MoMo: Conditioned Contrastive Representation Learning for Preference-Modulated Planning

Anonymous authors
Paper under double-blind review

Abstract

Temporally contrastive representation learning induces a latent structure that reduces long-horizon planning to inference in a low-dimensional linear system. However, existing contrastive planning work learns a single latent geometry that cannot distinguish among multiple valid behaviors, thereby preventing it from trading task efficiency against risk exposure for the same start-goal query. We introduce MoMo, a *preference-conditioned* contrastive planner allowing a scalar user preference to continuously modulate plan conservativeness at inference time, without retraining. MoMo learns a joint conditioning of the representation geometry and latent prediction operator via Feature-Wise Linear Modulation and low-rank neural modulation, respectively. We show that our formulation preserves the probability density ratio encoded in the representation space, which is required for inference-driven contrastive planning, while retaining its inference-time efficiency. Across six environments, MoMo smoothly adapts plan safety to user preferences, yielding improved temporal and preferential consistency compared to state-augmentation baselines.

1 Introduction

Robotic systems are increasingly deployed in safety-critical domains such as healthcare and disaster response, where behaving either too aggressively or too conservatively can be costly (Bolli & Asada, 2025; Cani et al., 2025; Leonard et al., 2014). For instance, a surgical robot may maintain greater clearance during a delicate phase of the operation, but favor a more direct maneuver when rapid intervention is required. In these settings, the same start and goal can admit multiple viable trajectories that trade task efficiency against risk exposure, as illustrated in Figure 1. This balance is not fixed *a priori* but varies with the mission context, ultimately reflecting a specified safety-efficiency planning trade-off. This creates a need for *preference-conditioned* planning, whereby a planner flexibly reshapes its behavior during inference, without retraining, in response to a safety preference that encodes the user's risk tolerance.

Existing approaches to risk-aware planning span model-based methods depending on known dynamics (Blackmore & Ono, 2009; Ono & Williams, 2008; Dai et al., 2019), learning-based methods with fixed risk specifications (Feng et al., 2025; Santara et al., 2017), and recent risk-conditioned policies that allow inference-time adaptation (Huang et al., 2021; Yoo et al., 2024; Zhang et al., 2023). Despite this progress, the intersection of inference-time preference control, planning under unknown dynamics, and scalability to high-dimensional or long-horizon settings remains underexplored. Contrastive Representation Learning for Planning (CRLP) offers a foundation for addressing this gap (Eysenbach et al., 2023; 2025). By learning representations from trajectory data whose geometry reflects temporal reachability, the latent structure reduces long-horizon planning to low-dimensional linear inference (Eysenbach et al., 2025), and has shown viability in robotic settings (Zheng et al., 2025b;a). This makes CRLP effective when dynamics are unknown and efficient inference-time planning is required. Existing contrastive planners, however, are preference-agnostic, encoding only temporal similarity and recovering a single behavioral mode for a given task. Extending CRLP to support preference-conditioning is nontrivial, as a preference-conditioned variant must preserve this structure while allowing it to vary meaningfully with user preferences. We approach this challenge by considering behavioral differences that arise at two levels: the organization of state representations

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

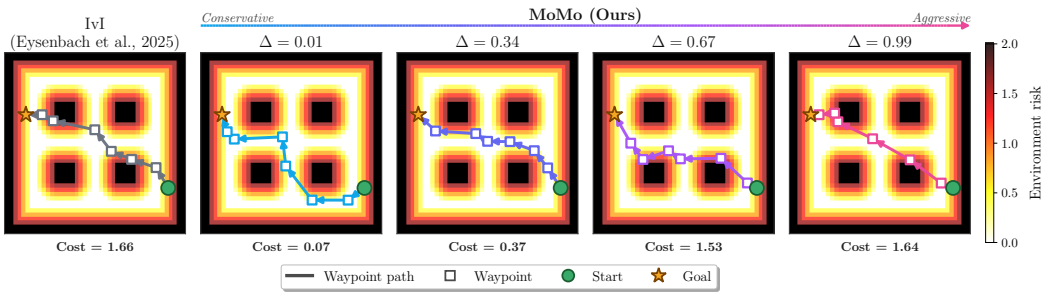

Figure 1: Preference-conditioned planning in a risk-structured environment with four obstacles and a cost function reflecting obstacle proximity. Different scalar safety preferences (Δ) modulate waypoint plans, ranging from low-risk detours to more direct, riskier routes.

and the transitions between them. Conditioning representations yields a preference-aware geometry, while conditioning transitions reshapes how trajectories evolve under varying preferences. To unify these components, we introduce MoMo, a preference-conditioned contrastive planner that jointly conditions the representation geometry and transition operator, enabling inference-time behavioral modulation while preserving the planning structure of CRLP. Training MoMo requires trajectory data with diverse behavioral coverage. Therefore, we curate training datasets offering rich coverage of the joint state-preference space for this purpose. In summary, our contributions are:

1. **Problem formulation.** We contextualize preference-conditioned planning within the CRLP framework and identify a conditioning scheme that preserves its inference-driven structure.
2. **Architecture.** We introduce MoMo, which modulates planning behaviors by jointly conditioning the encoder and the latent transition operator.
3. **Empirical results.** We show that MoMo enables behavioral modulation of plans in response to user preference across six environments spanning a diverse set of tasks, while retaining the inference-time planning efficiency.

2 Related Work

Model-Based Risk-Aware Planning. Classical model-based methods provide explicit safety mechanisms but rely on known dynamics. Chance-constrained trajectory optimization enforces user-specified bounds on failure probability, allocating risk across the planning horizon (Blackmore & Ono, 2009; Ono & Williams, 2008; Dawson et al., 2020). Risk-aware control bounds tail risk through conditional value-at-risk (CVaR) constraints (Hakobyan et al., 2019), while control barrier functions filter inputs to maintain safety invariants (Vahs et al., 2023). These approaches enable safety guarantees but assume known, often linearized dynamics and become expensive in long-horizon settings, motivating learned alternatives that remove the dependence on explicit dynamics models.

Adaptive Safe Policy Learning. Learning-based methods relax dependence on known dynamics by extracting behaviors from trajectory data. Feng et al. (2025) combine graph-based waypoint planning with safe goal-conditioned RL policies for multi-agent navigation, enforcing safety through graph pruning (Parimi & Williams, 2026). Huang et al. (2021) and Yoo et al. (2024) introduce risk-conditioned policies enabling inference-time adaptation across a continuous risk spectrum but rely on direct policy conditioning, with limited scalability over longer horizons. Offline decision-transformer approaches like (Liu et al., 2023) condition on desired return and constraint thresholds for zero-shot adaptation across cost budgets, extending to signal temporal logic specifications (Guo et al., 2024) and constraint-conditioned actor-critic learning (Guo et al., 2025). Benchmark suites like DSRL (Liu et al., 2024) have accelerated progress, but center on safety-constrained control rather than granular preference modulation. Mao et al. (2024) follow a different approach, learning conditioned terrain costmaps for downstream planning. These methods address risk adaptability at the policy or costmap level, rather than via learned representations that support inference-driven planning.

Preference-Conditioned and Multi-Objective RL. Multi-objective RL addresses tradeoffs by learning policies or value functions conditioned on preferences over vector-valued returns (Yang et al.,

2019; Cai et al., 2023). Offline variants such as PEDA, together with the D4MORL benchmark, extend this to preference-conditioned decision transformers trained on annotated demonstrations (Zhu et al., 2023), while recent work offers coverage of the Pareto front for unseen preference adaptation (Liu et al., 2025). These methods cast the problem as optimization over vector-valued returns, whereas our setting studies how different safety preferences reshape plans for a fixed start-goal pair.

Latent Planning. Beyond policy-level adaptation, learned representations support tractable planning in high-dimensional settings. Early work learns latent dynamics from observations for control and online planning (Watter et al., 2015; Hafner et al., 2019), with subsequent methods studying planning directly in latent spaces (Park et al., 2024; Zhang et al., 2025). Reeves & Williams (2024) extend this to stochastic, risk-aware planning by learning linear latent dynamics and propagating uncertainty through probabilistic flow tubes (Dong & Williams, 2012) sampled from a variational autoencoder (Kingma & Welling, 2019). These approaches highlight the value of learned representations for planning under unknown dynamics, yet do not condition them on user preferences.

Contrastive Planning and Conditional Representation Learning. Our work builds on CRLP. Time-contrastive networks and successor representations demonstrate how temporal structure in trajectory data can be captured in learned representations (Sermanet et al., 2018; Dayan, 1993). Eysenbach et al. (2023) show that contrastive learning can be interpreted as goal-conditioned RL, with subsequent work improving performance from offline data (Zheng et al., 2025b;a). Eysenbach et al. (2025) show that temporally contrastive representations induce a Gauss-Markov latent structure, reducing long-horizon planning to efficient inference over intermediate waypoints in a lower-dimensional linear system. Separately, conditional representation learning reshapes embeddings as a function of auxiliary variables. Examples include masked subspaces (Veit et al., 2017), kernel-based objective reweighting (Ma et al., 2022; Tsai et al., 2022), Feature-wise Linear Modulation (FiLM) (Perez et al., 2017) for controlling intermediate activations, and HyperNetworks (Ha et al., 2016) for conditioned network parameters. These conditioning methods must be applied carefully to avoid disrupting the latent structure that enables inference-based planning, as analyzed in Section 3.3.

3 Problem Setup and Preliminaries

3.1 Preference-Conditioned Planning from Trajectory Data

Consider a system with state space $\mathcal{S} \subseteq \mathbb{R}^d$ and per-step cost $\delta_t \in \mathbb{R}_{\geq 0}$ penalizing proximity to constraints such as obstacles. Preference-conditioned planning seeks a learned mapping $\mathcal{P}$ that, given a query $\mathbf{q} = (s_0, s_g, \Delta)$ with $s_0, s_g \in \mathcal{S}$ and preference $\Delta \in [0, 1]$, returns a sequence of n waypoints $s_{1:n} = \mathcal{P}(\mathbf{q})$ from s_0 to s_g. Δ specifies the desired safety behavior, which we quantify using a CVaR-based measure in Section 4.1. The planner is trained on an offline dataset

$$\mathcal{D} = \left\{(\tau_i, \Delta_i)\right\}_{i=1}^{N}, \quad \tau_i = \left(s_t^i, \delta_t^i\right)_{t=0}^{T_i - 1}, \tag{1}$$

where each trajectory is collected under an unknown behavioral policy and annotated with the scalar preference label $\Delta_i \in [0, 1]$ summarizing the trajectory's safety. We require the preference-conditioned planner $\mathcal{P}$ to satisfy three properties:

1. **Temporal consistency.** Waypoints correspond to dynamically feasible short-horizon transitions.
2. **Preference alignment.** For a fixed (s_0, s_g), varying Δ should modulate behavior such that lower values of Δ yield plans whose induced trajectory exhibits lower risk.
3. **Efficient inference.** $\mathcal{P}$ produces preference-conditioned plans $s_{1:n}$ at test time without retraining, and the computational cost of producing that plan scales gracefully in n.

These requirements motivate a planning-oriented latent representation learned directly from trajectory data, whose geometry captures temporal reachability and can be queried efficiently at inference time.

3.2 Contrastive Planning from Trajectory Data

Our method builds on CRLP (Eysenbach et al., 2023; 2025), which learns a latent space whose geometry reflects temporal reachability. Given an anchor state s, a positive future s_+ is sampled from the discounted state-occupancy distribution $p_\gamma(s_+|s)$ (Eysenbach et al., 2023), using a discounting factor

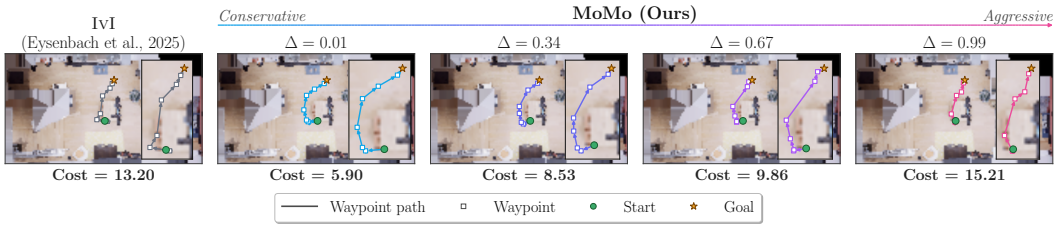

Figure 2: Preference-conditioned planning on the 29-dimensional *Ant* agent in an AI Habitat environment. Risk is defined as a linear function of obstacle proximity within an influence radius. For a fixed start and goal, increasing Δ reduces the planned obstacle clearance, resulting in a more efficient path.

