FACTORED DIFFUSION POLICIES:COMPOSITIONALLY GENERALIZED ROBOT CONTROL WITH A SINGLE SCORE NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Robotic tasks are typically specified by a tuple of factors, such as the goal to be reached, the obstacles to be avoided, and the appearance of the scene, and collecting expert demonstrations for every combination of factor values grows combinatorially. We present *factored diffusion policies*: a single diffusion network trained with per-factor null-token dropout, whose score decomposes additively across factors. When the factors are approximately conditionally independent given the action and observation, the composed score approximates the joint score uniformly over all factor combinations, so a training set in which every factor value appears at least once suffices: the task budget grows with the sum of the factor cardinalities rather than their product. The decomposition residual equals the action-gradient of the factors' interaction log-ratio and is measurable from forward passes, so the assumption can be checked per task, and the same structure yields a closed-loop trajectory-tube certificate. We evaluate our approach using drone racing simulations where a drone must navigate through gates while balancing speed and path curvature against tracking overshoot which can cause gate crashes. Trained on such a cover, the factored network passes several times more held-out gates than the same architecture without dropout, replicated across disjoint covers, seeds, and two and three factors, while the same composition rule over separately trained networks generalizes far worse. Injected factor interaction raises the measured residual and degrades composed inference, and our experiments indicate that the gain comes from the factored training, and the decomposition supplies the task budget, the certificate, and the approximate conditional independence check.

1 INTRODUCTION

A standard approach to robot policy learning treats the task as a monolithic input z and trains a single policy network $p(a \mid z)$ on a finite set of training tasks $\mathcal{Z}_{\text{train}}$ by collecting expert demonstrations for each $z \in \mathcal{Z}_{\text{train}}$. Typical robotic task specifications have a factored structure, for example, a navigation task is specified by the relative pose of the target, obstacles to be avoided, and also the appearance of the environment; a landing task by allowed approach angles and environment conditions like wind; a racing task by the track geometry and the gate sizes. A task with K such factors is a tuple $z = (z_1, \ldots, z_K) \in \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$, where $\mathcal{Z}_i$ is the set of values factor i can take. Without exploiting the factored structure, the cost of data collection and training grows as $\prod_i |\mathcal{Z}_i|$. In this paper, we explore how the factored structure of task specifications can be exploited to improve data efficiency of training and generalization to unseen factor combinations.

The cost of monolithic conditioning is unavoidable in general, but it could shrink dramatically when the factors are *approximately conditionally independent given an action–observation pair* (a, o): knowing (a, o) and the value of one factor does not appreciably help infer the others (Assumption 1). For example, for a generalizable drone racing policy, the track and the gate-opening size can be expected to be approximately conditionally independent given the drone state (o) and a planned action chunk (a): the spatial route of the planned trajectory picks out the track while its speed and precision pick out the gate size. Under this assumption, each factor's effect on the policy can be learned independently and combined post-hoc, dropping the training-task budget from $\prod_i |\mathcal{Z}_i|$ to $O\big(\sum_i |\mathcal{Z}_i|\big)$ and giving compositional generalization to unseen factor combinations. The remainder

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

of the paper turns this idea into a constructive method for diffusion policies, an approximation bound, and a closed-loop certificate.

A single diffusion network with approximate conditional independence of factors. Diffusion policies are a leading method for learning control policies from expert demonstrations (Chi et al., 2023; Reuss et al., 2024): they are trained to approximate the *score* of $p(a \mid o, z)$, the gradient $\nabla_a \log p$ in action space. The crucial property is that density factorization under conditional independence becomes *addition* in score space, and *approximate* conditional independence becomes a bounded norm on the gradient of the residual factor interaction (Theorem 1). If the factors are approximately independent given (a, o), then

$$\nabla_a \log p(a \mid o, z_1, \ldots, z_K) \;\approx\; s_\varnothing + \textstyle\sum_{i=1}^{K} \Delta_i,$$

where $s_\varnothing$ is the unconditional score and Δ_i is the score correction for factor i alone (Definition 1). This additive composition follows from the log-density structure; in contrast, a behavior-cloning policy outputs a point estimate, and does not admit such a decomposition. We exploit additive composition by training a *single shared* score network with per-factor null-token dropout, so that each Δ_i is learned from data covering only one factor at a time, and at inference we compose $s_{\mathrm{comp}} = s_\varnothing + \sum_i \Delta_i$ (Definition 1); the same network also supports joint readout, which we compare against throughout. This identity is the algebraic backbone of classifier-free guidance (Ho & Salimans, 2021) and compositional image generation (Liu et al., 2022; Du et al., 2023), where it composes a single concept or text-prompt factor with the unconditional score. Our setting, has two points of distinction: First, the policy runs in closed loop: the composed score field denoises samples into actions, which a tracking controller executes on the plant, and what ultimately matters is a task-level event such as reaching a goal or gate passage. Second, we quantify the gap when conditional independence holds only approximately: Theorem 1 bounds $\|s - s_{\mathrm{comp}}\|$ by $2\sqrt{GM}$ uniformly in z, where G bounds the residual factor-interaction strength (Assumption 1) and M bounds the curvature of the factor-interaction log-ratio. Because this bound holds for every $z \in \prod_i \mathcal{Z}_i$, including factor combinations absent from training, and s_{comp} needs only the unconditional score plus one per-factor correction, training data in which every factor value appears at least once suffices: $O(\sum_i |\mathcal{Z}_i|)$ tasks against $\Omega(\prod_i |\mathcal{Z}_i|)$ for a joint model. The residual this theorem bounds is measurable and it is the action-gradient of the factors' interaction log-ratio (Proposition 1), and therefore, the independence assumption can be checked per task.

Scores also make certification possible. The per-factor structure also yields a closed-loop guarantee. Fix a nominal training task z^{nom} and let the *nominal trajectory* be the closed-loop state trajectory the system would follow under the true joint score on z^{nom}. Under (i) bounded factor-interaction (Assumption 1), (ii) per-factor Lipschitz score (Assumption 3), and (iii) a contracting tracking controller (Assumption 2), the closed-loop trajectory generated by the composed policy on a (possibly held-out) test task stays within a tube of radius R around the nominal (Theorem 2). R aggregates four sources: the decomposition residual $2\sqrt{GM}$ from Theorem 1, a score-matching error $(2K-1)\eta$ on training data, a per-factor extrapolation term and a process disturbance from sensor noise and unmodeled dynamics. The transfer from score-space to state-space error passes through an ODE-sensitivity factor C_{ode}, for which we give two computable bounds following standard methods: analytical Grönwall (uniform over tasks) and a path-dependent linear time-varying (LTV) bound using per-step Jacobians along the nominal path. If the nominal trajectory completes the task with a robustness margin of at least R, then the realized trajectory under the composed policy does so as well. In our experiments the rigorous radius does not clear the physical gate margins, so we use the certificate as a robustness diagnostic rather than a deployment-level task-success guarantee.

Experimental validation with drone racing. We study the approach on drone racing: a drone (13-state underactuated Newton–Euler quadrotor with 4 inputs) has to fly through the gates of a race track in prescribed order, and a task is specified by a track identity z_1, a gate aperture z_2, and in some experiments a third factor z_3, the expert speed. The quadrotor accelerates only by tilting its thrust axis, so the policy has to balance speed and path curvature against tracking overshoot, and an overshoot exceeding the aperture half-width at a crossing fails that gate. Our experiments validate the potential for compositional generalization from sum-sized task coverage: a single score network trained with per-factor dropout on tasks in which every factor value appears at least once succeeds on the majority of held-out factor combinations. First, the shared network passes 87% of held-out gates (26 of 30)

under composed inference, against 100% for an oracle trained directly on the held-out tasks, while the same composition rule over separately trained per-factor networks in the style of Wang et al. (2024); Patil et al. (2024) passes 10%. This leads us to conclude that parameter sharing, not the additive score alone, helps with generalization. Second, a jointly conditioned network without factor dropout matches ours; the performance gap between the two training modes appears at the sum-sized coverage Theorem 1 describes, where dropout raises held-out passage from 3–16% to 36–50% with two factors and from 5–11% to 25–32% with three. Third, the gain comes from the factored training, not from the inference rule: composed inference nearly matches joint readout of the same network at dense coverage and trails it at sparse coverage, as the error accounting predicts, since the composed score combines $2K-1$ weighted network evaluations and so carries $(2K-1)\eta$ of score-matching error where joint readout carries η. Fourth, the residual Theorem 1 bounds is measurable, and when we inject a controlled factor interaction into the expert data it rises monotonically while composed held-out passage falls from 87% to 37% and joint readout is unaffected. Additionally, we present a preliminary study of visual policies that takes the factored conditioning to raw pixels: on single-gate traversal from an onboard RGB camera and a noisy gyro, with four factors (venue, gate color, gate ID, approach side), factor conditioning raises success over a baseline without factor input and transfers zero-shot to a venue held out of training entirely.

Contributions. (i) a factored diffusion policy realized as a single shared score network with per-factor null-token dropout, with a compositional generalization bound (Theorem 1) under which the training-task requirement grows with $\sum_i |\mathcal{Z}_i|$ rather than $\prod_i |\mathcal{Z}_i|$; (ii) an identity expressing the decomposition residual as the action-gradient of the factors' interaction log-ratio (Proposition 1), which makes the independence assumption checkable per task from forward passes alone; (iii) a trajectory-tube certificate (Theorem 2) chaining score error through the sampling ODE and the closed-loop tracking dynamics, used as a robustness diagnostic; (iv) controlled experiments that separate the mechanisms: a jointly conditioned no-dropout baseline, sum-sized coverage at two and three factors, and a controlled factor-interaction sweep (Section 4).

2 Problem of Generalizable Control from Demonstrations

Consider a robot with state $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ and discrete-time dynamics

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

where $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input applied at step t. A *task* is specified by a tuple of factors $z = (z_1, \ldots, z_K) \in \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$. Each task z carries a *success predicate* ϕ_z over closed-loop trajectories $x_{0:T}$ that determines whether the task is completed. We assume an expert that generates demonstration data for a finite training set $\mathcal{Z}_{\text{train}} \subsetneq \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$. The expert produces a command $a_t \in \mathcal{A}$ at each step, and a low-level *tracking controller* κ generates the final plant input $u_t = \kappa(x_t, a_t)$. We make the standard assumption that the closed-loop system formed by κ and f is contracting (Lohmiller & Slotine, 1998) or incrementally input-to-state stable (Angeli, 2002), i.e., small perturbations to a_t produce bounded state deviations (Assumption 2 in Appendix A.1). The command a_t may be a single setpoint, a trajectory, or an entire plan. At each step the policy also receives an observation $o_t \in \mathcal{O}$, which may include any subset of the robot state and a perceptual input (e.g., a camera image). The factored decomposition we develop in Section 3 is independent of the observation modality. Expert demonstrations are available only for a finite training set $\mathcal{Z}_{\text{train}}$ costing $O(|\mathcal{Z}_{\text{train}}|)$. For this paper, we assume the factors for the training tasks $z \in \mathcal{Z}_{\text{train}}$ are chosen independently randomly, that is, the training-task distribution factorizes as $p_{\text{gen}}(z) = \prod_{i=1}^{K} p_{\text{gen}}(z_i)$. This is a property of the data-collection protocol that holds by design. The goal is to design a policy $\pi_\theta(o_t, z)$ that generalizes from $\mathcal{Z}_{\text{train}}$ to the full product space $\mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$, including tasks for which no expert demonstration or training data exists.

Running example: drone racing. In drone racing, a quadrotor has to autonomously pass through a gate using onboard IMUs and a forward-facing RGB camera. The state $x \in \mathbb{R}^{13}$ (position, attitude quaternion, velocity, body angular rate) evolves under control $u \in \mathbb{R}^4$ (collective thrust, body torque) according to Newton–Euler dynamics. We study two instantiations that share these dynamics but differ in factors and observation: a *state-based full-race* setting ($K = 2$: track $\times$ gate-opening size, Section 4.1) and a *vision-based single-gate* setting ($K = 4$: venue, gate color, gate ID, approach side, Appendix C). The tracking controller κ is an $\mathrm{SE}(3)$ geometric controller (Lee et al., 2010) with

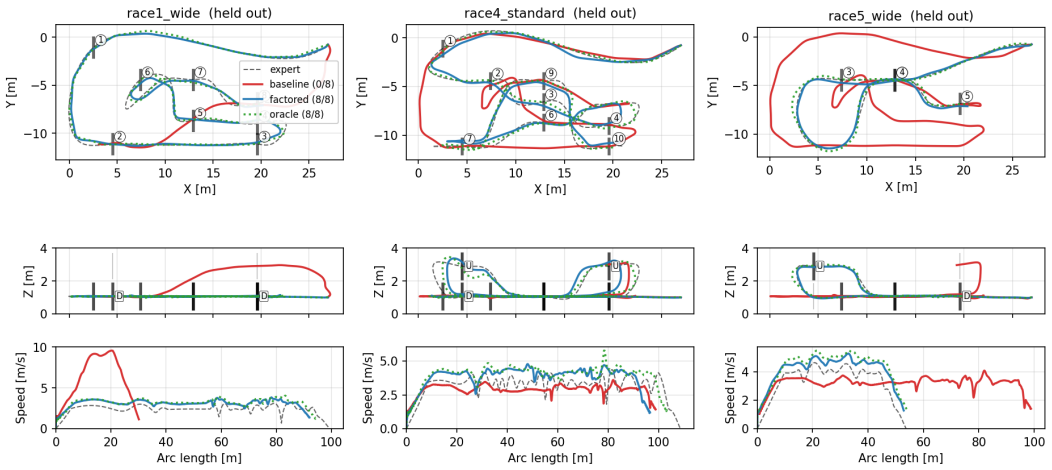

Figure 1: **Compositional generalization on held-out drone-racing tasks.** Closed-loop trajectories on three (track, gate-size) pairs not seen during training; top-down (X–Y), side view (X–Z), and speed vs. arc length. Black dashed: expert reference. Green dotted: oracle (trained on the same pair). Red: unfactored baseline. Blue: factored compositional policy $s_{\text{comp}} = s_{\varnothing} + \Delta_1 + \Delta_2$. The factored policy tracks expert and oracle to ~ 0.1 m; the baseline diverges. Aggregate over 20 tasks: Section 4.1.

fixed gains. The success predicate ϕ_z requires the drone's center to pass within a gate half-width r of each gate center (in the case of full-race, in the prescribed order). In the state-based full-race setting, $r = r(z_2)$ varies with the gate-opening factor z_2; in the vision-based single-gate setting, r is fixed. Every $z \in \mathcal{Z}_{\text{train}}$ requires careful hand-tuning of the trajectory and a speed profile or training a specialized policy, or multiple expert demonstrations.

3 Compositional Generalization via Conditional Independence

Generalizing from $\mathcal{Z}_{\text{train}}$ to the full product space $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$ requires additional structure: each factor's effect on the policy must be learnable independently and combinable post hoc. The key hypothesis for our approach to work is that the factors are *approximately conditionally independent given an action–observation pair* (a, o), that is, knowing (a, o) and the value of one factor does not help infer the others beyond what (a, o) alone reveals. For example, z_1 = track shape governs the direction in a while z_2 = gate aperture governs its speed. These are two largely disjoint channels in (a, o), so neither factor reveals the other beyond what (a, o) already does. Formally, let $p(a, o, z)$ denote the joint distribution over action–observation–task tuples induced by the training data: sample a task $z \sim p_{\text{gen}}$, roll out the expert policy on task z, and record each action–observation pair produced along the trajectory. The forward-noising process of Section 3.1, $a_\sigma = \alpha_\sigma a + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ independent of (a, o, z), extends this to a noised joint $p_\sigma(a_\sigma, o, z)$ at every $\sigma \in [0, \sigma_{\max}]$, with $\sigma = 0$ recovering the clean joint. The conditional posterior $p_\sigma(z \mid a_\sigma, o)$ and the per-factor marginals $p_\sigma(z_i \mid a_\sigma, o)$ are obtained from the noised joint via Bayes' rule. We measure the interaction among factors through the *interaction log-ratio*

$$g_\sigma(z, a_\sigma, o) \;:=\; \log \frac{p_\sigma(z \mid a_\sigma, o)}{\prod_{i=1}^{K} p_\sigma(z_i \mid a_\sigma, o)}, \qquad z = (z_1, \ldots, z_K), \quad \sigma \in [0, \sigma_{\max}]. \tag{2}$$

Conditional independence at noise level σ holds at (a_σ, o) if and only if $g_\sigma(z, a_\sigma, o) = 0$ for all z.

