Exploring the Intrinsic Geometry of Diffusion Models with Constrained Inverse Kinematics

Miguel Angel Rogel Garcia*, Phone Thiha Kyaw*, and Jonathan Kelly
Institute for Aerospace Studies, University of Toronto, Toronto, Canada
{miguel.rogel, phone.thiha, jonathan.kelly}@robotics.utias.utoronto.ca
*Equal contribution.

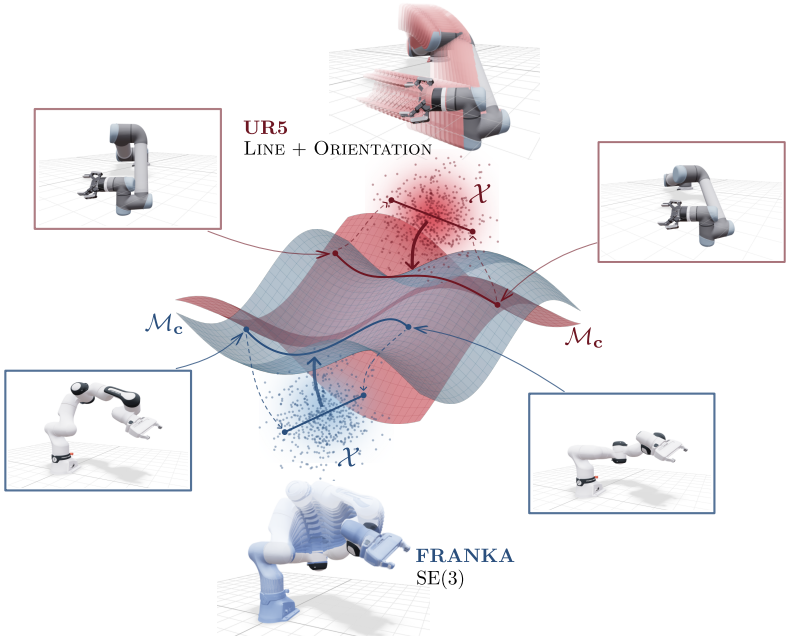

Fig. 1: **Diffusion models capture the geometry of constraint manifolds in inverse kinematics.** Given a constraint family and a target pose, we analyze the learned diffusion model along two axes: estimating the intrinsic dimension of the model's learned distribution and the behaviour of linear interpolation between samples in latent space. The estimated intrinsic dimension matches the analytical dimension of the underlying constraint manifold. Latent interpolation stays close to the constraint manifold, further indicating that the model has encoded the manifold's geometry.

***Abstract*—Recent studies suggest that diffusion models can recover geometric structure in the data manifolds they are trained on, yet the supporting evidence has so far come mostly from natural-image data, where the underlying geometry itself is unknown. We study this question in a setting where the geometry is analytically tractable: *constrained inverse kinematics* (IK). Each task-space constraint defines a configuration-space manifold with known intrinsic dimension, giving direct ground truth for evaluating the geometry learned by the model. For each of the 6-DoF UR5 and 7-DoF Franka, we train a single conditional diffusion model across seven constraint families, spanning solution manifolds from discrete IK branches to self-motion manifolds. Our empirical results reveal that the intrinsic dimension recovered from the model's score function matches the analytical degrees of freedom of the corresponding constraint manifold across both robots. Moreover, linear interpolation in the latent space leads to generated solutions that remain close to the appropriate constraint manifold, indicating that the learned representation further captures geometric structure of the constraint family beyond intrinsic dimension alone. Constrained IK therefore offers a controlled setting for studying the intrinsic geometry learned by diffusion models.**

I. Introduction

Diffusion models have become the dominant generative framework for high-dimensional data. Although they are trained only to denoise, a growing line of work reveals that their internal representations recover surprisingly rich geometric structure. The score function aligns with the normal bundle of the data manifold and recovers its intrinsic dimension [28], which itself varies with the conditioning input [13]. Pullback metrics on feature spaces yield semantically meaningful tangent directions [20]. Riemannian metrics built from the score admit geodesics that remain on the data manifold [7, 2]. Taken together, these findings suggest that diffusion models can encode aspects of the geometry of the distributions on which they are trained.

Much of this evidence has been collected on natural-image manifolds, where the intrinsic dimension itself is an estimated quantity [22, 4] and the topology is recovered only indirectly. We complement this line of work by studying the *constrained*

inverse kinematics setting. Here, the data manifold is given analytically across a parametric family, as the IK solution set forms a constraint manifold of known dimension and topology. This manifold varies smoothly with a small conditioning vector, and we ask how much of that variation is reflected in the geometry of the trained model's latent space.

While prior generative IK solvers have focused on sample quality [1, 19, 3, 16, 29], we instead treat constrained IK as a controlled setting for analyzing diffusion-model geometry. For each of the 6-DoF UR5 and the 7-DoF Franka manipulator, we train a single diffusion model across all constraint types and analyze the latent space it induces. Our empirical findings show that the intrinsic dimension estimated from the learned score function matches the analytically known intrinsic dimension of the constraint manifold for every constraint type on both robots. We further show that solutions generated from linear interpolation in the latent space remain close to the corresponding constraint manifold, indicating that the learned representation captures the geometric structure of the entire constraint family beyond dimensionality.

Contributions. (i) We present constrained IK as a controlled setting for diffusion-model interpretability in which the data manifold is given analytically across a parametric family of varying intrinsic dimension. (ii) We train a single conditional diffusion model that encodes this entire family, with the dimension of its latent representations matching the analytical ground truth for every member. (iii) We provide evidence that linear interpolation in the learned latent space remains close to the constraint manifold, with generated motion largely following the local geometry of the manifold, demonstrating that the latent representation reflects more than intrinsic dimension alone.

II. Background

This section reviews the relevant technical background on constraint manifolds, diffusion models and intrinsic dimensionality. A notation glossary is provided in Appendix A.

A. Constraint manifolds and their intrinsic geometry

Constrained inverse kinematics provides a setting in which the geometry of the data distribution is available analytically. This makes constrained IK well-suited for studying the internal structure of diffusion models: rather than inferring the data manifold from samples, we can compare the learned representation against a manifold whose geometry is known a priori. Unlike in classical IK, a task-space constraint need not specify a single end-effector pose; it may fix only some of the task-space degrees of freedom and leave the rest free, yielding a constraint manifold in configuration space. Such constraint sets are well characterised in the robot kinematics literature, with their dimension, redundancy structure, and connectivity available in closed form for a wide range of manipulators [5, 25, 17, 18, 12]. We adopt this body of analytical results as our ground truth.

Let $\mathbf{q} \in \mathbb{R}^n$ denote the joint configuration of a manipulator with n actuated joints, and let $f : \mathbb{R}^n \to \mathrm{SE}(3)$ denote its forward kinematics. For a task-space specification $\mathbf{c}$ that restricts $f(\mathbf{q})$ to a subset $\mathcal{S}_\mathbf{c} \subseteq \mathrm{SE}(3)$, the set of configurations satisfying the constraint is

$$\mathcal{M}_\mathbf{c} \;=\; \{\mathbf{q} \in \mathbb{R}^n : f(\mathbf{q}) \in \mathcal{S}_\mathbf{c}\} \;\subset\; \mathbb{R}^n. \tag{1}$$

When $\mathcal{S}_\mathbf{c}$ is a smooth submanifold of $\mathrm{SE}(3)$ of codimension k and f has full-rank Jacobian on $\mathcal{M}_\mathbf{c}$, the implicit function theorem gives $\mathcal{M}_\mathbf{c}$ the structure of a smooth embedded submanifold of configuration space with *intrinsic dimension* $n - k$ [14]. The codimension k equals the number of task-space coordinates fixed by the constraint; as k increases, the feasible set loses degrees of freedom and $\mathcal{M}_\mathbf{c}$ shrinks from higher-dimensional continuous families through two- and one-dimensional ones down to a finite point set. The constraints we consider range from $k = 6$ for a full $\mathrm{SE}(3)$ point-target down to $k = 1$ for a single-coordinate restriction. These include both purely positional restrictions (point, line, or plane in $\mathbb{R}^3$) and combinations involving end-effector orientation in $\mathrm{SO}(3)$.

The topology of $\mathcal{M}_\mathbf{c}$ also depends on the kinematic redundancy of the manipulator relative to the constraint. For the non-redundant 6-DoF UR5 under a full $\mathrm{SE}(3)$ pose constraint, $\mathcal{M}_\mathbf{c}$ collapses to a finite set of configurations: the classical IK branches for 6R manipulators [21, 23]. For the redundant 7-DoF Franka under the same constraint, the extra joint promotes $\mathcal{M}_\mathbf{c}$ to a one-dimensional self-motion manifold along which the arm can reconfigure while the end-effector pose stays fixed [5].

Every quantity we use to study the learned latent space (the dimension of $\mathcal{M}_\mathbf{c}$, its tangent space $\mathcal{T}_\mathbf{q}\mathcal{M}_\mathbf{c}$, and the constraint error from a generated configuration to $\mathcal{S}_\mathbf{c}$) is available in closed form from f and $\mathbf{c}$, rather than estimated from samples. The same cannot be said of natural-image manifolds, where the dimension being recovered is itself an estimate. The analytical $\mathcal{M}_\mathbf{c}$ serves as the reference against which we compare both the score-based intrinsic dimension estimate in Section III-A and the on-manifold behaviour of latent-space interpolations in Section III-B.