$\gamma \in (0, 1)$ to control the sampling distribution horizon. Negative pairs are formed from an anchor and negative future s_- by sampling independently from the product of marginal distributions. Let $\psi\colon \mathcal{S} \to \mathbb{R}^{d_z}$ and $\phi\colon \mathcal{S} \to \mathbb{R}^{d_z}$ denote future and anchor encoders, respectively, with representation dimension d_z. For a batch of B positive pairs $\{(s^i, s^i_+)\}_{i=1}^B$, we define the logits

$$S_{ij} = \exp\Big(-\tfrac{1}{2}\big\|\phi(s^i) - \psi(s^j_+)\big\|_2^2\Big), \tag{2}$$

with diagonal matrix entries scoring positive pairs and off-diagonal entries scoring bootstrapped negative pairs (Eysenbach et al., 2025). The symmetrized InfoNCE objective (van den Oord et al., 2019) trains the encoders to distinguish the positive from negative pairs in the sampled batch:

$$\mathcal{L}_{\text{NCE}} = -\sum_{i=1}^{B}\left[\log\left(\frac{S_{ii}}{\sum_{j=1,\, j\neq i}^{B} S_{ij}}\right) + \log\left(\frac{S_{ii}}{\sum_{j=1,\, j\neq i}^{B} S_{ji}}\right)\right]. \tag{3}$$

Optimizing (3) pulls temporally related states together and pushes unrelated states apart, so that latent distance reflects temporal reachability. While standard approaches constrain representations to the unit hypersphere (Eysenbach et al., 2023), following Eysenbach et al. (2025) we instead bound the expected norm $\frac{1}{d_z}\,\mathbb{E}_{s\sim p(s)}\big[\|\psi(s)\|_2^2\big] \leq c$, enforced via regularization, where c is the constraint target (Eysenbach et al., 2025). The anchor encoder is parameterized as a linear transformation of the future encoder, such that $\phi(s) = A\psi(s)$, where $A \in \mathbb{R}^{d_z \times d_z}$ is a latent predictor mapping representations to their γ-discounted future. Crucially, learned optimal logits encode the density ratio

$$S^*_{ij} \propto \frac{p_\gamma(s^j_+ \mid s^i)}{p(s^j_+)}. \tag{4}$$

The representation space learned in (Eysenbach et al., 2025) therefore encodes how likely it is to see a certain future state given a starting state. Combined with the assumption that the marginal distribution of representations is isotropic Gaussian and the linear parameterization of $\phi(s)$ in terms of $\psi(s)$, this property induces a Gauss-Markov latent structure that supports tractable planning. Eysenbach et al. (2025) derive that the latent dynamics from $z = \psi(s)$ to its γ-discounted future representation z' as

$$p(z' \mid z) = \mathcal{N}\Big(\tfrac{c}{c+1}Az,\ \tfrac{c}{c+1}\,I\Big). \tag{5}$$

A planning query between start and goal representations z_0, z_g is solved by inferring intermediate latent waypoints $z_{1:n}$ from the Markov chain-factorized Gaussian posterior $p(z_{1:n} \mid z_0, z_g)$. Under (5), this posterior has mean $\mu = \Lambda^{-1}\eta$ and covariance $\Sigma = \Lambda^{-1}$, where Λ is a block-tridiagonal matrix defined by A and c, and η encodes start and goal embeddings (Eysenbach et al., 2025), defined in Appendix A.3. Long-horizon planning, therefore, reduces to solving a sparse nd_z-dimensional linear system in $\mathcal{O}(nd_z^3)$ time rather than optimizing trajectories in the original observation space. Following (Eysenbach et al., 2025), the inferred latent waypoints $z_{1:n}$ are decoded to the observation space plan $s_{1:n}$ by nearest-neighbor lookup in a held-out set $\mathcal{D}_{\text{held}} \subset \mathcal{D}$, mapping each latent waypoint to the observation $s \in \mathcal{D}_{\text{held}}$ whose representation $\psi(s)$ is closest.

3.3 Preference Conditioning in Contrastive Planners

Extending contrastive planning to support preference conditioning requires care. The Gauss-Markov planning structure depends on the density-ratio property of learned logits in (4), and any preference-conditioned extension must preserve this property. Consider a conditioned logit mirroring (2) with

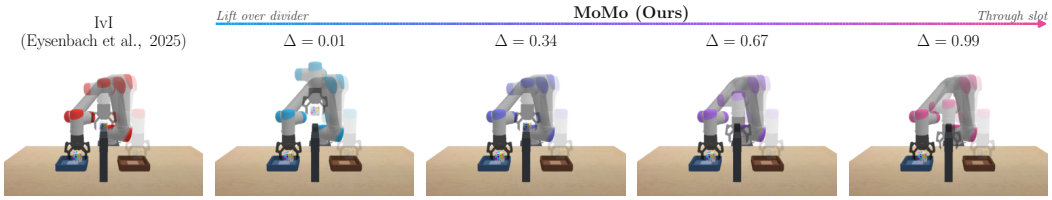

Figure 3: Preference-conditioned planning on a pick-and-place *Manipulation* task, where the UR5 robot must move a cube between start and goal regions over a gate divider. MoMo shows behavioral gradation between safely lifting above and efficiently sliding through the gate as Δ increases.

preference-dependent encoders ϕ_Δ and ψ_Δ,

$$S_{ij}^{\Delta} = \exp\Big(-\tfrac{1}{2}\big\|\phi_\Delta(s^i) - \psi_\Delta(s_+^j)\big\|_2^2\Big). \tag{6}$$

If the anchor encoder is parameterized as $\phi_\Delta(s) = A_\Delta \psi_\Delta(s)$, then conditioning can act through two distinct mechanisms. It can deform the representation geometry via ψ_Δ, and it can modify how trajectories evolve through that space via the latent transition matrix A_Δ. We show in Appendix A.1 that this preference conditioning at the level of the logits preserves the density-ratio property, justifying the joint-conditioning architecture designed for MoMo.

4 MoMo: Preference-Conditioned Contrastive Planning

We now introduce MoMo, a preference-conditioned contrastive planning framework built on CRLP. Given an offline trajectory dataset, MoMo learns a conditioned encoder and prediction matrix that jointly reshape the planning geometry using the queried safety preference. This section defines the trajectory-level preference statistic (Section 4.1), the conditioning mechanisms (Sections 4.2, 4.3), and the training objective (Section 4.4). Algorithm 1 summarizes the training and inference pipeline.

4.1 Safety Preference from Trajectory Costs

For preference-conditioned planning, we summarize each trajectory's safety behavior with a scalar statistic. Given a trajectory τ_i, let $(\delta_t^i)_{t=0}^{T_i-1}$ denote the sequence of per-step costs, where $\delta_t^i \in \mathbb{R}_{\geq 0}$ penalizes constraint exposure such as obstacle proximity. Zero-cost segments arise in both conservative and risk-tolerant behavior whenever the agent follows an efficient path through safe regions, so averaged costs fail as a faithful summary by diluting the preference signal. Thus, we use the upper tail of the empirical cost distribution, following prior work in risk-conditioned planning that treats tail-risk metrics as a natural conditioning signal (Santara et al., 2017). Letting $q_\alpha(\tau_i)$ denote the α-quantile of $(\delta_t^i)_{t=0}^{T_i-1}$, and $\mathcal{T}_\alpha(\tau_i) = \{t \in \{0, \dots, T_i - 1\} : \delta_t^i \geq q_\alpha(\tau_i)\}$ index upper-tail steps, we define the raw tail-risk statistic as the empirical CVaR (Rockafellar & Uryasev, 2000),

$$\widetilde{\Delta}_i = \mathrm{CVaR}_\alpha(\tau_i) = \frac{1}{|\mathcal{T}_\alpha(\tau_i)|} \sum_{t \in \mathcal{T}_\alpha(\tau_i)} \delta_t^i, \tag{7}$$

This isolates meaningful risk interactions in high-cost segments, excluding low-cost segments common to both conservative and risk-tolerant behaviors. We wrap $\{\widetilde{\Delta}_i\}$ to lie in $[0, 1]$ across the training dataset, so that $\Delta_i = 0$ corresponds to the most risk-averse trajectory and $\Delta_i = 1$ to the most risk-tolerant. Each trajectory thus carries a single label, shared across all states in the trajectory, setting the preference context for every contrastive pair sampled from it. This implies that the same state can be seen under different preferences and will accordingly have different conditioned representations. Since a planner must reason about temporal sequences, this captures which states are safely reachable from one another rather than solely the independent risk of each state. The conditioning variable setup using the cost-computed Δ as a signal for behavioral modulation is discussed in Appendix A.4.

4.2 Preference-Conditioned Latent Geometry

The CRLP encoder organizes states on temporal reachability. Under preference-conditioning, the same state may play different roles depending on the desired safety level. A state near an obstacle,

for example, is a plausible waypoint under risk-tolerant preferences but unlikely under a conservative preference. The encoder must therefore jointly capture temporal reachability and preferential similarity. Applying a condition-dependent mask to shared embeddings (Veit et al., 2017) leaves intermediate features unchanged, limiting the expressivity with which preference can reshape the representation. Instead, we use Feature-Wise Linear Modulation (FiLM) (Perez et al., 2017), which applies Δ-dependent affine transformations to the encoder's hidden-layer activations, allowing the preference to reshape features hierarchically throughout the network. Let $h^{(l)}$ denote the pre-activation at layer l, before the nonlinearity. A FiLM generator F_ξ with parameters ξ takes input Δ and produces scale and shift vectors $\{(\lambda_\Delta^{(l)}, \beta_\Delta^{(l)})\}$ for each encoder layer l, modulating the activation

$$a_\Delta^{(l)} = \text{swish}\left(\lambda_\Delta^{(l)} \odot h^{(l)} + \beta_\Delta^{(l)}\right), \tag{8}$$

where $\odot$ denotes an elementwise product. We use the Swish nonlinearity (Ramachandran et al., 2017) following Eysenbach et al. (2025). The scale $\lambda_\Delta^{(l)}$ controls the strength of each feature channel under the queried preference, while the shift $\beta_\Delta^{(l)}$ adjusts each feature's bias, effectively gating its contribution to downstream layers. Applying this modulation across all encoder layers defines the preference-conditioned future-state encoder $\psi_\Delta(s)$ with FiLM parameters generated by $F_\xi(\Delta)$.

4.3 Preference-Conditioned Latent Predictions

To support varying latent transition dynamics, we parameterize a preference-dependent prediction matrix A_Δ as a shared base matrix with a low-rank preference-dependent perturbation. This reflects the inductive bias that dominant temporal structure is shared across preferences, while conditioning shifts which transitions are locally favored. Let $A_0 \in \mathbb{R}^{d_z \times d_z}$ denote the shared matrix, and $U, V : [0,1] \to \mathbb{R}^{d_z \times r}$ with $r < d_z$ be networks mapping the Δ to rank-r matrices. We define

$$A_\Delta = A_0 + U(\Delta)V(\Delta)^\top. \tag{9}$$

$V(\Delta)$ defines a preference-dependent r-dimensional subspace of the representation space, and $U(\Delta)$ specifies how this subspace contributes to the predicted future representation. Conceptually, the perturbation reads r preference-relevant directions out of the current representation and writes r corrections into the predicted future. Sharing A_0 across preferences grounds universal learned dynamics, preventing the transition structure from drifting unpredictably as Δ varies. The parameterization also reduces the number of preference-dependent outputs from d_z^2 to $2rd_z$, thereby improving scalability to higher-dimensional representations and mitigating overfitting when the preference axis is sparsely sampled. Combined with ψ_Δ, this defines the conditioned anchor encoder $\phi_\Delta(s) = A_\Delta \psi_\Delta(s)$, completing the conditioned logit S_{ij}^Δ in (6). In practice, F_ξ, U, and V all operate on a shared Fourier feature embedding of Δ (Tancik et al., 2020). Full architectural details appear in Appendix E.

