MoMo: Conditioned Contrastive Representation Learning for Preference-Modulated Planning

Yusuf Syed, Viraj Parimi, Brian Williams

Massachusetts Institute of Technology
Cambridge, MA 02139
{yusufs,vparimi,williams}@mit.edu

Abstract

Temporally contrastive representation learning induces a latent structure capable of reducing long-horizon planning to inference in a low-dimensional linear system. However, existing contrastive planning work learns a single latent geometry which cannot distinguish multiple valid behaviors 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 that is required for inference-driven contrastive planning, further retaining its inference-time efficiency. Across six environments, MoMo smoothly adapts plan safety according to user preferences, yielding improved temporal and preferential consistency over 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 [4, 6, 27]. For instance, a surgical robot may maintain a larger clearance during a delicate phase of an 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 exposure to risk, as illustrated in Figure 1. This balance is not fixed *a priori*, but varies with mission context, ultimately reflecting a specified safety-efficiency tradeoff for planning behavior. This creates a need for *preference-conditioned* planning, in which a planner flexibly reshapes its behavior at inference time, without retraining, in response to a scalar safety preference encoding the user's risk tolerance.

Existing approaches to risk-aware planning span model-based methods that depend on known dynamics [3, 37, 8, 10, 23, 7], learning-based methods with fixed risk specifications [14, 46], and recent risk-conditioned policies that allow inference-time adaptation [22, 58, 60, 35]. 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 promising foundation for addressing this gap [13, 12]. By learning representations from trajectory data whose geometry reflects temporal reachability, CRLP induces a latent structure reducing long-horizon planning to low-dimensional linear inference [12], and has shown practical viability in robotic settings [62, 61]. This makes CRLP attractive when dynamics are unknown and efficient inference-time planning is required. Existing contrastive planning methods, however, are preference-agnostic, encoding temporal similarity alone and recovering only a single mode of behavior 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 a user-specified preference. We approach this challenge by considering behavioral differences arising at two levels: the organization of states in the representation space, and the transitions between them. Conditioning representations yields a preference-aware geometry, while conditioning the transition operator reshapes how trajectories evolve under varying preferences.

To unify these components, we introduce MoMo, a preference-conditioned contrastive planning framework that jointly conditions the representation geometry and the 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 richer 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

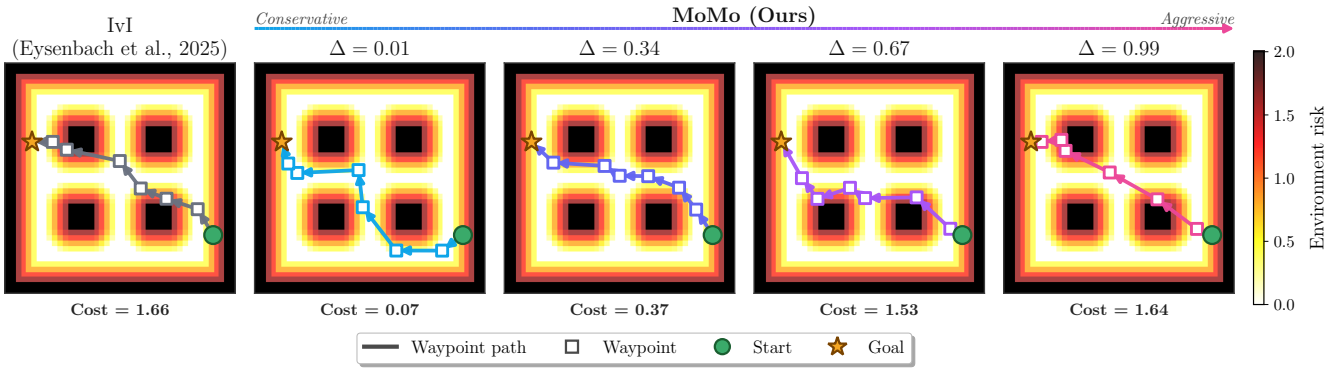

Figure 1: Preference-conditioned planning in a risk-structured environment with four obstacles and a cost function reflecting obstacle proximity. Different scalar safety preferences (Δ) modulate waypoint plans, ranging from low-risk detours to more direct, riskier routes.

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 [3, 37, 8, 10, 23, 7]. Risk-aware control bounds tail risk through conditional value-at-risk (CVaR) constraints [20], while control barrier functions filter control inputs to maintain safety invariants [53]. These approaches enable principled safety guarantees but assume known, often linearized dynamics and become expensive in high-dimensional or long-horizon settings, motivating learning-based alternatives that remove the dependence on explicit dynamics models.

Adaptive Safe Policy Learning. Learning-based methods relax dependence on known dynamics by extracting behaviors directly from trajectory data. Feng, Parimi, and Williams [14] combine graph-based waypoint planning with safe goal-conditioned RL policies for multi-agent navigation, but enforce safety through graph pruning, which can disconnect goals beyond high-risk regions [38]. Huang et al. [22] and Yoo, Park, and Woo [58] introduce risk-conditioned policies that allow inference-time adaptation across a continuous risk spectrum, but rely on direct policy conditioning, with limited scalability over longer horizons. In the offline setting, decision-transformer approaches such as Liu et al. [30] condition on desired return and constraint thresholds for zero-shot adaptation across cost budgets, with extensions to safety verification [60], signal temporal logic specifications [16], and constraint-conditioned actor-critic learning [17]. Benchmark suites such as DSRL [29] have accelerated progress in this area, but center on safety-constrained control rather than granular preference modulation. Mao et al. [35] follow a different approach, learning preference-conditioned terrain costmaps for downstream planning. Across these methods, preference or risk adaptability operates at the level of policies or costmaps, not through a learned representation whose structure supports planning by inference.