B. Conditional diffusion models and DDIM inversion

Conditional generative models typically learn a family of distributions on a fixed support; with conditioning inputs ranging from text prompts in image synthesis [24] to state observations in robotic policies [6]. In constrained inverse kinematics, the support itself varies with the conditioning input. Rather than fitting a separate model per constraint, we train a single model whose conditioning input $\mathbf{c}$ selects both the constraint type and the task-space target (Section III). Varying $\mathbf{c}$ thus changes the conditional data distribution, and in our setting, it also changes the dimension and topology of the manifold on which that distribution is supported.

Diffusion models learn this distribution by reversing a Gaussian noising process. For a clean sample $\mathbf{x}_0$ and variance schedule $\bar{\alpha}_t$, the forward process, defined by

$$\mathbf{x}_t \;=\; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 \;+\; \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

produces latent variable $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^n$. A denoising network $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{c})$ is trained to predict $\boldsymbol{\epsilon}$ from the corrupted sample, noise level t, and condition $\mathbf{c}$ [11]. Up to a time-dependent

TABLE I: **The seven task-space constraints considered in this work.** "Fixed" denotes the number k of task-space DoFs imposed by each constraint, and "Free" denotes the ID $n-k$ of the joint-space solution manifold, given for the 6-DoF UR5 and the 7-DoF Franka.

			Free (ID)	
Constraint	End-effector quantity fixed	Fixed	UR5	Franka
PLANE	one position coordinate	1	5	6
LINE	two position coordinates	2	4	5
POSITION	three position coordinates	3	3	4
ORIENTATION	three rotation coordinates	3	3	4
PLANE + ORIENTATION	one position coord. + full rotation	4	2	3
LINE + ORIENTATION	two position coords. + two rotation coords.	4	2	3
SE(3)	full pose	6	0	1

scaling, this predicted noise is an estimator of the Stein score of the conditional noised marginal [27, 26],

$$s_\theta(\mathbf{x}_t, t, \mathbf{c}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{c}) \approx -\frac{\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{c})}{\sqrt{1-\bar{\alpha}_t}}. \quad (3)$$

At small noise levels, the score points toward higher-likelihood regions of $\mathcal{M}_\mathbf{c}$ and, in expectation, aligns with its normal bundle [28]. The trained score field therefore gives us a direct way to ask whether the learned conditional distribution has the same local dimension as the analytical manifold $\mathcal{M}_\mathbf{c}$.

The trained model also gives us a deterministic sampler. DDIM [26], a non-Markovian reformulation of the reverse process, generates samples by iterating the deterministic update

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\hat{\mathbf{x}}_0(\mathbf{x}_t, t, \mathbf{c}) + \sqrt{1-\bar{\alpha}_{t-1}}\,\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{c}), \quad (4)$$
$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\theta}{\sqrt{\bar{\alpha}_t}}.$$

Iterating (4) from $\mathbf{x}_T$ to $\mathbf{x}_0$ defines a deterministic map $g_\mathbf{c} : \mathcal{X} \to \mathbb{R}^n, \mathbf{x}_T \mapsto \mathbf{x}_0$, which is locally invertible under the small-step approximation $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_{t-1}, t-1, \mathbf{c}) \approx \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{c})$ [26], with generated samples concentrated near $\mathcal{M}_\mathbf{c}$ when the learned score closely approximates the true score. Solving (4) for $\mathbf{x}_t$ yields the forward-time *DDIM inversion* update

$$\begin{aligned}\mathbf{x}_t \approx\ & \sqrt{\bar{\alpha}_t}\,\hat{\mathbf{x}}_0(\mathbf{x}_{t-1}, t-1, \mathbf{c}) \\ & + \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_{t-1}, t-1, \mathbf{c}),\end{aligned} \quad (5)$$

which iterated from $\mathbf{x}_0$ recovers a latent $\mathbf{x}_T = g_\mathbf{c}^{-1}(\mathbf{x}_0)$ that maps back to $\mathbf{x}_0$. Stopping inversion at an intermediate noise level $\bar{t} < T$ instead returns a *partial* latent $\mathbf{x}_{\bar{t}}$ that retains most of $\mathbf{x}_0$'s structure; we use this latent variable for the interpolation experiments in Section III-B.

C. Intrinsic dimension estimation

The manifold hypothesis posits that most high-dimensional data found in practice concentrates on a much lower-dimensional submanifold, whose intrinsic dimension (ID) counts the local factors of variation [10, 9]. Estimating this dimension matters in practice: it lower-bounds the sample complexity of learning [22], and linear directions in latent space often carry semantic meaning when the dimension is small [20]. A first family of ID estimators infers the dimension from data-space neighbourhood statistics: local PCA on the nearest neighbours [10], maximum-likelihood on Poisson-process neighbour counts [15], or the closed-form 1NN/2NN ratio of TwoNN [8]. A second, more recent family uses the score function of a trained diffusion model directly. Stanczuk et al. [28] show that, as the noise level $t \to 0$, the score $s_\theta(\mathbf{x}_t, t)$ at a noised sample $\mathbf{x}_t$ points back toward the manifold $\mathcal{M}_\mathbf{c}$, in a direction normal to it. Stacking the score at K independent perturbations of $\mathbf{x}_0$ therefore yields a matrix whose top singular vectors span the normal space $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}_\mathbf{c}$.

We adopt the method proposed in [28] to estimate the ID of the learned latent space. Concretely, for a sample $\mathbf{x}_0$, one draws K independent Gaussian perturbations at a single time t, evaluates the network on each, and stacks the resulting score vectors as the columns of

$$S = \left[s_\theta(\mathbf{x}_t^{(1)}, t) \ \dots\ s_\theta(\mathbf{x}_t^{(K)}, t) \right] \in \mathbb{R}^{n \times K}.$$

Recall that $\mathcal{M}_\mathbf{c}$ has codimension k in $\mathbb{R}^n$, so its normal space at any point is k-dimensional. The left singular vectors of S corresponding to the k largest singular values then span $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}_\mathbf{c}$, so $\hat{d} = n - \hat{k}$ estimates the intrinsic dimension, with $\hat{k}$ recovered heuristically as the index of the largest consecutive drop in the singular value spectrum $\sigma_1 \geq \cdots \geq \sigma_n$,

$$\hat{k} = \operatorname*{argmax}_{i=1,\dots,n-1} (\sigma_i - \sigma_{i+1}). \quad (6)$$

Two hyperparameters control the estimator: the time t, which must be small enough for the normal-bundle alignment to hold yet large enough that $\mathbf{x}_t$ stays in the network's training distribution, and the perturbation count K, which must exceed the codimension k for the normal space to be fully spanned. In our experiments, within a useful intermediate range of t, $\hat{d}$ recovers the analytical ID, while values of t too small or too large degrade the estimate. The construction above, which we call Score-SVD, assumes $\mathcal{M}_\mathbf{c}$ is positive-dimensional. It therefore does not directly apply to 0-dimensional data manifolds such as finite, isolated point sets, which is the case for the UR5 under a full SE(3) pose constraint. In Section III-A, we modify the Score-SVD approach to estimate the ID in this 0-dimensional case.

III. GEOMETRIC STRUCTURES IN THE LATENT SPACE

Throughout this work, we study the geometry of the learned latent space under a family of progressively tighter task-space

constraints, evaluated on two manipulators with different kinematic redundancy. The seven constraints (Table I) range from fixing one position coordinate for a plane constraint, to fixing the complete $\mathrm{SE}(3)$ pose. Each constraint is evaluated on both the non-redundant 6-DoF UR5 and the redundant 7-DoF Franka, yielding fourteen constraint–manipulator conditions in total. The two manipulators are chosen because they produce qualitatively different solution-set geometries under identical constraints: for the UR5 under a full $\mathrm{SE}(3)$ pose constraint, IK solutions form a finite set of disconnected configuration branches, whereas the additional redundant joint of the Franka induces a one-dimensional self-motion manifold. Precise constraint definitions are given in Appendix B-B.

A. Recovering ID across a family of constraints

If the diffusion model has learned the geometry of the constraint family, we would expect the ID recovered from its score function at a configuration $\mathbf{q} \in \mathcal{M}_\mathbf{c}$ to match the analytical degree of freedom $n - k$ for every conditioning input. We test this hypothesis across the fourteen constraint–manipulator conditions; for each robot, we train a single conditional diffusion model on all seven constraints of Table I. For each family, we sample task-space targets and compute their analytical IK solutions as training data; the conditioning vector $\mathbf{c}$ encodes both the constraint type and the target. Full training details are given in Appendix C-C.

We estimate the intrinsic dimension of $\mathcal{M}_\mathbf{c}$ using the Score-SVD procedure of Section II [28]. Figure 2 shows the resulting estimates for every constraint–manipulator condition, together with two neighbourhood-based baselines (local MLE [15] and PCA with a 90% variance threshold [10]) and the analytical ID as ground truth. On both robots, the Score-SVD estimate matches the analytical ID for every condition with ID > 0. Both baselines underestimate the analytical dimension by a gap that widens as the ID grows, with PCA only matching the true value at the smallest dimensions. The SE(3) condition for UR5 is a special case: $\mathcal{M}_\mathbf{c}$ collapses to a finite set of isolated IK branches with ID $= 0$. Equivalently, the tangent space is zero-dimensional, so every nonzero ambient direction is normal. In this case, the standard Score-SVD rule cannot detect the relevant spectral gap, as Score-SVD assumes $\mathcal{M}_\mathbf{c}$ is positive dimensional. We therefore modify the estimator to recover the analytical ID $= 0$ of this discrete case. Our variant, Score-SVD†, appends a virtual zero singular value $\sigma_{n+1} = 0$ to the descending spectrum: when the smallest singular value σ_n is well separated from this zero value, the argmax in (6) selects position n, returning $\hat{k} = n$ and hence $\hat{d} = 0$ (implementation details of Score-SVD† can be found in Appendix D-A).