Assumption 1 (Bounded factor interaction)**.** There exists $G \geq 0$ such that for every noise level $\sigma \in [0, \sigma_{\max}]$, every action–observation pair (a_σ, o) in the support of p_σ, and every $z \in \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$, $|g_\sigma(z, a_\sigma, o)| \leq G$.

Even when the training prior $p_{\text{gen}}(z) = \prod_i p_{\text{gen}}(z_i)$ factorizes, conditioning on the action a can induce dependence between factors (the collider effect), and conditional diffusion models can in general violate strict conditional independence (Gaudi et al., 2025). Assumption 1 quantifies the strength of such effects via G, and Proposition 1 makes the resulting score residual measurable. On our state-based benchmark the norm of the measured score residual averages 0.45 on training tasks

and 1.08 on held-out tasks, in the network's output units: small where the assumption is interpolated, and roughly doubled where it is extrapolated (Section 4).

3.1 Diffusion Policies, Denoising ODEs, and ODE Sensitivity

Let $p(a \mid o, z)$ denote the conditional distribution over commands $a \in \mathcal{A}$ given observation o and task z, induced by the expert demonstrations. We learn to approximate $p(a \mid o, z)$ with a diffusion model, which operates on its *score function*, the gradient of the log-density with respect to the action. The score is the operational object on which the compositional structure can be implemented.

A diffusion model defines a forward noising process $a_\sigma = \alpha_\sigma a_0 + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ and $\alpha_\sigma^2 + \sigma^2 = 1$ ($\sigma = 0$ is clean data, large σ is near-Gaussian), where $a_0 \sim p(a \mid o, z)$. The score at noise level σ is

$$s(a_\sigma, \sigma, o, z) \;:=\; \nabla_{a_\sigma} \log p_\sigma(a_\sigma \mid o, z), \tag{3}$$

where p_σ is the marginal density of a_σ. A neural network s_θ approximates s via denoising score matching. Actions are generated by solving the reverse sampling ODE

$$\frac{da_\sigma}{d\sigma} = f_{\mathrm{dr}}(\sigma)\, a_\sigma + \tilde{g}(\sigma)\, s_\theta(a_\sigma, \sigma, o, z) \tag{4}$$

backward from $\sigma_{\max}$ (near-Gaussian) to $\sigma \approx 0$ (near-clean), where f_{dr} and $\tilde{g}$ are schedule-dependent drift and diffusion coefficients. In practice the integration is performed with a discrete-time scheduler such as DDIM (Denoising Diffusion Implicit Models Song et al., 2021a), which under a deterministic-seed setting realizes (4). At inference on an unseen task the policy runs a *composed learned* score that differs from the true joint score, and we need to bound how much this difference affects the generated command. The bound below is a standard Grönwall argument on (4); similar arguments appear in convergence analyses for diffusion samplers (Song et al., 2021b; Chen et al., 2023; Benton et al., 2024). Throughout the sensitivity analysis both denoising chains start from a shared initial sample $a_{\sigma_{\max}} = a'_{\sigma_{\max}} = a^\star$; since (4) is a probability-flow ODE this makes the sampled action a deterministic function of (o, z) and yields a per-task certificate. Section 4.1 verifies that the conclusions transfer to fresh-seed sampling at deployment.

Lemma 1 (ODE Sensitivity)**.** *Let $s, s' : \mathcal{A} \times [0, \sigma_{\max}] \to \mathcal{A}$ be two score fields with (a) $\|s - s'\|_\infty \le \varepsilon$ and (b) s is $L_a(\sigma)$-Lipschitz in its first argument. Let $\{a_\sigma\}, \{a'_\sigma\}$ be denoising paths generated by the reverse ODE* (4) *from a shared initial noise $a_{\sigma_{\max}} = a'_{\sigma_{\max}}$. Then*

$$\|a_0 - a'_0\| \;\le\; C_{\mathrm{ode}} \cdot \varepsilon, \;\; C_{\mathrm{ode}} \coloneqq \int_0^{\sigma_{\max}} |\tilde{g}(\sigma)| \, \exp\Big(\int_0^{\sigma} \big[|f_{\mathrm{dr}}(\sigma')| + |\tilde{g}(\sigma')|\, L_a(\sigma') \big] d\sigma' \Big)\, d\sigma. \tag{5}$$

Lemma 1 is conservative: the scalar Lipschitz bound $L_a(\sigma)$ collapses the directional structure of the score Jacobian, and the Grönwall integral compounds the worst case at every noise level. Linearizing the discrete DDIM map around the nominal denoising trajectory $\{\bar{a}_k\}$ yields a state-transition matrix whose operator norm replaces the scalar Lipschitz product, capturing the anisotropy of the per-step Jacobians and inter-step cancellation Fan et al. (2016). The resulting *path-dependent LTV* constant $C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a})$ (Lemma 2) is about five orders of magnitude tighter than the Grönwall constant in our experiments (Section 4.1).

3.2 Factored Score Decomposition

Definition 1 (Composed score)**.** Let $z = (z_1, \ldots, z_K)$ be a task. For each $i \in \{1, \ldots, K\}$, the *score correction* for factor i is $\Delta_i(a_\sigma, \sigma, o, z_i) = s(a_\sigma, \sigma, o, z_i) - s(a_\sigma, \sigma, o)$, where $s(\cdot, z_i)$ is the score conditioned on factor i alone (other factors marginalized via null-token substitution) and $s(\cdot)$ is fully unconditional. The *composed score* for z is

$$s_{\mathrm{comp}}(a_\sigma, \sigma, o, z) = s(a_\sigma, \sigma, o) + \sum_{i=1}^{K} \Delta_i(a_\sigma, \sigma, o, z_i). \tag{6}$$

Theorem 1 (Decomposition error bound)**.** *Suppose Assumption 1 holds and for every $\sigma \in [0, \sigma_{\max}]$, the interaction log-ratio g_σ defined in* (2) *is twice differentiable in a_σ with $\|\nabla^2_{a_\sigma} g_\sigma(z, a_\sigma, o)\|_{\mathrm{op}} \le M$ uniformly in σ, a_σ, o, z. Then for all $a_\sigma,\ \sigma,\ o,\ z$,*

$$\|s(a_\sigma, \sigma, o, z) - s_{\mathrm{comp}}(a_\sigma, \sigma, o, z)\| \le 2\sqrt{GM}. \tag{7}$$

The composed score s_{comp} approximates the true joint score $s(\cdot, z)$ within $2\sqrt{GM}$ for *every* $z \in \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$, including tuples absent from $\mathcal{Z}_{\text{train}}$. Computing s_{comp} requires $K+1$ learned components (Definition 1): the unconditional score $s(\cdot)$ and one per-factor score $s(\cdot, z_i)$ for each factor i. Learning $s(\cdot, z_i)$ requires training data in which the value z_i appears with at least one setting of the other factors. A training set of size $|\mathcal{Z}_{\text{train}}| = O\big(\sum_i |\mathcal{Z}_i|\big)$ in which every factor value appears in at least one training task therefore suffices to evaluate s_{comp} on all $\prod_i |\mathcal{Z}_i|$ tasks. A joint (non-compositional) model that learns $s(\cdot, z)$ directly requires training tasks covering all $\Omega\big(\prod_i |\mathcal{Z}_i|\big)$ factor combinations.

We emphasize that this is a factor-value *coverage* statement: it counts which training tasks must appear, not how many demonstrations per task are needed to estimate each score to a target accuracy; estimation quality enters separately through the measured score-matching error η in (9). The quantity Theorem 1 bounds has an exact form, which makes the assumption checkable.

Proposition 1. *If the scores in Definition 1 are exact, that is* $s(a_\sigma, \sigma, o, z_i) = \nabla_{a_\sigma} \log p_\sigma(a_\sigma \mid o, z_i)$ *and* $s(a_\sigma, \sigma, o) = \nabla_{a_\sigma} \log p_\sigma(a_\sigma \mid o)$*, then for all* a_σ, σ, o, z*,*

$$s(a_\sigma, \sigma, o, z) - s_{\text{comp}}(a_\sigma, \sigma, o, z) = \nabla_{a_\sigma} g_\sigma(z, a_\sigma, o). \tag{8}$$

The residual therefore vanishes identically under conditional independence and is controlled by Theorem 1 otherwise. Evaluated on a trained network rather than on exact scores, its norm, which we denote ε_D, estimates $\|\nabla_{a_\sigma} g_\sigma\|$ together with the network's own score error, which cannot be separated from it; ε_D is an estimator of the assumption's violation rather than an exact reading of it. It needs only forward passes and no ground-truth score, and it can be computed for a factor tuple before deploying on it.

The decomposition $s_{\text{comp}} = s + \sum_i \Delta_i$ is not fully determined by training data: a constant can shift between s and any Δ_i without changing s_{comp}. We resolve this via *factor dropout* during training, replacing each factor conditioning independently with a learned null token with probability p_{drop}, so s is identified as the all-null behavior and each Δ_i as the residual when only factor i is revealed (Appendix A.3). A network trained with $p_{\text{drop}} = 0$ also receives every factor, but only ever jointly.

3.3 Trajectory Tube Certificate

In our setting the expert controller is deterministic: for each (o, z) there is a unique reference chunk $a_0^*(o, z) \in \mathcal{A}$, so $p(a_0 \mid o, z) = \delta(a_0 - a_0^*(o, z))$, where δ denotes the Dirac delta distribution (a point mass at a_0^*). The noised density is then a Gaussian centered at $\alpha_\sigma a_0^*$, and the true score has the closed form $s(a_\sigma, \sigma, o, z) = -\frac{a_\sigma - \alpha_\sigma\, a_0^*(o,z)}{\sigma^2}$. This closed form makes the score-matching error η (9) directly measurable from training data.

Theorem 2 additionally needs a per-factor Lipschitz regularity of the trained score with respect to each factor z_i (Assumption 3); this is a mild architectural property that supplies the per-factor sensitivity constants L_i. The remaining ingredient is the network's approximation error. The score-matching error η is a *measured* quantity: the worst-case gap between s_θ and the true score on the training set,

$$\eta \;:=\; \sup_{\substack{a_\sigma,\, \sigma,\, o \\ c \,\in\, \mathcal{Z}_{\text{train}} \cup \{\varnothing\}}} \|s_\theta(a_\sigma, \sigma, o, c) - s(a_\sigma, \sigma, o, c)\|, \tag{9}$$

where c ranges over all conditioning configurations used in the composed score (unconditional, each single-factor, and the joint); how η is estimated is discussed in Appendix F. Expanding the composed score (Definition 1) gives $s^{\text{comp}} = \sum_{i=1}^K s(\cdot, z_i) - (K-1)\, s_\varnothing$, so the triangle inequality gives

$$\|s_\theta^{\text{comp}} - s^{\text{comp}}\| \le (2K-1)\eta. \tag{10}$$

Theorem 2 (Factored trajectory tube)**.** *Under Assumptions 1, 2, and 3, and either (a) the* $L_a(\sigma)$*-Lipschitz condition of Lemma 1, or (b) the Jacobian regularity of Lemma 2, let* x_t^{nom} *be the closed-loop state trajectory under the* true *joint score* $s(\cdot, z^{\text{nom}})$ *for a nominal task* $z^{\text{nom}} \in \mathcal{Z}_{\text{train}}$*, and let* x_t *be the trajectory under the* composed learned *score* $s_\theta^{\text{comp}}(\cdot, z)$ *with per-factor perturbations* $\|z_i - z_i^{\text{nom}}\| \le \delta_i$ *for each* $i \in \{1, \ldots, K\}$*, starting from* $\|x_0 - x_0^{\text{nom}}\| \le \delta_0$*. Let* η *be the score-matching error* (9)*, measured on* $\mathcal{Z}_{\text{train}}$*. Then*

$$\|x_t - x_t^{\text{nom}}\| \le \lambda^t\, \delta_0 + \frac{B_\kappa\, C_{\text{ode}}\, \varepsilon_s + w}{1 - \lambda}, \tag{11}$$

where the total score perturbation decomposes as $\varepsilon_s \le 2\sqrt{GM} + (2K-1)\eta + \sum_{i=1}^{K} L_i\,\delta_i$*, with terms from Theorem 1, the network approximation error* (10)*, and the factor perturbation.*

C_{ode} is instantiated either by the Grönwall bound $C_{\mathrm{ode}}^{\mathrm{ana}}$ (Lemma 1, uniform over z) or by the path-dependent LTV bound $C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a}^{(z)})$ (Lemma 2, per task); under hypothesis (b) the bound is first-order in ε_s (Appendix F). After the transient decays the error is bounded by $R_{\mathrm{ss}} = (B_\kappa\, C_{\mathrm{ode}}\varepsilon_s + w)/(1-\lambda)$ (Corollary 1). For drone racing this specializes to a per-gate criterion: with $R = \delta_0 + R_{\mathrm{ss}}$, if the nominal trajectory passes through every gate center and $R < r(z_2)$, then the trajectory generated by the composed learned score passes every gate. In our experiments, the rigorous R_{ss} does not clear the gate half-widths and only an empirical variant does (Appendix B), so it serves as a robustness diagnostic rather than a gate passage certificate; the factored bound $\sum_i L_i\delta_i$ is compared with the joint bound in Appendix F.

4 Experiments: Agile Aerial Robotics

Drone racing is a demanding benchmark for generalizable control: track geometry, gate-opening size, reference speed, visual venue, and gate color can all be varied independently. In practice each track is a separate engineering problem; competition teams spend days to weeks per configuration (Kaufmann et al., 2023; Song et al., 2023), and joint generalization across these axes is not standard. Section 4.1 holds out unseen track $\times$ gate-size pairs on state-based full races; a preliminary vision-based single-gate study is in Appendix C.

All our runs use a 0.5 kg, 25 cm racing-class quadrotor, simulated as a 6-DOF Newton–Euler rigid body at 600 Hz. Three diffusion policies differing only in factor conditioning are discussed in the following: ***Baseline*** (no factor input); ***Factored*** (ours: per-factor embeddings with $p_{\mathrm{drop}}{=}0.1$ null-token dropout; composed score $s_{\mathrm{comp}}{=}s_{\varnothing} + \sum_i \Delta_i$, Definition 1); ***Oracle*** (factored, trained on all tasks including held-out; architectural upper bound). Baseline and Factored never see the held-out tasks.

4.1 State-Based Full-Race Composition

The quadrotor must complete full multi-gate races (2–10 gates per track) using state-based control. The diffusion model generates a trajectory of 32 keypoints $a^j = (p_j, v_{\mathrm{speed},j}) \in \mathbb{R}^4$ per task; at inference, keypoints are linearly interpolated to 256 dense waypoints (velocity from unit tangent $\times$ speed; yaw from $\mathrm{atan2}(v_y, v_x)$) and tracked by the low-level $\mathrm{SE}(3)$ controller. Eight race tracks (race1–8) share a common gate layout but different gate sequences. Two factors define a task: track identity $z_1 \in \{\text{race1}, \ldots, \text{race8}\}$ and gate aperture (opening size) $z_2 \in \{\text{narrow } (r{=}0.3\,\mathrm{m}), \text{standard } (0.762\,\mathrm{m}), \text{wide } (1.0\,\mathrm{m})\}$. The two factors are approximately conditionally independent: z_1 sets the geometric raceline, while z_2 primarily modulates the speed profile along that path (narrower gates demand slower passage). Of $8 \times 3 = 24$ tasks, 20 are feasible; per-task max feasible speed is found by binary search. The denoiser is a 1D temporal ConvNet (4.3M parameters) with $\mathrm{Emb}(8, 64)$ for track and $\mathrm{Emb}(3, 64)$ for gate-size, each with a learned null token, trained from 50 DAgger-style rollouts per task. Six (track, gate-size) pairs are held out from training: (race1, wide), (race4, standard), (race5, wide), (race6, narrow), (race7, standard), (race8, wide). Every factor level appears in at least one training task: each track in 1–2 of its feasible gate sizes, and narrow / standard / wide in 3 / 6 / 5 training tasks respectively. Only the joint pairs are unseen. Coverage is thinnest for narrow, which is feasible on four tracks and held out on one of them; §B.3 shows that this is also where passage is weakest. Track descriptions, the speed maps, per-task results, ablations, and certificate results appear in Appendix B.