4.4 Preference-Conditioned Contrastive Objective

For planning, the representation structure must encode the likelihood of observing future states from an anchor state, given its preference context. Therefore, anchor-future pairs are scored under the preference label of the anchor, such that for a batch of preference-labeled positive pairs $\{(s^i, s_+^i, \Delta_i)\}_{i=1}^B$, the conditioned logit is defined as $S_{ij}^{\Delta_i}$ using (6). Positive pairs are sampled following Eysenbach et al. (2025), with each pair inheriting Δ_i of its source trajectory. This ensures that positive pairs (s^i, s_+^i) are pulled together under their shared preference, while negative pairs are pushed apart under the anchor preference. Extending the symmetrized InfoNCE loss with preference conditioning yields

$$\mathcal{L}_{\text{MoMo}} = -\sum_{i=1}^{B} \left\{ \log\left(\frac{S_{ii}^{\Delta_i}}{\sum_{j=1,\, j\neq i}^{B} S_{ij}^{\Delta_i}}\right) + \log\left(\frac{S_{ii}^{\Delta_i}}{\sum_{j=1,\, j\neq i}^{B} S_{ji}^{\Delta_j}}\right) \right\}. \tag{10}$$

Minimizing $\mathcal{L}_{\text{MoMo}}$ jointly optimizes the encoder parameters, FiLM generator, low-rank modulation networks, and shared matrix A_0, which together define the planner $\mathcal{P}$ from Section 3.1. Under the same assumptions made by Eysenbach et al. (2025), for any sampled pair (s^i, s_+^j) and preference $\Delta \in [0,1]$, the optimal conditioned similarity satisfies the following as shown in Appendix A.1,

$$\left(S_{ij}^\Delta\right)^\star \propto \frac{p_\gamma(s_+^j \mid s^i, \Delta)}{p(s_+^j)}. \tag{11}$$

Algorithm 1 MoMo: Preference-Conditioned Contrastive Planning. Color denotes conditioning of the representation geometry and latent prediction matrix; black follows Eysenbach et al. (2025)

TRAINING

Input: offline trajectories $\mathcal{D} = \{(\tau_i, \Delta_i)\}_{i=1}^{N}$

for each minibatch **do**

  Sample preference-labeled positive pairs $\{(s^i, s_+^i, \Delta_i)\}_{i=1}^{B}$ from discounted state-occupancy

  Encode anchors and candidate futures with ψ_{Δ_i}

  Compute conditioned prediction matrix A_{Δ_i} via (9)

  Form similarities $S_{ij}^{\Delta_i}$ and update parameters by minimizing (10) under the norm constraint

INFERENCE

Input: query $\mathbf{q} = (s_0, s_g, \Delta)$, waypoint count n

  Encode endpoints $z_0 = \psi_\Delta(s_0),\ z_g = \psi_\Delta(s_g)$

  Generate conditioned prediction matrix A_Δ via (9)

  Solve $z_{1:n} = \Lambda_\Delta^{-1}\eta_\Delta$, where $\Lambda_\Delta, \eta_\Delta$ follow (Eysenbach et al., 2025) with ψ_Δ, A_Δ

  Decode $z_{1:n}$ to $s_{1:n}$ via nearest-neighbor retrieval under ψ_Δ in $\mathcal{D}_{\text{held}}$

return Preference-conditioned plan $s_{1:n}$

This is the conditioned analog of (4), preserving the contrastive planning density-ratio property. The representation space now encodes the likelihood of observing a given future state from a starting state, given the user-preference set at runtime. This distinction from (4) enables behavioral modulation while keeping inference-driven planning tractable under MoMo. The modifications to CRLP achieving this density-ratio encoding operate via both the conditioning architectures and the conditioned variant of InfoNCE. Eysenbach et al. (2025) further assume that the marginal distribution over representations is isotropic Gaussian, which we argue remains consistent with our approach in Appendix A.2, supported by quantitative evidence. At inference, a query $\mathbf{q} = (s_0, s_g, \Delta)$ is resolved by computing A_Δ and the conditioned endpoints, solving the Gauss-Markov posterior for $z_{1:n}$, and nearest-neighbor decoding, outlined in Algorithm 1. In practice, goal-reaching policies trained over short horizons can then be used to track these conditioned waypoints for a complete trajectory.

5 Experiments

We evaluate MoMo across a range of environments to address the following questions:

Q1. Does MoMo produce temporally consistent plans by preserving the representation structure required for inference-driven planning?

Q2. Does MoMo produce plans that respond meaningfully to the user preference, or does conditioning collapse to a single behavior mode?

Q3. Does MoMo maintain effective preference-conditioned planning across diverse task settings, including higher-dimensional and longer-horizon tasks?

Environments and Baselines. We evaluate MoMo on four seeds, six environments, and four agents (Table 7), including three navigation tasks for a 2D *Point* agent (an obstacle field, a Gaussian cost contour, and a scene from AI Habitat (Straub et al., 2019; Puig et al., 2023)), a quadcopter *Drone* performing an exploration orbit around a central safe column, a quadruped *Ant* navigating an AI Habitat scene, and a UR5 *Manipulator* performing table-top pick-and-place with obstacle clearance. For each environment, we generate datasets at multiple risk levels and label them following Section 4.1, adapting D4RL (Fu et al., 2021) and FSRL (Liu et al., 2024). Details are in Appendix D.

We compare MoMo against four baselines. IvI (Eysenbach et al., 2025) is the preference-agnostic baseline producing a single plan per start-goal query. The remaining baselines apply a state augmentation (SA) approach, appending preference Δ_i as an additional observation dimension over three representation backbones: IvI-SA uses IvI's contrastive representations (Eysenbach et al., 2025), PCA-SA uses linear principal-component projections of raw observations, and VIP-SA uses representations from VIP (Ma et al., 2023), which encode temporal distances. Together, these isolate whether preference conditioning can be achieved via SA across various representation types.

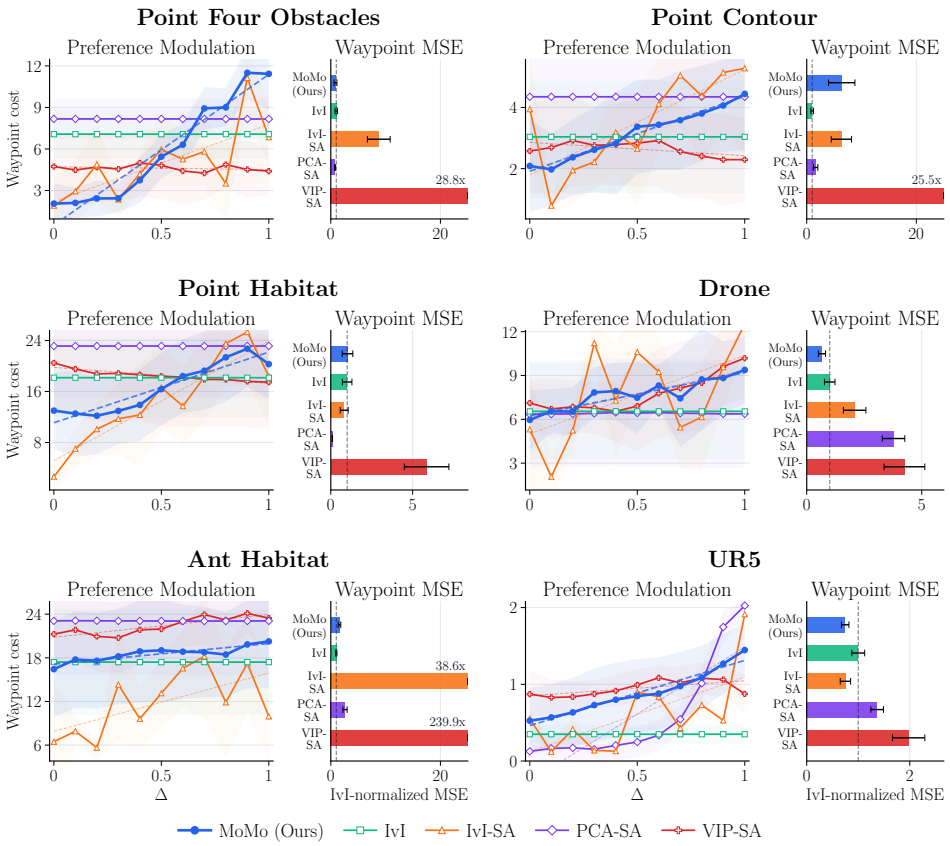

Figure 4: **Preference modulation and planning fidelity across six environments**. (Left) For each environment, we plot the waypoint cost as a function of preference Δ; dashed lines show each method's modulation trend. (Right) We report waypoint reconstruction MSE normalized by IvI's MSE (vertical reference line at 1.0), where bars beyond 1.0 indicate baselines that fit the dataset's waypoints worse than the preference-agnostic IvI. Bars labeled with multipliers (e.g., $28.8\times$) extend beyond the displayed axis range. MoMo produces a near-monotonic trend between Δ and waypoint cost while maintaining MSE comparable to IvI.

Demonstrating Consistent Preference Modulation. For **Q1**, we measure the waypoint reconstruction MSE relative to IvI, the inference-driven planning reference. As shown in Figure 4, MoMo achieves MSE comparable to IvI across most environments. At the same time, the SA baselines reach up to $28.8\times$ and $38.6\times$ higher on Point Four Obstacles and Ant Habitat, respectively. This indicates that joint conditioning better preserves the representation structure required for planning by inference.

For **Q2**, we query start-goal pairs across multiple preferences and plot the sum of waypoint costs in the left panel of Figure 4, observing a near-monotonic relationship between Δ and realized cost across all environments. MoMo's per-query variance is comparable to IvI's, indicating consistent modulation across individual queries. Different SA backbones fail in qualitatively different ways. PCA-SA mostly collapses toward flat responses, indicating that linear projections lack sufficient structure to encode preference at all. IvI-SA produces noisy but correlated responses, showing that contrastive representations can encode preference under SA but yield unstable plans, which we examine further in Appendix B.4. Finally, VIP-SA produces erratic responses with high MSE, suggesting that the temporal-distance representations are not preserved by SA-based conditioning.

Figure 1 illustrates the modulation of risk exposure for a given start-goal query, retrieving viable conditioned waypoint sequences which can be tracked by a standard goal-reaching policy to generate complete trajectories. Increasing Δ corresponds to more direct, efficient routes, at the expense of greater realized risk. We further analyze how our conditioning method achieves this modulation by deforming the latent manifold in Appendix B.2. MoMo is also able to demonstrate a mode-switching behavior in the left panel of Figure 1 by recovering a different, safer approach for the same start and goal. Across four training seeds, MoMo demonstrates an intended increase in plan cost from

conservative to aggressive queries in all seed-specific evaluations. Some state-augmentation baselines produce larger cost changes on particular tasks, but their preference responses are less consistent across environments and seeds. Full multi-seed results are reported in Appendix B.3.

Robustness across Tasks and Modalities. To evaluate **Q3**, we test MoMo along four robustness axes, with quantitative metrics for all six environments reported in Table 6. To test whether MoMo works across different environmental risk structures, we evaluate the *Point* agent in two longer-horizon settings: Point Contour, with dense Gaussian cost-field contours, and Point Habitat, with an indoor scene rendered using AI Habitat. As shown in Figure 4, MoMo maintains a Spearman correlation $\rho \geq 0.91$ between Δ and cost on both, with the smoothest modulation curves of any baseline.

