PLANNING IS TEST-TIME SCALING: A WORLD PLANNING MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective decision-making requires more than predicting the future. An agent must also know how to use predicted outcomes to improve its actions. We introduce the World Planning Model (WPM), a learned iterative decision process that couples future prediction with action refinement in a single network. Starting from an initial action proposal, WPM predicts its consequences and uses the resulting future representation to update the action, repeatedly applying the same operator to produce progressively refined decisions. By training prediction and refinement jointly, WPM learns not only task-relevant dynamics, but also how predictive information should be translated into action updates. This replaces explicit candidate sampling, scoring, and selection with implicit refinement, while retaining the ability to allocate additional computation at test time through more refinement steps. In offline goal-conditioned control from pixels, WPM matches or exceeds explicit planners built on the same model on most environments. Furthermore, WPM's success rises with refinement depth on most environments without retraining. Controlled ablations further indicate that predictive supervision improves the learned decision process. These results suggest a general alternative to both one-shot policy execution and explicit search: learning a reusable planning operator whose computation can be scaled at test time.

1 INTRODUCTION

World models enable agents to anticipate the consequences of candidate actions before execution, which provides a key inductive bias for executive control (Ha & Schmidhuber, 2018; LeCun, 2022). In modern robotics and sequential decision-making, this capability is primarily exploited through two paradigms: explicit planning (Chua et al., 2018; Hafner et al., 2019; Hansen et al., 2022; 2024) and policy learning (Hafner et al., 2020; 2021; 2025; Janner et al., 2019). Explicit planning methods use world models as action-conditioned dynamics models to evaluate and refine candidate trajectories with optimizers such as the cross-entropy method (de Boer et al., 2005) and model predictive path integral control (Williams et al., 2015). Learned policy priors often guide these candidates (Hansen et al., 2022; 2024). In contrast, policy learning methods use world models to generate imagined experience for training, which shifts computation from online decision-making to offline learning. Recent work further blurs this distinction by transferring dynamics priors from large-scale video pre-training (Du et al., 2023; Wu et al., 2024; Cheang et al., 2024; Hu et al., 2025) and joint observation-action generative models (Guo et al., 2024; Li et al., 2025; Zhu et al., 2025) to downstream control. However, across this evolving design space, a central challenge is to characterize how learned world knowledge is utilized throughout the decision-making pipeline. This work examines how predictive modeling interacts with policy execution and how this interplay shapes downstream task performance.

Examining how world knowledge is used raises an open question about the roles of inference-time reasoning and knowledge learned during training. On one hand, sampling-based planning methods retain flexible test-time reasoning, but iterative candidate generation and rollout evaluation impose steep runtime latency, which raises the need for mechanisms that directly learn candidate improvement rather than relying on brute-force evaluation loops. On the other hand, actor architectures trained on imagined experience shift optimization entirely into the training phase. Yet, during deployment, these lookahead-free policies lack an adaptive reasoning mechanism to invest additional computational budget when complicated decisions demand deeper evaluation. Crucially,

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.

while large-scale video pre-training and joint dynamics modeling expand what the model knows, scaling predictive knowledge does not resolve how that knowledge is engaged at inference time. Rather than relying solely on zero-shot policy rollout or costly online search, we use world models to support a learnable decision process with flexible and controllable test-time computation, while making each evaluation step as useful as possible.

Motivated by these observations, we propose reframing decision-making around a learned, iteratively applicable planning operator. Rather than mapping states directly to actions, our approach focuses on refining action proposals based on their predicted outcomes. Our central hypothesis is that action-conditioned representations of the future contain useful signals about the quality of a proposed action, and that a learned refiner can use these signals to make targeted corrections without relying on expensive sampling-based evaluation. A key distinction is that predicting the outcome of an action is not the same as knowing how to improve that action. Accurate forward prediction supports reliable simulation, but does not by itself provide a mechanism for policy improvement. Building on this distinction, we investigate whether jointly training forward outcome prediction and future-conditioned action refinement can connect these two capabilities, using predicted future outcomes as the basis for action updates.

Building on this formulation, we introduce the World Planning Model (WPM), which combines initial action proposal, forward outcome prediction, and action refinement. A key feature of WPM is that the predictive and refinement modules are trained together. The refiner learns to use predicted future outcomes to improve action proposals, while the world model learns task-relevant representations that support these updates while remaining consistent with the underlying dynamics. The key idea is to jointly learn both what future information is useful for decision-making and how that information should be used to modify an action trajectory. As a result, the training objective is not limited to accurate reconstruction or future prediction. It also encourages the predicted representations to provide information that directly supports better action refinement.

A key design goal of WPM is to retain action refinement without the high inference cost of sampling-based planning. Rather than generating and separately scoring many rollout candidates at each refinement step, WPM uses task objectives to directly shape action updates. Predicted future outcomes therefore serve not only to assess whether an action is promising, but also to indicate how it should be improved. During training, the refinement policy learns to translate this feedback into effective updates, that reduces the need for extensive candidate evaluation at test time. We refer to this as implicit selection through learned refinement: part of the search process is learned during training, while the model can still adjust its action iteratively at inference time. This preserves flexible test-time computation without collapsing decision making into a single feedforward prediction.

Under this formulation, WPM supports controllable test-time computation by applying the same learned planning operator for multiple refinement steps. At each step, the model predicts the outcome of the current action proposal and uses this prediction to refine the action for the next iteration. The number of refinement steps can be adjusted at test time without changing the model parameters. This iterative process allows WPM to use additional computation when needed while keeping the model itself fixed. It also provides a direct way to study how refinement depth affects decision quality, including when additional iterations improve performance and when the gains saturate.

Based on the above motivation and formulation, we summarize our main contributions as follows:

- **World and planner co-adaptation.** We propose WPM, which jointly learns action-conditioned prediction and future-conditioned refinement to investigate how grounded predictive representations can support action improvement.
- **Implicit selection through learned refinement.** We develop an integrated proposal-and-correction mechanism that internalizes task-directed preferences within action refinement, providing an alternative to explicit candidate ranking at deployment.
- **Iterative test-time planning.** We formulate a recurrent planning process with fixed model parameters and study how refinement depth affects decision quality and inference cost, including when additional refinement is beneficial and where its gains begin to saturate.
- **Empirical study.** On eight visual OGBench environments, WPM matches or exceeds explicit planners built on the same model on most environments, and its success rises with refinement depth without retraining.