Preference-Conditioned and Multi-Objective RL. Multi-objective RL (MORL) addresses tradeoffs by learning policies or value functions conditioned on preferences over vector-valued returns [57, 5]. Offline variants such as PEDA, together with the D4MORL benchmark, extend this to preference-conditioned decision transformers trained on annotated demonstrations [63], while recent work emphasizes comprehensive coverage of the Pareto front for unseen preference adaptation [28]. 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 offer a route to tractable planning in high-dimensional settings. Early work learns latent dynamics from observations for control and online planning [56, 19], with subsequent methods studying planning directly in latent spaces [24, 39, 59]. Reeves and Williams [44] extend this to stochastic, risk-aware planning by learning linear dynamics in latent space and propagating uncertainty through probabilistic flow tubes [11] sampled from a variational autoencoder [25]. 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 contrastive representations for planning. Time-contrastive networks and successor representations demonstrate how temporal structure in trajectory data can be captured in learned representations [48, 9, 1]. Eysenbach et al. [13] show that contrastive learning can be interpreted as goal-conditioned RL, with subsequent work improving performance from offline data [62, 61]. Eysenbach et al. [12] show that temporally contrastive representations induce a Gauss-Markov latent structure in which long-horizon planning reduces to efficient inference over intermediate waypoints in a lower-dimensional linear system. Separately, conditional representation learning offers tools for reshaping embeddings as a function of auxiliary variables. Examples include masked subspaces [55], kernel-based objective reweighting [32, 52, 36], Feature-wise Linear Modulation (FiLM) [40] for controlling intermediate activations, and HyperNetworks [18] for generating network parameters from auxiliary inputs. These conditioning methods must be applied carefully so as not to disrupt the latent structure enabling 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 from an offline dataset

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

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 [13, 12], 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 $p_\gamma(s_+|s)$ [13], using a discounting factor $\gamma \in (0, 1)$ which controls the effective temporal horizon of the sampling distribution. Negative pairs are formed from an anchor and negative future s_- by sampling independently from the product of marginal distributions $p(s)p(s_-)$.

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 [12]. The symmetrized InfoNCE objective [54, 12] trains the encoders to distinguish the positive from negative pairs in the sampled batch:

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

Optimizing (3) pulls temporally related states together and pushes unrelated states apart, so that latent distance reflects temporal reachability. While standard contrastive learning constrains representations to the unit hypersphere [13], following Eysenbach et al. [12] 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 a regularization term and dual gradient descent, where c is the constraint target [12]. The anchor encoder is parameterized as a linear transformation of the future encoder, such that $\phi(s) = A\psi(s)$, where $A \in \mathbb{R}^{d_z \times d_z}$ acts as a latent predictor mapping representations to their γ-discounted future. Crucially, the optimal logits learned under (3) 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 [12] 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. [12] derive that the resultant latent dynamics

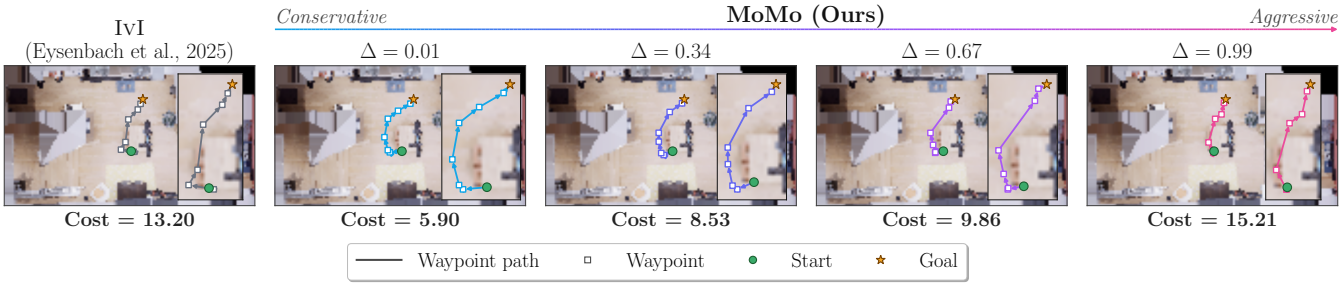

Figure 2: Preference-conditioned planning on the 29-dimensional *Ant* agent in an AI Habitat environment. Risk is defined as a linear function of obstacle proximity within an influence radius. For a fixed start and goal, increasing Δ reduces the planned obstacle clearance for a more efficient path.

from $z = \psi(s)$ to its γ-discounted future representation z' become

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

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 [12], 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 [12], 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 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, while preserving the density-ratio structure required for planning by inference. The remainder of 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. We therefore use the empirical cost distribution upper-tail, following prior work in risk-conditioned planning that treats tail-risk metrics as a natural conditioning signal [46, 22]. Letting $q_\alpha(\tau_i)$ denote the empirical 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 the upper-tail steps, we define the raw tail-risk statistic as the empirical CVaR [45] of the per-step costs,

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

This averages over the highest cost portions of the trajectory, isolating behaviorally meaningful interactions with risk while excluding the zero-cost segments common to both conservative and risk-tolerant behaviors. We wrap these raw values $\{\widetilde{\Delta}_i\}$ 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 preference label, shared across all states in the trajectory, which sets the preference context for every contrastive pair sampled from it. We assume that $\mathcal{D}$ comprises trajectories that do not take on additional risk without commensurate efficiency benefit. This makes the cost-computed Δ a meaningful conditioning variable for behavioral modulation.

4.2 Preference-Conditioned Latent Geometry

In CRLP, the encoder organizes states according to temporal reachability alone. Under preference-conditioning, the same state may play different roles depending on the desired safety level. A state near an obstacle, for example, may be a plausible waypoint under a risk-tolerant preference but an unlikely one under a conservative preference. The encoder must therefore jointly capture temporal reachability and preferential similarity. Schemes that apply a condition-dependent mask to a shared embedding [55] leave the intermediate features unchanged, limiting the expressivity with which preference can reshape the representation.

We instead modulate intermediate representations using Feature-Wise Linear Modulation (FiLM) [40], which applies Δ-dependent affine transformations to the encoder's hidden layer activations, allowing the preference to reshape features hierarchically through 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

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

where $\odot$ denotes an elementwise product. We use the Swish nonlinearity [43] following Eysenbach et al. [12]. 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 the learned dynamics to a common baseline, 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 $2rd_z$, which aids scalability to higher-dimensional representations and mitigates 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 [51]. 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. [12], 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

$$\mathcal{L}_{\text{MoMo}} = \sum_{i=1}^{B} \Bigg\{ \log \frac{S_{ii}^{\Delta_i}}{\sum_{\substack{j=1 \\ j\neq i}}^{B} S_{ij}^{\Delta_i}} + \log \frac{S_{ii}^{\Delta_i}}{\sum_{\substack{j=1 \\ j\neq i}}^{B} S_{ji}^{\Delta_j}} \Bigg\}. \tag{10}$$

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. [12], for any sampled pair (s^i, s_+^j) and preference $\Delta \in [0,1]$, the optimal conditioned similarity satisfies

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

as shown in Appendix A.1. This is the conditioned analogue of (4), preserving the density-ratio property of