B. Linear latent interpolation stays close to $\mathcal{M}_\mathbf{c}$

In Section III-A, we show that the intrinsic dimensionality of the latent space matches the analytical degrees of freedom of the constraint manifold. However, matching dimensionality alone does not fully explain the underlying geometry of the learned representation. A latent space could be dimensionally accurate yet geometrically curved, such that straight-line paths in latent coordinates decode into trajectories that leave $\mathcal{M}_\mathbf{c}$. We therefore investigate whether the diffusion model learns a non-linear mapping that locally linearizes the constraint manifold, translating curves on $\mathcal{M}_\mathbf{c}$ into straight lines in latent space. Specifically, we evaluate the extent to which linear interpolations in the DDIM-inverted latent space remain 'on-manifold' when decoded. Unlike image-based evaluations, our setting admits an exact, analytical test: given a generated configuration $\mathbf{q}$, we measure the forward kinematics residual, i.e., the distance from the end-effector pose to the task-space constraint set $\mathcal{S}_\mathbf{c}$. Constraint satisfaction thus becomes a quantitative property rather than a perceptual judgment.

For each of the fourteen constraint–manipulator conditions, we sample 30 task-space targets. At each target, we select five pairs of IK solutions whose joint-space separations span the empirical distribution, from the tenth percentile to the maximum observed separation; the widest pair represents an extreme boundary case. For the UR5 under the full $\mathrm{SE}(3)$ constraint, the widest pairs may connect IK branches that are disconnected in joint space, with no continuous pose-preserving path between them. In contrast, for the Franka, the widest pairs span the full extent of a continuous self-motion manifold.

Latent interpolation between IK solutions: For each IK pair, we invert both configurations to an intermediate noise level of the partial DDIM process. We then linearly interpolate their latent codes and decode the interpolants to joint configurations. Across all constraints, the decoded joint-space paths remain close to the corresponding constraint manifolds. The average position constraint error, measured by forward kinematics, ranges from sub-millimetre to a few millimetres; rotational errors for orientation-constrained tasks are reported in Appendix E-A.

Figures 3 and 4 show linear interpolation results for four representative constraint–manipulator conditions, spanning cases from high to low ID; the remaining conditions are reported in Appendix E-B. We observe that interpolation errors are larger at lower noise levels and decrease toward zero at higher noise levels. The errors also decrease as the joint-space separation between IK solutions becomes smaller. The dotted line in Figure 3 indicates the noise level at which the estimated ID begins to diverge from the analytical ID, coinciding with the level at which the interpolation error drops to near zero across all pairs. *Linear paths in latent space thus appear to remain close to the constraint manifold only once the intermediate-noise representation expands beyond the analytical ID. This suggests that the DDIM representation locally flattens the constraint geometry along interpolation directions, allowing simple linear paths in latent space to decode into motions that remain close to $\mathcal{M}_\mathbf{c}$.*

For the UR5 under the full $\mathrm{SE}(3)$ constraint, the feasible configurations form a finite set of disconnected IK branches. Any continuous joint-space path between two branches must therefore leave the constraint manifold along its interior. We observe that the decoded latent interpolation is consistent with this behaviour: constraint error stays small along the decoded samples, while the joint-space trajectory makes a sharp

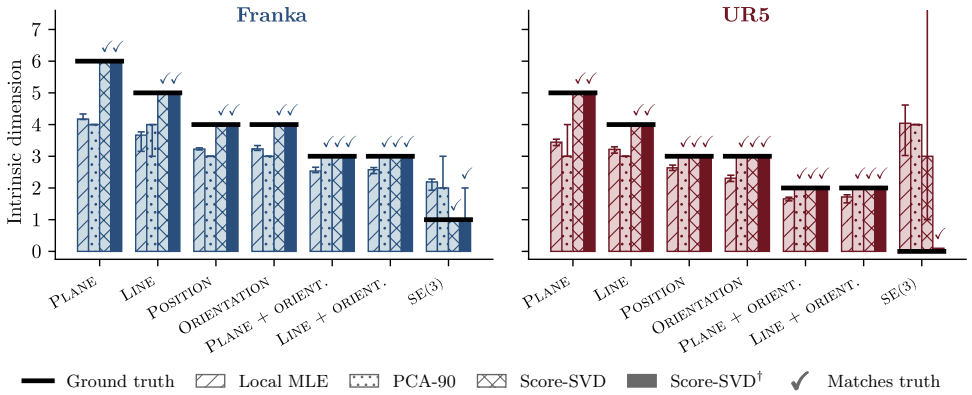

Fig. 2: **Intrinsic dimension estimation.** We compare the ID computed by four estimators on seven task-space constraints per robot. Bars show the per-target median with 95% bootstrap CIs.

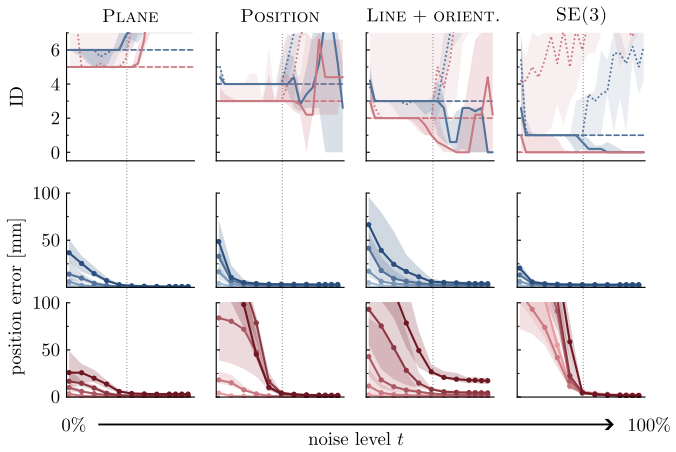

Fig. 3: **Linear interpolation in latent space for Franka and UR5.** *Top*: Estimated intrinsic dimension versus noise t; - - - / - - - Ground truth; · · · · / · · · · Score-SVD; —— / —— Score-SVD†. *Bottom*: Position error of interpolated samples; darker colours indicate IK interpolation pairs are further away from each other. Dotted vertical lines indicate the intermediate noise level at which ID estimation diverges from ground truth, coinciding with interpolation error converging towards zero.

transition when switching between branches. This suggests that the latent interpolation is consistent with the disconnected topology of the constraint set rather than artificially constructing an infeasible continuous bridge in joint space.

Latent interpolation is consistent with local flattening of constraint geometry: The interpolation results suggest that the model does more than recover the ID of each constraint manifold; it generates joint-space motions that are largely consistent with the local constraint geometry. To test this, we examine the map $g_{\mathbf{c}}$ along the latent interpolation path $\mathbf{x}(\alpha) = (1-\alpha)\,\mathbf{x}_a + \alpha\,\mathbf{x}_b, \alpha \in [0,1]$, where subscripts a and b denote the two interpolation endpoints at a fixed noise level t, with unit direction $\hat{\mathbf{v}} = (\mathbf{x}_b - \mathbf{x}_a)/\|\mathbf{x}_b - \mathbf{x}_a\|$. At each point on this path, we measure how much the generated configuration $\mathbf{q}(\alpha) = g_{\mathbf{c}}(\mathbf{x}(\alpha))$ changes under an infinitesimal step along $\hat{\mathbf{v}}$,

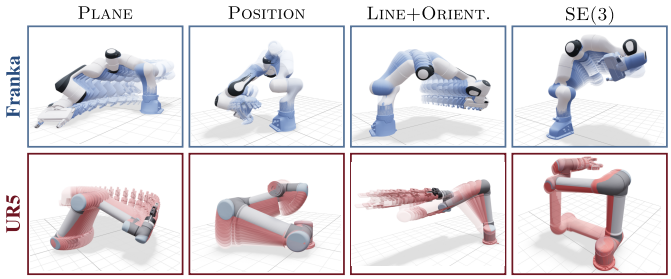

Fig. 4: **Visualization of linear interpolation between IK solutions in latent space.** End-effector poses of the generated configurations are plotted in task space; opacity encodes the interpolation parameter, fading smoothly from one endpoint (fully opaque) to the other (fully transparent). The generated trajectory remains close to $\mathcal{M}_{\mathbf{c}}$ across constraint types and both robots.

giving the directional sensitivity

$$\rho(\alpha) = \|Dg_{\mathbf{c}}(\mathbf{x}(\alpha))\,\hat{\mathbf{v}}\|_2. \tag{7}$$

Because $\mathbf{q}$ is expressed in raw joint angles, ρ has units of radians per latent unit; small values mean a unit latent step produces only a small change in joint space. We also report the corresponding joint-space speed with respect to the interpolation parameter,

$$\nu(\alpha) = \|Dg_{\mathbf{c}}(\mathbf{x}(\alpha))(\mathbf{x}_b - \mathbf{x}_a)\|_2, \tag{8}$$

which measures how quickly $\mathbf{q}$ changes as α moves from one endpoint to the other. Across all conditions, the median directional sensitivity ρ_{med} stays below 1 radian per latent unit, with an overall median of 0.350. The per-condition median of ν ranges from 0.03 to 1.45 radians per unit interpolation parameter, with an overall median of 0.82 across both manipulators (Table II). Notably, both the sensitivity and the speed for the UR5 under SE(3) remain close to zero, indicating that small movements in the latent space stay within a single discrete branch.

