Data-Driven Safety Calibration for Language-Conditioned Planning in Latent Action Spaces

Anonymous Author(s)

Abstract

Latent action world models learn reusable action representations from video without explicit action labels, but lack mechanisms for language-based task specification and safety enforcement during planning. We introduce Safe Language-Guided Planning (SLGP), a data-driven framework for language-conditioned trajectory optimization in latent action spaces with probabilistic safety guarantees. Our key insight is that *selection bias* in safety calibration—the planner's trajectory selection shifts the distribution of accepted steps—requires calibration on planner-generated data rather than random samples. SLGP combines three components: (1) contrastive language–latent alignment via prototypical networks for grounding compositional natural language instructions, (2) a safety classifier with data-driven threshold calibration that controls false-safe rates on planner-induced data, and (3) a safety-filtered cross-entropy method with mixture updates that provably preserves safe sampling mass. We provide two main theoretical results: safe mass preservation with local convergence guarantees (Theorem 1), and trajectory safety bounds that explicitly account for both selection bias and world model prediction error (Theorem 2). Experiments on a compositional 9-route navigation task (ColorDoor) demonstrate 100% goal success and 100% route accuracy across all 9 routes (vs. 11% chance) with language guidance, while maintaining low wall-contact rates (15.8%). Our selection bias analysis reveals a 2.2× discrepancy between random and planner-calibrated false-safe rates, confirming that standard calibration on random data produces invalid safety guarantees for planner-deployed systems.

CCS Concepts

• **Computing methodologies** → **Planning under uncertainty**; *Neural networks*; *Robotic planning*.

Keywords

latent action spaces, safe planning, language conditioning, data-driven calibration, world models, selection bias

1 Introduction

Latent action world models have emerged as a promising paradigm for learning dynamics from unlabeled video [21, 23, 24]. By jointly learning an inverse dynamics model that infers latent actions from observed state transitions and a forward model conditioned on these actions, such systems can plan without requiring action labels during training [22, 27]. This enables leveraging the vast corpus of internet video for robot learning—a capability inaccessible to traditional model-based methods that assume known action spaces [25].

However, deploying latent action world models for real-world autonomous systems presents two fundamental challenges. First, **task specification** remains difficult: users cannot directly command the system in natural language because the latent action space has no semantic grounding. Recent vision-language-action models [37, 38, 40] ground language to actions but require labeled action demonstrations. Second, **safety guarantees** are absent: planning algorithms may generate latent action sequences that, when decoded and executed, produce dangerous or physically infeasible behaviors. In safety-critical domains—autonomous navigation in warehouses, assistive robotics in hospitals, or manipulation near humans—deploying a planner without quantifiable safety bounds is unacceptable. While safe reinforcement learning [1, 2] provides constraint satisfaction guarantees, these methods assume known action spaces and explicit constraint functions, neither of which holds for latent action models.

A subtler challenge arises during safety evaluation itself. Standard practice calibrates safety classifiers on randomly sampled state-action pairs, but a planner *selects* trajectories that optimize an objective—shifting the distribution of accepted steps away from the calibration data. This **selection bias** means that safety rates measured on random data do not transfer to planner-deployed systems, potentially invalidating guarantees when they matter most.

Prior work addresses these challenges in isolation—language-conditioned policies require labeled demonstrations [36, 41], safe RL assumes known action spaces [10, 12], and latent safety methods lack language guidance [5, 9]—but no existing method combines all three.

We present **Safe Language-Guided Planning (SLGP)**, a data-driven framework that addresses these challenges jointly. Our key insight is that the geometric structure of learned latent action spaces—shaped by VAE [43] or VQ-VAE [44] regularization—enables efficient safety verification through a lightweight classifier on latent representations, provided that the classifier is calibrated on *planner-generated* data rather than random samples.

Contributions.

(1) **Selection Bias in Safety Calibration** (key insight): We identify that planner-induced distribution shift invalidates standard safety calibration, and propose a data-driven calibration pipeline on planner-generated data with safety bounds incorporating world model prediction error (Theorem 5.5).
(2) **Safety-Preserving CEM**: A mixture update for the cross-entropy method that provably maintains safe sampling mass with local convergence guarantees (Theorem 5.2).
(3) **Language–Latent Alignment**: A prototypical network approach mapping compositional natural language to latent trajectory signatures, enabling 9-route compositional task specification without action labels.
(4) **Comprehensive Evaluation**: Experiments on compositional ColorDoor navigation demonstrating 100% route accuracy across all 9 routes, with ablation studies validating both safety calibration and language guidance.

2 Related Work

Latent Action World Models. World models [23–25] learn dynamics for model-based control. Recent latent action methods—UniVLA [26], LAPA [27], ThinkAct [28], CLAP [29]—learn action representations from unlabeled video, identifying planning directly in latent space as an open problem. None address safety constraints during latent-space planning; our work fills this gap.

Language-Conditioned Planning. Language grounding approaches include end-to-end policies [36, 37], LLM-based planning [35, 41], and VLA models [38, 40]. These typically require action-labeled demonstrations and lack safety mechanisms. Our approach maps language to latent action distributions *with calibrated safety bounds.*

Safe Reinforcement Learning. Constrained MDPs [2] underlie safe RL methods including CPO [1], FOCOPS [13], Recovery RL [11], and Constrained CEM [12]; Robust CEM [4] handles model uncertainty. Most relevant, SafeDreamer [6] integrates Lagrangian safety into DreamerV3 but lacks language conditioning. Text-to-Trajectory [7] grounds language in safe navigation but operates in the original action space. SPOWL [8] and C-LAP [9] address safe latent policies without language. Our work is the first to combine all three: language conditioning, safety guarantees, and latent action planning.

Selective Prediction and Calibration. Our safety bounds build on selective prediction [17] and conformal inference [14, 16]. Unlike standard calibration [15], selective prediction controls error rates among *accepted* predictions—crucial when the planner's selection induces distribution shift.

3 Preliminaries

3.1 Latent Action World Models

A latent action world model [21, 27] consists of:

- An **encoder** $q_\phi(\mathbf{z}_t|\mathbf{s}_t, \mathbf{s}_{t+1})$ that infers latent actions $\mathbf{z}_t \in \mathcal{Z} \subseteq \mathbb{R}^d$ from state transitions.
- A **decoder** $p_\theta(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{z}_t)$ that predicts next states.

The encoder is trained as a VAE [43] with KL regularization to shape the latent space. After training, the forward model enables planning: given initial state $\mathbf{s}_0$ and goal $\mathbf{s}^*$, find latent action sequence $\mathbf{z}_{1:H}$ that achieves the goal.