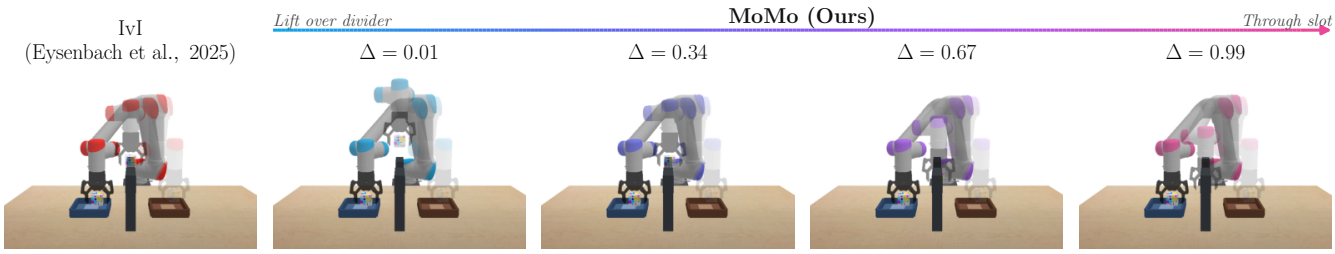

Figure 3: Preference-conditioned planning on a pick-and-place *Manipulation* task, where the UR5 robot must move a cube between start and goal regions over a gate divider. MoMo shows behavioral gradation between safely lifting above and efficiently sliding through the gate as Δ increases.

Algorithm 1 MoMo: Preference-Conditioned Contrastive Planning. Color denotes conditioning of the representation geometry and latent prediction matrix; black follows Eysenbach et al. [12]

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
end for

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 [12] with ψ_Δ, A_Δ replacing ψ, A
  Decode $z_{1:n}$ to $s_{1:n}$ via nearest-neighbor retrieval under ψ_Δ in $\mathcal{D}_{\text{held}}$
return Preference-conditioned plan $s_{1:n}$

the base contrastive planner. The representation space now encodes how likely it is to observe a given future state from a starting state, under the user-preference set at runtime. This distinction from (4) enables behavioral modulation while keeping inference-driven planning tractable under MoMo. Eysenbach et al. [12] further assume that the marginal distribution over representations is isotropic Gaussian, which we argue remains consistent with our approach in Appendix A.2. 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 as outlined in Algorithm 1. In practice, standard goal-reaching policies trained over short horizons can 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 six environments spanning four agents, summarized in Table 2, including three navigation tasks for a 2D *Point* agent (an obstacle field, a Gaussian cost contour, and an indoor scene from AI Habitat [49, 47, 50, 41]), 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 offline trajectory datasets at multiple discrete agent risk levels and annotate them with preference labels following Section 4.1. Dataset construction adapts D4RL [15] and FSRL [29]. Details about the datasets can be found in the Appendix D. We compare MoMo against four baselines. IvI [12] 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 [12], PCA-SA uses linear principal-component projections of raw observations, and VIP-SA uses representations from VIP [33] which encode temporal distances. Together, these isolate whether preference conditioning can be achieved via SA across various representation types.

Demonstrating Consistent Preference Modulation. To evaluate **Q1** and **Q2**, we train MoMo across environments and query across a continuum of Δ values

