ensure stable gradient propagation. The dynamics and reward models share the same architecture but are trained independently. For the total loss of world model is $\mathcal{L}_{\text{world}} = \alpha_c \mathcal{L}_c + \alpha_r \mathcal{L}_r$, we set $\alpha_c = \alpha_r = 0.5$ to maintain equal contribution of dynamic prediction and reward prediction. For exponential decay schedule, we constrain the covariance within $[e^{-10}, e^{0.2}]$. Details are provided in Table 2.

4.4.3 Data Generation with Imaginary Planning. The specific parameter setting is listed in Table 3.

4.4.4 Transformer-based Sequential Policy. For training the Transformer based sequential policy, we use the Adam optimizer with a learning rate of 3×10^{-4} and employ the ReLU activation function. Detailed parameter settings are provided in Table 4.

Parameters	Value
γ_{mpc}	0.99
H_I	10
H_m	10
H_{con}	10
N_{mpc}	3

Table 3: Hyperparameters of Imaginary Data Augmentation.

Parameters	Value
Layers	4
Dropout	0.01
Embedding	256
Attention heads	4
Context length	20

Table 4: Hyperparameters of Sequential Policy

5 CONCLUSION

This work developed Imaginary Planning Distillation (IPD), a novel framework that bridges the gap between supervised learning and reinforcement learning (RL). By integrating implicit dynamic programming with explicit model predictive control (MPC), IPD enables Transformer-based policies to transcend the limitations of suboptimal offline datasets. IPD utilizes an uncertainty-aware world model and a quasi-optimal value function to augment datasets with reliable imagined rollouts, significantly augmenting the dataset. In addition, we propose an optimal value gradient-guided action imitation objective that further integrates planning into the policy learning process. For inference, we replace the conventional return-to-go with a learned value function, facilitating dynamic guidance and stable decision-making. Our evaluation of IPD on the D4RL benchmark reveals consistent performance improvements over existing value-based and Transformer-based offline RL methods. Comprehensive ablation studies affirm the contributions of MPC-driven data augmentation, value-guided imitation, and learned return-to-go prediction. Additionally, our analysis of scaling laws offers valuable insights into the benefits of imaginary data augmentation.

In summary, IPD offers a comprehensive and principled approach that integrates "imaginary" planning and sequence modeling for offline RL. Thus, this work paves the way for more effective policy learning in real-world decision-making tasks.

ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China grants 62303389, 62373289; Guangdong Basic and Applied Basic Research Funding grants 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039 and Youth Talent Support Program of Guangdong Association for Science and Technology grant SKXRC2025463.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. 2023. Is conditional generative modeling all you need for decision-making?. In *The 11th International Conference on Learning Representations*.

[3] Fengshuo Bai, Yu Li, Jie Chu, Tawei Chou, Runchuan Zhu, Ying Wen, Yaodong Yang, and Yuanpei Chen. 2025. Retrieval dexterity: Efficient object retrieval in clutters with dexterous hand. *arXiv preprint arXiv:2502.18423* (2025).

[4] Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. 2025. RAT: Adversarial Attacks on Deep Reinforcement Agents for Targeted Behaviors. *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), 15453–15461.

[5] Fengshuo Bai, Hongming Zhang, Tianyang Tao, Zhiheng Wu, Yanna Wang, and Bo Xu. 2023. PiCor: Multi-Task Deep Reinforcement Learning with Policy Correction. In *AAAI Conference on Artificial Intelligence*. 6728–6736.

[6] Fengshuo Bai, Rui Zhao, Hongming Zhang, Sijia Cui, Shao Zhang, bo xu, Lei Han, Ying Wen, and Yaodong Yang. 2025. STAR: Efficient Preference-based Reinforcement Learning via Dual Regularization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[7] David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. 2022. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems* 35 (2022), 1542–1553.

[8] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[10] Bernd Frauenknecht, Artur Eisele, Devdutt Subhasish, Friedrich Solowjow, and Sebastian Trimpe. 2024. Trust the Model Where It Trusts Itself–Model-Based Actor-Critic with Uncertainty-Aware Rollout Adaption. In *International Conference on Machine Learning*.