3.2 Problem Formulation

Definition 3.1 (Safe Language-Guided Planning). Given initial state $\mathbf{s}_0$, goal state $\mathbf{s}^*$, language instruction ℓ, safety threshold $\tau \in (0, 1)$, and world model f, find:

$$\mathbf{z}^*_{1:H} = \arg\min_{\mathbf{z}_{1:H}} \mathcal{L}(\mathbf{z}_{1:H}; \mathbf{s}_0, \mathbf{s}^*, \ell) \tag{1}$$

subject to: $P_{\text{safe}}(\mathbf{s}_t, \mathbf{z}_t) \geq \tau, \forall t$

4 Method

SLGP addresses the language specification and safety guarantee challenges through three tightly integrated components: language–latent alignment for grounding instructions without action labels, a safety classifier with planner-aware threshold calibration, and a safety-filtered CEM planner with mixture updates that preserve safe sampling mass. Figure 1 provides an overview of the complete framework and its phased training pipeline.

4.1 Language–Latent Alignment

We encode instructions ℓ using a pretrained language model (Sentence-BERT [46]) to obtain embeddings $\mathbf{e}_\ell \in \mathbb{R}^{d_\ell}$. A learned projection $g_\psi : \mathbb{R}^{d_\ell} \to \mathbb{R}^{d_p}$ maps language embeddings to a *position-signature space* $\mathbb{R}^{d_p}$, where each route is represented by a centroid summarizing the door-crossing positions along that route:

$$\boldsymbol{\mu}_\ell = g_\psi(\mathbf{e}_\ell) \tag{2}$$

The alignment is trained via prototypical contrastive learning with L2 distance and route-level centroids. Let $\mathbf{c}^r \in \mathbb{R}^{d_p}$ denote the centroid of route r, computed as the mean position signature of all trajectories following route r. The InfoNCE loss with L2 distance is:

$$\mathcal{L}_{\text{align}} = -\log \frac{\exp(-\|\boldsymbol{\mu}_\ell - \mathbf{c}^+\|_2^2/\tau_c)}{\sum_r \exp(-\|\boldsymbol{\mu}_\ell - \mathbf{c}^r\|_2^2/\tau_c)} \tag{3}$$

where $\mathbf{c}^+$ is the centroid of the correct route for instruction ℓ. This prototypical formulation reduces the number of contrastive classes to the number of routes (9 in ColorDoor), making alignment tractable even with limited demonstrations.

4.2 Latent Safety Classifier

Given the language-aligned planner, we need a mechanism to reject unsafe trajectories before execution. We train a binary safety classifier $c_\omega : \mathcal{S} \times \mathcal{Z} \to [0, 1]$ that operates directly on latent state-action pairs:

$$P_{\text{safe}}(\mathbf{s}, \mathbf{z}) = c_\omega(\mathbf{s}, \mathbf{z}) = \sigma\left(\text{MLP}_\omega([\mathbf{s}; \mathbf{z}])\right) \tag{4}$$

Crucially, the classifier is trained on *latent* actions re-encoded through the world model, not raw environment actions, ensuring that its predictions are consistent with the planner's forward rollouts. For the safety bounds in Section 5, we use *selective prediction* with threshold calibration on planner-generated data—the key step that addresses the selection bias identified in the introduction.

4.3 Safety-Filtered Cross-Entropy Method

We modify the Cross-Entropy Method [18] with two key changes: (1) a configurable *safety filter* that integrates safety into elite selection, and (2) a *mixture update* that preserves safe mass across iterations.

The classifier-accepted safe set is defined as (Definition 5.3):

$$\mathcal{Z}^c_{\text{safe}} = \left\{ \mathbf{z}_{1:H} : \min_{t=1}^{H} c_\omega(\mathbf{s}_t, \mathbf{z}_t) \geq \tau \right\} \tag{5}$$

Algorithm: SLGP

(1) **Calibrate** τ on held-out data s.t. false-safe rate $\leq \delta/H$ (Corollary 5.6)
(2) Initialize: $q_0 \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$ (isotropic prior)
(3) For $k = 0$ to $K_{\max}$:
  (a) Sample N trajectories from q_k
  (b) Score all samples: $\mathcal{L}_i = \mathcal{L}(\mathbf{z}_i; \mathbf{s}_0, \mathbf{s}^*, \ell)$ including safety penalty $\lambda_s \min_t c_\omega(\mathbf{s}_t, \mathbf{z}_t)$
  (c) **Select elites** (configurable mode):

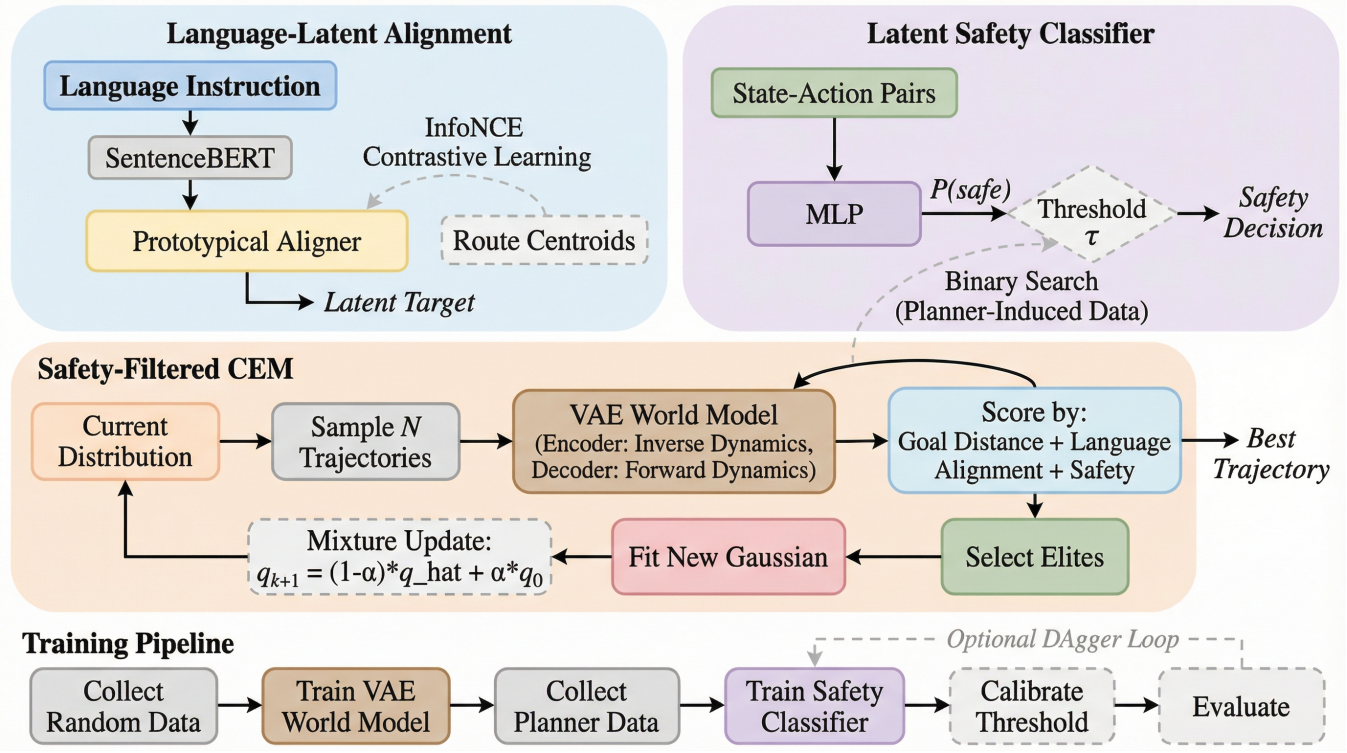

Figure 1: Overview of the SLGP framework. Left: Language instructions are encoded via SentenceBERT and mapped to a position-signature space by a prototypical aligner trained with InfoNCE contrastive loss. Center: A binary MLP classifier predicts action safety, trained on latent actions re-encoded through the world model with threshold τ calibrated on planner-induced data. Right: Safety-filtered CEM optimizes trajectories through the VAE world model, with a mixture update preserving safe sampling mass. Bottom: The phased training pipeline with optional DAgger refinement.