Small joint-space motion alone does not imply that the interpolation respects the constraint. We therefore ask whether the direction the decoded configuration moves in preserves

TABLE II: **Median joint-space sensitivity and tangent overlap along latent interpolation paths.**

	Constraint	Sensitivity		Tangent overlap			
		ρ_{med}	ν_{med}	$\tau^{\mathbf{q}}_{\text{med}}$	$\tau^{\mathbf{q}}_{\text{p5}}$	$\tau^{\mathbf{q}}_{\text{rand}}$	$\Delta^{\mathbf{q}}_{\text{rel}}$
FRANKA	POS	0.540	0.78	1.000	0.999	0.756	+32%
	PLANE	0.244	1.20	1.000	0.868	0.926	+8%
	LINE	0.299	0.90	0.997	0.794	0.845	+18%
	ORIENT	0.474	0.71	0.998	0.949	0.756	+32%
	PLANE+ORIENT.	0.169	0.82	0.995	0.875	0.655	+52%
	LINE+ORIENT.	0.221	0.78	0.987	0.775	0.655	+51%
	SE(3)	0.647	0.82	1.000	0.998	0.378	+165%
UR5	POS	0.703	1.05	1.000	0.999	0.707	+41%
	PLANE	0.441	0.97	1.000	0.988	0.913	+10%
	LINE	0.388	1.45	0.998	0.846	0.816	+22%
	ORIENT	0.356	0.62	0.997	0.931	0.707	+41%
	PLANE+ORIENT.	0.343	0.81	0.997	0.830	0.577	+73%
	LINE+ORIENT.	0.304	1.19	0.961	0.472	0.577	+66%
	SE(3)	0.010	0.03	0.000	0.000	0.000	–

the constraint error to first order. Let $h_{\mathbf{c}}$ denote the constraint error map, with $\mathcal{M}_{\mathbf{c}} = \{\mathbf{q} : h_{\mathbf{c}}(\mathbf{q}) = 0\}$, and let $\mathbf{u}(\alpha) = Dg_{\mathbf{c}}(\mathbf{x}(\alpha))\,\widehat{\mathbf{v}}$ be the decoded joint-space direction at $\mathbf{q}(\alpha) = g_{\mathbf{c}}(\mathbf{x}(\alpha))$. We measure the fraction of $\mathbf{u}(\alpha)$ lying in the null space of the constraint Jacobian,

$$\tau^{\mathbf{q}}(\alpha) = \frac{\|\Pi_{\ker Dh_{\mathbf{c}}(\mathbf{q}(\alpha))}\mathbf{u}(\alpha)\|_2}{\|\mathbf{u}(\alpha)\|_2}, \tag{9}$$

where $\Pi_{\ker Dh_{\mathbf{c}}(\mathbf{q}(\alpha))}$ denotes orthogonal projection onto that null space. Intuitively, $\tau^{\mathbf{q}}$ measures how much of the generated configuration moves *along* the constraint manifold rather than *away* from it. A value near one means the direction lies in the constraint tangent space, leaving any existing constraint error unchanged; when $\mathbf{q}(\alpha)$ lies on $\mathcal{M}_{\mathbf{c}}$, this null space coincides with $\mathcal{T}_{\mathbf{q}(\alpha)}\mathcal{M}_{\mathbf{c}}$. For comparison, a random unit direction in $\mathbb{R}^n$ has expected squared projected length $(n-k)/n$ onto an $(n-k)$-dimensional tangent space, giving the root-mean-square baseline $\tau^{\mathbf{q}}_{\text{rand}} = \sqrt{(n-k)/n}$. We defer full definitions and derivations of ρ, ν, and $\tau^{\mathbf{q}}$ to Appendix E-D.

Empirically, $\mathbf{u}(\cdot)$ is small and aligns closely with the analytical tangent space of $\mathcal{M}_{\mathbf{c}}$, as measured respectively by $\rho = \|\mathbf{u}(\cdot)\|$ and $\tau^{\mathbf{q}}$ in Table II. The median tangent overlap is $\tau^{\mathbf{q}}_{\text{med}} \approx 1.00$ across both manipulators, and all conditions with ID > 0 exceed the random-direction. UR5 under SE(3) is excluded since its tangent space is zero-dimensional, making the tangent overlap zero by construction; the disconnected-branch behaviour observed above is consistent with this. The medians remain close to the constraint tangent space, but the lower tail is weaker; in some conditions the 5th-percentile overlap $\tau^{\mathbf{q}}_{p5}$ falls slightly below the random baseline, reflecting a minority of interpolation directions no better than random at staying tangent. Since the tangent space is evaluated at the sample $\mathbf{q}(\alpha)$, which need not lie on $\mathcal{M}_{\mathbf{c}}$, $\tau^{\mathbf{q}} \approx 1$ indicates that the generated motion stays tangent to $\mathcal{M}_{\mathbf{c}}$ at a fixed offset, preserving whatever constraint error is already present. Together, the small ρ and the large $\tau^{\mathbf{q}}$ suggest that linear interpolation in the latent space succeeds because it produces joint-space motion that changes slowly and largely follows the local linearized constraint geometry of $\mathcal{M}_{\mathbf{c}}$, even when the generated configurations are not exactly feasible.

IV. Discussion and Limitations

We studied the latent-space geometry of a conditional diffusion models trained across seven task-space constraints on two manipulators, using the analytically known constraint manifold as ground truth. The intrinsic dimension recovered from the model's score function matches the analytical degrees of freedom for every positive-dimensional condition, and a small modification of the estimator extends this to the degenerate case in which the solution set collapses to a finite collection of disconnected IK branches. Beyond dimension, linear interpolation in the partial-DDIM latent space decodes into joint-space paths that stay close to the constraint manifold and move along directions aligned with its tangent space. Together these results indicate that the model captures aspects of each manifold's local geometry, not only its dimension, in a setting where that geometry is known rather than estimated.

Limitations: Our analysis treats each constraint condition as having a single, fixed dimension and tangent direction, whereas these can change near joint limits and singular configurations; handling that variation is left to future work. Our interpolation result is also not uniform: on a few constraints, a small number of latent directions drift off the manifold about as much as random ones would. Because we use straight-line paths in latent space, accuracy drops as the two interpolated solutions grow farther apart, where the manifold curves too much for a linear path to follow; geodesic alternatives are a natural next step. Finally, our empirical scope covers only two manipulators, seven constraint families, and one model per robot, leaving more complex robots and other generative models open for analysis.

References

[1] Barrett Ames, Jeremy Morgan, and George Konidaris. IKFlow: Generating diverse inverse kinematics solutions. *IEEE Robotics and Automation Letters*, 7(3):7177–7184, 2022.

[2] Simone Azeglio and Arianna Di Bernardo. What's inside your diffusion model? a score-based Riemannian metric to explore the data manifold. *arXiv preprint arXiv:2505.11128*, 2025.

[3] Raphael Bensadoun, Shir Gur, Nitsan Blau, and Lior Wolf. Neural inverse kinematic. In *International Conference on Machine Learning*, pages 1787–1797. PMLR, 2022.

[4] Bradley C.A. Brown, Anthony L. Caterini, Brendan Leigh Ross, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. Verifying the union of manifolds hypothesis for image data. In *International Conference on Learning Representations*, 2023.

[5] Joel W. Burdick. On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. In *International Conference on Advanced Robotics*, pages 25–34. Springer, 1989.

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[7] Willem Diepeveen, Georgios Batzolis, Zakhar Shumaylov, and Carola-Bibiane Schönlieb. Score-based pullback Riemannian geometry: Extracting the data manifold geometry using anisotropic flows. In *International Conference on Machine Learning*, 2025.

[8] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140, 2017.

[9] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[10] Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on computers*, 100(2):176–183, 1971.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[12] Zachary Kingston, Mark Moll, and Lydia E. Kavraki. Exploring implicit spaces for constrained sampling-based planning. *The International Journal of Robotics Research*, 38(10-11):1151–1178, 2019.

[13] Henry Kvinge, Davis Brown, and Charles Godfrey. Exploring the representation manifolds of Stable Diffusion through the lens of intrinsic dimension. In *ICLR Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[14] John M. Lee. *Introduction to Smooth Manifolds*. Springer, 2 edition, 2012.

[15] Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

[16] Oliver Limoyo, Filip Marić, Matthew Giamou, Petra Alexson, Ivan Petrović, and Jonathan Kelly. Generative graphical inverse kinematics. *IEEE Transactions on Robotics*, 41:1002–1018, 2024.

[17] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[18] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 2017.

[19] Suhan Park, Mathew Schwartz, and Jaeheung Park. NodeIK: Solving inverse kinematics with neural ordinary differential equations for path planning. In *International Conference on Control, Automation and Systems*, pages 944–949, 2022.