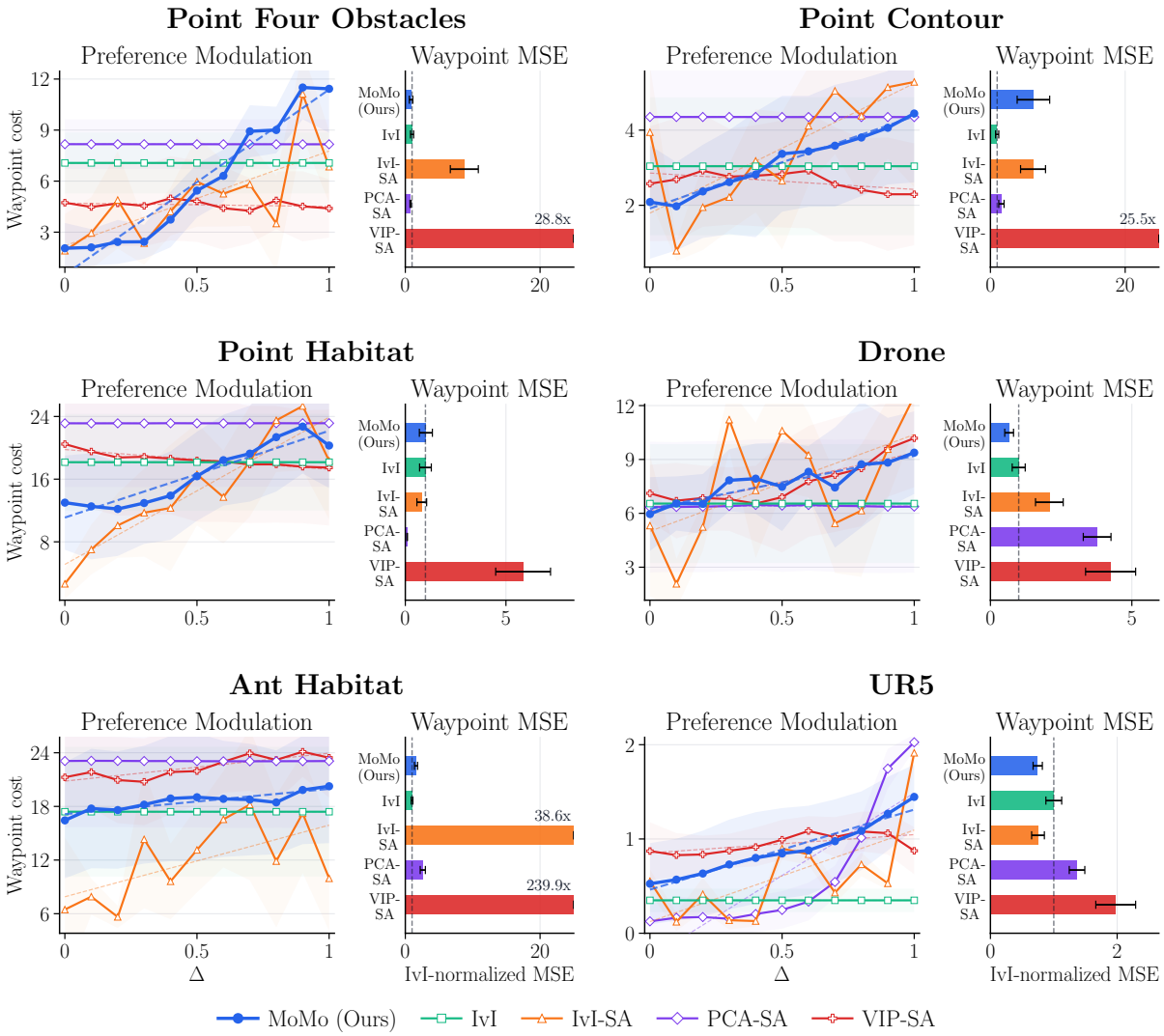

Figure 4: Preference modulation and planning fidelity across six environments. For each environment, the left panel plots waypoint cost as a function of preference Δ; the right panel reports waypoint reconstruction MSE normalized by IvI's MSE (vertical reference line at 1.0), where bars beyond 1.0 indicate baselines that fit the dataset's waypoints worse than the preference-agnostic IvI. Dashed lines in the left panels show each method's modulation trend. Bars labeled with multipliers (e.g., 28.8×) extend beyond the displayed axis range. MoMo produces a near-monotonic trend between Δ and waypoint cost while maintaining MSE comparable to IvI; IvI is preference-agnostic and produces a flat trend; the SA baselines show inconsistent or non-monotonic responses with higher MSE.

using Algorithm 1. For **Q1**, we measure waypoint reconstruction MSE relative to IvI as an inference-driven planning reference. As shown in the right panel of Figure 4, MoMo is able to achieve MSE comparable to IvI across most environments, while the SA baselines reach up to 28.8× and 38.6× higher on Point Four Obstacles and Ant Habitat, respectively. This indicates that joint conditioning preserves the representation structure required for planning by inference, whereas SA over alternative backbones disrupts it.

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 that modulation occurs at the individual query level rather than as a weak global average. 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.2. 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, traded for greater realized risk. We further analyze how our conditioning method is able to achieve this modulation via deformation of 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.

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 1. To test whether MoMo works across different environment 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 through 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 continues to modulate plans monotonically ($\rho = 0.87$), while 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 SA-based methods cannot.

To test representational capacity in high dimensions, we evaluate the 29-dimensional *Ant* quadruped navigating an AI Habitat scene. Figure 2 shows the resulting plans transitioning from cautious detours around obstacles at low Δ to direct routes at high Δ. While VIP-SA achieves a marginally higher correlation ($\rho = 0.87$ vs. 0.82), its waypoint MSE is significantly higher than IvI's, indicating that the plans themselves are not temporally consistent, whereas MoMo achieves the smoothest modulation curve across all baselines. While MoMo's correlation drops from $\rho = 0.99$ on the 2-D Point environment to $\rho = 0.82$ here, the modulation remains monotonic and the waypoint MSE is comparable to IvI. This graceful degradation across an order of magnitude in observation dimensionality suggests the approach is amenable to scaling. 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 1. These analyses demonstrate that MoMo effectively performs preference-conditioned planning in complex settings, and is robust across a range of modalities.

6 Conclusion

We introduced MoMo, a preference-conditioned contrastive planning framework that jointly conditions the representation geometry and latent transition matrix, enabling inference-time control over user preferences while preserving the planning structure of CRLP. By conditioning at the logit level using a preference context statistic that summarizes behavioral tendencies within each trajectory, we preserve the density-ratio encoding in the latent structure required by CRLP. 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 strong preference-cost correlation across diverse task settings including effective scaling to high-dimensional and longer-horizon tasks. More broadly, our work shows that representation-level conditioning, as opposed to policy- or cost-level conditioning, can support preference-conditioned planning at scale. Despite these encouraging results, there is further space for improvement upon the limitations outlined in Appendix C. One direction is to develop dual risk-and-goal-conditioned RL policies to track the conditioned waypoints, potentially mapping representations directly to actions to avoid noisy decoding. Another is to construct a general preference context embedding rather than a single scalar metric, to capture richer trajectory-level information. A third is to analyze the effects of network architecture on conditioned representation quality for planning.

References

[1] Barreto, A.; Dabney, W.; Munos, R.; Hunt, J. J.; Schaul, T.; van Hasselt, H. P.; and Silver, D. 2017. Successor Features for Transfer in Reinforcement Learning. In *Advances in Neural Information Processing Systems 30*, 4058–4068.

[2] Betser, R.; Gofer, E.; Levi, M. Y.; and Gilboa, G. 2026. InfoNCE Induces Gaussian Distribution. arXiv:2602.24012.

[3] Blackmore, L.; and Ono, M. 2009. *Convex Chance Constrained Predictive Control Without Sampling.*

[4] Bolli, R.; and Asada, H. H. 2025. 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)*, 713–719.

[5] Cai, X.-Q.; Zhang, P.; Zhao, L.; Bian, J.; Sugiyama, M.; and Llorens, A. J. 2023. Distributional Pareto-Optimal Multi-Objective Reinforcement Learning. In *Advances in Neural Information Processing Systems.*

[6] Cani, J.; Koletsis, P.; Foteinos, K.; Kefaloukos, I.; Argyriou, L.; Falelakis, M.; Pino, I. D.; Santamaria-Navarro, A.; Čech, M.; Severa, O.; Umbrico, A.; Fracasso, F.; Orlandini, A.; Drakoulis, D.; Markakis, E.; Varlamis, I.; and Papadopoulos, G. T. 2025. TRIFFID: Autonomous Robotic Aid For Increasing First Responders Efficiency. arXiv:2502.09379.

[7] Dai, S.; Schaffert, S.; Jasour, A.; Hofmann, A.; and Williams, B. 2019. Chance Constrained Motion Planning for High-Dimensional Robots. In *2019 International Conference on Robotics and Automation (ICRA)*, 8805–8811.

[8] Dawson, C.; Jasour, A.; Hofmann, A. G.; and Williams, B. C. 2020. Provably Safe Trajectory Optimization in the Presence of Uncertain Convex Obstacles. *CoRR.*

[9] Dayan, P. 1993. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 5(4): 613–624.

[10] de Groot, O.; Ferranti, L.; Gavrila, D. M.; and Alonso-Mora, J. 2025. Scenario-based motion planning with bounded probability of collision. *The International Journal of Robotics Research*, 44(9): 1507–1525.

[11] Dong, S.; and Williams, B. 2012. Learning and Recognition of Hybrid Manipulation Motions in Variable Environments Using Probabilistic Flow Tubes. *International Journal of Social Robotics*, 4(4): 357–368.

[12] Eysenbach, B.; Myers, V.; Salakhutdinov, R.; and Levine, S. 2025. Inference via Interpolation: Contrastive Representations Provably Enable Planning and Inference. arXiv:2403.04082.