Main results. Table 1 reports closed-loop gate passage across all 20 feasible tasks (104 gates), split into training (74 gates over 14 tasks) and held-out (30 gates over 6 tasks). Every row comes from one evaluation run on the listed checkpoints, and each gate is scored independently, so a missed gate does not affect the scoring of later ones. The factored model with composed inference passes $98/104$ (94%) overall and $26/30$ (87%) held out, against $103/104$ and $30/30$ for an oracle trained directly on the held-out tasks (Figure 1). The four missed held-out gates fall on three combinations, two of them on (race6, narrow), which the oracle passes $3/3$: this is a generalization failure rather than a control limit. Narrow is the thinnest-covered aperture, feasible on four tracks and held out on one of them.

Table 1: Gate passage on full multi-gate races: trained on 14 of the 20 feasible tasks, 6 held out. One evaluation run; each gate scored independently; no row crashes. The K-network row applies the same composition rule as ours with 12 separately trained networks. The no-dropout rows are our architecture trained with $p_{\text{drop}} = 0$, receiving the full task tuple on every sample.

Method	All (104)	Training (74)	Held-out (30)
Baseline (no factor input)	19/104 (18%)	16/74 (22%)	3/30 (10%)
K-network composition (12)	61/104 (59%)	58/74 (78%)	3/30 (10%)
No dropout, joint readout	101/104 (97%)	72/74 (97%)	29/30 (97%)
No dropout, composed	94/104 (90%)	66/74 (89%)	28/30 (93%)
Factored, composed (ours)	**98/104 (94%)**	**72/74 (97%)**	**26/30 (87%)**
Factored, joint readout (ours)	**101/104 (97%)**	**72/74 (97%)**	**29/30 (97%)**
Oracle (all 20 tasks)	103/104 (99%)	73/74 (99%)	30/30 (100%)

Shared parameters plus factor dropout are the mechanism. Table 1 separates the candidate explanations. The K-network row applies the identical composition rule over twelve separately trained networks and passes 10% of held-out gates where the shared network passes 87%, so the additive formula alone does not explain the result; since it changes parameter sharing, data sharing and optimization difficulty at once, we cannot draw conclusions from it. The controlled comparison is the no-dropout network, which differs from ours only in p_{drop}: on the 14-task split it matches our joint readout exactly, $101/104$ and $29/30$. At this coverage the gains therefore come from giving one shared network factor information, not from dropout and not from test-time composition, and the 14-task split is an interpolation benchmark rather than a test of Theorem 1. Thinning coverage separates them (Table 2). At a sum-sized cover of 8 tasks in which every factor value appears, dropout raises held-out passage from 3–16% to 36–50% of the 58 uncovered gates, on two disjoint covers and at two training seeds; with a third factor z_3, the expert speed as a fraction (0.70, 0.85, 1.00) of each task's feasible maximum, 8 of the 60 feasible tuples form a sum-sized cover and dropout raises passage on the 266 uncovered gates from 5–11% to 25–32%. Composed inference never beats joint readout of the same network: it trails by three gates on the 14-task split (98 against 101 of 104) and further at the sum-sized covers (13–14 against 21–29 of 58; at $K = 3$, 46 and 48 against 74 and 72 of 266), consistent with its $(2K - 1)\eta$ score-matching term. What composition provides is the analysis: the sum-rather-than-product task budget is proved through the composed score, Theorem 2's radius factors into ODE sensitivity times a per-factor error budget only because the score is a sum of per-factor terms, and the residual is measurable per task (Proposition 1). Full K-network details are in Appendix B.6.

Table 2: Held-out gate passage as training coverage thins, joint readout. Each held-out set is every feasible combination outside the cover, so compare within a row. Covers A and B are disjoint.

Factors	Training tasks	Cover, seed	$p_{\text{drop}} = 0.1$	$p_{\text{drop}} = 0$
$K = 2$	14 of 20 (Table 1)	–, 42	29/30	29/30
	11 of 20	–, 42	27/42	21/42
	8 of 20 (sum-sized)	A, 42	22/58	4/58
		A, 43	21/58	6/58
		B, 42	24/58	2/58
		B, 43	29/58	9/58
$K = 3$	8 of 60 (sum-sized)	A, 42	74/266	23/266
		A, 43	85/266	29/266
		B, 42	72/266	14/266
		B, 43	66/266	17/266

Composition under forced factor interaction. To test Assumption 1 where it fails by construction, we add a non-additive term $\alpha \cdot \text{narrowness}(z_2) \cdot \text{curvature}(z_1)$ to the expert speed profile, leaving geometry and gate positions unchanged so the success criterion is identical, and retrain at each α on the 14-task split. As α grows the measured residual ε_D rises monotonically and composed held-out passage falls, gently to $\alpha = 0.75$ and then sharply, while joint readout is unaffected (Table 3). One seed per level brackets the transition between $\alpha = 0.75$ and 1.0 without fixing a threshold.

Table 3: Injected factor interaction, 14-task split, one seed per level. ε_D is the training-task mean residual.

α	0	0.25	0.5	0.6	0.75	1.0
ε_D	0.45	0.47	0.52	0.55	0.58	0.64
Composed, held-out (of 30)	26	26	24	23	22	11
Joint readout, all (of 104)	101	101	101	101	101	101

Certificates. We instantiate Theorem 2 with the path-dependent LTV sensitivity of Lemma 2, obtaining finite task-wise sensitivity bounds that are six orders of magnitude tighter than the uniform Grönwall bound. The resulting norm-based tubes remain conservative relative to the physical gate margins, so we treat them as a robustness diagnostic rather than a deployment-level certificate. Further details on empirical measurements and residual-gap analysis are in Appendix B.

Vision-based single-gate traversal (Appendix C). A jointly factor-conditioned diffusion policy, trained without factor dropout and driving from a single onboard camera and a noisy gyro, reproduces the state-based racing picture on a four-factor task space. Within a venue, where every factor value is seen and only their combinations are held out (the setting of Theorem 1), factor input adds little: 59% against 58% held-out success for a baseline without it, the same interpolation regime as the state-based experiments reported above. Under distribution shift it adds a lot: 63% against 42% on the training venues and 37% against 25% on a venue absent from training, the latter a factor *value* the policy never saw and so outside the theorem's coverage. As in the state-based task, the crash rate is governed by training length rather than by factor dropout, which at matched early stopping leaves crashes unchanged and costs success.

5 Related Work

Diffusion Policy (Chi et al., 2023) and its multi-task, 3D, and flow-matching extensions (Reuss et al., 2024; Ze et al., 2024; Black et al., 2025) treat task variation as a monolithic conditioning input. Compositional diffusion in image generation (Liu et al., 2022; Du et al., 2023; Ho & Salimans, 2021; Wang et al., 2023) established the additive score identity we adapt to control. Several recent works also compose diffusion policies for robotics, each on a different compositional axis: heterogeneous data sources (PoCo, Wang et al., 2024), test-time policy mixtures (GPC, Cao et al., 2026), latent behavioral modes (FDP, Liu et al., 2026), input modalities (Modality-Composable DP, Cao et al., 2025), or base motion skills for few-shot adaptation (DSE, Patil et al., 2024). All combine K separately trained networks at inference, whereas we use a single shared score network with per-factor null-token dropout, named factors with held-out combination generalization, and a closed-loop trajectory-tube certificate.

Certified-robustness frameworks (Fazlyab et al., 2019; Everett et al., 2021; Astorga et al., 2023) treat the policy monolithically; sequential skill composition (Liang et al., 2024; Mishra et al., 2023) operates across time rather than across factors; champion-level drone racing (Kaufmann et al., 2023; Song et al., 2023; Foehn et al., 2021) tunes per track; domain randomization (Tobin et al., 2017; Sadeghi & Levine, 2017) and task-conditioned multi-task RL randomize or condition over factors at training without explicit per-factor composition. Extended discussions are in Appendix E.

6 Limitations

All experiments are in simulation and within a single robotic domain (drone racing); transfer of the method to manipulation or to real hardware is not yet established. The analysis rests on Assumption 1: the benchmark factors were chosen to be approximately conditionally independent. The measured residual roughly doubles on held-out tasks, and under an injected interaction composed inference degrades while joint readout does not (Table 3); a domain whose factors interact strongly by nature is not tested. The trajectory-tube certificate should be read as a robustness diagnostic rather than a closed task-success guarantee: the rigorous steady-state radius clears no gate half-width (Appendix B.9), and only the empirical variant does. The certificate moreover closes only for the state-based full-race policy and does not transfer to the vision-based single-gate policy: a gap that is structural to operator-norm path analysis on image-conditioned scores. Closing it likely requires a non-norm-based, manifold-aware sensitivity analysis which we leave to future work.

AI USE STATEMENT

In this work we used generative AI tools (LLM-based coding assistants) for the following tasks: designing and providing feedback on experiments, implementing experiments, interpreting the experimental results, drafting and editing parts of the paper, creating figures, identifying and summarizing related literature. We did not use generative AI tools to develop the theoretical framework or to formulate or prove Theorems. Translation, dataset cleaning and qualitative analysis are not applicable to this work.

We have reviewed all AI-assisted work. Every reference was checked individually against its published record before inclusion. Every number reported in the paper is produced by a released script from a released checkpoint under a fixed seed, and the central results were independently re-run by a second author on a separate machine. AI-assisted parts of the code were reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The supplementary material contains the code for data generation, training, evaluation, and the certificate and decomposition-error measurements, together with the trained checkpoints and the scene assets for the vision benchmark. The expert demonstrations are not included because of their size, but they can be regenerated with the included data-collection scripts. The supplementary documents the commands and settings used for the main experiments; further experimental details are given in Section 4 and the appendix. Data generation, training, and evaluation are seeded, and results we report across multiple seeds are stated as such. Because the policy is evaluated in closed loop, small numerical differences in network inference across GPUs and software versions can change individual rollouts; the central results were independently re-run by a second author on a separate machine and reproduced the reported findings.

ETHICS STATEMENT

All experiments are in simulation. Autonomous-flight policies that generalize from fewer training tasks have potential positive applications in inspection, search-and-rescue and delivery, and dual-use risks in surveillance and weaponised aircraft that are common to autonomous-aviation research. Standard deployment-stage mitigations such as geofencing, operator authentication and regulatory compliance apply and are outside the scope of this simulation-only work. No human subjects or personal data are involved.

REFERENCES

David Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

Angello Astorga, Chiao Hsieh, P. Madhusudan, and Sayan Mitra. Perception contracts for safety of ml-enabled systems. *Proc. ACM Program. Lang.*, 7(OOPSLA2), October 2023. doi: 10.1145/3622875. URL https://doi.org/10.1145/3622875.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 36916–36936, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9edce07129d2ec4c511a8783eb3d81b5-Paper-Conference.pdf.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010.

Jiahang Cao, Qiang Zhang, Hanzhong Guo, Jiaxu Wang, Hao Cheng, and Renjing Xu. Modality-composable diffusion policy via inference-time distribution-level composition. *arXiv preprint arXiv:2503.12466*, 2025.

Jiahang Cao, Yize Huang, Hanzhong Guo, Qiang Zhang, Rui Zhang, Weijian Mai, Mu Nan, Jiaxu Wang, Hao Cheng, Jingkai SUN, Gang Han, Wen Zhao, Yijie Guo, Qihao Zheng, Xiao Li, Chunfeng Song, Ping Luo, and Andrew Luo. Compose your policies! improving diffusion-based or flow-based robot policies via test-time distribution-level composition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TnLFRhLuZ6.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling Is as Easy as Learning the Score: Theory for Diffusion Models with Minimal Data Assumptions. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/chen2023iclr-sampling/.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/du23a.html.

Michael Everett, Golnaz Habibi, Chuangchuang Sun, and Jonathan P. How. Reachability analysis of neural feedback loops. *IEEE Access*, 9:163938–163953, 2021.

Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Locally optimal reach set over-approximation for nonlinear systems. In *Proceedings of the 13th International Conference on Embedded Software*, pp. 1–10, 2016.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/95e1533eb1b20a97777749fb94fdb944-Paper.pdf.

Philipp Foehn, Angel Romero, and Davide Scaramuzza. Time-optimal planning for quadrotor waypoint flight. *Science robotics*, 6(56):eabh1221, 2021.

Sachit Gaudi, Gautam Sreekumar, and Vishnu Boddeti. Coind: Enabling logical compositions in diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 32159–32191, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Geometric tracking control of a quadrotor uav on se (3). In *49th IEEE conference on decision and control (CDC)*, pp. 5420–5425. IEEE, 2010.

Yangge Li, Chenxi Ji, Jai Anchalia, Yixuan Jia, Benjamin C. Yang, Daniel Zhuang, and Sayan Mitra. Lyapunov perception contracts for operating design domains. In *Proceedings of the 7th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 283 of *Proceedings of Machine Learning Research*, pp. 1053–1065, 2025.

Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16467–16476, June 2024.

Chaoqi Liu, Haonan Chen, Sigmund H. Høeg, Shaoxiong Yao, Yunzhu Li, Kris Hauser, and Yilun Du. Flexible multitask learning with factorized diffusion policy. *IEEE Robotics and Automation Letters*, 11(4):4697–4704, 2026. doi: 10.1109/LRA.2026.3664611.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 423–439, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19790-1.

Winfried Lohmiller and Jean-Jacques E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34 (6):683–696, 1998.

Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2905–2925. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/mishra23a.html.

Omkar Patil, Anant Sah, and Nakul Gopalan. Composing diffusion policies for few-shot learning of movement trajectories. *arXiv preprint arXiv:2410.17479*, 2024.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal Diffusion Transformer: Learning Versatile Behavior from Multimodal Goals. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.121.

Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. In *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017. doi: 10.15607/RSS.2017.XIII.034.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yunlong Song, Angel Romero, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Science Robotics*, 8(82):eadg1462, 2023.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Lirui Wang, Jialiang Zhao, Yilun Du, Edward Adelson, and Russ Tedrake. PoCo: Policy Composition from and for Heterogeneous Robot Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.127.

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for (score-based) text-controlled generative models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35331–35349. Curran Associates, Inc., 2023. doi: 10.52202/075280-1535. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6f125214c86439d107ccb58e549e828f-Paper-Conference.pdf.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.067.

A Formal Assumptions and Proofs

This appendix collects the full formal statement of the cascade-contraction assumption and the proofs of all results stated in the main paper. Each result is restated for self-containedness; numbering refers to the main-paper labels.

A.1 CASCADE CONTRACTION ASSUMPTION

Assumption 2 (Cascade contraction)**.** The closed-loop system consisting of the tracking controller κ and the plant dynamics f is contracting: for any two command sequences $\{a_t\}$ and $\{a'_t\}$ and corresponding state trajectories $\{x_t\}$ and $\{x'_t\}$ satisfying $x_{t+1} = f(x_t, \kappa(x_t, a_t))$,

$$\|x_{t+1} - x'_{t+1}\| \leq \lambda \|x_t - x'_t\| + B_\kappa \|a_t - a'_t\| + w, \tag{12}$$

where $\lambda \in [0, 1)$ is the per-step contraction rate, $B_\kappa > 0$ is the command-to-state gain, and $w \geq 0$ is the per-step disturbance from sensor noise and unmodeled dynamics.

This assumption is a property of the tracking controller and plant, independent of how the commands a_t are generated (by a diffusion policy, a hand-tuned planner, or any other source). Contraction ensures that state deviations caused by command perturbations remain bounded rather than accumulating over time. On its own, contraction does not yield a useful certificate: it bounds state deviation in terms of *command* perturbation $\|a_t - a'_t\|$, but says nothing about how large that perturbation is or how it depends on the task. The factored diffusion policy fills this gap by decomposing the command perturbation into per-factor contributions $\sum L_i \delta_i$, each of which can be bounded independently. The certificate (Theorem 2) chains the two: factorization controls the commands, contraction propagates commands to states.

A.2 PER-FACTOR LIPSCHITZ SCORE

Assumption 3 (Per-factor Lipschitz score)**.** For each factor $i \in \{1, \ldots, K\}$,

$$\|s_\theta(\cdot, z_i) - s_\theta(\cdot, z'_i)\| \leq L_i \|z_i - z'_i\| \tag{13}$$

for all $z_i, z'_i \in \mathcal{Z}_i$ and all (a_σ, σ, o), where L_i is the Lipschitz constant of the i-th conditioning branch.