[20] Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of Riemannian geometry. In *Advances in Neural Information Processing Systems*, volume 36, pages 24129–24142, 2023.

[21] Donald Lee Pieper. *The Kinematics of Manipulators Under Computer Control*. Stanford University, 1969.

[22] Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021.

[23] Madhusudan Raghavan and Bernard Roth. Inverse kinematics of the general 6R manipulator and related linkages. *Journal of Mechanical Design*, 115(3):502–508, 1993.

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[25] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009.

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[27] 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*, 2021.

[28] Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *International Conference on Machine Learning*, 2024.

[29] Zeyu Zhang and Ziyuan Jiao. IKDiffuser: A diffusion-based generative inverse kinematics solver for kinematic trees. *arXiv preprint arXiv:2506.13087*, 2025.

APPENDIX A
NOTATION

TABLE III: Summary of notation used throughout the paper.

Symbol	Meaning
Manipulator and kinematics	
$\mathbf{q}$	joint configuration of the manipulator
n	number of actuated joints (6 for the UR5, 7 for the Franka); the same symbol also denotes the dimension of the sine–cosine score space (12 and 14) on which Score-SVD acts
$f : \mathbb{R}^n \to \mathrm{SE}(3)$	forward kinematics map of the manipulator
$\mathrm{SE}(3),\ \mathrm{SO}(3)$	special Euclidean and special orthogonal groups
$\mathbf{p},\ \mathbf{R}$	end-effector position and orientation, $f(\mathbf{q}) = (\mathbf{p}(\mathbf{q}), \mathbf{R}(\mathbf{q}))$
$p_x,\ p_y,\ p_z$	end-effector position coordinates
$\mathbf{e}_x,\ \mathbf{e}_y,\ \mathbf{e}_z$	end-effector body axes
$d(\mathbf{R}, \mathbf{R}^\star)$	geodesic angle between two orientations on $\mathrm{SO}(3)$
$\mathbf{c}$	conditioning input: constraint family and target
$\mathbf{p}^\star,\ \mathbf{R}^\star$	target end-effector position and orientation
$\mathbf{m}$	selection vector $(m_1, \ldots, m_6) \in \{0,1\}^6$ marking the constrained coordinates ($k = \|\mathbf{m}\|_1$)
$\mathcal{S}_\mathbf{c}$	task-space constraint set in $\mathrm{SE}(3)$
$\mathcal{M}_\mathbf{c}$	constraint manifold in configuration space
k	codimension; number of fixed task-space coordinates
$n - k$	intrinsic dimension of the constraint manifold
ID	intrinsic dimension $n - k$; abbreviation used in tables and figures
$h_\mathbf{c}$	constraint error map; zero set is $\mathcal{M}_\mathbf{c}$
$Dh_\mathbf{c}$	Jacobian of the constraint map
$\mathcal{T}_\mathbf{q}\mathcal{M}_\mathbf{c}$	tangent space of $\mathcal{M}_\mathbf{c}$ at a configuration
$\mathcal{N}_{\mathbf{x}_0}\mathcal{M}_\mathbf{c}$	normal space of $\mathcal{M}_\mathbf{c}$ at a sample
$\Pi_{\ker Dh_\mathbf{c}}$	orthogonal projector onto the tangent space
Diffusion model	
$\mathbf{x}_0,\ \mathbf{x}_t$	clean and noised samples at time t
$\mathcal{X}$	latent (noise) space of the diffusion model
$t,\ T$	diffusion timestep and maximum timestep
$\bar{\alpha}_t$	variance schedule at time t
$\boldsymbol{\epsilon}$	injected Gaussian noise
$\hat{\boldsymbol{\epsilon}}_\theta$	denoising network with parameters θ
s_θ	score of the conditional noised marginal
$p_t(\mathbf{x}_t \mid \mathbf{c})$	conditional noised marginal density
$\hat{\mathbf{x}}_0$	predicted clean sample from a noised one
$g_\mathbf{c},\ g_\mathbf{c}^{-1}$	DDIM decoder and its inverse (inversion)
$\bar{t}$	intermediate noise level for partial inversion
N	number of training configurations (dataset size)
Dataset and training	
$\mathbf{q}_\mathrm{rand}$	random joint configuration used to sample reachable target poses
$r,\ r_\mathrm{min},\ r_\mathrm{max}$	cylindrical workspace radius $r = \sqrt{x^2 + y^2}$ and its bounds
$z_\mathrm{min},\ z_\mathrm{max}$	workspace height bounds
$\phi(\mathbf{q})$	sine–cosine joint encoding $(\sin\mathbf{q}, \cos\mathbf{q})$, of dimension $2n$
$(\hat{\mathbf{a}}, \hat{\mathbf{b}})$	model estimates of $(\sin\mathbf{q}, \cos\mathbf{q})$ at inference
$\beta,\ \beta_\mathrm{min},\ \beta_\mathrm{max}$	per-step forward-process noise variance (sigmoid schedule) and its range
η	DDIM stochasticity parameter ($\eta = 0$, deterministic)
Intrinsic dimension estimation	
K	number of score perturbations
K_s	number of samples drawn per target ($K_s = 5$)
S	stacked score matrix, $S \in \mathbb{R}^{n \times K}$
σ_i	singular values of S, in descending order
$\hat{d},\ \hat{k}$	estimated intrinsic dimension and codimension
σ_{n+1}	virtual zero singular value in Score-SVD†
NN	neighbourhood size of the local MLE and PCA baselines
Latent interpolation and local flattening	
$\mathbf{x}_a,\ \mathbf{x}_b$	latent-space interpolation endpoints
α	interpolation parameter in $[0, 1]$
$\mathbf{x}(\alpha)$	linear interpolation path in latent space
$\mathbf{q}(\alpha)$	decoded joint configuration along the path, $g_\mathbf{c}(\mathbf{x}(\alpha))$
$\widehat{\mathbf{v}}$	unit direction between latent endpoints
$Dg_\mathbf{c}$	Jacobian of the DDIM decoder, $Dg_\mathbf{c} \in \mathbb{R}^{n \times 2n}$
$\rho(\alpha)$	decoder directional sensitivity
$\nu(\alpha)$	joint-space speed along the path
$\mathbf{u}(\alpha)$	decoded joint-space direction
$\tau^\mathbf{q}(\alpha)$	tangent overlap of the decoded direction
$\tau^\mathbf{q}_\mathrm{rand}$	random-direction tangent-overlap baseline, $\sqrt{(n-k)/n}$
$\mathbf{r}$	random unit vector on the sphere S^{n-1}
S^{n-1}	unit sphere in joint space
$\rho_\mathrm{med},\ \nu_\mathrm{med},\ \tau^\mathbf{q}_\mathrm{med}$	median sensitivity, speed, and tangent overlap

continued on next page

TABLE III – continued from previous page

Symbol	Meaning
$\tau^{\mathbf{q}}_{\mathrm{p5}}$	5th-percentile tangent overlap (lower tail)
$\Delta^{\mathbf{q}}_{\mathrm{rel}}$	relative tangent-overlap improvement over baseline
lin, sph	straight-line and spherical (slerp) latent interpolation
Constraint families (Table I)	
PLANE	one position coordinate fixed ($k = 1$)
LINE	two position coordinates fixed ($k = 2$)
POSITION	position fixed, orientation free ($k = 3$)
ORIENT.	orientation fixed, position free ($k = 3$)
PLANE+ORIENT.	one position coordinate and full rotation ($k = 4$)
LINE+ORIENT.	two position and two rotation coordinates ($k = 4$)
SE(3)	full end-effector pose fixed ($k = 6$)

APPENDIX B
MATHEMATICAL BACKGROUND

This section provides the geometric and kinematic preliminaries underlying the analysis in our work. We first review the smooth-manifold machinery used to equip the constraint manifold $\mathcal{M}_{\mathbf{c}}$ with a dimension, a tangent space, and a normal space (Appendix B-A). We then define the seven task-space constraints of Table I precisely, specifying the task-space set $\mathcal{S}_{\mathbf{c}}$ and the constraint error map $h_{\mathbf{c}}$ for each (Appendix B-B).

A. Smooth manifolds

Every constraint manifold in this paper is the solution set of a system of smooth equations, so we briefly recall the geometry of such sets [14]. A smooth manifold is a topological space that is locally diffeomorphic to a Euclidean space. A subset $\mathcal{M} \subseteq \mathbb{R}^n$ is an *embedded submanifold* of codimension k if, near each of its points, it is the common zero set of k *local defining functions*, smooth scalar functions whose differentials are linearly independent, so that $\mathcal{M}$ inherits the smooth structure of the ambient space and has intrinsic dimension $n - k$. Each constraint manifold we study takes exactly this form, so its dimension and tangent geometry follow directly from the defining functions.

Let $h_{\mathbf{c}} : \mathbb{R}^n \to \mathbb{R}^k$ be smooth with $k \leq n$. A value is *regular* if the Jacobian $Dh_{\mathbf{c}}(\mathbf{q}) \in \mathbb{R}^{k\times n}$ has full row rank k at every $\mathbf{q}$ in its preimage. When $\mathbf{0}$ is a regular value, the regular value theorem, a consequence of the implicit function theorem, makes the preimage

$$\mathcal{M} = \{\mathbf{q} \in \mathbb{R}^n : h_{\mathbf{c}}(\mathbf{q}) = \mathbf{0}\} = {h_{\mathbf{c}}}^{-1}(\mathbf{0}) \tag{10}$$

an embedded submanifold of $\mathbb{R}^n$ of dimension $n - k$ [14]. The codimension k equals the number of independent scalar constraints imposed by $h_{\mathbf{c}}$.