[13] Eysenbach, B.; Zhang, T.; Salakhutdinov, R.; and Levine, S. 2023. Contrastive Learning as Goal-Conditioned Reinforcement Learning. arXiv:2206.07568.

[14] Feng, M.; Parimi, V.; and Williams, B. 2025. Safe Multi-Agent Navigation Guided by Goal-Conditioned Safe Reinforcement Learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 16869–16875. IEEE.

[15] Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2021. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

[16] Guo, Z.; Zhou, W.; and Li, W. 2024. 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*, 17003–17019. PMLR.

[17] Guo, Z.; Zhou, W.; Wang, S.; and Li, W. 2025. Constraint-Conditioned Actor-Critic for Offline Safe Reinforcement Learning. In *International Conference on Learning Representations.*

[18] Ha, D.; Dai, A.; and Le, Q. V. 2016. HyperNetworks. arXiv:1609.09106.

[19] Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning Latent Dynamics for Planning from Pixels. arXiv:1811.04551.

[20] Hakobyan, A.; Kim, G. C.; and Yang, I. 2019. Risk-Aware Motion Planning and Control Using CVaR-Constrained Optimization. *IEEE Robotics and Automation Letters*, 4(4): 3924–3931.

[21] Hobson, E. W. 1913. 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.

[22] Huang, X.; Feng, M.; Jasour, A. M.; Rosman, G.; and Williams, B. C. 2021. Risk Conditioned Neural Motion Planning. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9057–9063.

[23] Huang, X.; Hong, S.; Hofmann, A.; and Williams, B. C. 2019. Online Risk-Bounded Motion Planning for Autonomous Vehicles in Dynamic Environments. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29(1): 214–222.

[24] Ichter, B.; and Pavone, M. 2018. Robot Motion Planning in Learned Latent Spaces. arXiv:1807.10366.

[25] Kingma, D. P.; and Welling, M. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4): 307–392.

[26] Kuffner, J.; and LaValle, S. 2000. 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, 995–1001 vol.2.

[27] Leonard, S.; Wu, K. L.; Kim, Y.; Krieger, A.; and Kim, P. C. 2014. Smart Tissue Anastomosis Robot (STAR): A Vision-Guided Robotics System for Laparoscopic Suturing. *IEEE Transactions on Biomedical Engineering*, 61(4): 1305–1317.

[28] Liu, R.; Pan, Y.; Xu, L.; Song, L.; You, P.; Chen, Y.; and Bian, J. 2025. Efficient Discovery of Pareto Front for Multi-Objective Reinforcement Learning. In *International Conference on Learning Representations*.

[29] Liu, Z.; Guo, Z.; Lin, H.; Yao, Y.; Zhu, J.; Cen, Z.; Hu, H.; Yu, W.; Zhang, T.; Tan, J.; and Zhao, D. 2024. Datasets and Benchmarks for Offline Safe Reinforcement Learning. *Journal of Data-centric Machine Learning Research*.

[30] Liu, Z.; Guo, Z.; Yao, Y.; Cen, Z.; Yu, W.; Zhang, T.; and Zhao, D. 2023. 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*, 21611–21630. PMLR.

[31] Lu, Y.; Zhang, G.; Sun, S.; Guo, H.; and Yu, Y. 2024. f-MICL: Understanding and Generalizing InfoNCE-based Contrastive Learning. arXiv:2402.10150.

[32] Ma, M. Q.; Tsai, Y.-H. H.; Liang, P. P.; Zhao, H.; Zhang, K.; Salakhutdinov, R.; and Morency, L.-P. 2022. Conditional Contrastive Learning for Improving Fairness in Self-Supervised Learning. arXiv:2106.02866.

[33] Ma, Y. J.; Sodhani, S.; Jayaraman, D.; Bastani, O.; Kumar, V.; and Zhang, A. 2023. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. arXiv:2210.00030.

[34] Ma, Z.; and Collins, M. 2018. Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency. arXiv:1809.01812.

[35] Mao, L.; Warnell, G.; Stone, P.; and Biswas, J. 2024. PACER: Preference-conditioned All-terrain Costmap Generation. *arXiv preprint arXiv:2410.23488*.

[36] Mu, E.; and Guttag, J. 2022. Conditional Contrastive Networks.

[37] Ono, M.; and Williams, B. C. 2008. 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*, 3427–3432.

[38] Parimi, V.; and Williams, B. C. 2026. Risk-Bounded Multi-Agent Visual Navigation via Iterative Risk Allocation. arXiv:2509.08157.

[39] Park, S.; Jeon, S.; and Park, J. 2024. 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.

[40] Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; and Courville, A. 2017. FiLM: Visual Reasoning with a General Conditioning Layer. arXiv:1709.07871.

[41] Puig, X.; Undersander, E.; Szot, A.; Cote, M. D.; Partsey, R.; Yang, J.; Desai, R.; Clegg, A. W.; Hlavac, M.; Min, T.; Gervet, T.; Vondrus, V.; Berges, V.-P.; Turner, J.; Maksymets, O.; Kira, Z.; Kalakrishnan, M.; Malik, J.; Chaplot, D. S.; Jain, U.; Batra, D.; Rai, A.; and Mottaghi, R. 2023. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots.

[42] Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F. A.; Bengio, Y.; and Courville, A. 2019. On the Spectral Bias of Neural Networks. arXiv:1806.08734.

[43] Ramachandran, P.; Zoph, B.; and Le, Q. V. 2017. Searching for Activation Functions. arXiv:1710.05941.

[44] Reeves, M.; and Williams, B. C. 2024. LaPlaSS: Latent Space Planning for Stochastic Systems. arXiv:2404.07063.

[45] Rockafellar, R. T.; and Uryasev, S. 2000. Optimization of conditional value-at risk. *Journal of Risk*, 3: 21–41.

[46] Santara, A.; Naik, A.; Ravindran, B.; Das, D.; Mudigere, D.; Avancha, S.; and Kaul, B. 2017. RAIL: Risk-Averse Imitation Learning. arXiv:1707.06658.

[47] Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; Parikh, D.; and Batra, D. 2019. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[48] Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; and Levine, S. 2018. Time-Contrastive Networks: Self-Supervised Learning from Video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1134–1141.

[49] Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; Clarkson, A.; Yan, M.; Budge, B.; Yan, Y.; Pan, X.; Yon, J.; Zou, Y.; Leon, K.; Carter, N.; Briales, J.; Gillingham, T.; Mueggler, E.; Pesqueira, L.; Savva, M.; Batra, D.; Strasdat, H. M.; Nardi, R. D.; Goesele, M.; Lovegrove, S.; and Newcombe, R. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv preprint arXiv:1906.05797*.