This is a mild regularity property of the trained network rather than a substantive task hypothesis. For finite discrete factor spaces $\mathcal{Z}_i$ (such as the eight tracks and three gate-opening sizes used in our experiments) the right-hand side of (13) is automatically finite: L_i is the maximum-over-pairs ratio $\max_{z_i \neq z'_i} \|s_\theta(\cdot, z_i) - s_\theta(\cdot, z'_i)\| / \|z_i - z'_i\|$ and is computed exactly by evaluating all factor pairs. For continuous factor spaces the bound follows from the network architecture: each z_i enters the network through a learned embedding followed by Lipschitz operations (linear layers with bounded weights, layer normalization, and standard nonlinearities), whose composition is Lipschitz in the input. What matters operationally is the *magnitude* of L_i, since this controls the per-factor term $L_i \delta_i$ in the tube radius of Theorem 2; we measure L_1, L_2 directly in Table 9.

A.3 IDENTIFIABILITY VIA FACTOR DROPOUT

The decomposition $s_{\text{comp}} = s + \sum_i \Delta_i$ (Definition 1) is not uniquely determined by training data alone. For any constants $c_i \in \mathbb{R}$ with $\sum_i c_i = 0$, the substitution $\tilde{s} := s + \sum_i c_i$, $\tilde{\Delta}_i := \Delta_i - c_i$, leaves the composed score $s + \sum_i \Delta_i$ unchanged but redistributes mass between the unconditional baseline and the per-factor corrections. Without an additional constraint, a learned s_θ could absorb arbitrary content into s that ought to live in Δ_i, in which case Theorem 1's $2\sqrt{GM}$ bound on the true compositional gap would not transfer to s_θ^{comp}.

Factor dropout pins the decomposition. We resolve this in training by independently replacing each factor conditioning z_i with a learned null token $\varnothing_i$ with probability p_{drop}, so that during training the network sees all 2^K subsets of factors. The unconditional score $s_\theta(\cdot, \varnothing_1, \ldots, \varnothing_K)$ is then trained to match the score of $p(a \mid o)$ marginalized over all factors (the all-null behavior), and each single-factor score $s_\theta(\cdot, \varnothing_1, \ldots, z_i, \ldots, \varnothing_K)$ is trained to match the score of $p(a \mid o, z_i)$, marginalized over the others. This pins both s and each $\Delta_i = s(\cdot, z_i) - s(\cdot)$ to their information-theoretic targets and removes the constant ambiguity above. The dropout probability p_{drop} plays the same role as in classifier-free guidance (Ho & Salimans, 2021); we use $p_{\text{drop}} = 0.1$ throughout. Without factor dropout, the per-factor corrections Δ_i measured from s_θ are not the targets to which Theorem 1 applies, and the $2\sqrt{GM}$ bound would not transfer to the learned model.

A.4 Proof of Theorem 1 (decomposition error bound)

Theorem (Decomposition error bound, Theorem 1 restated)**.** *Suppose Assumption 1 holds and for every $\sigma \in [0, \sigma_{\max}]$, the interaction log-ratio g_σ (2) is twice differentiable in a_σ with $\|\nabla^2_{a_\sigma} g_\sigma(z, a_\sigma, o)\|_{\mathrm{op}} \le M$ uniformly in σ, a_σ, o, z. Then $\|s(a_\sigma, \sigma, o, z) - s_{\mathrm{comp}}(a_\sigma, \sigma, o, z)\| \le 2\sqrt{GM}$ for all a_σ, σ, o, and $z = (z_1, \ldots, z_K)$.*

Proof. Step 1 (gradient bound on g_σ). Fix $\sigma \in [0, \sigma_{\max}]$ and (z, a_σ, o); let $u := \nabla_{a_\sigma} g_\sigma(z, a_\sigma, o)$; assume $\|u\| > 0$ (the bound is trivial otherwise). With $v := u/\|u\|$ and the slice $\phi(t) := g_\sigma(z, a_\sigma + tv, o)$,

$$\begin{aligned}
\phi(t) &= \phi(0) + t\,\phi'(0) + \int_0^t (t-s)\,\phi''(s)\,ds && \text{[Taylor, integral remainder]}\\
&\ge \phi(0) + t\,\|u\| - \tfrac{1}{2}Mt^2 && [|\phi''| \le M;\ \phi'(0) = \|u\|]\\
&\ge -G + t\,\|u\| - \tfrac{1}{2}Mt^2. && [|\phi(0)| \le G \text{ by Assumption 1}]
\end{aligned}$$

Since $\phi(t) \le G$ as well, $\|u\| \le 2G/t + Mt/2$ for every $t > 0$; minimizing at $t^* = 2\sqrt{G/M}$ gives $\|\nabla_{a_\sigma} g_\sigma(z, a_\sigma, o)\| \le 2\sqrt{G\,M}$ uniformly in σ, z, a_σ, o.

Step 2 (score decomposition identity). For all (a_σ, σ, o, z), suppress these arguments with $(\cdot)$ for brevity. For any conditioning variable c, Bayes' rule gives $p_\sigma(a_\sigma \mid o, c) = p_\sigma(a_\sigma \mid o) \cdot p_\sigma(c \mid a_\sigma, o)/p_\sigma(c \mid o)$. Since $p_\sigma(c \mid o)$ does not depend on a_σ, taking ∇_{a_σ} log yields the score decomposition identity

$$s(\cdot, c) = s(\cdot) + \nabla_{a_\sigma} \log p_\sigma(c \mid a_\sigma, o). \tag{14}$$

Then:

$$\begin{aligned}
s(\cdot, z) - s_{\mathrm{comp}} &= \big[s(\cdot) + \nabla_{a_\sigma} \log p_\sigma(z \mid a_\sigma, o)\big] - s_{\mathrm{comp}} && \text{[(14) with } c = z]\\
&= \nabla_{a_\sigma} \log p_\sigma(z \mid a_\sigma, o) - \sum_{i=1}^{K} \Delta_i && \text{[Def. 1; cancel } s(\cdot)]\\
&= \nabla_{a_\sigma} \log p_\sigma(z \mid a_\sigma, o) - \sum_{i=1}^{K} \nabla_{a_\sigma} \log p_\sigma(z_i \mid a_\sigma, o) && \text{[Def. 1; (14) with } c = z_i]\\
&= \nabla_{a_\sigma}\, g_\sigma(z, a_\sigma, o). && \text{[interaction log-ratio (2), at noise level } \sigma]
\end{aligned}$$

Taking norms and applying Step 1, $\|s(\cdot, z) - s_{\mathrm{comp}}\| = \|\nabla_{a_\sigma} g_\sigma(z, a_\sigma, o)\| \le 2\sqrt{G\,M}$. □

Remark 1 (Lifting the bounded-interaction assumption to noise levels). Assumption 1 and Theorem 1's curvature condition are stated on the noised log-ratio g_σ because the proof above invokes Bayes' rule on p_σ and reads the gap $s(\cdot, z) - s_{\mathrm{comp}}$ as $\nabla_{a_\sigma} g_\sigma$ at every noise level the reverse ODE traverses. A clean-action bound $|g_0| \le G$ does not in general imply $|g_\sigma| \le G$ for $\sigma > 0$: Gaussian convolution alters the conditional densities $p_\sigma(z \mid a_\sigma, o)$ relative to $p(z \mid a, o)$, and g_σ interpolates between g_0 (at $\sigma \to 0$) and a fully marginalized log-ratio over observations alone (at $\sigma \to \sigma_{\max}$, where action information is washed out). Two grounds for why the lifted bound is realistic in our setting:

(i) Closed form for deterministic experts. Section 3.3 establishes $p(a_0 \mid o, z) = \delta(a_0 - a_0^\star(o, z))$, so $p_\sigma(a_\sigma \mid o, z) = \mathcal{N}(\alpha_\sigma a_0^\star(o, z), \sigma^2 I)$ and the noised posterior $p_\sigma(z \mid a_\sigma, o)$ is a softmax over Gaussian likelihoods centered at the per-task expert actions $\{\alpha_\sigma a_0^\star(o, z)\}_z$. The log-ratio g_σ then has a closed form, and uniform boundedness in σ reduces to a finite-diameter condition on the expert-action set $\{a_0^\star(o, z) : z \in \prod_i \mathcal{Z}_i\}$ for each o, a standing requirement of bounded-action policies.

(ii) Empirical evidence. The decomposition error $\varepsilon_D = \|s_{\mathrm{joint}} - s_{\mathrm{comp}}\|$ measured on the trained network (Table 9) estimates $\|\nabla_{a_\sigma} g_\sigma\|$ at the noised (a_σ, σ, o) encountered during sampling (Proposition 1), together with the network's own score error. Theorem 1 bounds that quantity by $2\sqrt{GM}$, so a small measured ε_D is consistent with the lifted assumption holding on the operating distribution but does not bound $2\sqrt{GM}$ from above. On training tasks the mean is 0.45; on held-out tasks it is 1.08.

A.5 Proof of Proposition 1 (residual identity)

Proposition (Proposition 1 restated)**.** *If the scores in Definition 1 are exact, that is* $s(a_\sigma, \sigma, o, z_i) = \nabla_{a_\sigma} \log p_\sigma(a_\sigma \mid o, z_i)$ *and* $s(a_\sigma, \sigma, o) = \nabla_{a_\sigma} \log p_\sigma(a_\sigma \mid o)$*, then for all* a_σ, σ, o, z*,* $s(a_\sigma, \sigma, o, z) - s_{\mathrm{comp}}(a_\sigma, \sigma, o, z) = \nabla_{a_\sigma} g_\sigma(z, a_\sigma, o)$*.*

Proof. By Bayes' rule, $\log p_\sigma(a_\sigma \mid o, z) = \log p_\sigma(z \mid a_\sigma, o) + \log p_\sigma(a_\sigma \mid o) - \log p_\sigma(z \mid o)$, and the last term does not depend on a_σ. Hence $s(\cdot, z) = \nabla_{a_\sigma} \log p_\sigma(z \mid a_\sigma, o) + s(\cdot)$, and in the same way $\Delta_i = \nabla_{a_\sigma} \log p_\sigma(z_i \mid a_\sigma, o)$. Subtracting (6) leaves ∇_{a_σ} of the logarithm of the ratio in (2). □

A.6 Proof of Lemma 1 (ODE sensitivity)

Lemma (ODE sensitivity, Lemma 1 restated)**.** *Let* $s, s' : \mathcal{A} \times [0, \sigma_{\max}] \to \mathcal{A}$ *be two score fields with (a)* $\|s - s'\|_\infty \le \varepsilon$ *and (b)* s *Lipschitz in its first argument with rate* $L_a(\sigma)$*. Let* $\{a_\sigma\}, \{a'_\sigma\}$ *be denoising paths generated by the reverse ODE* (4) *from a shared initial noise* $a_{\sigma_{\max}} = a'_{\sigma_{\max}}$*. Then* $\|a_0 - a'_0\| \le C_{\mathrm{ode}}\, \varepsilon$*, with* C_{ode} *given by* (5)*.*

Proof. Define the error along the denoising path: $e(\sigma) = a_\sigma - a'_\sigma$. By the shared initial noise, $e(\sigma_{\max}) = 0$. The reverse ODE (4) gives

$$\frac{de}{d\sigma} = f_{\mathrm{dr}}(\sigma)\, e(\sigma) + \tilde{g}(\sigma)\big[s(a_\sigma, \sigma) - s'(a'_\sigma, \sigma)\big].$$

Split the score difference:

$$s(a_\sigma, \sigma) - s'(a'_\sigma, \sigma) = \underbrace{\big[s(a_\sigma, \sigma) - s(a'_\sigma, \sigma)\big]}_{\text{same field, different points: } \le L_a(\sigma)\, \|e\|} + \underbrace{\big[s(a'_\sigma, \sigma) - s'(a'_\sigma, \sigma)\big]}_{\text{different fields, same point: } \le \varepsilon}.$$

Setting $\mu(\sigma) = |f_{\mathrm{dr}}(\sigma)| + |\tilde{g}(\sigma)|\, L_a(\sigma)$:

$$\Big\| \frac{de}{d\sigma} \Big\| \le \mu(\sigma)\, \|e\| + |\tilde{g}(\sigma)|\, \varepsilon.$$

This is a linear differential inequality with forcing term, integrated backward from $\sigma_{\max}$ (where $\|e\| = 0$) to $\sigma = 0$. Grönwall's inequality gives

$$\|e(\sigma)\| \le \int_\sigma^{\sigma_{\max}} |\tilde{g}(\sigma')|\, \varepsilon\, \exp\Big(\int_\sigma^{\sigma'} \mu(\sigma'')\, d\sigma'' \Big)\, d\sigma'.$$

Evaluating at $\sigma = 0$ yields $\|a_0 - a'_0\| = \|e(0)\| \le C_{\mathrm{ode}} \cdot \varepsilon$. □

Remark 2. Unlike the standard Grönwall setting (same ODE, different initial conditions), here both denoising paths start from the *same* initial noise ($e(0) = 0$). The divergence is driven entirely by the score-field mismatch ε, which enters as a forcing term. The Lipschitz constant L_a then amplifies this forcing through feedback: the two paths drift apart, causing the *same* score field to evaluate differently at the two locations, which further increases the drift. The exponential in C_{ode} (5) reflects this feedback amplification.

A.7 Path-dependent LTV sensitivity (Lemma 2)

DDIM linearization. Discretize the reverse ODE (4) with K DDIM steps. Let $\bar{a}_k$ ($k = 0, 1, \ldots, K$) be the nominal denoising trajectory from $a^\star$ at noise levels $\sigma_0 = \sigma_{\max} > \sigma_1 > \cdots > \sigma_K = 0$. Each step is

$$a_{k+1} \;=\; c_1(k)\, a_k \;+\; c_2(k)\, \hat{\varepsilon}_\theta(a_k, \sigma_k), \tag{15}$$

where c_1, c_2 are scheduler-determined coefficients (for DDIM: $c_1(k) = \sqrt{\bar{\alpha}_{k+1}/\bar{\alpha}_k}$, $c_2(k) = \sqrt{1 - \bar{\alpha}_{k+1}} - \sqrt{\bar{\alpha}_{k+1}(1 - \bar{\alpha}_k)/\bar{\alpha}_k}$, in the notation of Song et al. (2021a)). The eps-prediction $\hat{\varepsilon}_\theta$ and the score s are related by $\hat{\varepsilon}_\theta(x, \sigma) = -\sqrt{1 - \bar{\alpha}_\sigma}\, s(x, \sigma)$, so the score-mismatch hypothesis $\|s - s'\|_\infty \le \varepsilon$ of Lemma 1 transfers to $\|\hat{\varepsilon}_\theta - \hat{\varepsilon}'_\theta\|_\infty \le \varepsilon$ (since $\sqrt{1 - \bar{\alpha}_\sigma} \le 1$). We state and measure the LTV bound below in $\hat{\varepsilon}_\theta$-form throughout, matching standard DDIM implementations. Linearizing around the nominal yields per-step Jacobians

$$M_k \;=\; c_1(k)\, I \;+\; c_2(k)\, J_k, \qquad J_k := \partial \hat{\varepsilon}_\theta / \partial a \big|_{\bar{a}_k, \sigma_k}, \tag{16}$$

and the discrete state-transition matrix from step k to K is

$$\Phi_{k,K} \;:=\; M_{K-1}M_{K-2}\cdots M_k, \qquad \Phi_{K,K} := I. \tag{17}$$

Lemma 2 (Path-dependent LTV sensitivity)**.** *Under the assumptions of Lemma 1 and the shared-seed condition $a_{\sigma_{\max}} = a'_{\sigma_{\max}} = a^\star$, and to first order in ε along the nominal path $\{\bar{a}_k\}_{k=0}^K$,*

$$\|a_K - a'_K\| \;\le\; C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a})\,\varepsilon \;+\; O(\varepsilon^2), \qquad C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a}) \;=\; \sum_{k=0}^{K-1} |c_2(k)|\,\|\Phi_{k+1,K}\|_2. \tag{18}$$

(*Indexing convention.*) The K reverse-sampling steps are indexed so that $k = 0$ is the noisy start ($\sigma_0 = \sigma_{\max}$) and $k = K$ is the clean end ($\sigma_K = 0$), matching the order in which DDIM evaluates them. The clean-end action error $\|a_K - a'_K\|$ in this lemma therefore corresponds to $\|a_0 - a'_0\|$ in the continuous-time Lemma 1, where $a_0 = a_{\sigma=0}$ is the clean action by the standard convention.