2 PRELIMINARIES

Offline goal-conditioned reinforcement learning. We study goal-conditioned control from reward-free offline data. The environment is a controlled Markov process $(\mathcal{S}, \mathcal{A}, \mathcal{T})$, with states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, and transition kernel $\mathcal{T}(s_{t+1} \mid s_t, a_t)$. A task is given by a goal g through the sparse reward $r(s, g) = \mathbf{1}[s \text{ satisfies } g]$, and the aim is a goal-conditioned policy π that maximizes $J(\pi) = \mathbb{E}_{g,\pi}\big[\sum_t \gamma^t r(s_t, g)\big]$. For visual control, the policy observes an image $x_t = \mathcal{O}(s_t)$ and a goal image g, without access to the underlying environment state. Learning uses a fixed dataset $\mathcal{D}$ of trajectories collected by an unknown behavior policy, with no further interaction with the environment (Levine et al., 2020). Hindsight relabeling (Kaelbling, 1993; Andrychowicz et al., 2017) samples a future observation $g = x_{t'}$, where $t' > t$, from the same trajectory to construct supervised (x_t, g, a_t) examples for goal-conditioned behavior cloning (Ghosh et al., 2021; Ding et al., 2019). Limited coverage makes generalization to unseen state-goal combinations difficult. Moreover, policies may consider actions outside the data distribution, where model and value estimates are unreliable, without being able to collect additional training data to correct these errors.

Explicit and implicit action selection. Explicit planning optimizes actions at inference time using predicted outcomes. Here, $a \in \mathcal{A}$ denotes a single action, while $\mathbf{a} = \{a_0, \ldots, a_{H-1}\} \in \mathcal{A}^H$ denotes a sequence of H actions, with H the planning horizon. A learned dynamics model $\hat{f}(x_t, \mathbf{a})$ predicts the resulting trajectory or future representation, and a scalar utility function $U(\cdot, g)$ scores its desirability, such as predicted return or negative distance to the goal. The planning objective is:

$$\mathbf{a}^\star \in \operatorname*{arg\,max}_{\mathbf{a} \in \mathcal{A}^H} U\big(\hat{f}(x_t, \mathbf{a}),\, g\big). \tag{1}$$

Model predictive control (MPC) approximately solves this optimization, typically with a sampling-based optimizer such as the cross-entropy method (de Boer et al., 2005) or MPPI (Williams et al., 2015), executes the first action or a short prefix, and replans after receiving a new observation.

A conventional policy directly outputs actions from the observation and goal, potentially using world model knowledge acquired during training. We focus on a different form of implicit selection, which uses predictive information during inference while retaining direct action output. A general formulation is:

$$\zeta = \mathcal{I}(x_t, g), \qquad \mathbf{a} \sim \pi(\cdot \mid x_t, g, \zeta), \tag{2}$$

where $\mathcal{I}$ represents inference-time computation using a world model, and ζ contains the resulting predictive information. The policy learns how this information should influence its output, with no explicit argmax over a scalar utility and no separate candidate-ranking step. This formulation allows additional computation to inform the decision beyond a one-shot policy evaluation. The following section specifies how WPM instantiates this process through learned action refinement.

3 METHODOLOGY

We introduce the World Planning Model (WPM), a unified architecture that integrates action proposal, future prediction, and action refinement within a single learned decision process. As shown in Figure 1, the WPM is composed of encoder ϕ_θ, forecaster F_θ, and actor A_θ.

3.1 PROBLEM DEFINITION

Following the offline goal-conditioned setting in Section 2, we seek to learn an iterative decision process that uses predicted outcomes to revise an action proposal before execution. Given a current image x_t and a goal image g, the process first produces a proposal and then updates it through a configurable number of prediction-and-refinement steps. The model parameters remain frozen as the inference budget changes.

We represent each proposal as an action chunk $\mathbf{a} = \{a_0, \ldots, a_{H_\mathrm{a}-1}\} \in \mathcal{A}^{H_\mathrm{a}}$, where a_h is a single environment action and H_a is the chunk length. Chunking is a choice of action representation rather than a requirement of implicit selection. With $H_\mathrm{a} = 1$, the process reduces to single-action refinement. The chunk length does not bound how far ahead the forecaster predicts.

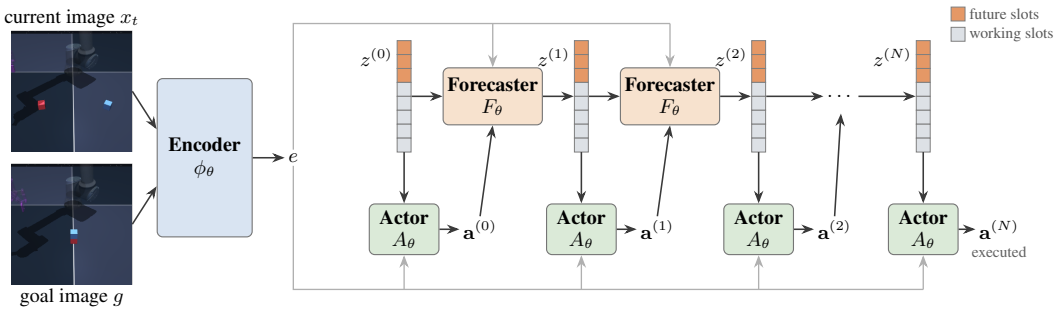


Figure 1: **WPM unrolled over refinement steps.** The encoder ϕ_θ reads the current image and the goal image once and gives the context e, which every block reads. The illustration here is `visual-cube-double`. Starting from a random latent $z^{(0)}$, the actor A_θ turns each latent into a proposal $\mathbf{a}^{(k)}$, and the forecaster F_θ writes the predicted consequence of that proposal into the next latent. Both blocks share their weights across steps, so the number of steps N can be set at test time. Furthermore, nothing is scored or selected.