[50] Szot, A.; Clegg, A.; Undersander, E.; Wijmans, E.; Zhao, Y.; Turner, J.; Maestre, N.; Mukadam, M.; Chaplot, D.; Maksymets, O.; Gokaslan, A.;

Vondrus, V.; Dharur, S.; Meier, F.; Galuba, W.; Chang, A.; Kira, Z.; Koltun, V.; Malik, J.; Savva, M.; and Batra, D. 2021. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[51] Tancik, M.; Srinivasan, P. P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J. T.; and Ng, R. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. arXiv:2006.10739.

[52] Tsai, Y.-H. H.; Li, T.; Ma, M. Q.; Zhao, H.; Zhang, K.; Morency, L.-P.; and Salakhutdinov, R. 2022. Conditional Contrastive Learning with Kernel. arXiv:2202.05458.

[53] Vahs, M.; Pek, C.; and Tumova, J. 2023. Belief Control Barrier Functions for Risk-aware Control. arXiv:2309.06499.

[54] van den Oord, A.; Li, Y.; and Vinyals, O. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748.

[55] Veit, A.; Belongie, S.; and Karaletsos, T. 2017. Conditional Similarity Networks. arXiv:1603.07810.

[56] Watter, M.; Springenberg, J. T.; Boedecker, J.; and Riedmiller, M. 2015. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. arXiv:1506.07365.

[57] Yang, R.; Sun, X.; and Narasimhan, K. 2019. A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation. In *Advances in Neural Information Processing Systems*.

[58] Yoo, G.; Park, J.; and Woo, H. 2024. 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.

[59] Zhang, J.; Bai, C.; Pan, W.; Liu, T.; and Guo, J. 2025. Local Path Optimization in The Latent Space Using Learned Distance Gradient. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 15940–15946. IEEE.

[60] Zhang, Q.; Zhang, L.; Xu, H.; Shen, L.; Wang, B.; Chang, Y.; Wang, X.; Yuan, B.; and Tao, D. 2023. SaFormer: A Conditional Sequence Modeling Approach to Offline Safe Reinforcement Learning. arXiv:2301.12203.

[61] Zheng, C.; Eysenbach, B.; Walke, H.; Yin, P.; Fang, K.; Salakhutdinov, R.; and Levine, S. 2025. Stabilizing Contrastive RL: Techniques for Robotic Goal Reaching from Offline Data. arXiv:2306.03346.

[62] Zheng, C.; Salakhutdinov, R.; and Eysenbach, B. 2025. Contrastive Difference Predictive Coding. arXiv:2310.20141.

[63] Zhu, B.; Dang, M.; and Grover, A. 2023. Scaling Pareto-Efficient Decision Making via Offline Multi-Objective RL. ICLR 2023 Poster.

A Conditioned Contrastive Theory

A.1 Density Encoding in Learned Logits

The Gauss–Markov inference-driven planning method outlined in Section 3.2 requires the learned scores 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 proving (4) holds for the non-conditioned InfoNCE objective as done in [12]. We define the logits function $g(s, s_+)$ such that $g(s^i, s_+^j) = S_{ij}$. Also let $\mathcal{B}_B = \{(s^i, s_+^i)\}_{i=1}^B$ denote a batch of B i.i.d. samples from $p(s, s_+)$, written compactly as $\mathcal{B}_B \sim p^B$. Start by considering the optimization problem which yields the optimal logits function,

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

Now consider the first term in the objective,

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

For compactness, define the batch denominator and its variation as

$$D_i^g(s) = \sum_{\substack{j=1\\ j\neq i}}^{B} g(s, s_+^j), \quad \delta D_i^g(s) = \sum_{\substack{j=1\\ j\neq i}}^{B} \delta g(s, s_+^j). \tag{14}$$

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

$$\begin{aligned}\delta J_1(g) &= \lim_{B\to\infty} \mathbb{E}_{\mathcal{B}_B\sim p^B}\left[\frac{1}{B}\sum_{i=1}^{B}\left(\frac{\delta g(s^i,s_+^i)}{g(s^i,s_+^i)} - \frac{\delta D_i^g(s^i)}{D_i^g(s^i)}\right)\right] \\ &= \lim_{B\to\infty}\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{s^i\sim p(s)}\left[\int \delta g(s^i, s_+)\, q_g(s^i, s_+)\, ds_+\right] \\ &= \iint \delta g(s, s_+)\, q_g(s, s_+)\, p(s)\, ds\, ds_+, \end{aligned} \tag{15}$$

where

$$q_g(s, s_+) = \frac{p(s_+ \mid s)}{g(s, s_+)} - k(s)p(s_+). \tag{16}$$

The function k is the large-batch normalization term

$$\begin{aligned} k(s) &= \lim_{B\to\infty} \mathbb{E}\left[\frac{B-1}{D_i^g(s)}\right] \\ &= \left[\int g(s, s_+)p(s_+)\, ds_+\right]^{-1}. \end{aligned} \tag{17}$$

which is independent of the 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 [21], the bracketed term must therefore vanish for all anchor and future states. Absorbing $k(s)$ as a free normalization of logits [12], we obtain the following optimal solution.

$$g^*(s, s_+) \propto \frac{p(s_+ \mid s)}{p(s_+)}. \tag{18}$$

Repeating this procedure for the second term in the original optimization problem $J_2(g)$ and noting that $p(s_+ \mid s)p(s) = p(s \mid s_+)p(s_+)$ 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) [12, 31, 34].

Extending this to the conditioned contrastive learning, define the conditional logits function as $\hat{g}(s, s_+; \Delta)$. With $\widetilde{\mathcal{B}}_B = \{(s^i, s^i_+, \Delta_i)\}_{i=1}^B \sim p(s, s_+, \Delta)^B$, it leads to the following problem according to the form in (12).

$$\max_{\hat{g}} \lim_{B\to\infty} \mathbb{E}_{\widetilde{\mathcal{B}}_B}\left[\frac{1}{B}\sum_{i=1}^{B}\left\{\log \frac{\hat{g}(s^i, s^i_+; \Delta_i)}{\sum_{\substack{j=1\\ j\neq i}}^{B} \hat{g}(s^i, s^j_+; \Delta_i)} + \log \frac{\hat{g}(s^i, s^i_+; \Delta_i)}{\sum_{\substack{j=1\\ j\neq i}}^{B} \hat{g}(s^j, s^i_+; \Delta_j)}\right\}\right]. \tag{19}$$