Proof. Let $e_k := a_k - a'_k$, so $e_0 = 0$ by the shared initial condition. Define the per-step eps-prediction forcing $\Delta\hat{\varepsilon}_k := \hat{\varepsilon}_\theta(a'_k, \sigma_k) - \hat{\varepsilon}'_\theta(a'_k, \sigma_k)$ and $b_k := c_2(k)\,\Delta\hat{\varepsilon}_k$; via the bridge $\hat{\varepsilon}_\theta = -\sqrt{1-\bar{\alpha}_\sigma}\,s$, the score-field mismatch assumption $\|s - s'\| \le \varepsilon$ gives $\|\Delta\hat{\varepsilon}_k\| \le \sqrt{1-\bar{\alpha}_k}\,\varepsilon \le \varepsilon$, and hence $\|b_k\| \le |c_2(k)|\,\varepsilon$.

$$\begin{aligned}
e_{k+1} &= M_k\,e_k + b_k + O(\|e_k\|^2) && \text{[linearize (15) around } \bar{a}_k\text{]}\\
e_K &= \sum_{k=0}^{K-1} \Phi_{k+1,K}\,b_k \;+\; O(\varepsilon^2) && \text{[discrete variation of constants from } e_0 = 0\text{;}\\
&&& \quad b_k = O(\varepsilon) \text{ absorbs the nonlinear remainder]}\\
\|e_K\| &\le \sum_{k=0}^{K-1} \|\Phi_{k+1,K}\|_2\,\|b_k\| + O(\varepsilon^2) && \text{[triangle inequality, then submultiplicativity of } \|\cdot\|_2\text{]}\\
&\le \sum_{k=0}^{K-1} |c_2(k)|\,\|\Phi_{k+1,K}\|_2\,\varepsilon + O(\varepsilon^2) && \text{[apply } \|b_k\| \le |c_2(k)|\,\varepsilon\text{].}
\end{aligned}$$

□

Remark 3 (Grönwall as a further relaxation). Applying the submultiplicativity inequality $\|\Phi_{k+1,K}\|_2 \le \prod_{j=k+1}^{K-1}\|M_j\|_2$ in (18) together with the triangle inequality $\|M_j\|_2 \le |c_1(j)| + |c_2(j)|\,L_a(\sigma_j)$ yields the discrete analog of the continuous Grönwall bound in Lemma 1; passing to the continuous-time limit recovers $C_{\mathrm{ode}}^{\mathrm{ana}}$ in (5) exactly. Hence $C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a}) \le C_{\mathrm{ode}}^{\mathrm{ana}}$, with strict inequality when (i) inter-step cancellation occurs (matrix product tighter than product of norms) and (ii) Jacobian structure is not captured by the scalar Lipschitz constant (discrete $\sigma_{\max}$ tighter than $L_a(\sigma_j)$). Both effects are typically present: Section 4.1 reports $C_{\mathrm{ode}}^{\mathrm{ana}} \approx 10^5$–$10^6$ (vacuous) versus $C_{\mathrm{ode}}^{\mathrm{LTV}} \approx 2$–$10$ on the $K = 50$ DDIM chain of our factored model, five to six orders of magnitude tighter.

Remark 4 (Remaining gap: nonlinear manifold contraction). Lemma 2 is a linearized worst-case bound: it (i) drops the $O(\varepsilon^2)$ Taylor remainder and (ii) assumes the score-error direction Δs_k aligns with $\Phi_{k+1,K}$'s top singular direction at every step. The linearization drop is an unavoidable feature of any sensitivity analysis; it is controlled as long as $\varepsilon \cdot C_{\mathrm{ode}}^{\mathrm{LTV}}$ remains small compared to the local radius of linearization. The direction-alignment assumption is *conservative*: empirically, Δs_k does not align with $\Phi_{k+1,K}$'s worst direction across steps, and the bound overestimates the measured action gap by a factor of order 10^1–10^2 (Section 4.1). Closing this gap would require either distributional assumptions on Δs_k (stochastic Grönwall) or an analysis that exploits the *manifold-contractive* character of the learned score field: empirically, the denoising ODE is more contractive along the data manifold than its Jacobian predicts, because the score field itself points back toward the data manifold. Neither refinement is pursued in this paper; we report $C_{\mathrm{ode}}^{\mathrm{LTV}}$ as the best worst-case bound achievable by norm-based path analysis and separately report the empirical amplification in Section 4.1.

A.8 Proof of Theorem 2 (factored tube)

Theorem (Factored trajectory tube, Theorem 2 restated)**.** *Under Assumptions 1, 3, and 2, and either the scalar Lipschitz condition $L_a(\sigma)$ of Lemma 1 (hypothesis (a)) or the path-Jacobian regularity of Lemma 2 under the shared-seed condition $a_{\sigma_{\max}} = a'_{\sigma_{\max}} = a^\star$ (hypothesis (b)), the closed-loop state error satisfies the trajectory-tube bound* (11) *with $\varepsilon_s \le 2\sqrt{GM} + (2K-1)\eta + \sum_{i=1}^{K} L_i\delta_i$.*

Proof. Define $e_t = x_t - x_t^{\mathrm{nom}}$.

Step 1 (score perturbation). The gap between the true joint score at the nominal factors and the composed learned score at the perturbed factors decomposes via three triangle-inequality steps, routed through the composed scores at the nominal task z^{nom} so that Assumption 3 is invoked on the trained network rather than on the unlearned true score:

$$\begin{aligned}
&\|s(\cdot, z^{\mathrm{nom}}) - s_\theta^{\mathrm{comp}}(\cdot, z)\| \\
&\quad \le \underbrace{\|s(\cdot, z^{\mathrm{nom}}) - s^{\mathrm{comp}}(\cdot, z^{\mathrm{nom}})\|}_{\le 2\sqrt{GM}\ \text{(Thm 1)}} + \underbrace{\|s^{\mathrm{comp}}(\cdot, z^{\mathrm{nom}}) - s_\theta^{\mathrm{comp}}(\cdot, z^{\mathrm{nom}})\|}_{\le (2K-1)\eta\ \text{(equation (10))}} \\
&\qquad + \underbrace{\|s_\theta^{\mathrm{comp}}(\cdot, z^{\mathrm{nom}}) - s_\theta^{\mathrm{comp}}(\cdot, z)\|}_{\le \sum_i L_i\delta_i\ \text{(Asm 3)}}.
\end{aligned}$$

The third term expands as $s_\theta^{\mathrm{comp}}(\cdot, z^{\mathrm{nom}}) - s_\theta^{\mathrm{comp}}(\cdot, z) = \sum_{i=1}^{K}\big[s_\theta(\cdot, z_i^{\mathrm{nom}}) - s_\theta(\cdot, z_i)\big]$ because both composed scores share the unconditional baseline $s_\theta(\cdot)$; Assumption 3 on each per-factor branch then gives $\|s_\theta(\cdot, z_i^{\mathrm{nom}}) - s_\theta(\cdot, z_i)\| \le L_i\|z_i^{\mathrm{nom}} - z_i\| \le L_i\delta_i$, and the triangle inequality yields the displayed bound. Hence $\varepsilon_s \le 2\sqrt{GM} + (2K-1)\eta + \sum_i L_i\delta_i$.

Step 2 (action perturbation). Under hypothesis (a), Lemma 1 gives $\|a_t - a_t^{\mathrm{nom}}\| \le C_{\mathrm{ode}}^{\mathrm{ana}}\varepsilon_s$ uniformly over tasks. Under hypothesis (b), Lemma 2 gives the tighter per-task bound $\|a_t - a_t^{\mathrm{nom}}\| \le C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a}^{(z)})\,\varepsilon_s + O(\varepsilon_s^2)$ along the nominal denoising trajectory $\bar{a}^{(z)}$ (the $O(\varepsilon_s^2)$ remainder inherited from Lemma 2's discrete linearization is controlled by Remark 4).

Step 3 (contraction). From (12) with $d = B_\kappa C_{\mathrm{ode}}\,\varepsilon_s + w$: $\|e_{t+1}\| \le \lambda\,\|e_t\| + d$. Unrolling: $\|e_t\| \le \lambda^t\delta_0 + d/(1-\lambda)$ (under hypothesis (b), the $O(\varepsilon_s^2)$ from Step 2 propagates additively through the per-step disturbance and the unrolled bound carries an $O(\varepsilon_s^2)$ correction in the steady-state term). □

B Full-Race Experiments: Detailed Results

This appendix collects the per-task numbers, ablations, and certificate diagnostics behind the headline §4.1 results. Subsections are organized roughly in the order of the main-text claims: tracks and reference speeds (§B.2), per-task and per-track outcomes (§B.3), the composed-vs-joint comparison (§B.4), score-field diagnostics (§B.5), the K-network baseline's training pipeline (§B.6), the DDIM step sweep (§B.7), the path-dependent LTV bound (§§B.8–B.10), and the 10-seed robustness study (§B.11) and residual-gap diagnostic (§B.12).

B.1 Training hyperparameters and compute

Both the state-based full-race policy (Section 4.1) and the vision-based single-gate policy (Appendix C) share a common training recipe except where noted. Diffusion uses a DDPM cosine schedule with 100 noise steps for training and 50-step DDIM at inference, AdamW with learning rate 10^{-4}, batch size 32, and 1500 epochs. All training and evaluation runs use a workstation with one NVIDIA GeForce RTX 4080 SUPER (16 GB VRAM), an AMD Ryzen 9 7900X CPU (12 cores, 24 threads), and 64 GB system RAM, running CUDA 12.1 with PyTorch 2.4.1. Wall-clock time: state-based training is ~ 30 min per model; vision-based training runs ~ 2 h per DAgger round and ~ 50 epochs total; the 12-network composition baseline of Section 4.1 (Appendix B.6) totals ~ 40 min across all networks. All path-Jacobian and seed-sweep certificate measurements (Tables 11 and 13) run on the same single GPU.

B.2 Race tracks and reference speed profiles

The eight race tracks share a common gate layout. Maximum feasible speeds are determined by binary search: for each candidate speed, the $\mathrm{SE}(3)$ expert flies the cubic-spline reference and every gate is checked for passage with the actual half-width $r(z_2)$. A combination is declared infeasible if no candidate speed (down to a small lower bound) clears every gate.

Table 4: Race tracks and maximum feasible speeds (m/s). "—" indicates infeasible at that gate size.

Track	Gates	Narrow	Standard	Wide	Description
race1	8	—	2.1	2.4	Complex, U-turns
race2	9	—	2.1	2.4	Corkscrew
race3	7	—	2.1	2.5	Right-to-left sweep
race4	10	—	3.3	3.5	Long winding
race5	5	2.2	3.6	3.9	Short loop
race6	3	2.3	4.1	4.6	Simple run
race7	2	8.2	10.0	10.0	Height change
race8	2	7.0	7.9	7.9	U-turn

The full task space has $8 \times 3 = 24$ tasks; 20 are feasible. Races 1–4 with narrow gates exceed the $\mathrm{SE}(3)$ controller's tracking precision and are omitted. Race 6 has the strongest gate-size speed differentiation ($2.0\times$ ratio between narrow and wide); races 7–8 are nearly speed-saturated, so the wide and standard maxima coincide.

B.3 Per-task and per-track results

Held-out task detail. Table 5 reports per-task gate passage on the six held-out (track, gate-size) pairs at $N = 50$. Pass/fail outcomes are invariant across the five DDIM seeds tested: different seeds produce different DDIM-generated reference trajectories, and the deterministic $\mathrm{SE}(3)$ controller plus plant tracks each one to a different state trajectory, but the trajectory variance across seeds stays within the gate's pass/fail margin in every case. This is evidence of robustness, not of upstream determinism.

Table 5: Held-out tasks (5 seeds, $\sigma = 0\%$). Generated mean speed vs. reference from the speed map.

		Gate passage		Speed [m/s]	
Track	Size	Factored	Oracle	Gen.	Ref.
race1	wide	**100%**	100%	2.2	2.5
race4	std	**100%**	100%	3.0	3.4
race5	wide	**100%**	100%	3.0	4.0
race6	nar	0%	0%	3.4	2.4
race7	std	**100%**	100%	5.8	9.5
race8	wide	**100%**	100%	5.9	7.9
Total		**5/6 (83%)**	5/6		

Both factored and oracle achieve 100% on 5 of 6 held-out tasks. Both fail (race6, narrow): the trained policy generates 3.4 m/s for race6 (reference: 2.4 m/s), at the edge of the SE(3) controller's ability to track slow speeds through race6's geometry. Since oracle (trained directly on race6 narrow) also fails $0/3$, the limit is a control / network-capacity constraint rather than a composition error; factored inherits the same constraint.

Speed fidelity. The gate-size score correction Δ_2 should produce monotonically increasing speeds as gate width increases. Table 6 confirms the ordering narrow $<$ standard $\leq$ wide on every track where narrow is feasible (with equality on race8, where standard and wide both generate 6.0 m/s). Generated speeds are conservative (below reference) but the relative ordering is always correct.

Per-track breakdown. Table 7 shows gate passage for the factored model per (track, gate-size), with held-out tasks in italic. Wide-gate tasks pass $45/46$ and standard $44/46$; narrow remains the

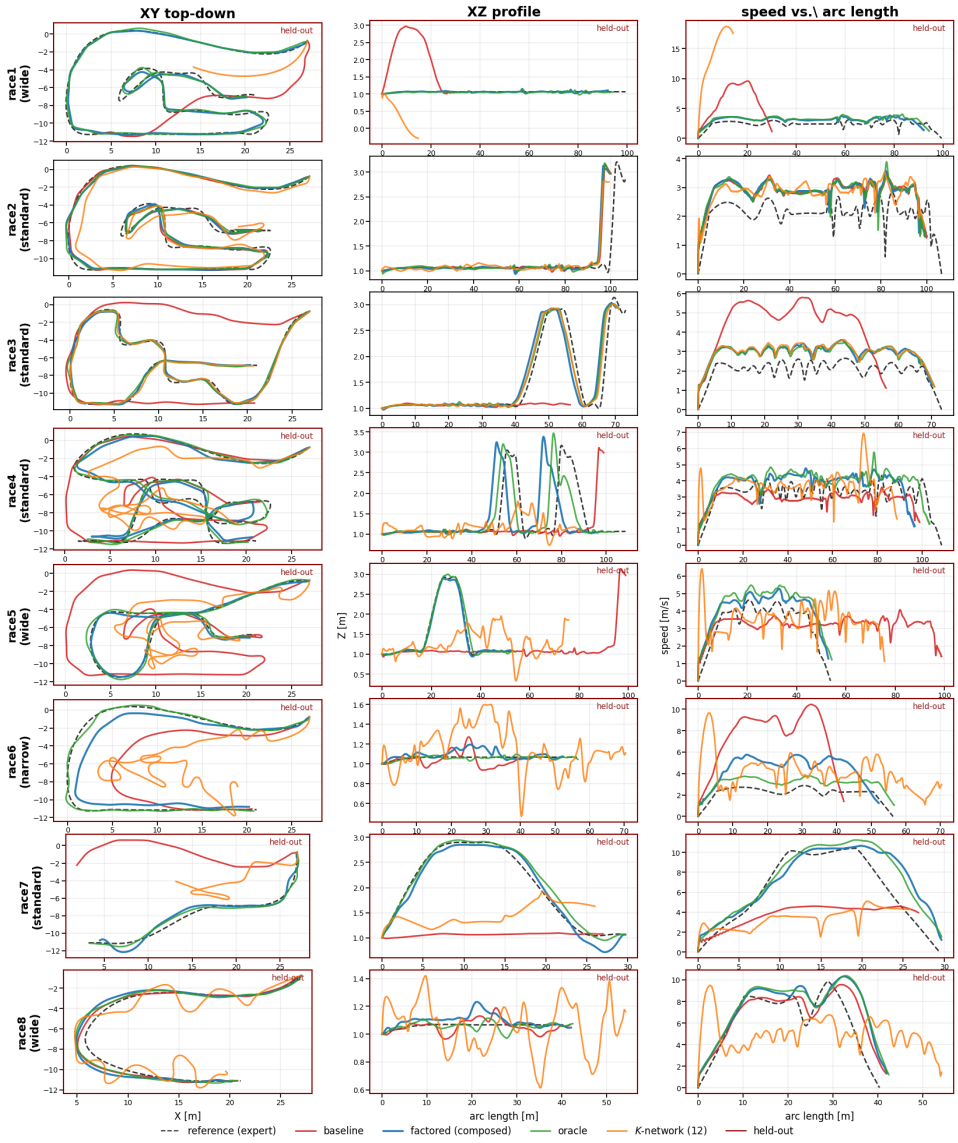

Figure 2: **Per-race closed-loop trajectories with all five methods overlaid.** One row per track; columns are XY top-down (geometric route), XZ side profile (height), and speed vs. arc length. For each race we pick the held-out joint pair where one exists (red border, "held-out" badge), else *standard* (race2, race3 have no held-out combo). Methods: expert reference (dashed black), baseline (red), factored composed (blue, ours), oracle (green), and the K-network composition baseline (orange). The factored policy tracks reference and oracle within $\sim$0.1 m on every held-out row except (race6, narrow), where the decomposition error is $3.5\times$ the gate-size sensitivity, the highest ratio of any task (Table 9). The baseline fails nearly every held-out gate ; the K-network composition produces erratic geometry and noisy speed profiles even on training tasks (race4_standard, race8_wide), corroborating Table 1.