[11] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219* (2020).

[12] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.

[13] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*. PMLR, 2052–2062.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. 1861–1870.

[15] Nicklas Hansen, Hao Su, and Xiaolong Wang. 2024. Td-mpc2: Scalable, robust world models for continuous control. In *The 12th International Conference on Learning Representations*.

[16] Nicklas Hansen, Xiaolong Wang, and Hao Su. 2022. Temporal difference learning for model predictive control. In *39th International Conference on Machine Learning*. 8387–8406.

[17] Lingxin Hao and Daniel Q Naiman. 2007. *Quantile regression*. Sage Publications.

[18] Zen Kit Heng, Zimeng Zhao, Tianhao Wu, Yuanfei Wang, Mingdong Wu, Yangang Wang, and Hao Dong. 2025. Boosting Universal LLM Reward Design through Heuristic Reward Observation Space Evolution. *arXiv preprint arXiv:2504.07596* (2025).

[19] Shengchao Hu, Ziqing Fan, Chaoqin Huang, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. 2024. Q-value regularized transformer for offline reinforcement learning. In *International Conference on Machine Learning*.

[20] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. 2021. How to train your robot with deep reinforcement learning: lessons we have learned. *International Journal of Robotics Research* 40, 4-5 (2021), 698–721.

[21] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems* 32 (2019).

[22] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 23, 6 (2021), 4909–4926.

[23] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2022. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*.

[24] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems* 32 (2019).

[25] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[26] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. 2020. Reinforcement learning with augmented data. *Advances in neural information processing systems* 33 (2020), 19884–19895.

[27] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[28] Long Ma, Yuanfei Wang, Fangwei Zhong, Song-Chun Zhu, and Yizhou Wang. 2024. Fast peer adaptation with context-aware exploration. In *41st International Conference on Machine Learning*.

[29] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. 2021. A graph placement methodology for fast chip design. *Nature* 594, 7862 (2021), 207–212.

[30] Volodymyr Mnih. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[32] Frank Nielsen. 2019. On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy* 21, 5 (2019), 485.

[33] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*. 4195–4205.

[34] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. 2022. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems* 35 (2022), 18143–18156.

[35] Zhongjian Qiao, Jiafei Lyu, Kechen Jiao, Qi Liu, and Xiu Li. 2025. Sumo: Search-based uncertainty estimation for model-based offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 20033–20041.

[36] Yihao Qin and Yiding Ji. 2025. An Efficient Bayesian Policy Exploration Approach for Reinforcement Learning Model Predictive Control. In *IEEE 19th International Conference on Control and Automation*. 460–465.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[38] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[39] Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. 2023. Model-Bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*. 33177–33194.

[40] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[41] Jianhao Wang, Wenzhe Li, Haozhe Jiang, Guangxiang Zhu, Siyuan Li, and Chongjie Zhang. 2021. Offline reinforcement learning with reverse model-based imagination. *Advances in Neural Information Processing Systems* 34 (2021), 29420–29432.

[42] Yuanfei Wang, Xiaojie Zhang, Ruihai Wu, Yu Li, Yan Shen, Mingdong Wu, Zhaofeng He, Yizhou Wang, and Hao Dong. 2025. AdaManip: Adaptive Articulated Object Manipulation Environments and Policy Learning. In *International Conference on Learning Representations*.

[43] Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. 2022. ToM2C: Target-oriented Multi-agent Communication and Cooperation with Theory of Mind. In *International Conference on Learning Representations*.

[44] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38–45.

[45] Yifan Wu, George Tucker, and Ofir Nachum. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361* (2019).

[46] Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. 2024. Elastic decision transformer. *Advances in Neural Information Processing Systems* 36 (2024).

[47] Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. 2023. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline RL. In *International Conf. on Machine Learning*. 38989–39007.

[48] Zifeng Zhuang, Dengyun Peng, Jinxin Liu, Ziqi Zhang, and Donglin Wang. 2024. Reinformer: Max-return sequence modeling for offline RL. In *41st International Conference on Machine Learning*.