If we interpret anchor sampling as selecting an index at random from the concatenated tuple of all observed states in the dataset, we see that 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 state since s_+ is unchanged, it is only for notational convenience. This yields the first component, built in the same manner as (13),

$$J_1(\hat{\hat{g}}) = \lim_{B\to\infty} \mathbb{E}_{\widehat{\mathcal{B}}_B \sim p(\hat{s}, s_+)^B}\left[\frac{1}{B}\sum_{i=1}^{B} \log \frac{\hat{\hat{g}}(\hat{s}^i, s^i_+)}{\sum_{\substack{j=1\\ j\neq i}}^{B} \hat{\hat{g}}(\hat{s}^i, s^j_+)}\right]. \tag{20}$$

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. Applying the same variational calculation as in (15), define

$$q_{\hat{\hat{g}}}(\hat{s}, s_+) = \frac{p(s_+ \mid \hat{s})}{\hat{\hat{g}}(\hat{s}, s_+)} - \hat{\hat{k}}(\hat{s})p(s_+), \tag{21}$$

where

$$\hat{\hat{k}}(\hat{s}) = \left[\int \hat{\hat{g}}(\hat{s}, s_+)p(s_+)\, ds_+\right]^{-1}. \tag{22}$$

The conditioned variation is then

$$\delta J_1(\hat{\hat{g}}) = \iint \delta\hat{\hat{g}}(\hat{s}, s_+)\, q_{\hat{\hat{g}}}(\hat{s}, s_+)\, p(\hat{s})\, d\hat{s}\, ds_+. \tag{23}$$

The first-order condition again requires the variational integrand to vanish pointwise. Hence,

$$(\hat{\hat{g}})^*(\hat{s}, s_+) \propto \frac{p(s_+ \mid \hat{s})}{p(s_+)}. \tag{24}$$

Reverting back to the original conditioned logit definiton, we have

$$(\hat{g})^*(s, s_+; \Delta) \propto \frac{p(s_+ \mid s, \Delta)}{p(s_+)}. \tag{25}$$

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

$$(\hat{g})^*(s, s_+; \Delta) \propto \frac{p(s, \Delta \mid s_+)}{p(s, \Delta)}, \tag{26}$$

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

A.2 Gaussianity of Representations

A second key assumption underlying CRLP is that the marginal distribution over representations is isotropic Gaussian [12]. In our formulation, we require this assumption to hold for all conditioned latent spaces. In other words, whilst Eysenbach et al. [12] 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 variance of the distribution could fluctuate over the preference continuum. We recognize that under conditioning, the density-ratio encoding and isotropic Gaussian assumptions face larger logit function approximation error as well as sampling error from the sparser joint state-preference space. However, we find empirically that planning performance remains consistent with [12], generating representations that keep inference-driven planning tractable, as evidenced in Section 5.

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

$$p(z_{1:n} \mid z_0, z_g) \propto p(z_g \mid z_n) \prod_{k=1}^{n} p(z_k \mid z_{k-1}), \tag{27}$$

in combination with (5) [12]. This yields the Gaussian distribution $z_{1:n} \sim \mathcal{N}(z_{1:n}; \mu = \Lambda^{-1}\eta, \Sigma = \Lambda^{-1})$, where

$$\Lambda = \text{tridiag}\left(-A,\ \frac{c}{c+1}A^\top A + \frac{c+1}{c}I,\ -A^\top\right), \tag{28}$$

$$\eta = \begin{bmatrix} z_0^\top A^\top & \mathbf{0} & \dots & \mathbf{0} & z_g^\top A \end{bmatrix}^\top. \tag{29}$$

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

B Further Studies

B.1 Representation Space Deformation

This section presents some further analysis of how conditioning affects the representation space structure. 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 extremities 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 a pure translation and rotation, so could be learned as a linear transformation. We observe that as the preferences differ, the residual rises and this rigid transformation is insufficient to describe 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.2 State Augmentation Comparison

A candidate conditioning mechanism for preference modulation of planning behavior is state augmentation, which is used as a comparison baseline. 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 (18), 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 preference context. Note that s_+ represents any possible future following marginalization in (15), 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 as a way of distinguishing positive and negative pairs, offering a degenerate solution to the optimization problem if the contrastive signal from remaining states is weak.

The learned Guass-Markov distribution therefore admits intermediate waypoints which share the same preference context as the augmented start and goal. The

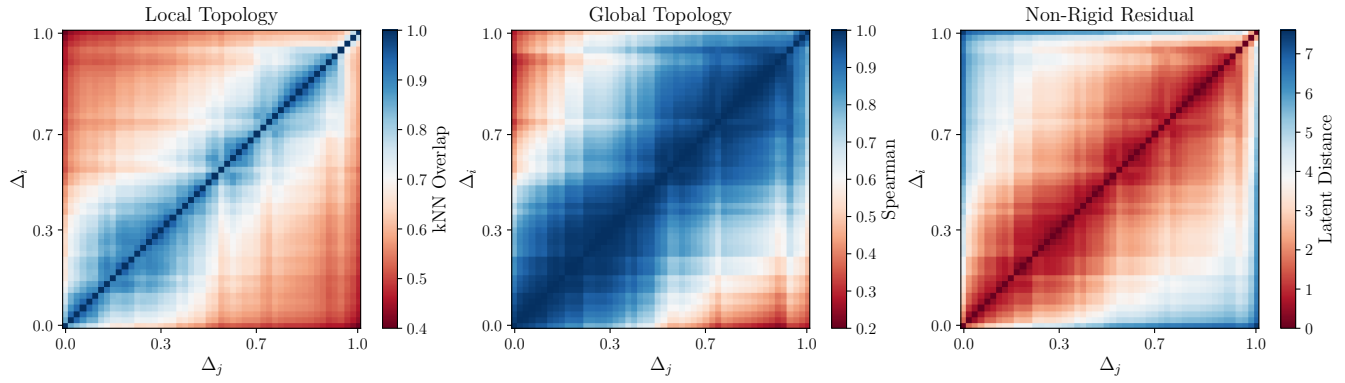

Figure 5: Matrix summaries of representation space manifold deformation for Point Four Obstacle environment under preference-conditioning. Plots characterize local changes via kNN overlap (left), global changes via Spearman coefficients (middle), and non-rigid transformation effects using residual latent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

key distinction here is that a single unified representation space has been learned as opposed to a conditioned deformation.

It is critical that with this formulation, state augmentation is able to encode precise temporal relations in the latent space. By inspection of Figure 4, we observe large waypoint MSE using state augmentation. This is obtained alongside weaker or unstable trends between user preference and realized plan costs, indicated by lower Spearman correlations as well as considerably higher roughness in Table 1. 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 1, 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 [12].