Table 6: Generated mean speed (m/s) by gate size (factored model). Reference in parentheses.

Track	Narrow	Standard	Wide
race1–5	—	2.0–3.0 (2.2–3.7)	2.3–3.3 (2.5–4.0)
race6	3.4 (2.4)	3.5 (4.2)	3.8 (4.8)
race7	5.0 (7.7)	5.8 (9.5)	6.1 (9.5)
race8	5.3 (7.0)	6.0 (7.9)	6.0 (7.9)

bottleneck at $9/12$, with two of its three misses on race8 narrow at 7 m/s through 0.3 m gates, at the dynamic limit of the SE(3) controller. The four held-out misses fall on three tasks, (race6, narrow), (race8, wide) and (race4, standard); the oracle trained on those tasks passes all three.

Table 7: Per-track gate passage (factored, composed, corrected scoring). *Italic* = held-out. "—" = infeasible.

Track	Narrow	Standard	Wide	Total
race1	—	8/8	*8/8*	16/16
race2	—	9/9	9/9	18/18
race3	—	7/7	7/7	14/14
race4	—	*9/10*	10/10	19/20
race5	5/5	5/5	*5/5*	15/15
race6	*1/3*	3/3	3/3	7/9
race7	2/2	*2/2*	2/2	6/6
race8	1/2	1/2	*1/2*	3/6
	9/12	44/46	45/46	**98/104**

B.4 Composed vs. joint inference

The headline §4.1 numbers use *composed* inference, $s_{\text{comp}} = s_{\varnothing} + \Delta_1(z_1) + \Delta_2(z_2)$ (Definition 1, requiring 3 forward passes per DDIM step). An alternative is *joint* conditioning: pass both factor embeddings to the shared network in a single forward pass. Table 8 compares the two modes on the same factored model and the full 20-task evaluation.

Table 8: Composed vs. joint inference (factored model, all 20 feasible tasks).

Inference	Gates	Rate	Error [m]
Joint ($s(z_1, z_2)$ directly)	101/104	97%	0.21
Composed ($s_{\varnothing} + \Delta_1 + \Delta_2$)	**98/104**	**94%**	0.24

Composed inference trails joint readout by three gates on this benchmark (98 against 101 of 104; 26 against 29 of 30 held out), and by more at sparser training coverage (Table 2). We do not claim that composed inference improves gate passage. The ordering is what the error accounting predicts: the composed score sums $2K - 1$ network evaluations and carries $(2K - 1)\eta$ of score-matching error where joint readout carries η, so thinning coverage, which raises η, widens the gap. Near-equality at the 14-task coverage is what Theorem 1 predicts when the decomposition error is small relative to the per-factor sensitivities, and at fixed coverage a widening gap points to that error rather than to η; the interaction sweep of Table 3 realizes exactly that. What composed inference buys is not accuracy but structure: a per-factor error budget for the certificate (Theorem 2), the null-token control used in §C, and a measurable residual, at a cost of $K+1$ forward passes per denoising step and greater sensitivity to the step count (§B.7).

B.5 Decomposition and Lipschitz diagnostics

Theorem 1 bounds the decomposition gap by $\|s_{\text{joint}} - s_{\text{comp}}\| \le 2\sqrt{GM}$, where G bounds the pointwise interaction log-ratio and M bounds the score Hessian. Rather than estimating G and M separately, we directly measure the left-hand side: the decomposition error $\varepsilon_D := \|s_{\text{joint}} - s_{\text{comp}}\|$

on random (action, timestep) pairs across all training tasks. Since Theorem 1 gives $\varepsilon_D \leq 2\sqrt{GM}$, a measured ε_D *lower*-bounds $2\sqrt{GM}$, and constrains the product GM from below rather than certifying it from above. We report ε_D in place of G and M because it is the quantity that actually propagates into the tube radius of Theorem 2, for which $2\sqrt{GM}$ is only a conservative surrogate; measuring the gap directly avoids that conservatism.

ε_D is the interaction, not a proxy for it. When factor dropout pins the conditional targets (Appendix A.3), so that $s_\varnothing$ matches $\nabla_a \log p(a \mid o)$ and each single-factor score matches $\nabla_a \log p(a \mid o, z_i)$, the per-factor correction of Definition 1 reduces to $\Delta_i = \nabla_a \log p(z_i \mid a, o)$, since the remaining term $\nabla_a \log p(z_i \mid o)$ vanishes for having no dependence on a. Subtracting the composed score from the joint one, the unconditional term cancels and

$$s_{\text{joint}} - s_{\text{comp}} \;=\; \nabla_a \log \frac{p(z_1, \ldots, z_K \mid a, o)}{\prod_{i=1}^{K} p(z_i \mid a, o)}.$$

The decomposition error is therefore the action-gradient of the factors' total correlation given (a, o), and vanishes identically under Assumption 1. So ε_D is a direct estimator of how badly that assumption is violated, not a proxy for it, and the assumption is falsifiable on any task where the score network can be queried. This is what licenses the per-task use of ε_D as a predictor of composition failure.

Table 9: Per-task decomposition error ε_D on the trained factored model, 50 noised expert states per task drawn with a fixed seed, in the network's output units; L_2 is the gate-size sensitivity $\|s(z_2) - s(z_2')\|$ measured on the same states, and the ratio ε_D / L_2 normalizes the residual by the weaker factor's own correction. Gates: composed inference, corrected scoring.

Task	Split	ε_D mean	ε_D max	ε_D / L_2	Gates
race1_standard	train	0.49	2.75	0.46	8/8
race2_standard	train	0.35	1.11	0.96	9/9
race2_wide	train	0.37	1.41	0.69	9/9
race3_standard	train	0.28	0.62	0.22	7/7
race3_wide	train	0.33	1.68	0.20	7/7
race4_wide	train	0.62	2.12	0.75	10/10
race5_narrow	train	0.67	2.08	1.10	5/5
race5_standard	train	0.62	2.40	0.44	5/5
race6_standard	train	0.36	1.63	0.51	3/3
race6_wide	train	0.46	2.61	0.40	3/3
race7_narrow	train	0.34	1.15	0.34	2/2
race7_wide	train	0.34	1.07	0.40	2/2
race8_narrow	train	0.59	2.16	0.88	1/2
race8_standard	train	0.42	1.60	0.44	1/2
race1_wide	held-out	0.62	2.53	0.79	8/8
race4_standard	held-out	1.06	4.76	1.50	9/10
race5_wide	held-out	0.95	2.98	0.75	5/5
race6_narrow	held-out	1.26	2.64	**3.51**	1/3
race7_standard	held-out	0.97	2.83	1.12	2/2
race8_wide	held-out	1.62	5.89	1.07	1/2
mean, train		0.45		0.6	72/74
mean, held-out		1.08		1.5	26/30

Three readings. The residual roughly doubles where the assumption is extrapolated rather than interpolated, 0.45 on training tasks against 1.08 held out. The bound of Theorem 1 is uniform in the factor tuple and does not itself predict that gap; it is consistent with the network-error component of ε_D that we cannot separate out, and we report it as an observation. Normalized by the gate-size sensitivity, the worst-performing task, (race6, narrow), is the extreme at 3.5 against a held-out mean of 1.5. The measurement supports no threshold: (race8, wide) drops a gate at a ratio of 1.07 and (race4, standard) passes 9 of 10 at 1.50, so what it establishes is that the assumption is checkable per task before deployment, not that ε_D predicts passage.

B.6 K-NETWORK COMPOSITION BASELINE

The K-network row of Table 1 reports a PoCo / DSE-style baseline (Wang et al., 2024; Patil et al., 2024) adapted to our factor decomposition: rather than null-token masking of one shared network, we train $1 + 8 + 3 = 12$ separate unconditional networks: an unconditional $\varepsilon_\varnothing$ on all 14 training tasks pooled, eight track-only networks $\{\varepsilon_{\text{track}_i}\}_{i=1}^{8}$ each on data with $z_1 = \text{race}_i$, and three gate-size-only networks $\{\varepsilon_{\text{gate}_j}\}_{j=1}^{3}$ each on data with $z_2 = \text{gate}_j$. These compose at inference via

$$\varepsilon^{K\text{-net}}_{\text{comp}}(x, t; z_1, z_2) = \varepsilon_{\text{track}_{z_1}}(x, t) + \varepsilon_{\text{gate}_{z_2}}(x, t) - \varepsilon_\varnothing(x, t).$$

This is structurally identical to our shared-network composition (Definition 1); the only difference is that each $\varepsilon_\bullet$ comes from a different network rather than from null-token masking of one shared network. Architecture (4.3M-parameter ConvNet), epochs (1500), optimizer, DDIM steps, normalization stats, $\mathrm{SE}(3)$ controller, carrot tracker, and gate-passage criterion are unchanged from the shared-network setup.

The result is $3/30$ (10%) on held-out gates against $26/30$ (87%) for the shared network (Table 1). The mechanism follows from the theory. In the shared-network approach, null-token dropout (Section 3.2) regularizes the network to learn per-factor score corrections Δ_i that are *additive-compatible*: the decomposition error ε_D on noised expert states averages 0.45 on training tasks (Table 9), small relative to the per-factor sensitivities. Separately trained networks have no such regularization; each $\varepsilon_{\text{track}_i}, \varepsilon_{\text{gate}_j}, \varepsilon_\varnothing$ is the score of a different marginal, fit independently, with no incentive for additive compatibility. Their sum $\varepsilon_{\text{track}} + \varepsilon_{\text{gate}} - \varepsilon_\varnothing$ is not constrained to approximate any joint score and, off-distribution, drifts arbitrarily.

On training tasks the K-network reaches $58/74$ (78%) against $72/74$, with its losses on race1_standard ($4/8$) and race6_standard ($2/3$), so the gap is concentrated on held-out combinations: no single network ever saw a (z_1, z_2)-conditioned target, and a sum of independently fit marginal scores is not constrained to approximate the joint score of a pair it never observed. The shared network avoids this because the joint training pairs provide a learning signal that the per-factor corrections must collectively reproduce. We note that this baseline changes parameter sharing, data sharing and optimization difficulty at once, which is why the controlled comparison in Section 4.1 rests on the no-dropout network rather than on this row.

B.7 DDIM STEP SWEEP

The headline §4.1 numbers use $N = 50$ DDIM steps. The trajectory-tube bound of Theorem 2 is driven by C_{ode} (eq. (5)), the amplification of per-step score-field mismatch into clean-action error across the reverse ODE. Grönwall's inequality yields the analytical upper bound $C_{\text{ode}} = \int |\tilde{g}| \exp(\int [|f_{\text{dr}}| + |\tilde{g}| L_a])\, d\sigma$, which grows exponentially with the denoising path length. We additionally measure the *empirical* amplification $C^{\text{emp}}_{\text{ode}} = \|a_0^{\text{comp}} - a_0^{\text{joint}}\| / \max_t \|s^{\text{comp}}(x_t) - s^{\text{joint}}(x_t)\|$ on the held-out composition set. Sweeping the DDIM step count $N \in \{10, 20, 30, 50\}$ (Table 10) exposes two findings.

Table 10: DDIM step sweep with the $p = 0.1$ factored checkpoint. $R^{\text{emp}}_{\text{ss}}$ plugs $C^{\text{emp}}_{\text{ode}}$ into Corollary 1 with B_κ=0.05, λ=0.9, w=0.01 (closed-loop constants defaulted, pending measurement). "Tube max" is the observed $\max_t \|x_t^{\text{comp}} - x_t^{\text{joint}}\|$ across all training tasks.

N	Comp. all	Comp. held-out	Joint all	$C^{\text{emp}}_{\text{ode}}$	$R^{\text{emp}}_{\text{ss}}$ [m]	Tube max [m]
10	88%	83%	91%	0.395	3.14	7.17
20	93%	83%	96%	0.136	1.15	3.47
30	94%	87%	96%	0.109	0.93	2.67
50	**94%**	**87%**	**97%**	**0.064**	**0.59**	**2.37**

Finding 1: empirical C_{ode} *decreases* with N, opposite to the analytical Grönwall bound. The analytical C_{ode} grows monotonically with N because each additional step adds a factor $\exp(|\tilde{g}(\sigma)| L_a(\sigma))$ to the cumulative amplification, as a worst-case over adversarial score fields. Empirically, $C^{\text{emp}}_{\text{ode}}$ drops from 0.40 at $N = 10$ to 0.06 at $N = 50$. The two composed and joint denoising chains stay close: finer discretization gives per-step mismatches more opportunity to cancel along the reverse ODE

rather than compound. The resulting empirical amplification is sub-unity, meaning the end-to-end keypoint deviation is *smaller* than the largest per-step score mismatch observed along the path. This sharply separates the certified (Grönwall) regime from the operating regime: the conservative assumption that a perturbation expands at rate L_a forever is never realized in a well-trained score model.

Finding 2: step count matters little for either mode. Compared on the same 104-gate task set, joint passage moves from 91% at $N = 10$ to 97% at $N = 50$ and composed from 88% to 94%; on held-out tasks composed goes from 83% to 87% and joint from 93% to 97%. Composed inference trails joint by three to five gates at every N, and the gap does not widen at coarse schedules. Composed inference evaluates three score fields per denoising step and sums them, so each step carries up to 3κ of approximation error (eq. (10)) against κ for joint readout, which accounts for the constant offset; the per-step budget is evidently small enough at every N tested that the reverse ODE does not drift off-manifold.

Operating point. We adopt $N = 50$ as the canonical operating point throughout the headline tables: composed passage is 94% overall and 87% on held-out tasks, with the empirical certificate radius $R_{\mathrm{ss}}^{\mathrm{emp}} = 0.59\,\mathrm{m}$ below the wide ($1.0\,\mathrm{m}$) and standard ($0.762\,\mathrm{m}$) gate half-widths. $N = 30$ gives the same composed passage at 90 network evaluations against 150, a $1.7\times$ speed-up. The empirical tube $\max_t \|x_t^{\mathrm{comp}} - x_t^{\mathrm{joint}}\| = 2.37\,\mathrm{m}$ at $N = 50$ still exceeds $R_{\mathrm{ss}}^{\mathrm{emp}}$; this gap is tracked to the defaulted values of B_κ, λ, w, which should be measured directly on the closed-loop $\mathrm{SE}(3)$ plant before the certificate is reported as tight.

B.8 Path-dependent LTV bound

Path-dependent LTV certificate via backward Lyapunov. Lemma 2 replaces the scalar per-step amplification σ_k with the spectral norm of the actual state-transition matrix $\Phi_{k+1,K} = M_{K-1}M_{K-2}\cdots M_{k+1}$ formed along a deterministic nominal DDIM trajectory $\{\bar{a}_k\}_{k=0}^{K}$ initialized from a fixed canonical $a_{\sigma_{\max}} = x^\star$. We compute this via the backward Gramian recursion $P_K = I,\ P_k = M_k^\top P_{k+1} M_k$ and form

$$C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a}) \;=\; \sum_{k=0}^{K-1} |c_2(k)|\,\sqrt{\|P_{k+1}\|_2} \;=\; \sum_{k=0}^{K-1} |c_2(k)|\,\|\Phi_{k+1,K}\|_2.$$

The Jacobians $J_k = \partial \varepsilon_\theta^{\mathrm{comp}} / \partial x|_{\bar{a}_k,\sigma_k}$ are computed by `torch.func.jacrev` (128×128; state dimension 32×4). A backward matrix recursion in float64 maintains $\|P_k\|_2$ at each step; cross-checking against explicit Φ products agrees to machine precision ($\approx 10^{-15}$). Table 11 reports $C_{\mathrm{ode}}^{\mathrm{LTV}}(\bar{a}^{(z)})$ aggregated over task conditions z and seeds.

Table 11: Path-dependent LTV bound $C_{\mathrm{ode}}^{\mathrm{LTV}}$ at one canonical seed $x^\star = \texttt{torch.randn(seed = 0)}$, aggregated over all 24 task combinations (18 training + 6 held-out, excluding 4 infeasible). Columns: mean, median, and max across combos at seed 0; the mean-across-seeds of the per-seed mean over combos (10 seeds) shows insensitivity to the canonical seed choice in the typical case. Last column: ratio to empirical $C_{\mathrm{ode}}^{\mathrm{emp}}$.