Design Motivation. As Section 1 argues, predicting an action's consequences does not by itself specify how the action should be changed, so we pair an action-conditioned forecaster with an actor that learns to use predictive information when producing the next proposal. The architecture is designed around two requirements. First, part of the latent state is supervised with future observations, so that the actor receives grounded predictive information, not only unconstrained recurrent features. Second, the same forecaster and actor are reused at every refinement step with a fixed latent structure, allowing the refinement loop to be run for a variable number of steps at test time.

3.2 World Planning Model (WPM)

Overall process. The encoder jointly processes the current observation and goal images into context tokens $e = \phi_\theta([x_t, g]) \in \mathbb{R}^{M \times d_e}$, M tokens of dimension d_e, computed once per decision and read at every refinement step. All model parameters are denoted by θ. The model also maintains an internal latent $z^{(k)} \in \mathbb{R}^{L \times d_z}$ of L slots of dimension d_z, where k indexes refinement steps.

We initialize $z^{(0)} \sim q_{\text{init}}$, every slot is drawn from a standard normal truncated and scaled to unit root-mean-square, the same normalization every slot receives when it is written. The actor then produces an initial distribution over action chunks and a proposal:

$$\pi^{(0)} = A_\theta(z^{(0)}, e), \qquad \mathbf{a}^{(0)} \sim \pi^{(0)}. \tag{3}$$

Each subsequent step uses the current proposal to update the latent, from which the actor produces a revised proposal:

$$\begin{aligned} z^{(k+1)} &= F_\theta(e, z^{(k)}, \mathbf{a}^{(k)}), \\ \pi^{(k+1)} &= A_\theta(z^{(k+1)}, e), \\ \mathbf{a}^{(k+1)} &\sim \pi^{(k+1)}, \qquad k = 0, \dots, N-1. \end{aligned} \tag{4}$$

Here, the model performs N refinement steps before executing $\mathbf{a}^{(N)}$. All steps concern the same decision, since the environment does not advance during refinement. Setting $N = 0$ executes the initial proposal, while larger N allocates more computation without changing the model parameters.

Forecaster. At refinement step k, the forecaster F_θ updates the latent state from the context e, the current latent $z^{(k)}$, and the action proposal $\mathbf{a}^{(k)}$, as in Equation (4). Starting from a random $z^{(0)}$, it rewrites all L slots at each step, with a learned step embedding indicating the refinement index.

We divide the slots into L_z future slots and $L - L_z$ working slots. The j-th future slot is trained against the target encoding of the observation at a predefined offset δ_j in the offline trajectory ($\delta_1 < \cdots < \delta_{L_z}$, some beyond the chunk), so it predicts where the proposal leads. The working slots have no target and serve as recurrent state for refinement. Both are rewritten after every action update.

Actor. The actor A_θ reads the whole latent and the context and outputs a distribution over action chunks, for the initial proposal and every refinement. Selection is implicit. The actor learns how predictive information should change its output and revises its proposal before execution.

3.3 INFERENCE

Explicit planning optimizes candidate action sequences under a utility, as in Equation (1). WPM instead has the actor generate the updated proposal from the latent, implementing Equation (2) with $\zeta = z^{(N)}$ and no ranking or argmax. Algorithm A.1 in Appendix A.4 gives the procedure.

3.4 TRAINING OBJECTIVES

We train the forecaster and actor jointly on offline trajectories, so that the forecaster learns what future information is useful for control and the actor learns how to use it for refinement. For training example b, let $\mathbf{u}_b$ be the recorded action chunk and $f_{b,j} = \bar{\phi}([x_{t+\delta_j}, g])$ the target representation of its future observation at offset δ_j, given by $\bar{\phi}$, a frozen copy of the image encoder ϕ_θ through which no gradient flows. Predictive losses on a separate grounding branch train the forecaster, and an action loss on the refinement loop trains the actor. Appendix A.2 writes every term in full.

Learning action-conditioned predictive representations. The grounding branch runs the forecaster on the recorded chunk, $\hat{z}_b = F_\theta(e_b, z_b^{(0)}, \mathbf{u}_b)$, without the actor, so its losses define what the future slots represent without changing the actor's proposals. We use a contrastive objective (van den Oord et al., 2018) rather than regression, which an average continuation can satisfy. Each future slot must pick out its own observed future among those of the batch, and the objective is formulated as:

$$\mathcal{L}_{\text{ground}} = -\frac{1}{L_z}\sum_{j=1}^{L_z}\frac{1}{W_j}\sum_b w_{b,j}\log\frac{s_{b,j}(f_{b,j})}{\sum_{b'\in\mathcal{N}(b)} s_{b,j}(f_{b',j}) + s_{b,j}(f_b^{\text{now}})}, \tag{5}$$

where $s_{b,j}(f) = \exp\big(c(\hat{z}_{b,j}, f)/\tau\big)$, c is the cosine similarity and τ a temperature. $\mathcal{N}(b)$ holds b's target and negatives from the batch, excluding nearby samples of the same trajectory, and the current observation $f_b^{\text{now}} = \bar{\phi}([x_t, g])$ is one more negative, encoded in the space of the targets, so that copying the present is penalized. The weights $w_{b,j}$, normalized by $W_j = \sum_b w_{b,j}$, favor futures that make progress toward the goal under a frozen goal-conditioned value function, as in advantage-weighted regression (Peng et al., 2019). The value function is used only for these weights. Because the recorded chunk is often predictable from the observation and goal, the forecaster could still ignore its action input, so an action-contrast loss $\mathcal{L}_{\text{ac}}$ requires the prediction under the recorded chunk to match the observed future better than the prediction under a chunk $\mathbf{u}_{\sigma(b)}$ of another example, which a forecaster that ignores the action cannot do. See Appendix A.2 for details.