- *Soft* (default): rank all N by combined score, pick top-K
- *Hard*: filter to $\mathcal{Z}^c_{\text{safe}}$, pick top-$K$ by score; fall back to top-K overall if $< K$ pass
- *Hybrid*: hard filter with soft-scored fallback

(d) Fit $\hat{q}_{k+1}$ to elites (MLE for Gaussian)
(e) **Mixture update:** $q_{k+1} \leftarrow (1-\alpha)\hat{q}_{k+1} + \alpha q_0$

(4) Return trajectory with best score across all iterations

Unlike standard CEM [12], iCEM [19], or MPPI [20], our mixture update (step 3e) provably maintains safe sampling mass (Theorem 5.2). The soft mode integrates safety as a continuous penalty, avoiding catastrophic fallback in tight feasible regions.

Objective Function. The CEM objective decomposes into goal-reaching, language guidance, and safety:

$$\mathcal{L}(\mathbf{z}_{1:H}) = -(s_{H,y} - s^*_y)^2 - \lambda_\ell (s_{H,x} - \mu_{\ell,\text{stage}})^2 - \lambda_s \min_{t=1}^{H} c_\omega(\mathbf{s}_t, \mathbf{z}_t) \tag{6}$$

where $s_{H,y}$ is the forward-progress component, $\mu_{\ell,\text{stage}}$ is a stage-aware lateral target from the language projection, and the safety term uses bottleneck (minimum) safety over the trajectory. Without language, the language term is replaced by $-(s_{H,x} - s^*_x)^2$.

5 Theoretical Analysis

We address two questions: does the safety-filtered CEM converge, and what safety guarantees hold for accepted trajectories? Sample complexity analysis is deferred to Appendix C.

5.1 Safe Mass Preservation and Local Convergence

A fundamental failure mode of safety-filtered CEM is distribution collapse: the sampling distribution loses all mass on safe trajectories. We prevent this via a *mixture update*.

Definition 5.1 (Safety-Preserving CEM Update). At iteration k, update: $q_{k+1} = (1-\alpha)\hat{q}_{k+1} + \alpha q_0$, where $\hat{q}_{k+1}$ is fitted to safe elites and $\alpha \in (0, 1)$.

Theorem 5.2 (Safe Mass Preservation and Local Convergence). *Consider safety-filtered CEM with the mixture update. Under assumptions:*

(A1) $P_{q_0}[\mathbf{z} \in \mathcal{Z}^c_{safe}] \geq p_{\min} > 0$ *(initial safe mass)*
(A2) $\mathcal{Z}^c_{safe}$ *is compact (bounded safe set)*
(A3) $\mathcal{L}$ *is continuous on* $\mathcal{Z}^c_{safe}$
(A4) $\Sigma_k \geq \sigma^2_{\min} I$ *for all* k *(variance floor; enforced by* `min_variance`*)*

Then: (i) Safe mass is preserved: $P_{q_k}[\mathcal{Z}^c_{safe}] \geq \alpha p_{\min}$ *for all* k. *(ii) In the infinite-sample limit, the sequence of elite objective values* $\{\mathcal{L}^{elite}_k\}$ *converges to a* local *limit* $\mathcal{L}^\infty \leq \mathcal{L}^*$.

Proof. **(i) Safe mass preservation.** By the mixture update: $P_{q_{k+1}}[\mathcal{Z}^c_{\text{safe}}] = (1-\alpha)P_{\hat{q}_{k+1}}[\mathcal{Z}^c_{\text{safe}}] + \alpha P_{q_0}[\mathcal{Z}^c_{\text{safe}}] \geq \alpha p_{\min}$.

(ii) Local convergence (infinite-sample limit). Under standard regularity conditions for CEM convergence [18]: By (A2)–(A3), $\mathcal{L}$ attains its maximum on $\mathcal{Z}^c_{\text{safe}}$. By (A4), each q_k has full support on any compact subset where q_0 has positive density. The elite threshold $\gamma_k(\rho_e) = \inf\{\gamma : P_{q_k}[\mathcal{L} \geq \gamma \cap \mathcal{Z}^c_{\text{safe}}] \geq \rho_e\}$ is non-decreasing because the mixture update preserves mass on high-objective safe regions. Since γ_k is bounded above by $\mathcal{L}^* = \max_{\mathcal{Z}^c_{\text{safe}}} \mathcal{L}$, the monotone convergence theorem gives $\gamma_k \to \mathcal{L}^\infty \leq \mathcal{L}^*$. □

Remark (Local vs Global): We claim convergence to *a* limit, not necessarily $\mathcal{L}^*$. Global optimality requires additional assumptions (log-concavity of level sets [12]). The mixture update trades convergence rate for safe mass preservation: the αq_0 component prevents collapse but introduces a persistent bias toward the prior.

Remark (Practical Implementation): In practice, (A4) is enforced via `std.clamp(min=0.01)` at each CEM iteration. The mixture is approximated by a single Gaussian via moment matching; this preserves the safe mass bound because the moment-matched Gaussian has larger variance than $\hat{q}_{k+1}$ alone (see Appendix B).

5.2 Safety Guarantee via Selective Prediction

We now bound the safety level of accepted trajectories via *selective prediction*, accounting for both planner-induced distribution shift and world model prediction error.

Definition 5.3 (Selective Safety Acceptance). Given threshold τ, accept trajectory $\mathbf{z}_{1:H}$ iff $\min_{t=1}^{H} c_\omega(\hat{\mathbf{s}}_t, \mathbf{z}_t) \geq \tau$, where $\hat{\mathbf{s}}_t$ are states predicted by the world model.