At a point $\mathbf{q} \in \mathcal{M}$, the tangent space is the kernel of the Jacobian,

$$\mathcal{T}_{\mathbf{q}}\mathcal{M} = \ker Dh_{\mathbf{c}}(\mathbf{q}) \subseteq \mathbb{R}^n, \tag{11}$$

a linear subspace of dimension $n - k$ collecting the velocities along which $h_{\mathbf{c}}$ is unchanged to first order. Its orthogonal complement is the normal space $\mathcal{N}_{\mathbf{q}}\mathcal{M} = (\mathcal{T}_{\mathbf{q}}\mathcal{M})^{\perp}$, the k-dimensional row space of $Dh_{\mathbf{c}}(\mathbf{q})$, equivalently the image of ${Dh_{\mathbf{c}}(\mathbf{q})}^{\mathsf{T}}$. The orthogonal projector onto the tangent space admits the closed form

$$\Pi_{\ker Dh_{\mathbf{c}}(\mathbf{q})} = \mathbf{I} - {Dh_{\mathbf{c}}(\mathbf{q})}^{\mathsf{T}}\Big(Dh_{\mathbf{c}}(\mathbf{q})\,{Dh_{\mathbf{c}}(\mathbf{q})}^{\mathsf{T}}\Big)^{-1} Dh_{\mathbf{c}}(\mathbf{q}), \tag{12}$$

which is well defined at every regular point, since $Dh_{\mathbf{c}}(\mathbf{q})\,{Dh_{\mathbf{c}}(\mathbf{q})}^{\mathsf{T}}$ is invertible.

These three objects are exactly what we measure against the trained model. The dimension $n - k$ is the analytical intrinsic dimension recovered by the score-based estimator of Section III-A, the normal space $\mathcal{N}_{\mathbf{q}}\mathcal{M}$ is the subspace that the score spans in Score-SVD, and the projector (12) defines the tangent overlap $\tau^{\mathbf{q}}$ of Section III-B.

B. Precise constraint definitions

We write the forward kinematics in components as $f(\mathbf{q}) = (\mathbf{p}(\mathbf{q}), \mathbf{R}(\mathbf{q}))$, with end-effector position $\mathbf{p}(\mathbf{q}) \in \mathbb{R}^3$ and orientation $\mathbf{R}(\mathbf{q}) \in \mathrm{SO}(3)$. A constraint $\mathbf{c} = (\mathbf{p}^\star, \mathbf{R}^\star, \mathbf{m})$ pairs a target pose $(\mathbf{p}^\star, \mathbf{R}^\star) \in \mathrm{SE}(3)$ with a *selection vector*

$$\mathbf{m} = (m_1, \ldots, m_6) \in \{0, 1\}^6,$$

whose entries indicate which task-space coordinates are constrained: $m_i = 1$ holds the i-th coordinate at its target value, while $m_i = 0$ leaves it free. The first three entries correspond to the position coordinates p_x, p_y, p_z and the last three to the end-effector's body axes $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$. Varying $\mathbf{m}$ thus yields the seven constraints in our family; how $\mathbf{m}$ enters the conditioning vector is described in Appendix C-C.

a) Position: Each position entry acts independently on its own axis. A fixed entry ($m_i = 1$) imposes $p_i(\mathbf{q}) = p_i^\star$, contributing the scalar residual $p_i(\mathbf{q}) - p_i^\star$; a free entry leaves that axis unconstrained.

b) Orientation: The three orientation entries are read jointly rather than per axis, and their effect depends only on how many are set.

If all three are set, the orientation is fully fixed, $\mathbf{R}(\mathbf{q}) = \mathbf{R}^\star$, removing all three rotational degrees of freedom. The error is measured by the angle between the current and target orientations,

$$d(\mathbf{R}(\mathbf{q}), \mathbf{R}^\star) = \arccos\Big(\tfrac{1}{2}\Big(\mathrm{tr}\Big({\mathbf{R}^\star}^{\mathsf{T}}\mathbf{R}(\mathbf{q})\Big) - 1\Big)\Big), \tag{13}$$

which is zero exactly when the two orientations coincide.

If two are set, one entry is free; let $\mathbf{e}_j$ be the body axis it indexes. The constraint fixes the spatial direction of that axis, $\mathbf{R}(\mathbf{q})\,\mathbf{e}_j = \mathbf{R}^\star\mathbf{e}_j$, removing two degrees of freedom while leaving rotation about $\mathbf{e}_j$ free. The error is simply the angle between the current and target directions of that axis. There is no one-entry case, since a single rotational coordinate has no axis-independent meaning.

Stacking the constrained residuals gives the constraint-error map $h_{\mathbf{c}} : \mathbb{R}^n \to \mathbb{R}^k$, where $k = \|\mathbf{m}\|_1$ counts the constrained

TABLE IV: **Precise definitions of the seven task-space constraints.** For each constraint we list the selection vector $\mathbf{m} = (m_1\, m_2\, m_3 \mid m_4\, m_5\, m_6)$, the constrained task-space set $\mathcal{S}_\mathbf{c}$, and the codimension $k = \|\mathbf{m}\|_1$. This table complements Table I, which reports the resulting intrinsic dimension $n - k$ for each manipulator.

Constraint	Selection $\mathbf{m}$	Constrained quantities ($\mathcal{S}_\mathbf{c}$)	k
PLANE	001 | 000	$p_z = p_z^\star$	1
LINE	011 | 000	$p_y = p_y^\star,\ p_z = p_z^\star$	2
POSITION	111 | 000	$\mathbf{p} = \mathbf{p}^\star$	3
ORIENTATION	000 | 111	$\mathbf{R} = \mathbf{R}^\star$	3
PLANE + ORIENTATION	001 | 111	$p_z = p_z^\star,\ \mathbf{R} = \mathbf{R}^\star$	4
LINE + ORIENTATION	011 | 110	$p_y = p_y^\star,\ p_z = p_z^\star,\ \mathbf{R}\mathbf{e}_z = \mathbf{R}^\star\mathbf{e}_z$	4
SE(3)	111 | 111	$\mathbf{p} = \mathbf{p}^\star,\ \mathbf{R} = \mathbf{R}^\star$	6

coordinates: one per constrained position axis, plus two or three for orientation depending on how many orientation entries are set. The configurations that satisfy the constraint form the constraint manifold $\mathcal{M}_\mathbf{c} = {h_\mathbf{c}}^{-1}(\mathbf{0})$, i.e., those $\mathbf{q}$ whose end-effector pose agrees with the target $(\mathbf{p}^\star, \mathbf{R}^\star)$ on the constrained coordinates. Wherever $\mathbf{0}$ is a regular value of $h_\mathbf{c}$, $\mathcal{M}_\mathbf{c}$ is a smooth manifold of dimension $n - k$ (Appendix B-A).

We report the constraint residual through the constrained error $h_\mathbf{c}(\mathbf{q})$. Because its position and orientation parts carry different units, we report them separately: the position part as the Euclidean error over the constrained axes in millimetres, and the orientation part as the full geodesic angle (13) when the orientation is fully fixed, or the single axis angle when only an axis direction is constrained.

Table IV lists the seven constraints together with their selection vectors and constrained sets. Every constraint except LINE + ORIENTATION leaves the orientation either fully fixed or fully free. LINE + ORIENTATION is the only case that constrains just an axis direction: it fixes the two position coordinates of the line and the direction of the end-effector z-axis, leaving translation along the line and rotation about that axis free. The codimension k is a property of the constraint alone; the intrinsic dimension $n-k$ also depends on the number of joints n, so the redundant 7-DoF Franka keeps one extra self-motion DoF relative to the 6-DoF UR5 under the same constraint (Table I).

APPENDIX C
DATASET AND TRAINING

This section details how the training data is generated and how the diffusion model is trained. We first describe the analytical inverse-kinematics solver and how the discrete solution branches and continuous self-motion manifold are recovered (Appendix C-A), then the sampling of task-space targets for each constraint family (Appendix C-B), and finally the network architecture and optimization (Appendix C-C) that underlie the main results and every model in Appendix D.

A. IK solver and solution recovery

Our training data consists of joint configurations that satisfy a given task-space constraint, obtained by solving inverse kinematics for sampled task-space targets (the sampling procedure is described in Appendix C-B). We use the analytical solver `IKFast`, which symbolically solves the inverse-kinematics equations of a fixed kinematic chain and returns all closed-form solution branches for a queried end-effector pose. An analytical solver is preferable to a numerical one here because, for each queried pose, it enumerates the discrete IK branches exhaustively and without an initial guess, so the recovered configurations are not tied to the basin of a particular seed. For a full $\mathrm{SE}(3)$ constraint this returns the solution set of $\mathcal{M}_\mathbf{c}$ exactly; for the partial constraints, where the free task-space DoFs are sampled or swept on a grid (Appendix C-B) and near-duplicates are removed from the pooled solutions, it yields a finite sampled approximation of $\mathcal{M}_\mathbf{c}$.