To test the inference-driven advantage of learning latent dynamics from a reduced feature set, we train the *Drone* task on a 6-dimensional subset of the 18-dimensional observation space (translational positions and velocities only). Here, MoMo's modulation performance is modest ($\rho = 0.43 \pm 0.34$), however PCA-SA's curve in Figure 4 collapses toward a single behavioral mode. This result highlights the performance of MoMo's joint conditioning approach in maintaining both temporal and preferential consistency on a reduced feature set where most SA-based methods cannot.

To test representational capacity in high dimensions, we evaluate the 29-dimensional *Ant* quadruped navigating in an AI Habitat scene. Figure 2 shows plans transitioning from cautious detours around obstacles at low Δ to direct routes at high Δ. While VIP-SA achieves a marginally higher correlation ($\rho = 0.84 \pm 0.08$ vs. 0.74 ± 0.15), its waypoint MSE is substantially higher than IvI's, indicating that the plans are not temporally consistent, whereas MoMo achieves the smoothest modulation curve across all baselines. Despite the lower correlation than on the 2-D Point environment (0.97 ± 0.02), MoMo retains a positive preference-cost relationship and IvI-comparable waypoint MSE. This graceful degradation suggests that MoMo can scale to higher-dimensional tasks.

Finally, to test that MoMo does not depend on the specific choice of preference variable, we replace the CVaR-based preference on the UR5 *Manipulator* with maximum end-effector height as a clearance proxy over a gate obstacle. As Figure 3 illustrates, MoMo recovers safe trajectories that maintain clearance above the divider, transitioning to aggressive plans passing directly through the gate as Δ increases, achieving the highest correlation as shown in Table 6. These analyses demonstrate that MoMo performs robust preference-conditioned planning across diverse settings.

Closed-loop Task Performance . While our primary focus is preference-conditioned waypoint planning by MoMo, we additionally demonstrate that these plans remain effective when executed by a low-level controller. In Point Four Obstacles, we use the same goal-conditioned SAC policy (Haarnoja et al., 2018) to execute the waypoint sequences produced by MoMo and the SA baselines, and compare the resulting closed-loop behavior against CCAC, CDT, and MODULI (Yuan et al., 2025). Across three seeds, 40 queries per seed and 11 preference values, MoMo+GCRL achieves a mean goal-reaching success rate of 96.4%, versus 90.3% for IvI-SA+GCRL and 79% for MODULI. Among successful rollouts, it also produces the largest change in closed-loop cost across the preference range (36.40) and the highest preference-cost correlation (0.94). Thus, the preference modulation of the inferred plans persists under closed-loop execution. Full results are reported in Appendix B.1.

6 Conclusion

We introduced MoMo, a preference-conditioned contrastive planning framework that jointly conditions the representation geometry and latent transition matrix while preserving the density-ratio structure required by CRLP. This is achieved through logit-level conditioning on a preference-context statistic that summarizes behavioral tendencies within each trajectory. Across six environments and four agent embodiments, FiLM-based encoder modulation combined with low-rank modulation of the transition matrix produces temporally consistent plans with preference-dependent cost responses across diverse task settings, including effective scaling to high-dimensional and longer-horizon tasks. More broadly, our work shows that representation-level conditioning can support preference-conditioned planning at scale. Despite these encouraging results, we identify a few areas for improvement as outlined in Appendix C. In particular, these include our reliance on sufficient joint state-preference coverage in the dataset and the Gaussianity assumptions inherited from Eysenbach et al. (2025).

REPRODUCIBILITY STATEMENT

We provide training and inference procedures for MOMO in Algorithm 1, with the theoretical assumptions and derivations given in Appendix A. The appendix further documents the model architecture, hyperparameters, dataset construction, preference-labeling procedure, evaluation protocol, training seeds, and computational setup used in our experiments. The anonymous supplementary material includes the implementation of MOMO together with instructions for training MOMO. Together, the paper, appendix, and supplementary code specify the method and experimental protocol for using MOMO.

REFERENCES

Roy Betser, Eyal Gofer, Meir Yossef Levi, and Guy Gilboa. Infonce induces gaussian distribution, 2026. URL https://arxiv.org/abs/2602.24012.

Lars Blackmore and Masahiro Ono. *Convex Chance Constrained Predictive Control Without Sampling*. 2009. doi: 10.2514/6.2009-5876. URL https://arc.aiaa.org/doi/abs/10.2514/6.2009-5876.

Roberto Bolli and H. Harry Asada. Elderly bodily assistance robot (e-bar): A robot system for body-weight support, ambulation assistance, and fall catching, without the use of a harness. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 713–719, 2025. doi: 10.1109/ICRA55743.2025.11127403.

Xin-Qiang Cai, Pushi Zhang, Li Zhao, Jiang Bian, Masashi Sugiyama, and Ashley Juan Llorens. Distributional pareto-optimal multi-objective reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=prIwYTU9PV.

Jorgen Cani, Panagiotis Koletsis, Konstantinos Foteinos, Ioannis Kefaloukos, Lampros Argyriou, Manolis Falelakis, Iván Del Pino, Angel Santamaria-Navarro, Martin Čech, Ondřej Severa, Alessandro Umbrico, Francesca Fracasso, AndreA Orlandini, Dimitrios Drakoulis, Evangelos Markakis, Iraklis Varlamis, and Georgios Th. Papadopoulos. Triffid: Autonomous robotic aid for increasing first responders efficiency, 2025. URL https://arxiv.org/abs/2502.09379.

Siyu Dai, Shawn Schaffert, Ashkan Jasour, Andreas Hofmann, and Brian Williams. Chance constrained motion planning for high-dimensional robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8805–8811, 2019. doi: 10.1109/ICRA.2019.8793660.

Charles Dawson, Ashkan Jasour, Andreas G. Hofmann, and Brian C. Williams. Provably safe trajectory optimization in the presence of uncertain convex obstacles. *CoRR*, 2020. URL https://arxiv.org/abs/2003.07811.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993. doi: 10.1162/neco.1993.5.4.613. URL https://doi.org/10.1162/neco.1993.5.4.613.

Shuonan Dong and Brian Williams. Learning and recognition of hybrid manipulation motions in variable environments using probabilistic flow tubes. *International Journal of Social Robotics*, 4 (4):357–368, 2012. doi: 10.1007/s12369-012-0155-x. URL https://doi.org/10.1007/s12369-012-0155-x.

Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, and Sergey Levine. Contrastive learning as goal-conditioned reinforcement learning, 2023. URL https://arxiv.org/abs/2206.07568.

Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via interpolation: Contrastive representations provably enable planning and inference, 2025. URL https://arxiv.org/abs/2403.04082.

Meng Feng, Viraj Parimi, and Brian Williams. Safe multi-agent navigation guided by goal-conditioned safe reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16869–16875. IEEE, May 2025. doi: 10.1109/icra55743.2025.11127461. URL http://dx.doi.org/10.1109/ICRA55743.2025.11127461.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021. URL https://arxiv.org/abs/2004.07219.

Zijian Guo, Weichao Zhou, and Wenchao Li. Temporal logic specification-conditioned decision transformer for offline safe reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17003–17019. PMLR, 2024. URL https://proceedings.mlr.press/v235/guo24j.html.

Zijian Guo, Weichao Zhou, Shengao Wang, and Wenchao Li. Constraint-conditioned actor-critic for offline safe reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nrRkAAAufl.

David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks, 2016. URL https://arxiv.org/abs/1609.09106.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels, 2019. URL https://arxiv.org/abs/1811.04551.

Astghik Hakobyan, Gyeong Chan Kim, and Insoon Yang. Risk-aware motion planning and control using cvar-constrained optimization. *IEEE Robotics and Automation Letters*, 4(4):3924–3931, 2019. doi: 10.1109/LRA.2019.2929980.

E. W. Hobson. On the fundamental lemma of the calculus of variations, and on some related theorems. *Proceedings of the London Mathematical Society*, s2-11(1):17–28, 1913. doi: https://doi.org/10.1112/plms/s2-11.1.17. URL https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s2-11.1.17.

Xin Huang, Meng Feng, Ashkan M. Jasour, Guy Rosman, and Brian Charles Williams. Risk conditioned neural motion planning. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9057–9063, 2021. URL https://api.semanticscholar.org/CorpusID:236912881.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, November 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

J.J. Kuffner and S.M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pp. 995–1001 vol.2, 2000. doi: 10.1109/ROBOT.2000.844730.

Simon Leonard, Kyle L. Wu, Yonjae Kim, Axel Krieger, and Peter C.W. Kim. Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing. *IEEE Transactions on Biomedical Engineering*, 61(4):1305–1317, 2014. doi: 10.1109/TBME.2014.2302385.

Ruohong Liu, Yuxin Pan, Linjie Xu, Lei Song, Pengcheng You, Yize Chen, and Jiang Bian. Efficient discovery of pareto front for multi-objective reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fDGPIuCdGi.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21611–21630. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23m.html.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

Yiwei Lu, Guojun Zhang, Sun Sun, Hongyu Guo, and Yaoliang Yu. f-micl: Understanding and generalizing infonce-based contrastive learning, 2024. URL https://arxiv.org/abs/2402.10150.

Martin Q. Ma, Yao-Hung Hubert Tsai, Paul Pu Liang, Han Zhao, Kun Zhang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Conditional contrastive learning for improving fairness in self-supervised learning, 2022. URL https://arxiv.org/abs/2106.02866.

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, 2023. URL https://arxiv.org/abs/2210.00030.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency, 2018. URL https://arxiv.org/abs/1809.01812.

Luisa Mao, Garrett Warnell, Peter Stone, and Joydeep Biswas. Pacer: Preference-conditioned all-terrain costmap generation. *arXiv preprint arXiv:2410.23488*, 2024. URL https://arxiv.org/abs/2410.23488.

Masahiro Ono and Brian C. Williams. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *2008 47th IEEE Conference on Decision and Control*, pp. 3427–3432, 2008. doi: 10.1109/CDC.2008.4739221.

Viraj Parimi and Brian C. Williams. Risk-bounded multi-agent visual navigation via iterative risk allocation, 2026. URL https://arxiv.org/abs/2509.08157.

Suhan Park, Suhyun Jeon, and Jaeheung Park. A constrained motion planning method exploiting learned latent space for high-dimensional state and constraint spaces. *IEEE/ASME Transactions on Mechatronics*, 29(4):3001–3009, 2024. doi: 10.1109/TMECH.2024.3399594.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. URL https://arxiv.org/abs/1709.07871.

Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimir Vondrus, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019. URL https://arxiv.org/abs/1806.08734.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

Marlyse Reeves and Brian C. Williams. Laplass: Latent space planning for stochastic systems, 2024. URL https://arxiv.org/abs/2404.07063.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at risk. *Journal of Risk*, 3:21–41, 2000. URL https://api.semanticscholar.org/CorpusID:854622.

Anirban Santara, Abhishek Naik, Balaraman Ravindran, Dipankar Das, Dheevatsa Mudigere, Sasikanth Avancha, and Bharat Kaul. Rail: Risk-averse imitation learning, 2017. URL https://arxiv.org/abs/1707.06658.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, 2018. doi: 10.1109/ICRA.2018.8462891. URL https://doi.org/10.1109/ICRA.2018.8462891.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020. URL https://arxiv.org/abs/2006.10739.

Yao-Hung Hubert Tsai, Tianqin Li, Martin Q. Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning with kernel, 2022. URL https://arxiv.org/abs/2202.05458.

Matti Vahs, Christian Pek, and Jana Tumova. Belief control barrier functions for risk-aware control, 2023. URL https://arxiv.org/abs/2309.06499.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks, 2017. URL https://arxiv.org/abs/1603.07810.

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images, 2015. URL https://arxiv.org/abs/1506.07365.