Learning action generation from predictive information. $\mathcal{L}_{\text{read}}$ supervises the initial proposal and each of the N_b refinement steps sampled for example b with the recorded chunk:

$$\mathcal{L}_{\text{read}} = \mathbb{E}_b\Big[\,\text{NLL}\big(\pi_b^{(0)}, \mathbf{u}_b\big) + \textstyle\sum_{k=1}^{N_b}\rho_k\,\text{NLL}\big(\pi_b^{(k)}, \mathbf{u}_b\big)\Big], \qquad \rho_k \propto \tfrac{1}{\Pr[N_b \geq k]}, \tag{6}$$

where $\text{NLL}(\pi, \mathbf{u})$ is the negative log-likelihood of the recorded chunk under the actor's discretized action distribution, averaged over its scalars, and ρ_k gives the rarer deep steps comparable weight. See Appendix A.2 for the per-bin form of NLL and for ρ_k. The target is the recorded chunk at every step, so the $N = 0$ proposal remains a goal-conditioned behavior-cloning policy and every further step is asked for the same answer. This does not by itself make the actor use the future slots. Early in training they carry almost no information, so an actor trained only on them learns to ignore them. The observed future, in contrast, is what the recorded chunk led to, so reading it lowers $\mathcal{L}_{\text{read}}$. We therefore write it over the future slots at the first refinement step for a random fraction p_{post} of training examples, setting $z_{b,j}^{(1)} \leftarrow m_b f_{b,j} + (1 - m_b)\, z_{b,j}^{(1)}$ with $m_b \sim \text{Bernoulli}(p_{\text{post}})$.

From observed to predicted futures. Observed futures are unavailable at deployment. To overcome this, we design a two-stage training recipe, described in Section 3.5. The first stage substitutes the observed future on every refined example. The second lowers p_{post}, so the actor trains mostly on predicted futures, and aligns them with the observed ones through the prior–posterior loss:

$$\mathcal{L}_{\text{pp}} = \frac{\sum_b m_b \frac{1}{L_z}\sum_{j=1}^{L_z}\big(1 - c\big(z_{b,j}^{\text{pr}},\, \text{sg}(f_{b,j})\big)\big)}{\sum_b m_b}, \tag{7}$$

where the prior $z_b^{\mathrm{pr}} = F_\theta(e_b, z_b^{(0)}, \mathrm{sg}(\mathbf{a}_b^{(0)}))$ is the forecaster's first-step prediction before substitution, the posterior $f_{b,j}$ is the observed future, and m_b keeps only the substituted examples.

The stop-gradient sg keeps $\mathcal{L}_{\mathrm{pp}}$ a forecasting loss. Without it, gradients from $\mathcal{L}_{\mathrm{pp}}$ could reach the actor through the forecaster's action input. The actor could then reduce prediction error by changing its proposal toward an action that makes the forecaster reproduce the observed future, effectively learning an inverse dynamics model of the forecaster. This creates a different learning signal from the intended refinement process. We want the forecaster to learn the consequences of a proposed action, and the actor to use those predicted consequences to improve the proposal under action supervision. Because the forecaster is imperfect, such actions can exploit its errors and need not stay close to the actions that actually produced the observed futures in the data. As a result, we stop the predictive gradient at the proposal, while $\mathcal{L}_{\mathrm{read}}$ keeps the actor anchored to the offline actions and trains it to use predictive information for refinement. Appendix A.2 traces the gradient.

3.5 Training procedure

We separate training into two stages so that the actor first learns to use informative future features, then adapts to the forecaster's own predictions. Training the full iterative loop from the outset risks the actor learning to ignore inaccurate future slots before the forecaster can provide useful predictions. The two stages optimize

$$\mathcal{L}_{\mathrm{stage\,1}} = \mathcal{L}_{\mathrm{read}} + \lambda_{\mathrm{ground}}\mathcal{L}_{\mathrm{ground}} + \lambda_{\mathrm{ac}}\mathcal{L}_{\mathrm{ac}}, \tag{8}$$

$$\mathcal{L}_{\mathrm{stage\,2}} = \mathcal{L}_{\mathrm{read}} + \lambda_{\mathrm{pp}}\mathcal{L}_{\mathrm{pp}} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}}. \tag{9}$$

Stage 1: learning to predict and use observed futures. We initialize from a converged goal-conditioned behavior-cloning policy and freeze its image encoder. Each example receives at most one refinement step, with $p_{\mathrm{post}} = 1$, so whenever refinement occurs, the actor reads observed-future features in the future slots. The action loss $\mathcal{L}_{\mathrm{read}}$ trains the actor to use these features to predict the recorded chunk. In parallel, the separate grounding branch trains the forecaster on recorded actions and their observed futures using $\mathcal{L}_{\mathrm{ground}}$ and $\mathcal{L}_{\mathrm{ac}}$.

Stage 2: learning to refine actions using predicted futures. We freeze the encoder ϕ_θ and the target encoder $\bar{\phi}$, lower p_{post} from 1 to 0.25, and train the loop over multiple refinement steps. The actor now mostly reads model-generated futures and learns through $\mathcal{L}_{\mathrm{read}}$ to use them for action refinement. An auxiliary loss $\mathcal{L}_{\mathrm{cons}}$ also trains the actor's representation to tell whether the future block it reads belongs to its own example or to another example of the minibatch.

In both stages, we sample the number of refinement steps N_b for each example rather than fixing it, so the depth can be varied at deployment. Appendix A.2 defines $\mathcal{L}_{\mathrm{cons}}$ and the depth schedule.

4 Experiments

Our experiments validate the claims and hypothesis of Section 1. **(Q1) Co-adaptation.** Does learning to predict the future, jointly with refinement, improve the policy, and does the actor use the futures it predicts? **(Q2) Implicit selection.** Does learned refinement compare favorably with explicit planning, whether it selects among the same model's candidates or plans over a separately trained world model, and OGBench's published methods? **(Q3) Test-time scaling.** Does success rise with refinement depth without retraining, and where does it saturate?

4.1 Setup

Benchmark and protocol. We use eight visual OGBench environments (Park et al., 2025a) with their own offline datasets, 1 001-step trajectories of 64×64 RGB frames with no goal or reward labels. Table A.5 lists each dataset with its size and our settings. Every number is a success rate under OGBench's evaluation protocol, five goals with 50 episodes each (250 episodes), on which nothing is tuned. Each row reports the mean and standard deviation of its three training seeds, at fixed checkpoints, and Appendix B.3 gives the protocol.