Recovering the configurations for a single target differs between the two manipulators by their kinematic redundancy. For the UR5, a full $\mathrm{SE}(3)$ pose is reached by at most eight discrete configurations; one solver call returns this finite branch set. For the Franka, the redundant joint promotes the solution set of a fixed pose to a one-dimensional self-motion manifold, which a single analytical query cannot return in closed form. We therefore choose the seventh joint to parameterize the redundancy and sweep it over 50 values spaced uniformly across its joint-limit interval $[-2.9671, 2.9671]$ rad, solving analytically with this joint held fixed at each value and collecting the in-limit solutions; this samples the self-motion manifold while still enumerating the discrete branches at every sweep value. This sweep provides a discretization of the self-motion manifold under the chosen free-joint parameterization; it is not assumed to be a globally uniform parameterization of the manifold, and may under-sample narrow components or regions near singular parameterizations.

The procedure above can return many near-identical configurations, both from the redundant-joint sweep and from the over-sampling of task-space targets (Appendix C-B). We therefore remove near-duplicate configurations in joint space: a candidate is kept only if its ℓ_2 distance to every already-kept configuration exceeds a tolerance of 0.05 rad. This distance is computed on the raw joint angles and therefore does not account for angle wrap-around, so a pair of physically close configurations straddling the $0/2\pi$ boundary of a UR5 joint would be treated as roughly 2π apart and both kept. In practice the analytical solver returns each physical solution once within a fixed joint interval, which spans less than 2π for the Franka and follows the native $[0, 2\pi)$ convention for the UR5, so such boundary-

TABLE V: **Reachable-workspace bounds used for task-space target sampling.** Targets are drawn within a cylindrical shell: radius $r = \sqrt{x^2 + y^2} \in [r_{\min}, r_{\max}]$ and height $z \in [z_{\min}, z_{\max}]$, in metres. Sampled positions outside these bounds are rejected.

Manipulator	$r_{\min}$	$r_{\max}$	$z_{\min}$	$z_{\max}$
UR5	0.20	0.70	-0.10	0.80
Franka	0.10	0.85	-0.10	1.10

split near-duplicates are rare; in any case, before training every configuration is mapped to the sine–cosine representation of Appendix C-C, which is free of the 2π wrap-around, so any residual boundary pair introduces no discontinuity for the model.

B. Task-space target sampling

Each target fixes its constrained task-space DoFs at the values of a sampled pose $(\mathbf{p}^\star, \mathbf{R}^\star)$, according to the selection vector $\mathbf{m}$ (Appendix B-B). Generating training data for a family therefore consists of (i) sampling a target by choosing values for its constrained DoFs, and (ii) populating the corresponding manifold $\mathcal{M}_\mathbf{c}$ by sweeping the free DoFs and solving IK at each, with near-duplicates removed from the recovered configurations as in Appendix C-A.

Two conventions are shared across all families. First, every sampled end-effector position is constrained to the reachable shell of Table V, and positions falling outside it are rejected; this keeps targets within the manipulator's workspace and away from near-base and near-singular regions. Second, sampling a target orientation $\mathbf{R}^\star$ directly in $\mathrm{SO}(3)$ would frequently yield poses unreachable for the non-redundant UR5; we instead obtain orientations as the rotation component $\mathbf{R}(\mathbf{q}_{\text{rand}})$ of the forward kinematics of random joint configurations, which biases sampled orientations toward the robot's reachable orientation set. This restricts each orientation to one the arm reaches in some configuration, but does not guarantee that the orientation is jointly reachable with a separately sampled position or partial-position constraint, so a small fraction of full poses remain infeasible and are removed by the joint-limit filtering of Appendix C-A.

The per-family data-generation procedures are summarized in Table VI. A target fixes the constrained DoFs that define $\mathbf{c}$, drawn across targets on a uniform grid (e.g. the plane height $p_z^\star$), by rejection sampling in the shell, or as $\mathbf{R}(\mathbf{q}_{\text{rand}})$ for orientations, while the remaining free DoFs are swept on a grid (ORIENTATION, LINE+ORIENTATION) or sampled at random (POSITION, PLANE, LINE, PLANE+ORIENTATION) only to generate the full-pose IK queries whose solutions populate $\mathcal{M}_\mathbf{c}$. The per-family counts in the final column fall below the product of the target count and per-target cap, because the joint-space near-duplicate removal of Appendix C-A and the joint-limit filtering together discard near-identical and out-of-range solutions; the reduction is largest for ORIENTATION and LINE+ORIENTATION, and is more pronounced on the Franka, whose tighter joint limits reject more candidate solutions.

C. Network and training details

The network is conditioned on the constraint $\mathbf{c} = (\mathbf{p}^\star, \mathbf{R}^\star, \mathbf{m})$ of Appendix B-B, encoded as a 15-dimensional conditioning vector that captures both the target and the constraint family. Its first three entries hold the target position $\mathbf{p}^\star$, normalized to the reachable shell of Table V; the next six encode the target orientation $\mathbf{R}^\star$ in a continuous 6-D rotation representation, which avoids the discontinuities of Euler-angle or quaternion parameterizations. The final six entries are the selection vector $\mathbf{m}$ (Appendix B-B), which identifies the constraint family of Table I. For families that leave position partially free, the unconstrained entries of $\mathbf{p}^\star$ are set to zero, so the network is conditioned only on the coordinates the constraint actually fixes.

The model is an *endpoint* diffusion model: it learns a distribution over the joint configuration $\mathbf{q}$ that solves the constraint, rather than over a task-space trajectory. To remove the 2π wrap-around discontinuity of raw joint angles, each configuration is mapped to a sine–cosine encoding $\phi(\mathbf{q}) = (\sin\mathbf{q}, \cos\mathbf{q}) \in \mathbb{R}^{2n}$, giving a 12-dimensional input for the 6-DoF UR5 and 14-dimensional for the 7-DoF Franka. The encoded endpoints are standardized to zero mean and unit variance per coordinate before diffusion. The network is trained with the noise-prediction (ϵ) objective of Ho et al. [11], minimizing the mean-squared error between the sampled noise and the network output. At inference we draw a sample and invert the per-coordinate standardization, giving a vector $(\hat{\mathbf{a}}, \hat{\mathbf{b}}) \in \mathbb{R}^{2n}$ whose two halves are the model's estimates of $(\sin\mathbf{q}, \cos\mathbf{q})$. These coordinates are unconstrained and need not lie on the unit circle, so we recover each joint angle as the polar angle of the corresponding generated 2-vector, $q_j = \mathrm{atan2}(\hat{a}_j, \hat{b}_j) \in (-\pi, \pi]$, which depends only on its direction and discards the radial magnitude, equivalently projecting $(\hat{a}_j, \hat{b}_j)$ onto the unit circle before decoding. The corresponding task-space pose follows from the forward kinematics $f(\mathbf{q})$.

a) Architecture: The denoiser is a conditional residual multilayer perceptron rather than a convolutional U-Net, since the endpoint carries no temporal structure to exploit. The noisy input is projected to a hidden width of 512 and passed through 10 residual blocks, each applying a pre-LayerNorm, a feature-wise linear modulation (FiLM) of the normalized activations, and a two-layer MLP with a $4\times$ inner expansion and Mish activations; a final LayerNorm and linear layer map back to the input dimension. The diffusion timestep t is embedded with a sinusoidal positional encoding followed by a two-layer MLP, the conditioning vector $\mathbf{c}$ is embedded by a separate three-layer MLP, and the two embeddings (each of width 384) are summed to form the modulation signal that drives the FiLM scale and shift in every block. This conditioning scheme lets a single network serve all seven constraint families. The selection vector $\mathbf{m}$ tells the model which task-space coordinates are active, so one set of weights covers the whole family of Table I. The model has 25.6M parameters.

TABLE VI: **Per-family data-generation procedures.** Each row defines a target by fixing the variables in column 2, drawn across targets on a uniform grid, by rejection sampling in the reachable shell of Table V, or from the forward kinematics of a random configuration, as the orientation $\mathbf{R}(\mathbf{q}_{\text{rand}})$ or the full pose $f(\mathbf{q}_{\text{rand}})$. The remaining task-space DoFs (column 3) are sampled or swept only to generate full-pose IK queries whose solutions lie on $\mathcal{M}_{\mathbf{c}}$. Column 4 is the number of targets and column 5 the per-target cap on the number of configurations kept under the greedy near-duplicate removal; for the SE(3) constraint no cap is applied and all enumerated branches are included. The final column reports the number of configurations kept per family after joint-limit filtering and near-duplicate removal.

Constraint	Fixed	Free DoFs (sampled / swept)	# targ.	Cap/targ.	Total* ($\times 10^3$)
PLANE	$p_z^\star$	(p_x, p_y) in shell, $\mathbf{R}$ random	400	600	240 / 240
LINE	$(p_y^\star, p_z^\star)$	p_x along line, $\mathbf{R}$ random	600	400	239 / 211
POSITION	$\mathbf{p}^\star$	$\mathbf{R}$ random orientations	3000	≤ 200	546 / 577
ORIENTATION	$\mathbf{R}^\star$	$\mathbf{p}$ grid ball about seed	800	600	274 / 106
PLANE+ORIENT.	$(p_z^\star, \mathbf{R}^\star)$	(p_x, p_y) in shell	1000	300	300 / 299
LINE+ORIENT.	$(p_y^\star, p_z^\star, \mathbf{R}^\star\mathbf{e}_z)$	p_x, rotation about $\mathbf{e}_z$ (30×30 grid)	800	500	240 / 95
SE(3)	$(\mathbf{p}^\star, \mathbf{R}^\star)$	none (discrete branches)	8000	all branches	62 / 255

*Configurations kept after joint-limit filtering and near-duplicate removal, reported as UR5 / Franka; each full-pose IK query may return several IK branches, so these totals fall below the product of columns 4 and 5.