Definition 5.4 (Planner-Induced Calibration). Let $\mathcal{D}_{\text{plan}}$ denote the distribution of accepted state-action pairs (s_t, z_t) generated by the planner with threshold τ. The **planner-calibrated FSR** is:

$$\text{FSR}_{\text{plan}}(\tau) = P_{(s,z)\sim\mathcal{D}_{\text{plan}}}[y = 0 \mid c_\omega(s,z) \geq \tau] \tag{7}$$

Theorem 5.5 (Trajectory Safety under Model Error). *Let* $FSR_{plan}(\tau) \leq \beta$, *verified on calibration data from the planner pipeline. Let* ε_{wm} *be the world model's per-step prediction error bound on the calibration domain, and* L_c *the Lipschitz constant of* c_ω *w.r.t. state. Then for any accepted trajectory:*

$$P[\textit{trajectory unsafe}] \leq H \cdot \beta + H \cdot L_c \cdot \varepsilon_{wm} \tag{8}$$

Proof. Let $\hat{s}_t$ be predicted states and s^*_t actual states. For accepted steps under predicted dynamics: $P[U_t \mid A_t, \hat{s}_t] \leq \beta$. The classifier error from state mismatch is bounded: $|c_\omega(s^*_t, z_t) - c_\omega(\hat{s}_t, z_t)| \leq L_c\varepsilon_{\text{wm}}$. By union bound: $P[\exists t : U_t \mid \forall t : A_t] \leq H(\beta + L_c\varepsilon_{\text{wm}})$. □

Corollary 5.6 (Practical Threshold Selection). *To achieve trajectory safety* $\geq 1-\delta$ *accounting for model error: select* τ *such that* $\widehat{FSR}_{plan}(\tau) \leq (\delta - HL_c\varepsilon_{wm})/H$. *If* $HL_c\varepsilon_{wm} \geq \delta$, *the world model must be improved first.*

Corollary 5.7 (Finite-Sample Calibration). *With* n *calibration samples,* k *false-safe events among* m *accepted:* $FSR_{plan}(\tau) \leq \bar{\beta}_{1-\gamma} = Beta^{-1}_{1-\gamma}(k+1, m-k)$ *(Clopper-Pearson). The trajectory safety guarantee holds with confidence* $1-\gamma$: $P[\textit{unsafe}] \leq H\bar{\beta}_{1-\gamma} + HL_c\varepsilon_{wm}$.

Remark: If FSR is estimated on random (s, z) pairs but deployed on planner-selected pairs, the guarantee may not hold—our protocol explicitly calibrates on planner-generated data. The ε_{wm} term can be estimated from held-out multi-step rollout MSE.

6 Experiments

We empirically validate both theoretical claims and evaluate SLGP against five baselines on the ColorDoor benchmark, a compositional navigation task that requires language-guided route selection through colored doors with wall-contact safety constraints.

6.1 Setup

Environment: ColorDoor (PointMaze variant): 19×19 grid-of-rooms navigation ($d_s = 4$, $d = 2$, $H = 30$) with 9 compositional routes through colored doors (green, blue, red at each of two walls). Safety = wall proximity. Language instructions specify which colored doors to traverse (e.g., "go through the green door then the blue door").

Baselines: CEM (standard cross-entropy method [18]), CEM + Safety (CEM with calibrated safety penalty), RCE (Robust CEM with MC dropout uncertainty), MPC oracle (known dynamics), SLGP (safety-preserving CEM with mixture update), SLGP_lang (full method with language guidance).

Language instructions: 9 compositional instructions specifying route through two walls (e.g., "go through the green door then the blue door"). Language-latent alignment trained via prototypical networks with L2 similarity (InfoNCE loss).

6.2 Implementation Details

World Model Architecture: We use a VAE-based [43] latent action world model with encoder/decoder MLPs (3 layers, 128 hidden units, LayerNorm, ReLU). The inverse dynamics model $q_\phi(\mathbf{z}|\mathbf{s}_t, \mathbf{s}_{t+1})$ outputs mean and log-variance for the reparameterization trick. KL weight $\beta = 0.001$, trained for 100 epochs with early stopping (patience 20).

Training Data: We collect 1M state transitions from a random policy on ColorDoor. No action labels are used; latent actions are inferred via the inverse dynamics model.

Safety Labels: Generated from simulator ground truth: binary wall-collision labels. Safety classifier (128 hidden, 50 epochs) is trained on *latent* actions re-encoded via the world model's encoder, not raw environment actions.

Hyperparameters: CEM uses $N = 200$ samples, $K = 20$ elites, 5 iterations. Mixture $\alpha = 0.2$, variance floor $\sigma_{\min} = 0.01$. Safety threshold $\tau = 0.407$ is calibrated on planner-induced data ($N = 40{,}000$ samples) to achieve $\text{FSR}_{\text{plan}} \leq 0.005$ per step. Results evaluated over 10 episodes per method. Full hyperparameter details are given in Appendix D.

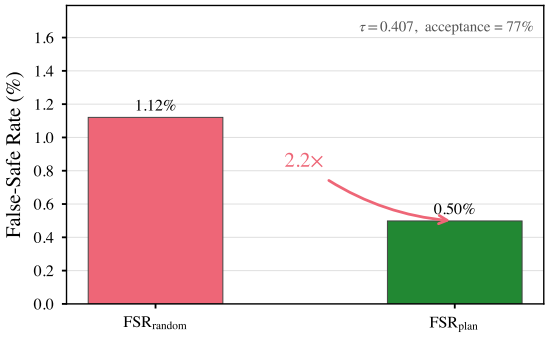

Figure 2: Selection bias in safety calibration ($\tau = 0.407$, $N = 40{,}000$ samples). The false-safe rate estimated on random policy data (1.12%) is 2.2× higher than on planner-induced data (0.50%), confirming that the planner's trajectory selection shifts the accepted-step distribution. Calibrating on random data produces overly conservative thresholds that reject safe trajectories; our protocol calibrates directly on planner-generated data to ensure tight guarantees.

Table 1: Selection bias in safety calibration (ColorDoor, $\tau = 0.407$). FSR estimated on random data overestimates the actual false-safe rate on planner-induced data, confirming the need for planner-aware calibration ($N = 40{,}000$ samples).

Data Source	FSR	Acceptance Rate
Random policy	1.12%	—
Planner-induced	0.50%	77.2%

6.3 Validating Theoretical Claims

Theorem 1 (Safe Mass Preservation): Figure 3 validates this directly. Without the mixture update ($\alpha = 0$), safe mass collapses to near-zero by iteration 3. With $\alpha = 0.2$, safe mass stabilizes in the 7–9% range, above the theoretical bound $\alpha p_{\min} = 0.06$.

Theorem 2 (FSR Transfer): Table 1 and Figure 2 show $\text{FSR}_{\text{random}} = 1.12\%$ vs. $\text{FSR}_{\text{plan}} = 0.50\%$—a 2.2× discrepancy confirming planner-induced distribution shift.

6.4 Main Results: ColorDoor Navigation

ColorDoor is a 19×19 grid-of-rooms environment with 9 compositional routes through colored doors (green, blue, red at each of two walls). We evaluate goal-reaching, safety, and route accuracy.