As in recent visuomotor policies (Zhao et al., 2023; Chi et al., 2023), the actor emits a chunk of $H_{\mathrm{a}} = 8$ actions over 256 bins per scalar, which Table A.8 compares with a Gaussian head. The

Table 1: **Effect of future-prediction supervision during training.** Success (%) on cube-double when training includes $\mathcal{L}_{\text{ground}}$ and $\mathcal{L}_{\text{ac}}$ (with) or omits both losses (without), evaluated at refinement depth N. We report mean $\pm$ standard deviation over seeds. Δ denotes the improvement from adding the two losses (with minus without).

depth	with $\mathcal{L}_{\text{ground}}, \mathcal{L}_{\text{ac}}$	without	Δ
$N = 0$	**49.3** ± 5.3	42.0 ± 3.5	$+7.3$
$N = 4$	**54.5** ± 8.3	37.9 ± 1.6	$+16.7$

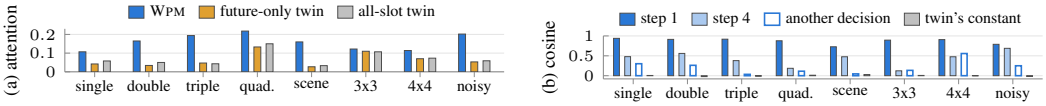


Figure 2: **Inside the loop.** For each environment in Table 2, we evaluate one checkpoint on held-out decisions. (a) Fraction of the action query's attention on the three future slots at $N = 4$. WPM fills these slots with learned future predictions. The future-only twin replaces only the future slots with fixed constants, and the all-slot twin replaces all eight slots. (b) Cosine similarity of the $+8$ slot to the embedding of the frame observed eight steps later. We compare WPM after refinement steps 1 and 4 (blue), the step-1 prediction against another decision's future (hollow, a cross-decision control), and the all-slot twin's constant (gray).

latent has $L = 8$ slots, the first $L_z = 3$ at offsets $\delta = (8, 16, 24)$, and Table A.4 lists the other settings. Policies execute the mean of the action distribution except on scene and puzzle, where that mean averages distinct button choices and we execute a sample at temperature 0.5.

Baselines. Three explicit planners solve Equation (1) over candidates drawn from the $N = 0$ proposal of the WPM checkpoint, which isolates the selection rule. Q-argmax keeps the best of eight chunks under a critic $Q(x_t, \mathbf{a}, g)$ trained with GCIQL on the same data, and CEM refits the proposal against it, so neither uses a world model. The imagined-futures planner scores the forecast future of each of eight chunks with a value head V_ψ distilled from the frozen GCIVL value (Park et al., 2025a), so it uses WPM's futures but ranks them instead of reading them. TD-MPC2 (Hansen et al., 2024) and LeWorldModel (Maes et al., 2026) are trained offline on the same data. Planners execute a sample at temperature 0.5, and Appendix B.4 gives their settings.

4.2 Co-adaptation of prediction and refinement

The future-prediction loss is a training-time effect. Stage 1 trains $\mathcal{L}_{\text{ground}}$ so that the future slots predict the observed future, and $\mathcal{L}_{\text{ac}}$ so that this prediction depends on the proposed chunk. Table 1 removes only these two losses. Success falls at both depths, and the gap grows from 7.3 points at $N = 0$ to 16.7 at $N = 4$, since refinement adds 5.2 points with the losses and costs 4.1 without them. Extra steps help only when the slots hold the consequence of the proposal, so the forecasting losses, not the added computation, make deeper refinement useful, and they still add 2.2 points on average at the fixed checkpoints, as Appendix C.2 shows.

The actor reads its future slots. If refinement uses the forecasts, changing them should change the decision, so the last two rows of Table 2 write other futures into the slots of WPM at $N = 4$. A plausible future of a wrong goal lowers success on all eight environments, by 15.7 to 72.3 points, so the actor follows its slots even against its goal. The true future at step 1 helps on the two cube-double environments and hurts on scene and puzzle-4x4, each at every seed. The slots are thus part of the decision, and a usable future helps more than a merely correct one.

What the future slots hold. Figure 2 tests whether the actor uses its future slots and whether they hold the realized future, against the two twins of Appendix C.2. At $N = 4$, the action query puts 11–22 % of its attention on the three future slots, more than either twin puts on its constants in every environment, though only by 1.2 points on puzzle-3x3, so the attention follows the forecasts rather than the slot positions. The first $+8$ prediction has cosine 0.73–0.94 with the embedding of the observation eight steps later, 0.35–0.88 higher than with another decision's future, so it carries decision-specific information. By step 4 the similarity falls to 0.12–0.69, since only the first prediction is trained against the observed future, yet success still rises with depth. Similarity to the observed future therefore does not guarantee a better decision, as the true-future row of Table 2 also shows. Appendix D extends this analysis to all future slots.

Table 2: **Implicit refinement against explicit planning.** Success (%) on OGBench's protocol, mean ± standard deviation over the training seeds of §4.1. Top: published OGBench results. Middle: WPM's proposal ($N = 0$), three planners that sample from it, and two world-model planners trained on our data, each at its best checkpoint. The last two rows are the diagnostics of §4.2. –: not published.