N	mean	median	max	$\overline{\text{mean}} \pm$ std (10 seeds)	$C^{\mathrm{LTV}}/C^{\mathrm{emp}}$
10	85.6	—	498	—	$110\times$
20	22.2	—	89	—	$61\times$
30	17.1	—	71	—	$53\times$
50	**9.68**	**4.38**	**52.0**	13.3 ± 4.6	$38\times$

Finding: the LTV bound tightens Grönwall by six orders of magnitude, but the remaining ~ 10–$100\times$ gap to empirical amplification is not closable by norm-based analysis. At $N = 50$, for the representative combo race3_standard, $C_{\mathrm{ode}}^{\mathrm{LTV}} \approx 4.0$ versus $C_{\mathrm{ode}}^{\mathrm{ana}} \approx 8.4 \times 10^5$. The LTV bound varies across combos from 3.35 (best) to 52.1 (worst, dominated by held-out combos race8_wide and race7_standard); the gap to the empirical per-combo amplification (which ranges 0.195–0.299) is 13–17$\times$ on typical combos, widening to $\sim 170\times$ on the worst-case combo. The LTV bound exploits inter-step cancellation: successive Jacobians do not share their worst singular directions, and the true state-transition matrix $\|\Phi_{k+1,K}\|_2$ is correspondingly less amplifying than naive per-step worst-case

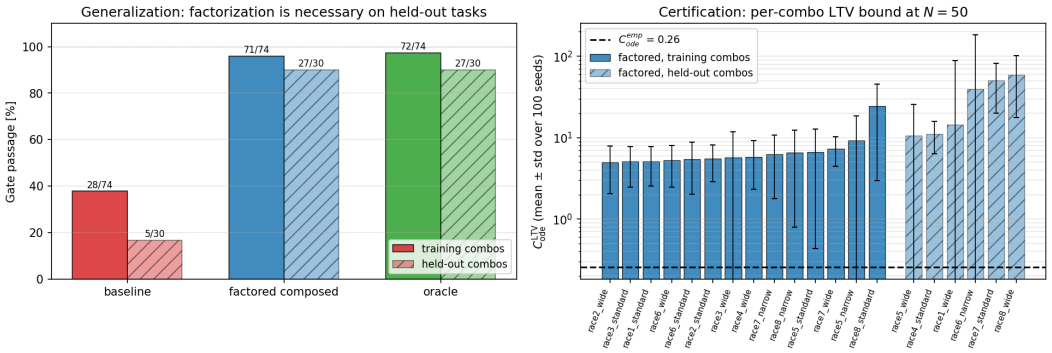

Figure 3: **Generalization (left) and certification (right) for the factored model at N=50.** Left: gate passage on training (solid) vs. held-out (hatched) tasks (same headline numbers as Table 1, plotted per task). Right: path-dependent $C_{\text{ode}}^{\text{LTV}}$ per task, mean $\pm$ std over 10 canonical seeds, sorted within each group. Training tasks cluster at $C \approx 5$ with low variance; held-out tasks have higher means and an order-of-magnitude wider seed spread, indicating that the certificate inherits the composed score's extrapolation uncertainty. Empirical $C_{\text{ode}}^{\text{emp}} \approx 0.26$ shown as dashed reference.

products would suggest. A diagnostic decomposition of the residual gap between the LTV bound and the empirical amplification appears in §B.12; the looseness is nonlinear and directional and cannot be closed by any norm-based bound.

B.9 Per-combo ε_s tightening

The R_{ss} formula (Corollary 1) depends on ε_s, the sup-bound on per-step score mismatch along the denoising path. A distribution-wide worst-case estimate yields $\varepsilon_s \approx 15.4$ for our model, which is $\sim 10\times$ larger than the actual maximum $\|\Delta\varepsilon_k\|$ encountered along any specific nominal. Under the deterministic-seed setup, the certificate is indexed by a fixed nominal trajectory $\bar{a}^{(z)}$, and the rigorous sup-bound $\varepsilon_s^{(z)} := \max_k \|\Delta\varepsilon_k^{(z)}\|$ need only hold along that trajectory. Table 12 shows the effect at $N = 50$: per-combo $\varepsilon_s^{(z)}$ ranges 1.20–2.06 (vs. default 15.4), and the resulting R_{ss} shrinks by the same factor of ~ 10, moving typical combos from ~ 30 m to ~ 3 m without any change to the LTV certificate.

Table 12: Effect of replacing the default scalar $\varepsilon_s = 15.4$ with per-combo $\varepsilon_s^{(z)} = \max_k \|\Delta\varepsilon_k^{(z)}\|$ measured along the nominal at $N = 50$, seed 0. Both values are rigorous sup-bounds for the respective scope (distribution-wide vs. along-nominal). Gate half-widths: 0.3 m (narrow), 0.762 m (standard), 1.0 m (wide). Combo set is 20 tasks spanning 8 tracks $\times$ 3 gate sizes.

	default	per-combo	ratio	closes gate
ε_s range	15.4	$[1.20, 2.06]$	$\sim 10\times$	—
R_{ss} (race3_std)	31.3 m	3.57 m	$\sim 10\times$	no
R_{ss} (best combo)	25.9 m	2.46 m	$\sim 10\times$	no
R_{ss} (worst combo)	207 m	17.0 m	$\sim 10\times$	no
# combos with $R_{\text{ss}} <$ gate	0/20	0/20	—	—

None of the 20 tasks achieves rigorous R_{ss} below its gate half-width even under the tightened ε_s. The residual factor of ~ 13 between R_{ss} and the observed deployment tubes is the same nonlinear manifold contraction quantified in §B.12. We report this calculation to separate looseness that is addressable by careful measurement (ε_s: $10\times$ tightening here) from looseness that is structural to norm-based Lipschitz analysis (nonlinear manifold contraction: $\sim 10\times$, requires a different certificate).

B.10 CONTINUOUS-TIME LIMIT

Running the backward-Lyapunov computation at $N \in \{100, 200\}$ (seed 0, all 20 combos) gives $C_{\text{ode}}^{\text{LTV}} = 5.60$ (mean over combos, min 1.95, max 29.1) at both N values, *identical to three decimal places*. The measured empirical amplification plateaus alongside: $C_{\text{ode}}^{\text{emp}} = 0.242$ (mean over combos) at $N{=}100$ versus 0.242 at $N{=}200$. Both have converged to the continuous-time probability-flow ODE limit; additional DDIM steps cannot tighten the bound or the observed amplification.

For race3_standard: $C_{\text{ode}}^{\text{LTV}}$ drops from 4.04 at $N{=}50$ to 2.26 at $N{=}100$ and stays there; the corresponding empirical $C_{\text{ode}}^{\text{emp}}$ goes $0.30 \to 0.27 \to 0.27$. The gap between certificate and empirical at the continuous-time limit is therefore $\sim 9\times$ for this combo, $\sim 23\times$ for the mean over combos, and $\sim 100\times$ for the worst combo. These are fundamental limits of norm-based path analysis on this trained score network; closing them requires either a training-side change (spectral regularization, rectified flow) or a non-norm-based certificate that captures manifold contraction.

B.11 SEED ROBUSTNESS OF THE LTV CERTIFICATE

Because the deterministic-seed analysis commits the certificate to a specific canonical $x^\star$, we measure how sensitive the numerical bound is to that choice via a 10-seed sweep at $N{=}50$. Table 13 reports per-combo $C_{\text{ode}}^{\text{LTV}}$ mean $\pm$ std and coefficient of variation (CV). A sharp dichotomy emerges: training combos are seed-robust (CV 0.4–0.7, typical mean ≈ 5); held-out combos are fragile (CV up to 5.2, with occasional single-seed explosions to hundreds).

Table 13: Per-combo $C_{\text{ode}}^{\text{LTV}}$ across 10 canonical seeds at $N{=}50$. "Mean" is over 10 seeds for that combo; "CV" $= \text{std}/\text{mean}$. Held-out combos were not seen during training.

combo	mean	std	CV	type
Training combos (14 shown, 14 total, sorted by mean):				
race2_wide	4.97	2.91	0.59	train
race3_standard (**ref.**)	5.14	2.65	0.52	train
race1_standard	5.14	2.60	0.51	train
race6_wide	5.26	2.80	0.53	train
race6_standard	5.42	3.40	0.63	train
race2_standard	5.52	2.65	0.48	train
race3_wide	5.73	6.11	1.07	train
race4_wide	5.81	3.48	0.60	train
race7_narrow	6.24	4.46	0.72	train
race8_narrow	6.58	5.78	0.88	train
race5_standard	6.62	6.18	0.93	train
race7_wide	7.36	2.94	0.40	train
race5_narrow	9.21	9.43	1.02	train
race8_standard	**24.39**	21.44	0.88	train
Held-out combos (6 total, sorted by mean):				
race4_standard	11.15	4.80	0.43	held-out
race5_wide	10.52	14.94	1.42	held-out
race1_wide	14.45	74.59	**5.16**	held-out
race6_narrow	39.57	142.75	**3.61**	held-out
race7_standard	51.10	31.08	0.61	held-out
race8_wide	59.58	41.90	0.70	held-out

Finding 5: the LTV certificate is seed-robust on training combos but not on held-out combos. Training combos give $C_{\text{ode}}^{\text{LTV}} = 6.8 \pm 4.9$ (mean across combos of per-combo means), with one outlier (race8_standard at 24.4). Held-out combos give $C_{\text{ode}}^{\text{LTV}}$ values an order of magnitude larger on average and CVs up to 5.2: certain seed–combo pairs drive $C_{\text{ode}}^{\text{LTV}}$ into the hundreds. Two operational consequences:

- A worst-case task-aggregate bound $\max_z C_{\text{ode}}^{\text{LTV}}(z)$, required for a certificate that covers all deployment conditions, is dominated by held-out outliers and does not reflect typical behavior.

- The gate-passage evidence (Section 4.1, 94% at $N{=}50$) shows held-out combos *do* pass the gate in closed-loop, despite their pathological certificate value. The certificate is conservative on held-out tasks, not wrong: the nonlinear manifold contraction absorbs what the worst-case bound over-predicts.

Future work could tighten the held-out certificate by either (i) increasing factor-dropout regularization during training to encourage flatter Jacobian spectra on unseen conditions, or (ii) training with rectified-flow / flow-matching objectives that explicitly straighten trajectories and reduce per-step Jacobian anisotropy.

B.12 Residual gap diagnostic

The LTV bound $C_{\text{ode}}^{\text{LTV}}$ still overestimates the empirical amplification by an order of magnitude. We decompose this residual along a nominal denoising path by comparing three quantities: the actual action gap, the *linearized* prediction $\|\sum_k c_2(k)\,\Phi_{k+1,K}\,\Delta\varepsilon_k\|$ that uses the measured per-step score errors $\Delta\varepsilon_k$ with their true directions, and the LTV bound $\sum_k |c_2(k)|\,\|\Phi_{k+1,K}\|_2\,\|\Delta\varepsilon_k\|$ that replaces directional propagation with operator-norm worst case.

Table 14: Source-of-looseness decomposition at $N = 50$, seed 0, mean over 14 training tasks.

Quantity	Value [m]	Ratio to **actual**
actual $\|a_K^c - a_K^j\|$	0.095	$1.0\times$
linearized $\|\sum_k c_2 \Phi \Delta\varepsilon\|$	0.562	$5.9\times$
LTV bound $\sum_k \|c_2\|\,\|\Phi\|\,\|\Delta\varepsilon\|$	0.938	$9.9\times$

Two distinct effects show up. The $5.9\times$ linearization gap is *nonlinear manifold contraction*: the score field points back toward the data manifold and shrinks errors faster than its Jacobian predicts, an effect that no operator-norm analysis on J_k can capture. The further $\sim 2\times$ from linearized to LTV bound is *directional alignment*: the LTV bound assumes $\Delta\varepsilon_k$ aligns with each $\Phi_{k+1,K}$'s top singular direction, which empirically it does not. Closing either gap requires a non-norm-based tool: distributional assumptions on $\Delta\varepsilon_k$, or a certificate that explicitly represents manifold distance.

C Preliminary Vision Study: Single-Gate Traversal

This study predates the corrections in the main text and is reported as a preliminary out-of-distribution transfer result, for two reasons stated up front. First, the checkpoints evaluated here were trained without factor dropout: the factor embeddings are jointly conditioned and the null tokens are untrained, so the models are factor-conditioned networks and the composition mechanism of Section 3.2 is not exercised. Second, zero-shot transfer to a venue absent from training evaluates a factor *value* the policy never saw, which Theorem 1 does not cover. The within-venue results below are the only ones in the recombination setting of the theorem.

The quadrotor must fly through a single race gate from diverse spawn positions (3–10 m range, $\pm 45^\circ$ azimuth, both approach sides) using only an onboard 320×240 RGB camera and a noisy gyro. The diffusion network predicts a chunk of $T = 16$ future body-frame offsets to the target gate; the first prediction is converted to a world-frame position estimate via the known gate location and gyro-integrated orientation, then tracked by the soft-preset $\mathrm{SE}(3)$ controller of Section 4. The visual encoder is a ResNet-18 with the last two blocks fine-tuned (13.9 M trainable parameters). Four factors span the task space: *venue* (4: Field, industrial, piazza, pool), *gate color* (3), *gate ID* (5), and *approach side* (± 1), for $4 \cdot 3 \cdot 5 \cdot 2 = 120$ tasks. We evaluate generalization along two axes: *within-venue* (held-out $(\text{color}, \text{gate}, \text{side})$ triples in seen venues, interpolation in the factor product) and *zero-shot venue transfer* (an entire venue held out, extrapolation across visual domain), comparing the three models defined in Section 4 (Baseline, Factored, Oracle). Only the within-venue axis instantiates the recombination setting of Theorem 1, where every factor value is seen in training and only their combinations are new. Zero-shot venue transfer evaluates a factor *value* absent from training, which lies outside the theorem's coverage guarantee; we report it as an empirical out-of-distribution property of the factor-conditioned architecture, not as a consequence of Theorem 1.

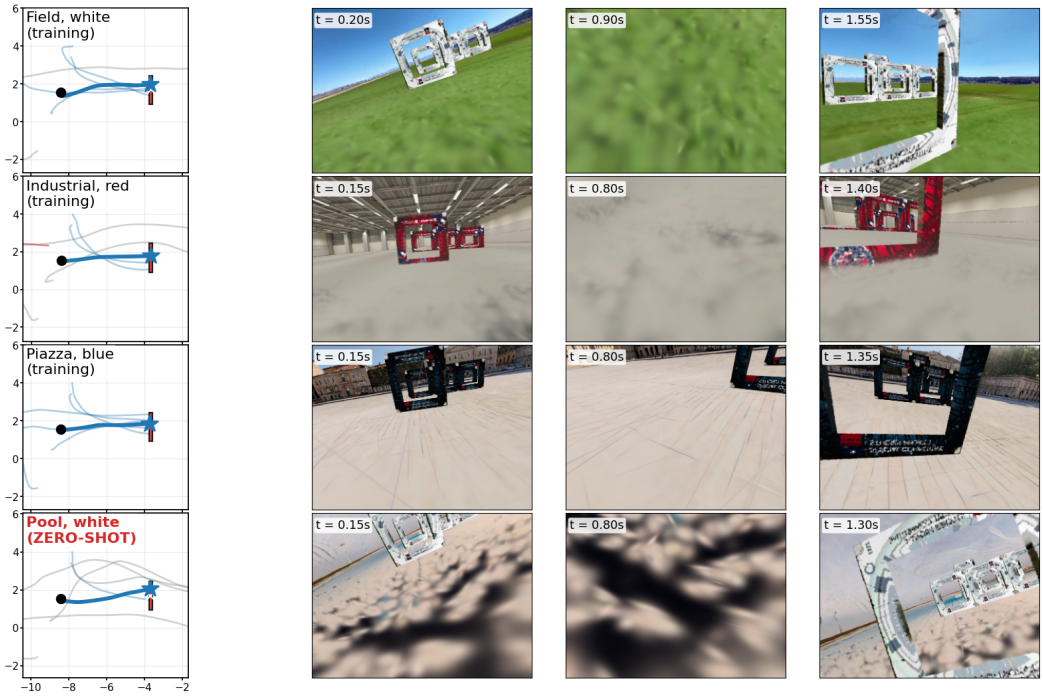

Figure 4: **Zero-shot venue transfer.** Each row shows the policy on a different (venue, gate-color) pair: Field–white, industrial–red, piazza–blue, and pool–white (*zero-shot*). Pool is entirely excluded from training; the policy never observes pool's photometric distribution (water, sky reflections), yet passes its gate (filled rectangle) from a single onboard RGB camera and a noisy gyro. *Left*: bird's-eye view of rollouts. Bold blue line = highlighted successful rollout (camera frames in that row are taken from this trajectory); thin blue = other successes; gray = timeouts; red = crashes. *Right*: three onboard camera chronological frames along the highlighted rollout.