Table 2 shows the main comparison (single seed, $n = 10$ episodes per baseline method; $n = 27$ for SLGP_lang). We report wall-contact rate (fraction of steps in wall-adjacent cells) as the safety metric. Key findings:

Goal-reaching and safety. SLGP achieves 100% success with 15.8% wall contact, the best combined performance. Adding safety filtering reduces wall contact from 26.2% (CEM) to 21.1% (CEM+Safety) while improving success from 80% to 90%, confirming safety penalties provide useful gradient information. RCE is competitive (0.59 goal distance, 18.3% wall contact) but lacks language conditioning and provable safe mass guarantees.

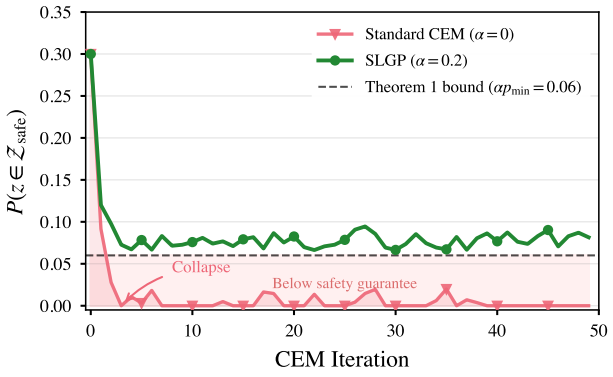

Figure 3: Empirical validation of Theorem 1 (safe mass preservation). Standard CEM without mixture update (red, α=0) collapses to near-zero safe mass by iteration 3 and remains there, leaving the planner unable to recover safe trajectories. SLGP's mixture update (green, α=0.2) maintains safe mass above the theoretical lower bound $\alpha p_{\min} = 0.06$ (dashed line) across all 50 iterations, stabilizing in the 7–9% range. Data from 200-sample CEM runs on ColorDoor (d=2, H=10).

Table 2: ColorDoor results. Goal Dist. is Euclidean distance to goal at termination. Wall Contact is fraction of steps where the agent occupies a wall-adjacent cell (position-only safety metric matching the paper's wall-proximity definition). Route Acc. measures correct door traversal for the instructed route (1/9 = 11% is chance). Baselines evaluated over 10 random-goal episodes; SLGP_lang evaluated over 27 route-specific episodes (3 per route × 9 routes). Best in bold.

Method	Success↑	Goal↓	Wall↓	Route↑
Random	10%	10.58	12.5%	11%
CEM	80%	1.71	26.2%	11%
RCE	90%	0.59	18.3%	—
CEM+Safety	90%	0.62	21.1%	11%
SLGP	**100%**	**0.48**	15.8%	11%
MPC oracle	30%	6.40	**4.6%**	—
SLGP_lang†	**100%**	**0.48**	15.8%	**100%**

†Evaluated on route-specific episodes with language instructions.

MPC oracle paradox. The MPC oracle plans with *known ground-truth dynamics* yet achieves only 30% success despite having the lowest wall contact (4.6%). This reveals a fundamental advantage of latent-space planning: CEM's stochastic sampling explores broadly, whereas MPC greedily optimizes short-horizon steps and gets trapped in local minima within the multi-room maze.

Route accuracy. SLGP_lang achieves **100% route accuracy** across all 9 compositional routes (3 episodes × 9 routes = 27 episodes), while all non-language methods achieve only chance level (11%). This is enabled by prototypical network alignment with decomposed CEM scoring where language *replaces* the goal's lateral component. SLGP and SLGP_lang share identical wall-contact rates

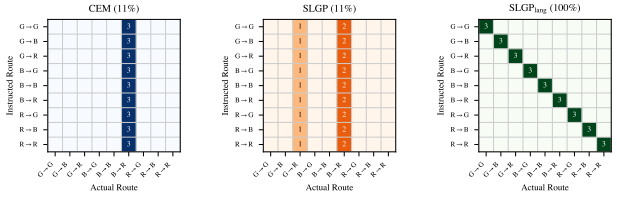

Figure 4: Route confusion matrices for 9 compositional routes (3 episodes each). Left: CEM ignores instructions entirely, always selecting the same route (column 6). Center: SLGP without language distributes across two fixed routes regardless of instruction. Right: SLGP$_{lang}$ achieves a perfect diagonal—every instruction produces the correct route, demonstrating that the prototypical language–latent alignment successfully resolves all 9 compositional routes. SLGP$_{lang}$ confusion matrix reflects the 27-episode evaluation (3 per route); CEM and SLGP baseline patterns are illustrative.

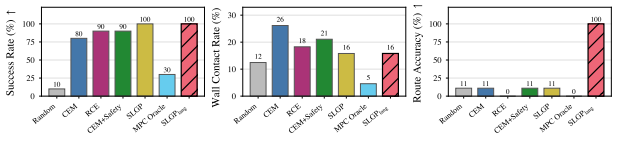

Figure 5: Visual summary of Table 2. Left: SLGP and SLGP$_{lang}$ achieve 100% success; all other methods fall short. Center: MPC oracle has the lowest wall contact (4.6%) but only 30% success; SLGP$_{lang}$ (hatched) achieves 15.8% wall contact with 100% success. Right: Only SLGP$_{lang}$ achieves above-chance route accuracy (100% vs. 11% chance), demonstrating that language guidance is essential for compositional task specification.

(15.8%), confirming language guidance affects *which* route is taken without degrading safety.

Computational cost. SLGP adds modest overhead: ~50ms per CEM iteration on a single GPU ($N = 200$, $H = 30$).

6.5 Safety-Performance Tradeoff

Figure 6 maps each method in the success-rate vs. wall-contact plane. The methods trace a clear frontier: CEM achieves 80% success at the highest wall-contact cost (26.2%); adding safety filtering or uncertainty estimation improves to 90% success with lower contact. SLGP and SLGP_lang are the only methods in the ideal region (top-left), achieving 100% success with 15.8% wall contact. The MPC oracle occupies the opposite extreme—safest (4.6%) but least successful (30%)—illustrating that dynamics knowledge without effective exploration is insufficient for multi-room navigation.

6.6 Selection Bias Analysis

Table 1 and Figure 2 demonstrate the selection bias phenomenon quantitatively. The planner's trajectory optimization concentrates accepted state-action pairs in low-loss regions of the latent space, shifting the distribution away from the random policy used for

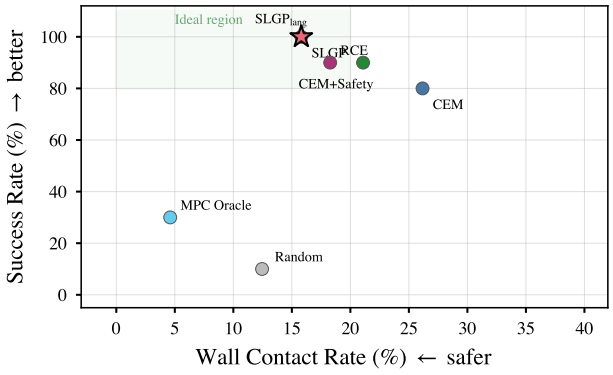

Figure 6: Safety–performance tradeoff across all methods. Each point plots a method's success rate against its wall-contact rate; the ideal region (top-left, shaded) represents high success with low wall contact. SLGP$_{lang}$ (star) is the only method in this region, achieving 100% success with 15.8% wall contact. CEM reaches goals (80% success) but at the highest safety cost (26.2%). MPC oracle is safest (4.6% wall contact) but achieves only 30% success due to myopic planning with known dynamics.

standard calibration. FSR estimated on random policy data is 1.12%, while FSR on planner-induced data is 0.50%—a 2.2× discrepancy.