Haiyu Wu, Randall Balestriero, and Morgan Levine. Visreg: Variance-invariance-sketching regularization for jepa training, 2026. URL https://arxiv.org/abs/2606.02572.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems*, 2019. URL https://openreview.net/forum?id=B1lR3HBeUB.

Gwangpyo Yoo, Jinwoo Park, and Honguk Woo. Risk-conditioned reinforcement learning: A generalized approach for adapting to varying risk measures. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16513–16521, Mar. 2024. doi: 10.1609/aaai.v38i15.29589. URL https://ojs.aaai.org/index.php/AAAI/article/view/29589.

Yifu Yuan, Zhenrui Zheng, Zibin Dong, and Jianye Hao. MODULI: Unlocking preference generalization via diffusion models for offline multi-objective reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73743–73766. PMLR, 2025. URL https://proceedings.mlr.press/v267/yuan25n.html.

Jiawei Zhang, Chengchao Bai, Wei Pan, Tianhang Liu, and Jifeng Guo. Local path optimization in the latent space using learned distance gradient. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15940–15946. IEEE, 2025. doi: 10.1109/iros60139.2025.11247535. URL http://dx.doi.org/10.1109/IROS60139.2025.11247535.

Qin Zhang, Linrui Zhang, Haoran Xu, Li Shen, Bowen Wang, Yongzhe Chang, Xueqian Wang, Bo Yuan, and Dacheng Tao. Saformer: A conditional sequence modeling approach to offline safe reinforcement learning, 2023. URL https://arxiv.org/abs/2301.12203.

Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive rl: Techniques for robotic goal reaching from offline data, 2025a. URL https://arxiv.org/abs/2306.03346.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding, 2025b. URL https://arxiv.org/abs/2310.20141.

Baiting Zhu, Meihua Dang, and Aditya Grover. Scaling pareto-efficient decision making via offline multi-objective rl. ICLR 2023 Poster, 2023. URL https://openreview.net/forum?id=Ki4ocDm364.

A Conditioned Contrastive Theory

A.1 Density Encoding in Learned Logits

The Gauss-Markov inference-driven planning method outlined in Section 3.2 requires learned logits to encode the probability ratio expressed in (4). This property must be preserved across the conditioning method. To prove that our training framework supports this, we begin by showing that (4) holds for the non-conditioned InfoNCE objective, as in (Eysenbach et al., 2025). We define the logits function $g(s, s_+)$ such that $g(s^i, s_+^j) = S_{ij}$. Start by considering the optimization problem, which yields the optimal logits function,

$$\max_{g(s,s_+)} \lim_{B\to\infty} \mathop{\mathbb{E}}_{\{(s^i,s_+^i)\}_{i=1}^B \sim p(s,s_+)} \left[\frac{1}{B} \sum_{i=1}^{B} \left\{ \log \left(\frac{g(s^i, s_+^i)}{\sum_{j=1, j\neq i}^{B} g(s^i, s_+^j)} \right) + \log \left(\frac{g(s^i, s_+^i)}{\sum_{j=1, j\neq i}^{B} g(s^j, s_+^i)} \right) \right\} \right]. \tag{12}$$

Now consider the first term in the objective,

$$J_1(g) = \lim_{B\to\infty} \mathop{\mathbb{E}}_{\{(s^i,s_+^i)\}_{i=1}^B \sim p(s,s_+)} \left[\frac{1}{B} \sum_{i=1}^{B} \log \left(\frac{g(s^i, s_+^i)}{\sum_{j=1, j\neq i}^{B} g(s^i, s_+^j)} \right) \right]. \tag{13}$$

Taking the functional variation, the solution to the optimization problem can be extracted as follows.

$$\begin{aligned} \delta J_1(g) &= \lim_{B\to\infty} \mathop{\mathbb{E}}_{\{(s^i,s_+^i)\}_{i=1}^B \sim p(s,s_+)} \left[\frac{1}{B} \sum_{i=1}^{B} \left(\frac{\delta g(s^i, s_+^i)}{g(s^i, s_+^i)} - \frac{\delta \sum_{j=1, j\neq i}^{B} g(s^i, s_+^j)}{\sum_{j=1, j\neq i}^{B} g(s^i, s_+^j)} \right) \right] \\ &= \lim_{B\to\infty} \frac{1}{B} \sum_{i=1}^{B} \mathop{\mathbb{E}}_{s^i \sim p(s)} \left[\int \left(\frac{\delta g(s^i, s_+)}{g(s^i, s_+)} p(s_+|s^i) - k(s^i) \delta g(s^i, s_+) p(s_+) \right) ds_+ \right] \\ &= \iint \delta g(s, s_+) \left[\frac{p(s_+|s)}{g(s, s_+)} - k(s) p(s_+) \right] p(s) ds ds_+ \end{aligned} \tag{14}$$

Note that in the second line, the expectation is rewritten over sampled anchors and integrated over futures, leaving a term

$$k(s^i) = \lim_{B\to\infty} \mathbb{E} \left[\frac{B-1}{\sum_{j=1, j\neq i}^{B} g(s^i, s_+^j)} \right] = \left[\int g(s^i, s_+) p(s_+)\, ds_+ \right]^{-1}, \tag{15}$$

which is independent of future realizations in the infinite-batch-sampling limit. The first order condition of optimality requires the final integral to vanish in a pointwise manner for all arbitrary variations $\delta g(s, s_+)$. From the fundamental lemma of the calculus of variations (Hobson, 1913), the bracketed term must therefore vanish for all anchor and future states. Absorbing $k(s)$ as a free normalization of logits (Eysenbach et al., 2025), we obtain the following optimal solution.

$$g^*(s, s_+) \propto \frac{p(s_+|s)}{p(s_+)} \tag{16}$$

Repeating this procedure for the second term in the original optimization problem $J_2(g)$ and noting that $p(s_+|s)p(s) = p(s|s_+)p(s_+)$ allows us to recover the same solution, ensuring consistency of the symmetrized objective. Using the function approximator $g(s, s_+) \approx \exp\{-\frac{1}{2}||A\psi(s) - \psi(s_+)||^2\}$, this proves the statement in (4) (Eysenbach et al., 2025; Lu et al., 2024; Ma & Collins, 2018).

Extending this to conditioned contrastive learning, define the conditioned logits function as $\hat{g}(s, s_+; \Delta)$. This leads to the following problem according to the form in (12),

$$\max_{\hat{g}(s,s_+;\Delta)} \lim_{B\to\infty} \mathop{\mathbb{E}}_{\{(s^i,s_+^i,\Delta_i)\}_{i=1}^B \sim p(s,s_+,\Delta)} \left[\frac{1}{B} \sum_{i=1}^{B} \left\{ \log \left(\frac{\hat{g}(s^i, s_+^i; \Delta_i)}{\sum_{j=1, j\neq i}^{B} \hat{g}(s^i, s_+^j; \Delta_i)} \right) + \log \left(\frac{\hat{g}(s^i, s_+^i; \Delta_i)}{\sum_{j=1, j\neq i}^{B} \hat{g}(s^j, s_+^i; \Delta_j)} \right) \right\} \right].$$

If we interpret anchor sampling as selecting a random index from the concatenated tuple of all observed states in the dataset, this implies both s^i and Δ_i. In other words, since the anchor state and preference context of each positive pair are sampled together, we can rewrite the anchor variable as $\hat{s} = [s^\top \ \Delta]^\top$ and accordingly an amended logit function $\hat{\hat{g}}(\hat{s}, s_+)$. Note that this is not an augmentation of the state, since s_+ remains unchanged; it is only for notational convenience. This yields the first component, built in the same manner as (13),

$$J_1(\hat{\hat{g}}) = \lim_{B\to\infty} \mathop{\mathbb{E}}_{\{(\hat{s}^i, s_+^i)\}_{i=1}^B \sim p(\hat{s}, s_+)} \left[\frac{1}{B} \sum_{i=1}^{B} \log \left(\frac{\hat{\hat{g}}(\hat{s}^i, s_+^i)}{\sum_{j=1, j\neq i}^{B} \hat{\hat{g}}(\hat{s}^i, s_+^j)} \right) \right].$$

Note that sampling from $p(s, s_+, \Delta)$ as done in the original optimization problem is equivalent to sampling from $p(\hat{s}, s_+)$ following the reasoning outlined above. This translates to a functional variation, analogous to (14),

$$\delta J_1(\hat{\hat{g}}) = \iint \delta\hat{\hat{g}}(\hat{s}, s_+) \left[\frac{p(s_+|\hat{s})}{\hat{\hat{g}}(\hat{s}, s_+)} - \hat{\hat{k}}(\hat{s}) p(s_+) \right] p(\hat{s}) d\hat{s} ds_+,$$

where $\hat{\hat{k}}(\hat{s})$ follows the definition corresponding to the formulation in (15). Once again, from variational calculus (Hobson, 1913), the bracketed term must vanish pointwise, yielding the proportionality of equivalent form to (16),

$$(\hat{\hat{g}})^*(\hat{s}, s_+) \propto \frac{p(s_+|\hat{s})}{p(s_+)}$$

Reverting to the original conditioned logit definition, we have

$$(\hat{g})^*(s, s_+; \Delta) \propto \frac{p(s_+|s, \Delta)}{p(s_+)}, \tag{17}$$

under the assumption that the sampling process is unchanged with the addition of a preference context variable. This is precisely the conditional probability ratio we intend to encode in the representation space, capturing the probability of observing a future state given both the anchor state and preference context. Repeating this procedure for $J_2(\hat{\hat{g}})$ using the same vector augmentation of the anchor yields

$$(\hat{g})^*(s, s_+; \Delta) \propto \frac{p(s, \Delta|s_+)}{p(s, \Delta)}, \tag{18}$$

which is equivalent to the first proportionality result by Bayes' theorem. As such, we preserve the required latent dynamics for inference-driven planning while directly reflecting a preference in the geometry of the representation space, as intended.

This modification to CRLP, proposed by MOMO, acts through both the architecture used to approximate the conditioned logit and the conditioned variant of the InfoNCE objective. State augmentation leaves the InfoNCE objective unchanged while increasing the encoder input dimension to include the preference label. This is different from MOMO's approach, which injects preferences by direct conditioning, using FiLM for the encoder and low-rank neural modulation for the predictor, giving preference-dependent deformations of the latent manifold, which are presented in Appendix B.2. Therefore, the two approaches give two different density ratios and encode these via two different architectural designs. FiLM and low-rank neural modulation are not the only ways to encode meaningful temporal and preference structures for conditioned planning, as presented in (17), since the architecture serves as an approximator of the optimal logit function. We propose this architecture as an elegant and efficient way to achieve a sufficiently low function-approximation error to capture the planning structure. Contrarily, SA encodes the probability of ending up in a future state with its own preference condition, given the current state and preference condition. This unified representation is observed to fail to plan waypoint sequences consistently and to modulate plans reliably, as quantified by reconstruction MSE and preference trends. These explanations are presented in Appendix B.4. In summary, MOMO's performance is thus driven by the joint effect of architectural and optimization modifications to CRLP, which is enabled by preservation of the density encoding in learned logits.

A.2 GAUSSIANITY OF REPRESENTATIONS

A second key assumption underlying CRLP is that the marginal distribution over representations is isotropic Gaussian (Eysenbach et al., 2025). In our formulation, we require this assumption

Table 1: Per-Δ covariance drift for MoMo across environments. Low, constant drift across the preference continuum constitutes more robust preservation of marginal distribution Gaussianity.