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 under each anchor's preference context, however this is not found to be 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 is further reliant on a 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, under the preference-dependent deformation of the latent space, nearest-neighbor retrieval could decode a safe latent waypoint back to an unsafe state. This limitation motivates an exploration of alternative decoding methods.

We also inherit assumptions from Eysenbach et al. [12], 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. [12], it remains open how function approximation and sparse sampling errors introduced by conditioning affect MoMo's planning performance.

Finally, data-driven planning is exposed to the quality of existing datasets, which are difficult to acquire with 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 provide coverage of 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 [15] 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

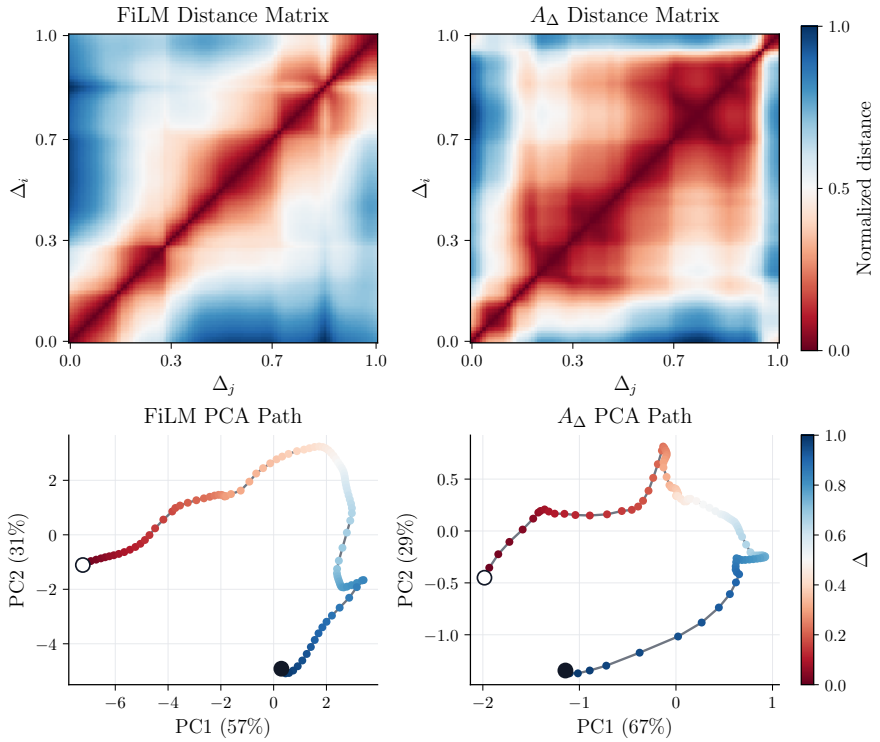

Figure 6: Summary of FiLM and transition matrix conditioning mechanism parameters for Ant Habitat environment. Across the preference spectrum, panels show the $\|.\|_2$ distance between FiLM generator outputs (top left), Frobenius norm $\|.\|_F$ of the conditioned transition matrix (top right), path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

sequences. Continuous trajectories between consecutive waypoints are then produced by a short-horizon goal-reaching policy. 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

$$R(s,a,g;w_\Delta) = -\Big[1 + \max\big(0,\, d_b(s',g) - d_b(s,g)\big) + w_\Delta \left(1 - \frac{d_o(s)}{\theta}\right) \mathbf{1}\{d_o(s) < \theta\}\Big]. \tag{30}$$

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 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, open map with a dense Gaussian cost contour, and extract a map from Replica Dataset FRL Apartment 5 loaded from AI Habitat to model realistic environments. For annotation of the continuous trajectories, we use a risk cost function which increases linearly with obstacle proximity inside 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 [29], 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, requiring the robot to retrieve a cube from a start bowl and move it to the goal bowl. Between the bowls, there is a divider obstacle with a slot such that safer trajectories correspond to high 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 the diverse set of preference behaviors, we apply varying obstacle inflation levels with Bi-directional Rapidly-exploring Random Trees [26] 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.

Table 1: Preference modulation profile metrics comparing MoMo to state augmentation baselines. ΔC measures endpoint contrast as the difference in total plan cost between $\Delta = 1$ and $\Delta = 0$ queries. ρ is Spearman correlation between Δ and mean cost curve. R_{norm} is cost-curve roughness defined as the sum of second-order finite differences in cost normalized by $|\Delta C|$.

Environment	$\Delta C \uparrow$				$\rho \uparrow$				$R_{\text{norm}} \downarrow$			
	Ours	IvI*	PCA*	VIP*	**Ours**	IvI*	PCA*	VIP*	**Ours**	IvI*	PCA*	VIP*
Point Four Obstacles	**9.37**	4.94	-0.01	-0.34	**0.99**	0.77	-0.95	-0.32	**1.32**	7.75	2.70	13.35
Point Contour	**2.36**	1.34	0.00	-0.28	**0.99**	0.85	-0.81	-0.57	**0.78**	10.14	2.78	5.56
Point Habitat	7.29	**15.71**	0.01	-3.00	0.91	**0.96**	0.67	-0.98	**1.42**	2.23	10.64	0.88
Drone	3.42	**7.21**	0.03	3.07	**0.87**	0.60	0.28	0.80	3.08	5.72	15.04	**1.27**
Ant Habitat	**3.82**	3.49	-0.01	2.19	0.82	0.63	-0.65	**0.87**	**1.64**	20.16	22.08	4.76
UR5	0.92	1.36	**1.90**	0.00	**1.00**	0.57	0.97	0.72	**0.29**	4.63	0.67	225.90

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.

Table 2: Summary of training datasets with sufficient state-preference space coverage.

Environment	**Collection framework**	**Risk levels**	**Observation dimension**	**Trajectories**	**Transitions**
Point Four Obstacle	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

E Model Training

We build off of the unconditioned model presented in [12], with key parameters summarized in Table 3. 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 which feeds these three modules. This allows for an efficient shared condition embedding to inform the modulation mechanisms.

Direct conditioning on the raw scalar Δ value may bottleneck performance due to the spectral bias of neural networks towards low frequency functions. This means that networks struggle to produce local fluctuations without affecting global behavior [42], 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 [51]. Exact implementation details are contained in our codebase, to be released following reviews.

Whilst our derived results and those in [12] exclude the positive pair from the denominator of the InfoNCE objective, standard empirical implementations sum over all indices to yield a softmax distribution and bound loss. In line with prior work we also adopt this convention [12, 2].

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

Table 3: 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