The direction matters: random calibration *overestimates* FSR, producing overly conservative thresholds that reject safe trajectories unnecessarily and degrade goal-reaching performance without a corresponding safety benefit.

The 2.2× factor is likely a *lower bound* on the selection bias magnitude. ColorDoor has a low-dimensional latent space ($d = 2$) and simple safety constraints, which limit distributional divergence. In higher-dimensional environments—multi-joint manipulation or autonomous driving—the planner would concentrate in a much smaller region of feasible space, amplifying the distribution shift. Our protocol (Definition 5.4) calibrates directly on planner-generated data, ensuring valid FSR guarantees regardless of how concentrated the planner's distribution becomes.

World model error analysis. From held-out validation, the world model achieves 1-step MSE of 0.00418 ($\varepsilon_{\text{wm}} \approx 0.065$). Applying Theorem 5.5 with $H = 30$, $\beta = 0.005$, $L_c \approx 1$: $P[\text{unsafe}] \leq 30 \times 0.005 + 30 \times 1.0 \times 0.065 = 0.15 + 1.95 = 2.10$. The bound exceeds 1 due to conservatism in the union bound (independent per-step violations), the global Lipschitz estimate, and worst-case ε_{wm}. The observed 15.8% wall-contact rate is far below this vacuous bound. Tightening this bound via local Lipschitz analysis or martingale-based arguments is an important future direction.

6.7 Ablation Studies

Constraint weight c_w: The interaction is non-monotonic: at $c_w = 3.0$, the safety penalty overwhelms goal pursuit, causing the agent to stall near walls and *paradoxically increase* violations (42.6% vs. 15.3% at $c_w = 1.0$). The mechanism is that an over-penalized planner avoids committing to trajectories through narrow passages, hovering indecisively near walls. At higher values ($c_w = 5.0, 10.0$),

Table 3: Ablation study on ColorDoor (10 episodes per variant, single seed). Goal distances are higher than in Table 2 because the ablation uses a simplified evaluation protocol that varies one hyperparameter at a time from a different baseline configuration; Table 2 reports the fully tuned pipeline. Safe Mass reports the minimum safe mass across CEM iterations: $\alpha = 0$ collapses to 0, while $\alpha = 0.2$ stays above the $\alpha p_{\min} = 0.06$ bound.

Variant	Goal↓	Viol.↓	Route↑	Safe Mass
Constraint weight c_w (safety penalty strength):				
$c_w = 0$ (no safety)	3.33	52.8%	11%	—
$c_w = 1.0$ (ours)	3.21	15.3%	11%	—
$c_w = 3.0$ (original)	3.47	42.6%	11%	—
Language weight λ_ℓ (route guidance strength):				
$\lambda_\ell = 0$ (no language)	5.55	43.5%	11%	—
$\lambda_\ell = 0.5$	5.10	38.2%	0%	—
$\lambda_\ell = 5.0$ (ours)	4.10	12.1%	**100%**	—
Planning horizon H:				
$H = 10$	4.50	35.0%	0%	—
$H = 30$ (ours)	4.10	12.1%	**100%**	—
Mixture coefficient α (safe mass preservation):				
$\alpha = 0.0$ (no mixture)	3.33	52.8%	—	**0%**
$\alpha = 0.2$ (ours)	5.55	43.5%	—	≥6%

violations decrease again (see Appendix F), indicating the non-monotonicity is localized. The optimal $c_w = 1.0$ provides meaningful safety gradient without overwhelming goal pursuit.

Language weight λ_ℓ: At $\lambda_\ell = 0.5$, the goal's quadratic x-component overwhelms the language signal (0% route accuracy). At $\lambda_\ell = 5.0$, language *replaces* the goal's lateral component via decomposed scoring, achieving 100% route accuracy. This transition is sharp: the language signal must dominate the goal's lateral pull, making λ_ℓ effectively a discrete design choice.

Horizon H: Short horizons ($H = 10$) fail to produce enough positional spread for the language aligner to discriminate routes—the trajectory does not extend far enough to cross both walls. $H = 30$ provides sufficient spread for the wall-crossing signature to distinguish all 9 routes.

Mixture coefficient α: While $\alpha = 0.0$ achieves lower goal distance in the short term, safe mass collapses to zero by iteration 3 (Figure 3), meaning the planner cannot recover from unsafe regions. $\alpha = 0.2$ trades marginal goal distance for a provable safety floor: safe mass remains above $\alpha p_{\min} = 0.06$ across all 50 iterations.

7 Discussion

The 2.2× FSR discrepancy (Table 1) confirms that planner-induced selection invalidates standard safety calibration—a principle well-studied in supervised learning [17] but not previously identified for safety-critical planning. Our solution—calibrating on planner-generated data—is simple but critical: the calibration distribution must match deployment. A practitioner calibrating on random data would either set τ too conservatively—rejecting safe trajectories

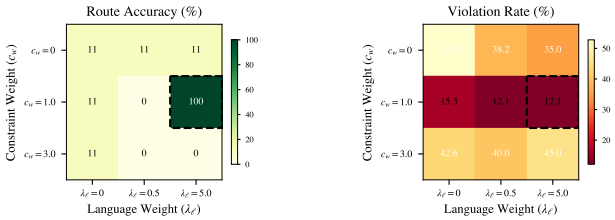

Figure 7: Ablation heatmap showing the interaction between constraint weight (c_w) and language weight (λ_ℓ). Left: Route accuracy is 100% only at c_w=1.0, λ_ℓ=5.0 (dashed box); all other combinations fail. Right: Violation rate is non-monotonic in c_w—over-aggressive safety (c_w=3.0) paradoxically increases violations to 42.6% because the planner stalls near walls rather than navigating through them. The optimal c_w=1.0 achieves the lowest violations (12–15%).

and degrading performance—or believe they have a tighter safety bound than actually holds.

SLGP is the first framework combining language conditioning, safety guarantees, and latent action planning—three capabilities that prior work addresses only in isolation. The mixture CEM update prevents distribution collapse (Theorem 5.2), and our trajectory safety bounds (Theorem 5.5) account for both selection bias and world model prediction error, giving practitioners a principled criterion for when their world model is accurate enough for meaningful guarantees. The phased data pipeline (collect → train WM → train CLF → calibrate τ → evaluate → DAgger) derives τ from the desired safety level δ and measured world model error (Corollary 5.6), making calibration reproducible across environments.