Environment	Δ					
	0.0	0.2	0.4	0.6	0.8	1.0
Point Four Obstacles	0.38	0.53	0.63	0.52	0.55	0.67
Point Contour	0.59	0.29	0.29	0.75	0.75	0.56
Point Habitat	0.53	0.46	0.51	0.60	0.54	0.52
Drone	0.76	0.76	0.73	0.71	0.69	0.69
Ant Habitat	2.76	3.02	3.11	3.73	4.57	8.41
UR5	0.95	0.96	0.96	0.97	0.98	0.99

to hold for all conditioned latent spaces. In other words, whilst Eysenbach et al. (2025) assumes $\psi \sim \mathcal{N}(\psi\,; \mu = 0, \Sigma = c \cdot I)$, we take the point-wise version $\psi_\Delta \sim \mathcal{N}(\psi_\Delta\,; \mu = 0, \Sigma = c \cdot I), \forall\, \Delta$. Using the representation norm constraint enforces this in expectation, implying that the distribution's variance could vary along the preference continuum. We recognize that under conditioning, the density-ratio encoding and isotropic Gaussian assumptions incur larger logit function-approximation error and sampling error due to the sparser joint state-preference space. However, we find empirically that planning performance remains consistent with (Eysenbach et al., 2025) across multiple agent embodiments and tested environments, thereby keeping inference-driven planning tractable, as evidenced in Section 5. This supports the assumption that any drift of the marginal distribution variance from the scaled identity matrix is not prohibitive for planning. Distribution sketching of representation spaces via constraints is a leading open research question in contrastive learning, recently addressed by VISReg (Wu et al., 2026).

To quantify this effect, we compute the covariance drift D_Σ from the isotropic Gaussian assumption using the sample covariance at specified Δ values, $\hat{\Sigma}_\Delta$, with a Frobenius norm

$$D_\Sigma = \frac{\|\hat{\Sigma}_\Delta - cI\|_F}{\|cI\|_F}. \tag{19}$$

We evaluate this covariance deviation across a range of Δ values across all environments in Table 1. The observed deviations are considered non-negligible, particularly in the Ant Habitat environment near the extremes of preference. However, they remain sufficiently small that we empirically find the temporal structure required for planning is preserved. We find that this covariance deviation is not correlated with higher reconstruction MSE, as evidenced by the Ant Habitat environment, where MSE is comparable to IvI despite a higher covariance deviation.

A.3 Planning Using Contrastive Representations

The distribution over intermediate latent waypoints $z_{1:n}$ from start and goal representations z_0, z_g, can be derived from the posterior

$$p(z_{1:n} \mid z_0, z_g) \propto p(z_g \mid z_n) \prod_{k=1}^{n} p(z_k \mid z_{k-1}), \tag{20}$$

in combination with (5) (Eysenbach et al., 2025). This yields the Gaussian distribution $z_{1:n} \sim \mathcal{N}(z_{1:n}\,; \mu = \Lambda^{-1}\eta, \Sigma = \Lambda^{-1})$, where

$$\Lambda = \text{tridiag}\left(-A,\, \frac{c}{c+1}A^\top A + \frac{c+1}{c}I,\, -A^\top\right), \tag{21}$$

$$\eta = \begin{bmatrix} z_0^\top A^\top & \mathbf{0} & \dots & \mathbf{0} & z_g^\top A \end{bmatrix}^\top. \tag{22}$$

Conditioned waypoints are resolved by redefining all representations as $z_{(.)} = \psi_\Delta(s_{(.)})$ and using A_Δ in place of A.

A.4 The Conditioning Variable

Part of the research gap in preference conditioning concerns the effective extraction of notions of safety from unlabeled offline datasets. In our work, Δ is computed as a statistic of the per-step cost distribution and serves as the conditioning variable for computing conditioned representations. This conditioning variable comes from both the label during training and the query during inference. The CVaR method is proposed to extract the most meaningful risk interactions within a trajectory, as explained in Section 4.1. Further work may address more extensive approaches to preference modeling; however, we focus on architectural and optimization modifications to CRLP for conditioning. The UR5 ablation demonstrates that Δ can be computed in different ways, and MoMo continues to learn meaningful representations, thereby enabling the integration of advancements in preference modeling with MoMo. We acknowledge that the adopted approach to preference labeling renders the conditioning mechanism dataset-dependent and contingent on a diverse, curated dataset, but concede this as a general limitation of learning from offline datasets, where the scope of observed behaviors is restricted. For inference-driven planning, the generated plans will further reflect the distribution of trajectories seen during training.

Using this conditioning variable Δ, the representation space can be reshaped by learning generalizable deformations of the latent manifold. We then evaluate the planner's ability to return a sequence of waypoints consistent with the safety preference, so that recovering the trend between the realized cumulative cost at inference and the user preference is the intended result. To provide another view of this result, we further quantify task success and obstacle clearance safety from closed-loop rollouts.

B Further Studies

B.1 Closed-Loop Task Performance

Protocol. The main experiments evaluate the preference-conditioned waypoint sequences produced by MoMo. Here, we additionally evaluate how those plans behave when executed in a closed loop. We use Point Four Obstacles and execute the waypoint sequences produced by MoMo and the three state-augmentation baselines using the same goal-conditioned RL policy trained with SAC (Haarnoja et al., 2018). We also compare against three methods that generate actions or trajectories directly, CCAC (Guo et al., 2025), CDT (Liu et al., 2023), and MODULI (Yuan et al., 2025).

All methods are evaluated using three training seeds, 40 start-goal queries per seed, and 11 uniformly spaced preference values. For CCAC, CDT, and MODULI, we map each method's conditioning input monotonically to the same normalized preference range. Success denotes reaching the goal in a closed loop. Cumulative cost is reported over successful trajectories.

Closed-loop preference response. Let $\bar{C}_{\mathrm{rollout}}(\Delta)$ denote the mean cumulative cost of successful trajectories at preference Δ. As in the waypoint evaluation, we measure the change in cost between the two preference extremes,

$$\Delta C_{\mathrm{rollout}} = \bar{C}_{\mathrm{rollout}}(1) - \bar{C}_{\mathrm{rollout}}(0), \tag{23}$$

and compute ρ_{rollout} as the Spearman correlation between the 11 queried preference values and the corresponding mean closed-loop costs. Because the set of successful trajectories can change with preference, we also report a second cost difference using only seed-query pairs that reach the goal at all 11 preference values. This allows us to test whether the preference-dependent change in cost persists when evaluated on the same set of queries across all preference levels. The aggregate results are summarized in Table 4.

The closed-loop results show that the preference modulation observed in the inferred waypoint sequences is preserved during execution. Across three seeds, MoMo reaches the goal in 96.4% of rollouts while producing the largest change in realized cost across the preference range ($\Delta C_{\mathrm{rollout}} = 36.40$) and the highest preference-cost correlation ($\rho_{\mathrm{rollout}} = 0.94$). Importantly, this trend is not driven by different queries succeeding at different preference values. Among the 86 seed-query pairs that succeed at all 11 values of Δ, the cost difference remains 32.74. Thus, changing Δ modifies the

Table 2: Closed-loop performance across 11 preference values on Point Four Obstacles. Success is the fraction of queries reaching the goal. Cost is the cumulative environment cost across successful rollouts, reported as the mean ± standard error across queries and seeds.

Δ	**MoMo+GCRL**		**MODULI**		**CCAC**		**CDT**	
	Success ↑	Cost ↓	Success ↑	Cost ↓	Success ↑	Cost ↓	Success ↑	Cost ↓
0.0	100.0%	25.50±1.90	100.0%	9.41±0.73	34.2%	24.11±1.28	33.3%	45.73±4.61
0.1	100.0%	21.69±1.79	95.0%	14.16±2.74	34.2%	24.14±1.27	33.3%	47.78±4.95
0.2	100.0%	20.77±1.39	91.7%	14.83±3.02	33.3%	23.88±1.34	40.0%	54.88±8.21
0.3	100.0%	22.02±1.55	88.3%	15.69±3.25	33.3%	23.92±1.33	39.2%	49.36±6.20
0.4	100.0%	25.74±1.95	85.8%	17.11±4.00	33.3%	23.96±1.33	36.7%	53.76±7.25
0.5	98.3%	33.03±1.73	81.7%	20.04±6.48	34.2%	24.56±1.73	39.2%	57.12±10.62
0.6	95.0%	35.80±2.22	76.7%	35.83±17.77	34.2%	23.96±1.35	36.7%	54.88±7.85
0.7	95.0%	42.35±2.31	70.0%	35.04±26.64	35.8%	24.70±1.37	36.7%	57.12±8.48
0.8	95.0%	43.67±3.08	65.8%	36.13±18.43	35.0%	24.07±1.35	36.7%	57.70±6.68
0.9	85.8%	60.46±3.13	58.3%	37.60±21.97	35.8%	24.51±1.30	36.7%	55.51±6.82
1.0	90.8%	61.90±3.04	55.8%	33.03±13.59	35.0%	24.14±1.25	36.7%	69.58±16.98
Mean	**96.4**%	–	79.0%	–	34.4%	–	36.8%	–

Table 3: Closed-loop execution of state-augmentation waypoint baselines using the same goal-conditioned SAC controller as MoMo. Evaluation protocol matches Table 2.

Δ	**IvI-SA+GCRL**		**VIP-SA+GCRL**		**PCA-SA+GCRL**	
	Success ↑	Cost ↓	Success ↑	Cost ↓	Success ↑	Cost ↓
0.0	90.0%	23.66±3.01	51.7%	40.20±3.37	70.0%	38.87±3.22
0.1	100.0%	29.36±3.93	50.8%	38.53±3.85	70.8%	39.27±3.27
0.2	96.7%	32.90±4.49	50.0%	38.39±2.52	70.8%	39.26±3.25
0.3	94.2%	32.61±7.04	51.7%	41.90±3.90	70.0%	38.85±3.22
0.4	96.7%	26.64±2.01	54.2%	40.54±3.74	70.0%	38.85±3.22
0.5	100.0%	26.69±1.71	53.3%	38.17±3.99	70.0%	38.85±3.22
0.6	90.0%	42.11±7.41	46.7%	37.56±4.84	70.0%	38.71±3.23
0.7	93.3%	41.39±2.42	49.2%	38.50±5.23	70.0%	38.71±3.23
0.8	95.8%	37.52±5.21	50.0%	39.47±3.83	70.0%	38.57±3.24
0.9	73.3%	60.22±5.81	48.3%	39.60±5.45	70.0%	38.59±3.24
1.0	63.3%	46.96±8.44	44.2%	37.50±5.72	70.0%	38.45±3.24
Mean	**90.3**%	–	50.0%	–	70.2%	–

executed behavior of the same start-goal queries rather than simply changing which queries reach the goal.

The comparisons also show why success and preference modulation need to be considered together. MODULI produces a well-ordered change in cost ($\rho_{\text{rollout}} = 0.90$), but its success decreases from 100% at $\Delta = 0$ to 55.8% at $\Delta = 1$. CDT similarly shows a strong preference-cost correlation when all successful rollouts are pooled, but only 25 of 120 seed-query pairs succeed at every preference value. For those same queries, the cost difference decreases from 23.86 to 1.83. In contrast, MoMo maintains both high success and a substantial change in cost when the evaluated queries are held fixed across the preference range.

The state-augmentation results further indicate that successful waypoint execution alone does not guarantee reliable preference control. IvI-SA retains relatively high success at 90.3%, but produces a smaller and less consistent change in realized cost ($\Delta C_{\text{rollout}} = 23.31$, $\rho_{\text{rollout}} = 0.81$). VIP-SA and PCA-SA show still weaker responses, with the realized cost either decreasing or remaining nearly unchanged as Δ increases. Together, these results show that MoMo's advantage extends beyond producing different waypoint sequences: the conditioned plans induce distinct, consistently ordered behaviors after closed-loop execution.

Table 4: Summary of closed-loop performance on Point Four Obstacles across three training seeds. All-Δ results use only seed-query pairs that reach the goal at every queried preference value.