	cube (play)				scene	puzzle (play)		cube (noisy)
method	single	double	triple	quadruple	play	3x3	4x4	double
GCBC (Ding et al., 2019)	5 ±1	1 ±1	15 ±2	8 ±1	12 ±2	0 ±0	10 ±1	–
GCIVL (Park et al., 2025a)	60 ±5	10 ±2	14 ±2	0 ±0	25 ±3	21 ±1	60 ±5	–
GCIQL (Kostrikov et al., 2022)	30 ±5	1 ±1	15 ±1	7 ±1	12 ±2	1 ±2	16 ±4	–
QRL (Wang et al., 2023)	41 ±15	5 ±0	16 ±1	5 ±1	10 ±1	1 ±1	0 ±0	–
CRL (Eysenbach et al., 2022)	31 ±15	2 ±1	17 ±2	4 ±1	11 ±2	0 ±0	10 ±6	–
HIQL (Park et al., 2023)	**89** ±0	39 ±2	21 ±0	14 ±1	49 ±4	**73** ±8	60 ±41	–
proposal alone ($N = 0$)	87.6 ±2.1	50.9 ±1.3	21.0 ±0.3	18.1 ±0.2	61.2 ±1.8	49.3 ±2.8	64.1 ±3.1	36.3 ±0.2
Q-argmax over 8 proposals	86.7 ±3.0	51.9 ±1.8	**23.6** ±1.1	16.1 ±0.2	61.3 ±1.0	48.3 ±3.6	65.3 ±1.5	25.7 ±4.2
CEM over proposals	80.3 ±0.5	50.9 ±0.6	21.0 ±1.4	15.1 ±1.0	60.3 ±1.2	42.9 ±0.9	57.3 ±2.7	30.0 ±2.2
planner over WPM's imagined futures	87.9 ±1.2	**54.3** ±2.2	22.0	16.0 ±1.8	62.7 ±0.8	52.4 ±2.9	68.5 ±2.9	26.1 ±1.4
TD-MPC2 (Hansen et al., 2024)	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.8 ±1.4	0.4 ±0.7	0.0
LeWorldModel + CEM (Maes et al., 2026)	0.1 ±0.2	0.0 ±0.0	2.0	0.0 ±0.0	0.0 ±0.0	0.4	0.0	0.4
WPM, $N = 4$ (ours)	88.8 ±0.4	52.3 ±0.6	20.6 ±0.8	**18.3** ±0.2	**63.8** ±0.5	50.1 ±2.9	**72.5** ±2.0	**41.1** ±1.4
WPM, $N = 4$, wrong-goal future	16.5 ±9.9	18.5 ±0.2	4.0 ±1.1	2.5 ±1.5	40.9 ±12.1	3.1 ±1.5	3.2 ±2.8	11.9 ±0.8
WPM, $N = 4$, true future at step 1	89.1 ±2.3	64.9 ±2.9	21.8 ±2.0	15.1 ±3.7	53.0 ±3.5	45.2 ±5.6	22.4 ±3.2	77.1 ±1.6

4.3 IMPLICIT SELECTION AGAINST EXPLICIT PLANNING

In Table 2, WPM improves on its own $N = 0$ proposal in seven of eight environments except for cube-triple. By contrast, explicit selection with Q-argmax, CEM, or WPM's imagined futures improves the proposal less consistently and often degrades it. Among the methods that start from this proposal, WPM therefore attains the highest mean success on five of eight environments, despite refining a single proposal rather than sampling and selecting among multiple candidates. Selection inherits the errors of its learned score. CEM, which optimizes that score hardest, falls below the proposal on six environments, while WPM optimizes no score and learns from the task loss how to use a forecast. This suggests that predicted outcomes can be used effectively inside the policy computation itself, instead of being externalized into a separate scoring-and-selection procedure.

Imagined-future planning retains an advantage on cube-double and puzzle-3x3, and Q-argmax on cube-triple. We additionally evaluate TD-MPC2 (Hansen et al., 2024) and LeWorldModel (Maes et al., 2026), which we re-implemented from the authors' papers and code, keeping their defaults where our setting allows, as Appendix B.4 describes. Both succeed on at most 2 % of episodes. Appendix C.5 shows both train as specified. LeWorldModel's planner is also 6 to 15 times slower than WPM, the inference cost that Section 1 set out to avoid. Overall, the comparison supports implicit forecast-guided refinement as a simpler and more reliable alternative to explicit test-time planning in this setting.

4.4 DEPTH AS A TEST-TIME SCALING AXIS

Because training samples the depth, Table 3 evaluates the same checkpoints from Table 2 at refinement depths $N = 0$ through 8, without retraining. Every environment improves over $N = 0$ at some nonzero depth, by 0.4 to 8.3 points at its best depth, the Δ of Table 3, although the gains are not monotonic. Overall, most of the benefit appears within the first few refinement steps, after which performance tends to saturate or decline slightly. The actor's action distribution likewise settles after two steps, as Appendix D.1 shows. Refinement depth therefore provides a test-time compute–performance tradeoff, but its optimal setting is task-dependent rather than universally increasing with depth. Each step adds one forecaster and actor pass, so $N = 4$ takes 3.4 to 5.3 times as long as $N = 0$, the cost Section 1 set out to study, as Table A.11 and Appendix C.3 detail. Even at $N = 4$, WPM is 6 to 15 times faster than LeWorldModel's planner, though TD-MPC2 and the batched candidate planners are about as fast or faster.

4.5 QUALITATIVE ANALYSIS

Figure 3 visualizes the first +8 forecast at selected decisions along a successful scene episode. As the proposed actions move the arm and manipulate the cube, the forecast changes accordingly and

Table 3: **Depth is a test-time scaling axis.** Success (%) at depths $N = 0$ to 8 using the same trained checkpoints. Δ is the gain of the best depth, in parentheses, over $N = 0$, with the depth picked on these episodes. Bold marks a gain that is positive at every seed. †Fewer than three seeds. ‡Fewer seeds than the rest of its row.

environment	$N = 0$	$N = 1$	$N = 2$	$N = 4$	$N = 8$	Δ
cube single	87.6 ±2.1	89.6 ±1.1	89.2 ±1.4	88.8 ±0.4	87.9 ±2.2	+2.0 (1)
cube double	50.9 ±1.3	51.3 ±0.6	54.0 ±0.8	52.3 ±0.6	51.9 ±1.2	**+3.1** (2)
cube triple†	21.0 ±0.3	21.2 ±0.0	21.2 ±0.6	20.6 ±0.8	22.2 ±0.8	**+1.2** (8)
cube quadruple	18.1 ±0.2	18.5 ±1.2	18.0 ±0.7	18.3 ±0.2	18.3 ±0.6	+0.4 (1)
scene	61.2 ±1.8	63.5 ±0.8	62.5‡ ±2.5	63.8 ±0.5	63.9 ±0.8	+2.7 (8)
puzzle 3x3	49.3 ±2.8	50.4 ±1.2	50.1 ±1.0	50.1 ±2.9	49.3 ±2.7	+1.1 (1)
puzzle 4x4	64.1 ±3.1	68.5 ±1.1	70.7 ±1.2	72.5 ±2.0	70.4 ±0.5	**+8.3** (4)
cube double, noisy	36.3 ±0.2	40.4 ±3.4	40.9 ±2.7	41.1 ±1.4	40.9 ±1.8	**+4.8** (4)