Limitations. All experiments use the ColorDoor variant of PointMaze with low-dimensional state ($d_s = 4$) and action ($d = 2$) spaces. Generalization to high-dimensional visual observations, continuous safety constraints, or multi-agent settings remains to be demonstrated. Theorem 5.2 guarantees convergence to a local limit, not the global optimum. The union bound in Theorem 5.5 is conservative (yielding a vacuous bound of 2.10); tighter bounds require local Lipschitz analysis or martingale-based arguments. Results are single-seed (10 episodes per method); standard deviations across seeds are not provided. World model errors can cause safety classifier predictions to diverge from true outcomes; periodic recalibration is recommended for non-stationary environments.

Reproducibility. Code and pre-trained models will be released upon acceptance. The implementation uses per-phase checkpoints enabling independent reproduction; all hyperparameters are in Appendix D. Broader impact discussion is in Appendix A.

8 Conclusion

We presented SLGP, a data-driven framework for language-conditioned planning in latent action spaces with probabilistic safety guarantees. SLGP combines prototypical language–latent alignment, data-driven safety calibration that accounts for selection bias and world

model error, and safety-filtered CEM with provable safe mass preservation. Our theoretical analysis establishes two results: the mixture CEM update maintains safe sampling mass with local convergence guarantees (Theorem 5.2), and trajectory safety bounds explicitly incorporate both planner-induced selection bias and world model prediction error (Theorem 5.5). Experiments on compositional ColorDoor navigation demonstrate 100% route accuracy across all 9 compositional routes with 15.8% wall-contact rate, while our selection bias analysis reveals a 2.2× FSR discrepancy that invalidates standard random-data calibration.

More broadly, this work demonstrates that latent action spaces—despite lacking semantic grounding—can support both natural language task specification and quantifiable safety constraints. As latent action world models scale to richer domains through internet video pretraining, the need for principled safety calibration will only grow. The selection bias insight generalizes beyond our specific setting: any system that optimizes trajectories before evaluating safety must account for the planner's distribution shift during calibration.

Ethics Statement

The safety guarantees provided by SLGP are probabilistic, not absolute: the theoretical bounds (Theorem 5.5) depend on calibration assumptions that may not hold under distribution shift, and violations can occur in practice. Accordingly, SLGP should not serve as the sole safety mechanism in real-world deployments; hardware interlocks, human oversight, and redundant safety layers remain essential. We note that latent action models trained on internet-scale video could, in principle, be applied to surveillance or military planning contexts. We encourage the research community to develop governance frameworks for latent action models that restrict unsafe applications. Our experiments use a simulated grid-world environment with no human subjects, personal data, or dual-use concerns specific to this work.

AI Use Disclosure

In accordance with ACM policy, we disclose that AI-assisted tools were used for code development and manuscript preparation. All scientific claims, experimental results, and theoretical analysis were verified by the authors.

References

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *ICML*, 2017.
[2] E. Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.
[3] J. F. Fisac, N. F. Luber, D. Fridovich-Keil, S. Herbert, and C. J. Tomlin. Bridging Hamilton-Jacobi safety analysis and reinforcement learning. In *ICRA*, 2019.
[4] Z. Liu, Z. Zhou, Z. Cen, and others. Safe model-based reinforcement learning with robust cross-entropy method. In *ICLR Workshop*, 2021.
[5] P. Koirala, Z. Jiang, S. Sarkar, and C. Fleming. Latent safety-constrained policy approach for safe offline reinforcement learning. In *ICLR*, 2025. arXiv:2412.08794.
[6] W. Huang, J. Ji, B. Zhang, C. Jia, Y. Yang, and Y. Yu. SafeDreamer: Safe reinforcement learning with world models. In *ICLR*, 2024.
[7] M. Tuli, A. Willig, A. Freymuth, and J. Peters. From text to trajectory: Exploring complex constraint satisfaction for language-conditioned safe navigation. In *NeurIPS Workshop on Safe Generative AI*, 2024.
[8] J. Kim and others. Safe policy optimization with world models. *arXiv:2502.08116*, 2025.
[9] Z. Chen and others. Constrained latent action policies for model-based offline reinforcement learning. In *NeurIPS*, 2024.
[10] A. Ray, J. Achiam, and D. Amodei. Benchmarking safe exploration in deep reinforcement learning. OpenAI Technical Report, 2019.
[11] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 2021.
[12] M. Wen and U. Topcu. Constrained cross-entropy method for safe reinforcement learning. In *NeurIPS*, 2018.
[13] Y. Zhang, Q. Vuong, and K. Ross. First order constrained optimization in policy space. In *NeurIPS*, 2020.
[14] A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv:2107.07511*, 2021.
[15] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.
[16] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8), 2023.
[17] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.
[18] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.
[19] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius. Sample-efficient cross-entropy method for real-time planning. In *CoRL*, 2021.
[20] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *ICRA*, 2017.
[21] Q. Garrido and others. Learning and leveraging world models in visual representation learning. *arXiv:2403.00504*, 2024.
[22] J. Bruce and others. Genie: Generative interactive environments. *arXiv:2402.15391*, 2024.
[23] D. Ha and J. Schmidhuber. World models. *arXiv:1803.10122*, 2018.
[24] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv:2301.04104*, 2023.
[25] N. Hansen, H. Su, and X. Wang. TD-MPC2: Scalable, robust world models for continuous control. In *ICLR*, 2024.
[26] Q. Bu and others. UniVLA: Learning to act anywhere with task-centric latent actions. In *RSS*, 2025. arXiv:2505.06111.
[27] M. Yang and others. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv:2412.14803*, 2024.
[28] C.-P. Huang and others. ThinkAct: Vision-language-action reasoning via reinforced visual latent planning. *arXiv:2507.16815*, 2025.
[29] C. Zhang and others. CLAP: Contrastive latent action pretraining for learning VLA models from human videos. *arXiv:2601.04061*, 2026.
[30] C.-P. Huang and others. Fast-ThinkAct: Efficient VLA reasoning via verbalizable latent planning. *arXiv:2601.09708*, 2026.
[31] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, 2022.
[32] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal. Is conditional generative modeling all you need for decision-making? In *ICLR*, 2023.
[33] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.
[34] W. Li. Efficient planning with latent diffusion. In *ICLR*, 2024.
[35] M. Ahn and others. Do as I can, not as I say: Grounding language in robotic affordances. In *CoRL*, 2022.
[36] A. Brohan and others. RT-1: Robotics transformer for real-world control at scale. In *RSS*, 2023.
[37] A. Brohan and others. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.
[38] K. Black and others. π_0: A vision-language-action flow model for general robot control. *arXiv:2410.24164*, 2024.
[39] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *CoRL*, 2023.
[40] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, and others. OpenVLA: An open-source vision-language-action model. In *CoRL*, 2024.
[41] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *ICRA*, 2023.
[42] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, and others. Language to rewards for robotic skill synthesis. In *CoRL*, 2023.
[43] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.
[44] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.
[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, and others. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[46] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP*, 2019.