Main results. Pool, visually the most distinct venue (water, reflections, sky), is held out entirely; both models train on Field + industrial + piazza (8,550 rollouts) and evaluate zero-shot on pool. Within-venue compositional generalization (Table 15, right) shows only a modest factored advantage: $+1.1$ pp on held-out (color, gate, side) triples and $+2$ pp on training combos. After two DAgger rounds both models exceed 58% on the small per-venue product space ($3 \times 5 \times 2 = 30$ combinations); the encoder has already adapted to each venue's photometry and the geometry signal generalizes across held-out triples on its own, leaving little room for the factor decomposition to help. Under zero-shot venue transfer (Table 15, left) the factor-conditioned model reaches 36.7% success on pool against 25.0% for the baseline, which we read as the geometric factors (gate, color, side) carrying the task across a photometric shift the encoder never saw. The crash-rate figures in the table should not be read as a property of the method: they depend strongly on how long the policy is trained. Retraining the pool split under the same recipe with the current code gives 20.8% crashes at the epoch-12 best-validation checkpoint (dropout off) and 30.8% with dropout at 0.1, against 5.8% for the released epoch-7 checkpoint, and across all six retrained models the crash rate rises as validation loss falls (Table 16). The released checkpoint is the least-fit of them. We therefore report success rates only, and leave a controlled study of training length and factor dropout on the vision benchmark to future work.

Certificates. We apply the same path-dependent LTV machinery as Section 4.1 (per-step DDIM Jacobians along the deterministic reverse path, backward Lyapunov recursion $P_s = M_s^\top P_{s+1} M_s$, $C_{\text{ode}}^{\text{LTV}} = \sum_s |c_2(s)| \sqrt{\|P_{s+1}\|}$). In this setting (DDIM 10), $C_{\text{ode}}^{\text{LTV}}$ is tightly clustered (mean 2074, range 2057–2088 across 18 within-distribution combos) and $\varepsilon_s \approx 1.95$, giving $R_{\text{ss}} \approx 2{,}023\,\text{m}$. Unlike the full race experiment of the previous section, here the the LTV bound does *not* get close to the empirical gap and this is not a particularly useful certificate.

Table 15: Vision-based single-gate results. *Left*: zero-shot venue transfer (pool held out; three train venues shown as in-distribution sanity check; success / (crash) across n trials). *Right*: within-venue compositional generalization at higher accuracy (All 4 venues, 24 held-out (venue, color, gate, side) combos, six per venue, 20% of the 120-combo space).

Model	3 train (n=180)	Pool 0-shot (n=120)
Baseline	41.7% (34.4%)	25.0% (36.7%)
Factored	**63.3% (10.6%)**	**36.7% (15.0%)**
Adv.	+21.6 pp / 3.2×	+11.7 pp / 2.4×

Model	Train	Held-out	Crash
Baseline	67.8%	58.3%	18.5%
Factored	**69.8%**	**59.4%**	**14.0%**

Table 16: Pool zero-shot, $n = 120$, same split and seed, retrained with the current code. Crash rate rises with fit; the released paper checkpoint is the epoch-7 best-validation model of a 50-epoch run.

Checkpoint	Dropout	Best epoch	Val. loss	Success	Crashes
released (paper)	off	7 of 50	0.079	42.5%	5.8%
50-epoch schedule	off	12 of 50	0.074	32.5%	20.8%
50-epoch schedule	0.1	12 of 50	0.073	35.0%	30.8%
20-epoch schedule	0.1	20 of 20	0.065	35.8%	33.3%
12-epoch schedule	0.02	12 of 12	0.070	24.2%	40.8%
12-epoch schedule	0.1, venue only	12 of 12	0.070	23.3%	40.0%

C.1 Ablations

We report two ablations that corroborate the mechanism by which the factored architecture decomposes appearance from geometry in the single-gate vision setting (Appendix C).

The venue embedding is doing useful work. Raising the venue null-token dropout from 0.1 to 0.4 hurts zero-shot pool by −3.4 pp on success rate and worsens crash rate by +22.5 pp. Replacing the trained venue embedding with the null token at evaluation (*vd=0.4 + null_venue eval*) degrades pool further to 22.5% success / 46.7% crash, showing that the venue row carries information even on a held-out venue. Weakening the venue embedding during training does not push the visual encoder toward distribution-invariant features; it degrades both in-distribution and zero-shot performance.

Backbone choice: DINO-v2 ViT-B with attention pooling. Replacing the fine-tuned ResNet-18 with a DINO-v2 ViT-B/14 backbone using attention pooling over 252 patch tokens, with the last transformer block fine-tuned, matches the ResNet-18 factored model on a different held-out venue (*Field*-holdout): 60.0% success on the three training venues vs. 35.8% zero-shot, with crash rate 19.2% and best validation loss 0.0522 (vs. 0.0608–0.0611 for the ResNet-18 factored / unfactored baselines on the same holdout). As a control, a *frozen* DINO-v2 ViT-S/14 backbone using only the cls token collapses both in-distribution and zero-shot (−19.2 pp success on pool, +49.2 pp crash relative to the fine-tuned ResNet-18): the cls token compresses away the spatial cues required for visual servoing, and a frozen backbone cannot adapt to gate localisation. The factored advantage is therefore not a backbone artefact: it persists when the visual encoder is swapped for a self-supervised ViT, provided enough capacity is left trainable for the encoder to acquire gate-localisation features.

D Factored vs. joint bound

A non-compositional approach that learns the joint score $s_\theta(\cdot, z)$ directly bounds the score perturbation by $L_{\text{joint}}\|\delta\|$, where L_{joint} is the Lipschitz constant over the product space $\mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$ and $\|\delta\| = \sqrt{\sum_i \delta_i^2}$. The joint bound has no decomposition error, so the total score perturbation is $\varepsilon_s^{\text{joint}} \le \eta + L_{\text{joint}}\|\delta\|$ (a single forward pass per inference, vs. $(2K-1)\eta$ for the composed score below).

The factored bound replaces this with $\varepsilon_s^{\text{fact}} \le 2\sqrt{GM} + (2K-1)\eta + \sum_i L_i\delta_i$. By Cauchy–Schwarz, $\sum_i L_i\delta_i \le \sqrt{\sum_i L_i^2}\,\|\delta\|$, with equality iff $(L_1, \ldots, L_K)$ and $(\delta_1, \ldots, \delta_K)$ are linearly dependent

($L_i \propto \delta_i$). The gap is substantial when per-factor sensitivities are heterogeneous and δ is not aligned with the largest L_i directions. The joint Lipschitz constant is itself bounded *above* by the per-factor RMS, $L_{\text{joint}} \le \sqrt{\sum_i L_i^2}$ (any two factor-product points are connected by a coordinate-axis path of combined length $\le \sqrt{\sum_i L_i^2}\,\|\delta\|$). This inequality can be strict: for the identity map $f(z_1, z_2) = (z_1, z_2)$ on $\mathbb{R}^2 \to \mathbb{R}^2$, $L_1 = L_2 = 1$ gives $\sqrt{L_1^2 + L_2^2} = \sqrt{2}$, but the joint operator norm is $L_{\text{joint}} = 1$, so $\sqrt{\sum_i L_i^2} > L_{\text{joint}}$ strictly. Consequently both $\sum_i L_i \delta_i$ and $L_{\text{joint}}\|\delta\|$ are dominated by $\sqrt{\sum_i L_i^2}\,\|\delta\|$, but neither is uniformly tighter than the other; their relative size depends on the alignment of δ with the Lipschitz directions of the two score fields. The factored certificate beats the joint certificate when the factored score perturbation is the smaller of the two:

$$2\sqrt{GM} \;+\; \sum_i L_i \delta_i \;<\; L_{\text{joint}} \|\delta\|,$$

a condition that depends on both the factor-interaction strength (G, M) and the alignment of δ with the dominant Lipschitz directions; it is not guaranteed by the structure alone. If the factor interaction is strong (G large) or the log-ratio is highly curved (M large), the decomposition cost dominates and the joint bound is tighter.

The *generalization* benefit of factorization (Theorem 1) is unconditional: it enables evaluation on unseen tasks where no joint model exists. The *certification* benefit is conditional: the factored tube is tighter than the joint tube only when the approximate conditional independence is strong enough ($2\sqrt{GM}$ small) *and* the per-factor weighting makes $\sum_i L_i \delta_i$ smaller than $L_{\text{joint}}\|\delta\|$.

E Extended Related Work

This appendix expands Section 5 with a per-method comparison of recent compositional-diffusion methods for control and a side-by-side summary table.

PoCo (Wang et al., 2024) merges K diffusion networks trained on different data sources (task / behavior / domain slices) via product-of-experts on noise predictions; the goal is fusing heterogeneous demonstration data, not generalizing to unseen factor combinations. Compose Your Policies (GPC) (Cao et al., 2026) takes test-time convex combinations of K pre-trained vision-language-action policies, with the weights set by task descriptors, and provides a per-step Grönwall-type improvement bound related to our Lemma 1; this composes whole policies rather than per-factor corrections of a shared one, and named factors are not part of the formalism. Factorized Diffusion Policy (FDP) (Liu et al., 2026) combines K expert U-Nets through a learned router; here "factor" means a latent behavioral mode discovered by clustering, not a named external attribute with known semantics. Modality-Composable DP (Cao et al., 2025) composes scores across visual modalities (RGB vs. point cloud), so the composition runs over input streams rather than task structure. Diffusion Score Equilibrium (DSE) (Patil et al., 2024) composes pretrained base diffusion policies for arm motions (lines, circles, pendulum swings) to fit a few demonstrations of a new movement skill, which again combines K separately trained networks, here over skills rather than named task factors.

We differ on four axes: (i) a single shared score network whose per-factor corrections are obtained by null-token dropout, rather than K separately trained networks plus a router or weighting scheme; (ii) named semantic factors indexing a product space, with demonstrated generalization to held-out factor combinations, rather than composition over latent modes, heterogeneous data sources, or whole policies; (iii) a formal trajectory-tube certificate that implies task-level gate passage, rather than a per-step action-error bound or an empirical collision rate; (iv) 3D racing-drone control rather than manipulation, autonomous driving, or 2D cars. The single shared-network realization matters beyond parameter count: per-factor null-token dropout is what makes the decomposition $s_{\text{comp}} = s_\varnothing + \sum_i \Delta_i$ identifiable from training data (Section 3.2), and it is this structure that enables the per-factor Lipschitz bounds L_i used in the certificate.

E.1 Certified robustness for neural-network controllers

Lipschitz-constant bounds (LipSDP, Fazlyab et al., 2019) compute global Lipschitz constants of feed-forward networks via semidefinite programming and certify outputs under bounded input perturbations. Reachability of neural-network controllers (ReachLP, Everett et al., 2021) propagates

Table 17: Compositional diffusion for control: prior work vs. ours. *Factor type*: whole policies (W), latent modes (L), or named semantic factors (S). *Certificate*: single-step action-error bound (A), task-success guarantee (T), or none.

Method	Domain	Networks	Factor type	# factors	Certificate
PoCo (Wang et al., 2024)	Manipulation	K	W	3 (slices)	none
GPC (Cao et al., 2026)	Manipulation	K	W	K	A
FDP (Liu et al., 2026)	Manipulation	K + router	L	latent	none
Mod.-Comp. DP (Cao et al., 2025)	Manipulation	K	W	modalities	none
DSE (Patil et al., 2024)	Arm motion	K	W	base skills	none
Ours	3D drone	**1 shared**	**S**	2	**T**

polytopic input sets through the controller step by step to bound reachable state sets. Black-box perception contracts (Astorga et al., 2023; Li et al., 2025) treat the perception–control stack as one nonlinear map and establish input/output assumption pairs that imply task-level safety. All three frameworks bound output variation against the entire input variation at once. In our setting the input variation is structured as a product over named factors, and the per-factor decomposition $\sum_i L_i \delta_i$ replaces $L_{\text{joint}} \|\delta\|$ with a tighter sum whenever the per-factor sensitivities L_i differ.

E.2 Sequential skill composition

SkillDiffuser (Liang et al., 2024) and Generative Skill Chaining (Mishra et al., 2023) compose pre-trained skills as a sequence in time: the trajectory is partitioned into segments, each generated by a different skill, and the composition runs along the temporal axis. Our composition is orthogonal: a single denoising step produces actions conditioned on all K factors simultaneously, and the additive structure is across factor scores at each step rather than across time.

E.3 Champion-level drone racing systems

Champion-level autonomous drone racing systems (Kaufmann et al., 2023; Song et al., 2023; Foehn et al., 2021) achieve sub-second lap times on a fixed track via per-track training, model-predictive control with track-specific cost functions, or extensive simulation-to-real fine-tuning. These systems beat the factored policy on absolute lap time but require re-tuning for each new track and gate-size combination. We trade peak speed for cross-track generalization from a training-task budget that grows as a sum rather than a product, with a closed-loop certificate that holds across $\mathcal{Z}_1 \times \mathcal{Z}_2$.

E.4 Domain randomization and task-conditioned multi-task RL

Domain randomization (Tobin et al., 2017; Sadeghi & Levine, 2017) trains a policy on a randomized distribution over environments to encourage invariance. Task-conditioned multi-task RL conditions the policy on a task descriptor and pools trajectories across tasks during training. Both approaches condition or randomize over factors at training time without explicit per-factor composition; the unfactored baseline in Section 4 represents this regime within our setup and reaches only 10% on held-out gates, indicating that randomization or generic conditioning alone does not yield held-out compositional generalization in our benchmark.

F Trajectory-Tube Certificate: Additional Material

Estimating the score-matching error. The denoising score-matching training loss is an unbiased estimate of the *expected* squared score error $\mathbb{E}[\|s_\theta - s\|^2]$ (Song et al., 2021b), which by Jensen's inequality lower-bounds η^2. Empirical sup-bounds on η are obtained by worst-case sampling over noised states drawn along the nominal denoising trajectory; the per-combo $\varepsilon_s^{(z)} := \max_k \|\Delta\hat{\varepsilon}_k^{(z)}\|$ measurements of Appendix B.9 are the operational instantiation of this estimate. A small training loss is necessary but not sufficient for small η.

Hypotheses (a) and (b). Under hypothesis (a) of Theorem 2 the bound (11) is a strict upper bound on the closed-loop state error. Under hypothesis (b) it is first-order in ε_s, with an $O(\varepsilon_s^2)$ Taylor

remainder that propagates additively through the contraction proof; the remainder is controlled when $\varepsilon_s \cdot C_{\mathrm{ode}}^{\mathrm{LTV}}$ is small relative to the local radius of linearization (Remark 4), which holds with a margin of about $10\times$ in the full-race experiments.

Corollary 1. *After the transient $\lambda^t \delta_0$ decays, the closed-loop state error satisfies $\|x_t - x_t^{\mathrm{nom}}\| \le R_{\mathrm{ss}}$ with C_{ode} from Theorem 2 and $R_{\mathrm{ss}} = (B_\kappa\, C_{\mathrm{ode}} \varepsilon_s + w)/(1-\lambda)$.*

Factored versus joint bound. The factored bound $\sum_i L_i \delta_i$ replaces the joint bound $L_{\mathrm{joint}}\|\delta\|$ with a per-factor weighted sum. Neither is uniformly tighter; the factored certificate beats the joint certificate when $2\sqrt{GM} + \sum_i L_i \delta_i < L_{\mathrm{joint}}\|\delta\|$, a condition on both the factor-interaction strength and the alignment of δ with the dominant Lipschitz directions (Appendix D).

Certifiable region. The gate-size factor z_2 controls R through Δ_2 (slowing the drone near narrow gates) and sets the threshold $r(z_2)$ (wider gates are easier to certify). The certifiable region $\{z_2 \in \mathcal{Z}_2 : r(z_2) > R\}$ grows as the score-perturbation budget ε_s shrinks, that is, as the decomposition residual $2\sqrt{GM}$ and the score-matching error η both decrease.