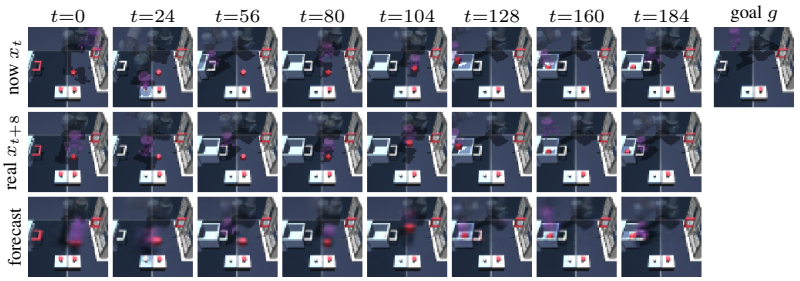


Figure 3: **What a future slot holds.** We show selected time points from a scene episode that WPM solves in 195 steps. Predicted slots are decoded only for visualization, and the policy operates on their latent representations directly.

anticipates the resulting scene eight steps later. In particular, it reflects where the arm and cube will move rather than simply reproducing their current configuration. These results indicate that the future slot captures the consequences of the proposed action rather than merely encoding the current observation. Appendix D.4 shows later refinement steps and an additional episode.

5 Conclusion

WPM turns planning into a learned refinement loop, in which an actor revises its proposal from a forecast of its consequence, with no candidate scored or selected. Trained jointly on offline data, the loop matches or exceeds explicit planners built on the same model on most visual OGBench environments, its success rises with depth without retraining, making depth an axis for test-time scaling, and it succeeds less often without its future-prediction losses. Appendix E discusses what these results do and do not yet show.

6 Related Work

World models in offline RL. Learned dynamics models support control either by planning over action sequences with MPC (Chua et al., 2018; Hafner et al., 2019; Hansen et al., 2024; 2026), often in learned latent spaces (Zhou et al., 2025; Maes et al., 2026), or by training policies on imagined rollouts (Hafner et al., 2020; 2025; Janner et al., 2019). In offline settings, model-based methods additionally constrain model use to reduce distribution shift (Yu et al., 2020; 2021), and training a model for predictive accuracy does not align it with how a planner uses it (Lambert et al., 2020; Grimm et al., 2020). Offline goal-conditioned RL instead learns policies or values directly from logged data (Kostrikov et al., 2022; Eysenbach et al., 2022; Park et al., 2023; 2025b).

Predicted futures have also been used explicitly for control, including as rollout trajectories, subgoals, or joint action-future predictions (Du et al., 2023; Black et al., 2024; Wu et al., 2024; Li et al., 2025; Zhang et al., 2026b). Most such methods generate a future prediction once and then use it for action generation or evaluation. Methods that allocate more computation at test time instead sample and verify multiple candidates (Kwok et al., 2026; Kim et al., 2026), iteratively refine a latent state of the action head without explicit future prediction (Tur et al., 2026), or refine plans against a fixed world model (Yang & Shan, 2026). WPM differs by placing prediction inside the policy's own refinement loop. Appendix F discusses related work in more detail.

REPRODUCIBILITY STATEMENT

Every architectural and optimization detail needed to rebuild the model is in §3 and Appendix A.2, which lists each loss term with its target, trained parameters and coefficient as the code runs it. The evaluation protocol (OGBench's episode list, the pairing and bootstrap procedure, the renderer pinning and the two-pass determinism rule) is in Appendix B.3, and the data pipeline and horizon rules are in Appendix B.2. All environments and datasets are OGBench's own (Park et al., 2025a). Every number in the paper is generated by scripts from the logged evaluations. The code and the configuration of every run are in the anonymous repository of Appendix B.

AI USE STATEMENT

In this work, we used generative AI tools to help write the code and to revise this paper, including its text, figures and tables. We have reviewed all AI-assisted code and revisions. Every number in the paper is generated by scripts from the logged evaluations of our own runs, and new references were checked against their venue pages. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Bhavya Kumar Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based RL. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

Hongjoon Ahn, Heewoong Choi, Jisu Han, and Taesup Moon. Option-aware temporally abstracted value for offline goal-conditioned reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2025.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 58757–58791, 2024. doi: 10.52202/079017-1873.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Marco Bagatella, Mert Albaba, Jonas Hübotter, Georg Martius, and Andreas Krause. Test-time offline reinforcement learning on goal-related experience. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026a.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026b.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 690–701, 2019.

Shuanghao Bai, Jing Lyu, Wanqi Zhou, Zhe Li, Dakai Wang, Lei Xing, Xiaoguang Zhao, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Badong Chen, and Shanghang Zhang. Latent reasoning VLA: Latent thinking and prediction for vision-language-action models. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research (TMLR)*, 2024.

Chethan Bhateja, Derek Guo, Dibya Ghosh, Anikait Singh, Manan Tomar, Quan Vuong, Yevgen Chebotar, Sergey Levine, and Aviral Kumar. Robotic offline RL from internet videos via value-function learning. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623, 2024.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2021.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jacob Varley, Alex Irpan, Benjamin Eysenbach, Ryan C. Julian, Chelsea Finn, and Sergey Levine. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2021.

Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Feilong Tang, Donglin Wang, and Haoang Li. Unified diffusion VLA: Vision-language-action model via joint discrete denoising diffusion process. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

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 *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Jinwoo Choi, Sang-Hyun Lee, and Seung-Woo Seo. Chain-of-goals hierarchical policy for long-horizon offline goal-conditioned RL. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 4754–4765, 2018.

Hojun Chung, Junseo Lee, and Songhwai Oh. Offline reinforcement learning with universal horizon models. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Ann. Oper. Res.*, 134:19–67, 2005.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2019.

Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 15298–15309, 2019.

Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2023.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. RvS: What is essential for offline RL via supervised learning? In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Ben Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

Amir-Massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494, 2017.

Gregory Farquhar, Tim Rocktäschel, Maximilian Igl, and Shimon Whiteson. TreeQN and ATreeC: Differentiable tree-structured models for deep reinforcement learning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2786–2793, 2017. doi: 10.1109/ICRA.2017.7989324.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2016.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2019.

Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. Generalized Decision Transformer for offline hindsight information matching. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2025.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the gap between TD learning and supervised learning – a generalisation point of view. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442, 2023.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Christopher Grimm, André Barreto, Satinder P. Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5541–5552, 2020.

Arthur Guez, Mehdi Mirza, Karol Gregor, Rishabh Kabra, Sébastien Racanière, Théophane Weber, David Raposo, Adam Santoro, Laurent Orseau, Tom Eccles, Greg Wayne, David Silver, and Timothy Lillicrap. An investigation of model-free planning. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2464–2473, 2019.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2024.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Learning massively multitask world models for continuous control. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P. van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2025.

Chi-Pin Huang, Yunze Man, Zhiding Yu, Min-Hung Chen, Jan Kautz, Yu-Chiang Frank Wang, and Fu-En Yang. Fast-ThinkAct: Efficient vision-language-action reasoning via verbalizable latent planning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jinu Hyeon, Woobin Park, Hongjoon Ahn, and Taesup Moon. Action-sufficient goal representations. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 12519–12530, 2019.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2022.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*, 2025.

Leslie Pack Kaelbling. Learning to achieve goals. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 1993.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018.

Jacky Kwok, Xilun Zhang, Mengdi Xu, Yuejiang Liu, Azalia Mirhoseini, Chelsea Finn, and Marco Pavone. Scaling verification can be more effective than scaling policy learning for vision-language-action alignment. In *Proc. European Conf. on Computer Vision (ECCV)*, 2026.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770, 2020.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2020.

Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 2022.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Robotics: Science and Systems (RSS)*, 2026a.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled Q-chunking. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026b.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Proc. Conf. on Robot Learning (CoRL)*, 2019.

Jason Yecheng Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. Offline goal-conditioned reinforcement learning via f-advantage regression. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: towards universal visual reward and representation via value-implicit pre-training. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *CoRR*, abs/2603.19312, 2026.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Naoki Morihira, Amal Nahar, Kartik Bharadwaj, Yasuhiro Kato, Akinobu Hayashi, and Tatsuya Harada. R2-Dreamer: Redundancy-reduced world models without decoders or augmentation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Ashvin V. Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2018.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proc. Conf. on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909, 2022.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-JEPA: Learning world models through object-level latent masking. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive RL. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in Atari games. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2015.

Evgenii Opryshko, Junwei Quan, Claas Voelcker, Yilun Du, and Igor Gilitschenski. Test-time graph search for goal-conditioned reinforcement learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Kwanyoung Park, Seohong Park, Youngwoon Lee, and Sergey Levine. Scalable offline model-based RL with action chunks. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026a.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2023.

Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline RL? In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2024a.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39737–39761, 2024b.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned RL. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025a.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes RL scalable. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2025b.

Seohong Park, Deepinder Mann, and Sergey Levine. Dual goal representations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026b.

Seohong Park, Aditya Oberai, Pranav Atreya, and Sergey Levine. Transitive RL: Value learning via divide and conquer. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026c.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Dean A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 1988.

Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task-agnostic offline reinforcement learning. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2015.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

Dikshant Shehmar, Matthew Kyle Schlegel, Matthew E. Taylor, and Marlos C. Machado. Laplacian representations for decision-time planning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, André Barreto, and Thomas Degris. The Predictron: End-to-end learning and planning. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3191–3199, 2017.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

Vlad Sobal, Wancong Zhang, Kynghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2025.

Gwanwoo Song, Kwanyoung Park, and Youngwoon Lee. Chunk-guided Q-learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Jonathan Spieler and Sven Behnke. Dream-MPC: Gradient-based model predictive control with latent imagination. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4732–4741, 2018.

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. In *Proc. European Conf. on Computer Vision (ECCV)*, 2026.

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, volume 29, pp. 2154–2162, 2016.

Mohammad Taufeeque, Philip Quirke, Maximilian Li, Chris Cundy, Aaron David Tucker, Adam Gleave, and Adrià Garriga-Alonso. Planning in a recurrent neural network that plays Sokoban. *arXiv preprint arXiv:2407.15421*, 2024.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

Yalcin Tur, Jalal Naghiyev, Haoquan Fang, Wei-Chuan Tsai, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Recurrent-depth VLA: Implicit test-time compute scaling of vision-language-action models via latent iterative reasoning. *arXiv preprint arXiv:2602.07845*, 2026.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Aravind Venugopal, Jiayu Chen, Xudong Wu, Chongyi Zheng, Benjamin Eysenbach, and Jeff Schneider. Occupancy reward shaping: Improving credit assignment for offline goal-conditioned reinforcement learning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control using covariance variable importance sampling. *CoRR*, abs/1509.01149, 2015.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

Zhihui Xie, Zichuan Lin, Deheng Ye, Qiang Fu, Wei Yang, and Shuai Li. Future-conditioned unsupervised pretraining for Decision Transformer. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2023.

Bochen Yang and Lianlei Shan. PearlVLA: Progressive embodied action-plan refinement in latent space. *arXiv preprint arXiv:2606.17924*, 2026.

Liu Yang, Kangwook Lee, Robert Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024a.

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 *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Sherry Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. Dichotomy of control: Separating what you can control from what you cannot. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024b.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative offline model-based policy optimization. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026a.

Zhuoyang Zhang, Shang Yang, Qinghao Hu, Luke J. Huang, James Hou, Yufei Sun, Yao Lu, and Song Han. ForeAct: Steering your VLA with efficient visual foresight planning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Bill Chunyuan Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Scaling goal-conditioned reinforcement learning with multistep quasimetric distances. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2026.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2023.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: world models on pre-trained visual features enable zero-shot planning. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2Goal: From world model to general goal-conditioned policy. In *Robotics: Science and Systems (RSS)*, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems (RSS)*, 2025.