Method	Success ↑	$\Delta C_{\text{rollout}}$ ↑	ρ_{rollout} ↑	All-Δ success	All-Δ ΔC ↑
MoMo+GCRL	**96.4%**	**36.40**	**0.94**	**86/120**	**32.74**
MODULI	79.0%	23.62	0.90	61/120	20.13
CCAC	34.4%	0.03	0.35	38/120	0.41
CDT	36.8%	23.86	0.90	25/120	1.83
IvI-SA+GCRL	90.3%	23.31	0.81	35/120	28.20
VIP-SA+GCRL	50.0%	−2.70	−0.39	25/120	0.92
PCA-SA+GCRL	70.2%	−0.43	−0.95	84/120	−0.43

Table 5: Mean minimum obstacle clearance across successful closed-loop rollouts, reported as mean $\pm$ standard error across queries and seeds. Values are in meters.

Method	Minimum clearance ↑
MoMo+GCRL	0.152 ± 0.012
MODULI	$\mathbf{0.178 \pm 0.049}$
CCAC	0.047 ± 0.020
CDT	0.028 ± 0.005
IvI-SA+GCRL	0.147 ± 0.011
VIP-SA+GCRL	0.121 ± 0.013
PCA-SA+GCRL	0.063 ± 0.011

Obstacle clearance. We also evaluate safety using minimum obstacle clearance. This measure is independent of the CVaR value used to define the preference label and directly measures how close an executed path comes to an obstacle.

MoMo achieves a mean minimum clearance of 0.152 ± 0.012 m, exceeding the three state-augmentation baselines as well as CCAC and CDT. This shows that changing the preference does not simply produce higher-cost paths that pass arbitrarily close to obstacles. MODULI achieves the largest clearance at 0.178 ± 0.049 m, but its success decreases from 100% at $\Delta = 0$ to 55.8% at $\Delta = 1$. MODULI therefore keeps more distance from obstacles, while MoMo reaches the goal more reliably across the preference range.

Overall, the closed-loop experiments show that the behavior produced by MoMo's waypoint plans remains after the goal-conditioned controller follows those plans. MoMo achieves the highest goal-reaching success and the largest change in cost across the preference range, while maintaining greater obstacle clearance than all compared methods except MODULI. The waypoint experiments in the main paper evaluate the plans directly, while these results show that the same preference-dependent behavior is also observed when the plans are executed.

B.2 Representation Space Deformation

This section presents further analysis of how conditioning affects the structure of the representation space. To characterize the deformation of the latent manifold, we embed a random collection of states across a range of preference contexts and consider different relation metrics between these conditioned representations.

To understand how the local topology of the latent space changes, we apply k-nearest-neighbors (kNN) in the representation space to a subset of states for a range of Δ values. For every (Δ_i, Δ_j) pair, we then compute the local overlap, defined as the average proportion of the k neighbors which are shared between the two Δ values. Values closer to one indicate that local neighborhoods are highly similar, whilst a value of zero implies the local topology has changed completely. An example of the resultant matrix for the four obstacle environment is provided in the left panel of Figure 5, where we observe strong local overlap for similar preference values, which then smoothly decreases as the difference between Δ_i and Δ_j widens. Importantly, the lowest overlap level, occurring at the

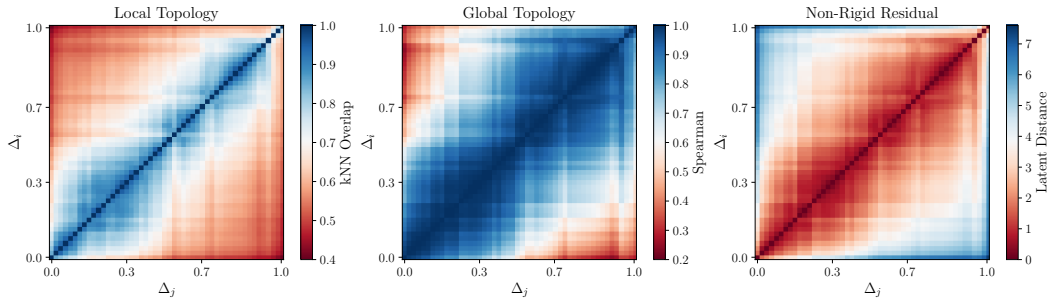

Figure 5: Matrix summaries of representation space manifold deformation for Point Four Obstacles environment under preference-conditioning. Plots characterize local changes via kNN overlap (left), global changes via Spearman's rank correlation (middle), and non-rigid transformation effects via residual latent distances (right).

extremes of zero and one, is non-negligible, at approximately 40%. This captures temporal relations in regions of the representation space shared across all preference contexts.

For measuring global changes in the latent space, we compute the distance of randomly sampled state pairs for a range of Δ values and take the Spearman correlation between these distances, as shown in the middle panel of Figure 5. Values closer to one indicate that the global distance ranking is preserved. We again observe the dissipation of this similarity further from the diagonal entries and the strongest global deformation of the space for highly mismatched preferences.

To motivate the source of these deformations, we further embed a set of states at each preference value. For every (Δ_i, Δ_j) pair, the two representation sets are centered, the best orthogonal alignment is then identified, and then the remaining RMS mismatch is reported in the right panel of Figure 5. A value of zero indicates that the deformation is purely translational and rotational, so that it can be modeled as a linear transformation. We observe that as preferences differ, the residual increases, and this rigid transformation is insufficient to capture the deformation. Instead, non-linear sources of preference-dependent warping are introduced. This further validates the choice of intermediate feature modulation in the conditioned encoder architecture. Together, these metrics point towards a healthy pattern in continuous preference modulation via deformation of the latent manifold.

We also analyze the FiLM and low rank modulated transition matrix, which produce these effective deformations. Top panels of Figure 6 demonstrate how the FiLM generator's output parameters and perturbed matrix elements remain similar for close Δ values, and gradually separate with changing preferences. We visualize the path of these conditioning parameters in the PCA-projected space. This yields a continuous path consistent across the two conditioning mechanisms and points to the overall stability of the joint conditioning mechanism employed in MoMo.

B.3 Preference Modulation Across Training Seeds

Protocol. We evaluate preference modulation of the inferred waypoint sequences over four training seeds, $\{0, 1, 2, 3\}$. For each seed, all methods are evaluated on the same set of start-goal queries at 11 uniformly spaced preference values. All environments use 12 waypoints except UR5, which uses 6. Metrics are first computed from the mean cost curve for each seed and then reported as mean $\pm$ sample standard deviation across seeds.

Let $\bar{C}_{k,j}$ denote the mean plan cost for seed k at preference Δ_j. We compute,

$$\Delta C_k = \bar{C}_{k,10} - \bar{C}_{k,0}, \tag{24}$$

$$\rho_k = \text{Spearman}\left((\Delta_j)_{j=0}^{10}, (\bar{C}_{k,j})_{j=0}^{10}\right), \tag{25}$$

$$R_{\text{norm},k} = \frac{\sum_{j=1}^{9} \left|\bar{C}_{k,j+1} - 2\bar{C}_{k,j} + \bar{C}_{k,j-1}\right|}{|\Delta C_k|}. \tag{26}$$

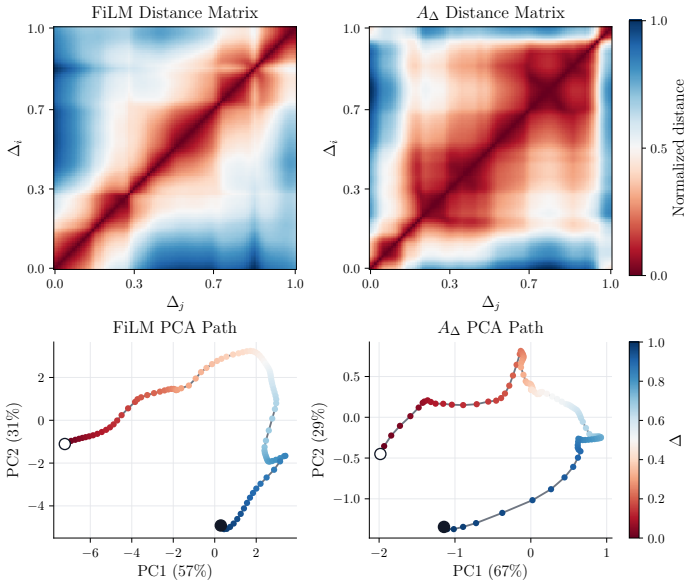

Figure 6: Summary of FiLM and transition matrix conditioning mechanism parameters for Ant Habitat environment. Across the preference spectrum, panels show the $\|.\|_2$ distance between FiLM generator outputs (top left), Frobenius norm $\|.\|_F$ of the conditioned transition matrix (top right), path of FiLM generator outputs across the first two principal components (bottom left), and the path of the flattened transition matrix elements across the first two principal components (bottom right).

A lower R_{norm} corresponds to a smoother cost curve. Since this normalization becomes sensitive when ΔC is close to zero, we interpret the three metrics together. Table 6 reports the resulting four-seed statistic for MOMO and the state-augmentation baselines.

Consistency across seeds. MOMO produces the intended preference response consistently across training seeds. In all 24 seed-specific evaluations, the waypoint cost increases from $\Delta = 0$ to $\Delta = 1$, and the preference-cost correlation is positive. It also achieves the lowest mean curve roughness in five of the six environments.

The state-augmentation baselines are less consistent across tasks and seeds. IvI-SA also exhibits an increasing-cost response and a positive correlation in 23 of 24 seed-specific evaluations, but its average R_{norm} across environments is 8.09, compared with 1.99 for MOMO. On Ant Habitat, for example, IvI-SA produces a larger change in cost (9.46 ± 6.43 compared with 3.25 ± 1.05), but with substantially higher roughness (11.73 ± 7.72 compared with 2.37 ± 0.54).

PCA-SA produces a strong response on UR5 ($\Delta C = 1.88 \pm 0.03$, $\rho = 0.93 \pm 0.06$), but remains nearly unresponsive on the other five environments. VIP-SA performs well on Drone and Ant Habitat, but reverses the direction of the preference response on Point Four Obstacles, Point Habitat, and UR5. It has jointly positive ΔC and ρ in 12 of the 24 seed-specific evaluations.

Variation across tasks. The multi-seed results also show that MOMO retains preference modulation across substantially different task settings. On the three Point environments, the mean preference-cost correlations are 0.97 ± 0.02, 0.99 ± 0.00, and 0.92 ± 0.04, while UR5 achieves 1.00 ± 0.00. On Ant Habitat, MOMO maintains a positive correlation of 0.74 ± 0.15 while producing the smoothest cost response among the compared methods. The drone setting is the most variable ($\rho = 0.43 \pm 0.34$), where VIP-SA achieves a stronger correlation.

Overall, the four-seed evaluation shows that MOMO provides the most consistent preference modulation across the full set of environments. While a baseline can achieve a larger cost range or correlation on an individual task, MOMO is the only method whose response remains correctly directed across all tested environments and training seeds. Generally, it produces smoother changes across the range of preferences.

B.4 State Augmentation Comparison

A candidate conditioning mechanism for modulating preference in planning behavior is state augmentation, which serves as a baseline for comparison. In this approach, we define a new state $\tilde{s} = [s^\top \ \Delta]^\top$ such that $\tilde{s}_+ = [s_+^\top \ \Delta_+]^\top$. Note that this is not equivalent to the rewriting of only the anchor as $\hat{s} = [s^\top \ \Delta]^\top$, which is done to perform a single marginalization step in the proof provided in Appendix A.1. For the IvI-SA baseline, simply substituting the augmented state pair $(\tilde{s}, \tilde{s}_+)$ into the sequence of steps provided from (12) to (16), then unpacking the augmentation in the final step, we arrive at the density ratio

$$(\tilde{g})^*(\tilde{s}, \tilde{s}_+) \propto \frac{p(s_+, \Delta_+ \mid s, \Delta)}{p(s_+, \Delta_+)}.$$

This ratio captures the odds of seeing a future in its own distinct preference context, given a starting state in another. Note that s_+ represents any possible future following marginalization in (14), not strictly a positive one. Since positive pairs are sampled from the same trajectory, this would imply that $(\tilde{g})^*$ is only non-zero when $\Delta = \Delta_+$. This risks collapsing the representation space, since the appended preference context can now be used independently to distinguish positive and negative pairs, thereby offering a degenerate solution to the optimization problem if the contrastive signal from the remaining states is weak.