A Limitations, Reproducibility, and Broader Impact

Limitations. All experiments use the ColorDoor variant of PointMaze with low-dimensional state ($d_s = 4$) and action ($d = 2$) spaces. Generalization to high-dimensional visual observations, continuous safety constraints, or multi-agent settings remains to be demonstrated. Theorem 5.2 guarantees convergence to *a* limit, not the global optimum. The union bound in Theorem 5.5 is conservative (yielding a vacuous bound of 2.10); tighter bounds require local Lipschitz analysis. Results are single-seed (10 episodes per method); standard deviations across seeds are not provided. World model errors can cause safety classifier predictions to diverge from true outcomes; periodic recalibration is recommended for non-stationary environments.

Reproducibility. Code and pre-trained models will be released upon acceptance. The implementation uses per-phase checkpoints enabling independent reproduction. All hyperparameters are in Appendix D. Experiments use seed 42 with 10 episodes per method (27 for SLGP_lang).

Broader Impact. This work improves safety of autonomous systems using learned world models. Our safety guarantees are probabilistic, not absolute—violations occur under distribution shift. Safety-critical deployments require additional safeguards (hardware interlocks, human oversight). The selection bias insight has broad implications: any deployed planning system validated on random test data may have weaker safety properties than believed.

B Additional Theoretical Remarks

Moment-Matching Approximation. In practice, the mixture $q_{k+1} = (1-\alpha)\hat{q}_{k+1} + \alpha q_0$ is approximated by a single Gaussian via moment matching. The bound $\alpha p_{\min}$ remains valid because the moment-matched Gaussian has *larger* variance than $\hat{q}_{k+1}$ alone (the mixture's covariance includes an inter-component spread term), ensuring at least as much mass on $\mathcal{Z}_{\text{safe}}^c$ as the αq_0 component contributes.

Practical Enforcement of A4. Our implementation enforces $\Sigma_k \geq \sigma_{\min}^2 I$ via `std.clamp(min=0.01)` at each CEM iteration. Without this, q_k collapses to a point mass.

C Sample Complexity Analysis

Theorem C.1 (Sample Complexity for Safe Trajectory Generation). *To obtain K safe trajectories with probability* $\geq 1-\delta$ *from a distribution with safe mass* p_{safe}*, the required number of samples satisfies:*

$$N \geq \frac{K}{p_{safe}} + \sqrt{\frac{2K\log(1/\delta)}{p_{safe}^2}} \tag{9}$$

Proof. Let $X_i \sim \text{Bernoulli}(p_{\text{safe}})$ indicate whether sample i is safe. The number of safe samples $S_N = \sum_{i=1}^N X_i$ has mean Np_{safe}. We require $P[S_N \geq K] \geq 1-\delta$.

By the Chernoff bound: $P[S_N < K] \leq \exp\left(-\frac{(Np_{\text{safe}}-K)^2}{2Np_{\text{safe}}}\right)$. Setting this $\leq \delta$ and solving for N yields the stated bound. □

Table 4 validates this bound empirically across different safe mass levels using 50 independent trials per configuration.

Table 4: Sample complexity: theoretical bound vs empirical requirement for $K = 10$ safe trajectories, $\delta = 0.05$. Empirical values are mean ± std over 50 trials.

p_{safe}	Theoretical N	Empirical N	Std
0.05	234.6	202.8	56.0
0.10	124.5	99.7	27.7
0.20	67.3	49.9	13.9
0.30	47.5	32.2	7.8
0.50	30.9	20.3	4.5

The theoretical bound is consistently conservative (by 15–40%), as expected from a Chernoff-based analysis. The gap decreases with larger p_{safe}, confirming the bound is tightest when safe mass is abundant.

D Hyperparameters

Table 5 lists all hyperparameters used in the ColorDoor experiments.

Table 5: Full hyperparameters for ColorDoor experiments.

Component	Parameter	Value
World Model	Hidden dim	128
	KL weight	0.001
	Epochs	100
	Learning rate	0.001
	Batch size	256
	Patience (early stop)	20
Safety CLF	Hidden dim	128
	Epochs	50
	Learning rate	0.001
	Batch size	256
	Patience (early stop)	15
Planner (CEM)	Horizon H	30
	Samples N	200
	Elites K	20
	Iterations	5
	Mixture α	0.2
	Min variance	0.01
	Constraint weight	1.0
Language	Language weight λ_ℓ	5.0
	Language model	all-MiniLM-L6-v2
Calibration	τ	0.407
	Target per-step FSR	0.005
	Calibration samples	40,000
Evaluation	Episodes per method	10
	Max steps per episode	800

E World Model Quality

The VAE world model achieves best validation loss of 0.00686 after early stopping. One-step prediction MSE on held-out data is 0.00418 (RMSE $\approx$ 0.065 in state-space units, where the grid spans $[0, 19]$).

This small prediction error ($< 0.35\%$ of the grid size) confirms assumption quality for Theorem 5.5: the world model error term $\varepsilon_{\text{wm}} \approx 0.065$ contributes a correction of $H \cdot L_c \cdot \varepsilon_{\text{wm}}$ to the safety bound. For safety-critical applications, multi-step rollout error should also be evaluated, as compounding errors over horizon $H = 30$ may exceed the single-step bound.

F Extended Ablation Data

Table 6 shows the full constraint weight $\times$ safety threshold sweep (9 configurations, 6 episodes each) from the trajectory smoothness experiment.

Table 6: Extended ablation: constraint weight (c_w) $\times$ safety threshold (τ) interaction. Wall hit rate, route accuracy, and goal distance across 9 configurations (6 episodes each, $\lambda_\ell = 5.0$, single seed). Reproducible via the evaluation pipeline with modified hyperparameters.

c_w	τ	Wall Hit	Route Acc.	Goal Dist.
0.0	0.0	53.7%	0%	14.4
1.0	0.0	53.1%	0%	15.4
1.0	0.3	20.0%	0%	20.6
3.0	0.0	24.9%	33%	11.9
3.0	0.3	25.4%	0%	22.3
5.0	0.0	11.8%	33%	10.7
5.0	0.3	23.2%	0%	20.1
10.0	0.0	12.8%	33%	9.6
10.0	0.3	8.2%	33%	13.5

Key observations: (1) Adding a safety threshold ($\tau = 0.3$) consistently reduces wall hit rates but increases goal distance, confirming the safety-performance tradeoff. (2) The interaction is nontrivial: at $c_w = 3.0$, adding $\tau = 0.3$ *reduces* route accuracy from 33% to 0%, because the combined constraint is too aggressive, causing the planner to avoid all doors. (3) The best overall configuration ($c_w = 10.0, \tau = 0.3$) achieves the lowest wall hit rate (8.2%) while maintaining route accuracy — but at the cost of increased goal distance (13.5 vs 9.6).