The learned Gauss-Markov distribution therefore admits intermediate waypoints that share the same preference context as the augmented start and goal. The key distinction here is that a single unified representation space has been learned, rather than a conditioned deformation.

It is critical that, with this formulation, state augmentation can encode precise temporal relations in the latent space. By inspection of Figure 4, we observe large waypoint MSE using state augmentation. We also observe less consistent preference responses across tasks, as reflected in unstable correlations, response directions, and curve roughness among the state augmentation baselines. If able to span a wider range of plan costs, it does not do so smoothly across the preference continuum. Therefore, across the metrics outlined in Table 6, MOMO demonstrates a consistent modulation of behaviors offering greater robustness across environments than the profiled baselines.

Under the injection of trajectory-level information into the augmented state, such that two identical original states can now lie on separate planes, state augmentation also becomes more susceptible to coverage issues in the held-out set for decoding, since every candidate state must be seen in the desired preference context.

C Limitations

In this section, we outline key limitations of our approach, both originating from preference conditioning and inherited from the inference-driven planning framework we build upon (Eysenbach et al., 2025).

Firstly, conditioning representations on Δ increases the computational complexity of a single training step from $\mathcal{O}(B)$ in (3) to $\mathcal{O}(B^2)$ in (10). This arises from the need to embed states within each anchor's preference context; this is not computationally prohibitive. Furthermore, at inference time, nearest-neighbor retrieval mandates that the representations of observations in $\mathcal{D}_{\text{held}}$ be computed for the queried preference. Since we address preferences on a continuous scale, these embeddings cannot be pre-computed and cached. Whilst this reduces the runtime efficiency of the contrastive planner, a fast batched forward pass through the architecture is also not found to be prohibitive.

The safety of the decoded waypoint sequence further relies on good coverage of the state space in the held-out set. It is also implicitly dependent on sufficient coverage of the joint preference and state space in the training set, since this is required to learn a generalizable conditioning mechanism. At inference time, if coverage in $\mathcal{D}_{\text{held}}$ is sparse, the preference-dependent deformation of the latent space could cause nearest-neighbor retrieval to decode a safe latent waypoint back into an unsafe state. This limitation motivates an exploration of alternative decoding methods.

We also inherit assumptions from Eysenbach et al. (2025), including that the learned logits encode the required density-ratio and that representations follow an isotropic Gaussian distribution. We justify these assumptions in Appendix A; however, as in Eysenbach et al. (2025), it remains open how

Table 6: Preference modulation profile metrics comparing MoMo to state augmentation baselines. ΔC is the difference in total plan cost between queries with $\Delta = 1$ and $\Delta = 0$. ρ is Spearman correlation between Δ and mean cost curve. R_{norm} is cost-curve roughness defined as the sum of absolute second-order finite differences in cost normalized by $|\Delta C|$. Values report mean $\pm$ sample standard deviation across four training seeds.

Environment	Metric	**MoMo**	IvI*	PCA*	VIP*
Point Four Obstacles	$\Delta C \uparrow$	**8.22±0.79**	5.53±3.72	−0.00±0.01	−0.54±0.79
	$\rho \uparrow$	**0.97±0.02**	0.75±0.07	−0.11±0.79	−0.38±0.55
	$R_{\text{norm}} \downarrow$	**1.48±0.13**	11.77±11.97	3.89±2.04	32.99±30.08
Point Contour	$\Delta C \uparrow$	2.68±0.34	**3.46±1.51**	−0.00±0.00	0.27±0.55
	$\rho \uparrow$	**0.99±0.00**	0.93±0.06	−0.06±0.73	0.37±0.65
	$R_{\text{norm}} \downarrow$	**0.75±0.06**	4.09±4.12	3.32±1.37	10.90±12.08
Point Habitat	$\Delta C \uparrow$	7.14±0.70	**15.05±2.71**	−0.01±0.01	−0.43±1.87
	$\rho \uparrow$	**0.92±0.04**	**0.92±0.07**	−0.01±0.66	−0.23±0.68
	$R_{\text{norm}} \downarrow$	**1.59±0.42**	3.32±2.72	9.12±7.66	11.45±9.08
Drone	$\Delta C \uparrow$	2.36±0.91	**4.52±5.73**	−0.01±0.06	2.83±1.62
	$\rho \uparrow$	0.43±0.34	0.39±0.35	0.00±0.35	**0.85±0.08**
	$R_{\text{norm}} \downarrow$	5.30±2.45	11.82±9.68	10.79±5.57	**1.71±0.96**
Ant Habitat	$\Delta C \uparrow$	3.25±1.05	**9.46±6.43**	0.01±0.03	5.97±3.99
	$\rho \uparrow$	0.74±0.15	0.64±0.19	−0.00±0.47	**0.84±0.08**
	$R_{\text{norm}} \downarrow$	**2.37±0.54**	11.73±7.72	13.70±10.59	2.69±1.52
UR5	$\Delta C \uparrow$	1.21±0.21	1.28±0.67	**1.88±0.03**	−0.76±0.55
	$\rho \uparrow$	**1.00±0.00**	0.54±0.21	0.93±0.06	−0.56±0.85
	$R_{\text{norm}} \downarrow$	**0.46±0.25**	5.81±3.95	0.68±0.02	56.91±112.66

Notes. *Ours* = MoMo; * denotes the corresponding state-augmentation baseline. Plain IvI is omitted because it is non-responsive ($\Delta C = 0$). All environments use 12 waypoints, apart from UR5, which uses 6.

errors from function approximation and sparse sampling introduced by conditioning affect MoMo's planning performance.

Finally, data-driven planning is constrained by the quality of existing datasets, which are difficult to acquire and provide dense coverage of the joint state-preference space. Furthermore, CRLP assumes all exploration of approaches to solving tasks has been performed, since no new temporal transitions external to the dataset can be extracted.

D Dataset Generation

We construct trajectory datasets that cover the joint state-preference space required to learn preference-conditioned planning representations. For each of the six tasks, we collect rollouts at multiple discrete risk levels and annotate each trajectory with the CVaR-based preference statistic from Section 4.1 except for the UR5 environment.

For the *Point* and *Ant* agents, we adapt the D4RL collection framework (Fu et al., 2021) to generate trajectories between randomized start and goal positions. The environment is discretized into a binary occupancy grid on which we perform Q-iteration to generate discrete waypoint sequences. A short-horizon goal-reaching policy then produces continuous trajectories between consecutive waypoints. We use a stochastic PD controller for the *Point* agent and a pre-trained SAC policy for the *Ant* agent. Q-iteration uses discrete state s corresponding to the position grid index, actions a in the cardinal and intercardinal directions, deterministic transitions to the next state s', goal g, and the reward

$$R(s, a, g\,; w_\Delta) = -\left[1 + \max\big(0,\, d_b(s', g) - d_b(s, g)\big) + w_\Delta \left(1 - \frac{d_o(s)}{\theta}\right) \mathbb{1}\big(d_o(s) < \theta\big)\right]. \quad (27)$$

In the above, d_b represents the shortest path distance between two cells (pre-computed via breadth-first search for all state pairs) and d_o gives the distance to the nearest obstacle, for which a penalty

Table 7: Summary of training datasets with sufficient state-preference space coverage.

Environment	Collection framework	Risk levels	Observation dimension	Trajectories	Transitions
Point Four Obstacles	D4RL	3	4	16,557	3.31M
Point Contour	D4RL	4	4	51,626	12.00M
Point Habitat	D4RL & Habitat	3	4	29,488	7.50M
Ant Habitat	D4RL & Habitat	3	29	20,186	7.50M
Drone	FSRL	4	18	7,200	2.04M
UR5	PyBullet	4	57	63,000	1.73M

is incurred within an influence radius θ. By varying w_Δ, the degree of risk-aversion baked into the Q-iteration waypoints is controlled. Rollouts are generated across discrete risk levels defined by distinct w_Δ values. We use a four-obstacle map, an open map with a dense Gaussian cost contour, and extract a map from the Replica Dataset FRL Apartment 5 loaded from AI Habitat to model realistic environments. For annotating continuous trajectories, we use a risk cost function that increases linearly with obstacle proximity within the influence radius.

The dataset for the *Drone* agent is produced using the data collection method and TRPO-Lagrangian policy for the *OfflineDroneCircle-v0* environment from FSRL (Liu et al., 2024), trained at different target orbit radii. Preference annotation uses a risk cost which increases linearly with radial distance from a safe central column of radius θ.

The *Manipulator* environment consists of a tabletop pick-and-place task in which the robot retrieves a cube from a start bowl and moves it to the goal bowl. Between the bowls, there is a divider with a slot, such that safer trajectories correspond to higher clearance above the obstacle, whilst more efficient, riskier motions slide through the divider. Trajectories are generated by simulation in PyBullet using a UR5 robot. To obtain a diverse set of preference behaviors, we apply varying levels of obstacle inflation to Bi-directional Rapidly-exploring Random Trees (Kuffner & LaValle, 2000) for planning. Cost is computed from a margin-adjusted sigmoid function of vertical clearance between the end-effector and divider, weighted by the end-effector's xy-position, and wrapped to the unit interval.

E Model Training

We build off of the unconditioned model presented in (Eysenbach et al., 2025), with key parameters summarized in Table 8. Since all conditioning modules, including the FiLM generator and two low-rank modulation networks, would generate an intermediate representation of the preference context, we introduce a separate condition-embedding network to feed these three modules. This enables efficient shared condition embedding to inform the modulation mechanisms.

Direct conditioning on the raw scalar Δ value may bottleneck performance due to neural networks' spectral bias towards low-frequency functions. This means that networks struggle to produce local fluctuations without affecting global behavior (Rahaman et al., 2019), an important phenomenon for producing generalizable representations from sparse coverage of the joint state-preference space. This merits a richer representation of the conditioning variable, which we address using Fourier features (Tancik et al., 2020). Exact implementation details are in our code supplement.

Whilst our derived results and those in (Eysenbach et al., 2025) exclude the positive pair from the denominator of the InfoNCE objective, standard empirical implementations sum over all indices to yield a softmax distribution and bounded loss. In line with prior work, we also adopt this convention (Eysenbach et al., 2025; Betser et al., 2026).

We train our models on an Ubuntu 22.04 machine with a GeForce RTX 4080 GPU and an Intel Core i9-14900K CPU, resulting in approximately 2 hours of training time for MoMo.

In this work, we used generative AI tools to adapt closed-loop benchmark codebases to our training setup. We have not used generative AI tools in the ideation process, paper writing, or for research and literature review. All AI-assisted code has been reviewed and tested.

Table 8: Model parameters used for MoMo.

Component	**Value**
Encoder, predictor, and conditioning models	
Representation dimension	$d_z = 4$ for *Point* and *Manipulator*; $d_z = 6$ for *Drone*; $d_z = 24$ for *Ant*
Encoder network	Residual MLP with 5 residual blocks; width 64; Swish activations
FiLM generator network	MLP with 5 hidden layers; width 128; Swish activations
Condition embedding input features	$[\Delta, \sin(2\pi\Delta), \cos(2\pi\Delta), \sin(4\pi\Delta), \cos(4\pi\Delta)]$
Condition embedding network	MLP with 2 hidden layers; width 16; Swish activations; 16-dimensional output linear embedding.
Low-rank A_Δ network	Separate U_Δ and V_Δ MLPs each with 2 hidden layers; width 64; ReLU activations
Low-rank A_Δ rank	$r = 2$
CVaR level	$\alpha = 0.9$
Optimization	
Batch size	256
Optimizer	Adam
Learning rate	3×10^{-4}
Global norm gradient clipping	1.0
Future sampling discount factor	$\gamma = 0.9$ for *Point, Ant, Manipulator*; $\gamma = 0.8$ for *Drone*
Maximum future horizon	200 steps
Training step updates	$500,000$
Representation norm constraint target	$c = 10$
Dual descent initialization	0.001