TABLE VII: **Network and training hyperparameters** (shared across both manipulators; only the input dimension differs).

Hyperparameter	Value
Hidden width	512
Residual blocks (depth)	10
MLP inner expansion	4×
Embedding width (time, cond)	384
Parameters	25.6M
Diffusion steps T	100
Noise schedule	sigmoid ($\beta \in [10^{-4}, 2 \times 10^{-2}]$)
Sampler	DDIM, 100 steps, $\eta = 0$
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	10^{-5}
Batch size	512
Training steps	500,000
Gradient clipping	1.0
Conditioning dropout	0.10
EMA decay (warm-up)	0.9995 (1,000 steps)

b) Diffusion and optimization: We use a discrete forward process of $T = 100$ steps whose per-step noise variances follow a sigmoid β schedule, interpolating from $\beta_{\min} = 10^{-4}$ to $\beta_{\max} = 2 \times 10^{-2}$; this schedule gave the lowest constraint error in our ablation. Samples are drawn with a deterministic DDIM sampler [26] using the full 100-step grid. The network is trained for 500,000 steps with AdamW (learning rate 2×10^{-4}, weight decay 10^{-5}) at batch size 512, with gradients clipped to unit norm. Each batch is drawn by a stratified sampler that balances the seven constraint families equally, so no family dominates the gradient. We apply classifier-free conditioning dropout, setting the entire conditioning vector to zero with probability 0.10, and maintain an exponential moving average of the weights (decay 0.9995, after a 1,000-step warm-up) that is used for all evaluation.

D. Training-set size ablation

In image classification, datasets of lower intrinsic dimension have been observed empirically to need fewer training samples to reach a given accuracy [22]. Although higher-dimensional manifolds are often harder to estimate statistically, our data-generation setting has an additional effect: low-codimension constraints leave larger feasible sets per target and are therefore easier to populate with valid IK samples. A constraint family of codimension k leaves an $(n-k)$-dimensional set of solutions for every task-space target, so a higher intrinsic dimension $n-k$ offers a larger family of valid configurations per target and could make the manifold $\mathcal{M}_{\mathbf{c}}$ easier to populate from limited data. We examine whether this holds in our setting by training a separate model on each constraint family, using the architecture and optimization of Appendix C-C, while varying the number of training configurations N from 500 to 40,000. Each model is then evaluated with the constraint-error procedure of Section III, and we report the position and orientation error of its valid samples as a function of N.

Figure 5 shows the resulting error curves, which are broadly consistent with the expectation above on both manipulators, although not as a strict rule. The low-codimension families PLANE (k=1) and LINE (k=2) reach low error almost immediately and gain little from additional data, whereas the high-codimension families PLANE+ORIENTATION and LINE+ORIENTATION (k=4) start with much larger error at N=500 and improve across the whole range, the latter falling from 64 to 21 mm and from 14.6° to 4.6° on the Franka. Ordering the families by codimension thus roughly tracks the amount of data each appears to need to reach a given error, in line with the view that larger solution families are easier to learn.

The two SE(3) conditions differ from this pattern, in opposite directions. On the Franka the redundant joint promotes the fixed pose to a one-dimensional self-motion manifold, so SE(3) behaves more like a low-codimension family despite k=6, with orientation error near its floor of

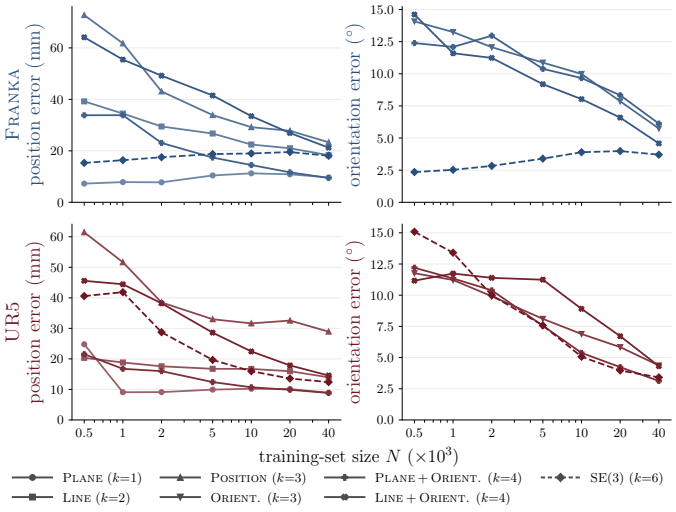

Fig. 5: **Constraint error versus training-set size.** Position error (left) and orientation error (right) of per-family models for Franka (top) and UR5 (bottom), against the number of training configurations N, with one curve per constraint family identified by its marker and drawn as a distinct shade of the manipulator's colour, from light to dark with increasing codimension k; the discrete SE(3) condition is dashed. Low-codimension families reach low error with little data, while high-codimension families improve across the whole range; the two SE(3) conditions are the exceptions, comparatively easy on the redundant Franka and hardest on the non-redundant UR5.

about 2.4° already at N=500 and changing little thereafter. On the UR5 the same condition appears the hardest, since a full pose admits only the eight discrete configurations of a non-redundant arm, so its error is among the largest at small N and approaches the continuous families only by N=40,000. The UR5 POSITION family is a further exception, its error settling above the others at large N rather than continuing to fall, suggesting that training efficiency is not explained by codimension alone, since POSITION shares its codimension with ORIENTATION, which trains without difficulty.

We finally compare the per-family models against the single jointly-trained model of Appendix C-C, which is conditioned on all seven families at once and underlies our main results. Table VIII reports this comparison at the largest training-set size N=40,000. The jointly-trained model achieves lower position and orientation error on every condition, most notably on the UR5 POSITION family, where the position error falls from 28.9 to 11.7 mm. Training across families thus appears to help most on the conditions the per-family models found hardest. The two models also differ in training length and dataset size, so this is a descriptive comparison of the trained models rather than a controlled study of joint versus per-family training.

APPENDIX D
INTRINSIC DIMENSION ESTIMATION

This section expands the intrinsic-dimension study of Section III-A. We first state the full Score-SVD† algorithm with its sample-selection procedure and a worked example of the discrete UR5 SE(3) case (Appendix D-A), then study the sensitivity to its two hyperparameters (Appendix D-B), and describe the neighbourhood-based baselines we compare against (Appendix D-C). We then report the full numerical results and score spectra across all conditions (Appendix D-D), and examine how the constraint error depends on the size of the training set (Appendix C-D).

Algorithm 1 Score-SVD† intrinsic-dimension estimate at a sample $\mathbf{x}_0$. Changes to Algorithm 1 of Stanczuk et al. [28] are shown in teal.

Require: conditional score model $s_\theta(\cdot, \cdot, \mathbf{c})$; sample $\mathbf{x}_0 \in \mathcal{M}_\mathbf{c}$; intermediate noise level t; perturbation count K

$n \leftarrow \dim(\mathbf{x}_0)$; $S \leftarrow [\,]$
for $i = 1, \dots, K$ **do**
  $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t^{(i)} \leftarrow \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}^{(i)}$ ▷ variance-preserving kernel
  $\mathbf{s}^{(i)} \leftarrow -\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t^{(i)}, t, \mathbf{c}) \,/\, \sqrt{1-\bar{\alpha}_t}$ ▷ score from ϵ-prediction
  append $\mathbf{s}^{(i)}$ as a column of S
end for
$\sigma_1 \geq \cdots \geq \sigma_n \leftarrow \mathrm{SVD}(S)$ ▷ singular values of $S \in \mathbb{R}^{n \times K}$
$\sigma_{n+1} \leftarrow 0$ ▷ virtual zero singular value
$\hat{k} \leftarrow \mathrm{argmax}_{i=1,\dots,n}\,(\sigma_i - \sigma_{i+1})$ ▷ search extended from $n-1$ to n
return $\hat{d} \leftarrow n - \hat{k}$

A. Score-SVD† algorithm, sample selection, and example

Section III-A introduced Score-SVD†, our variant of the score-based estimator of Stanczuk et al. [28] that resolves the discrete, ID = 0 case. Algorithm 1 states the full procedure at a single sample $\mathbf{x}_0$, written as a modification of Algorithm 1 of Stanczuk et al. [28], with the changes highlighted in teal. The score is read from the *conditional*, *variance-preserving*, ϵ-parameterized network of Appendix C-C rather than from an unconditional variance-exploding score model, so the K perturbations are drawn with the forward kernel $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$ of Ho et al. [11] and the score recovered as $s_\theta = -\hat{\boldsymbol{\epsilon}}_\theta / \sqrt{1-\bar{\alpha}_t}$, evaluated at an intermediate t rather than in the $t \to 0$ limit (the sensitivity to t is studied in Appendix D-B). We also modify the estimator itself, appending a virtual zero singular value $\sigma_{n+1} = 0$ to the descending spectrum and extending the gap search of (6) from $i = 1, \dots, n-1$ to $i = 1, \dots, n$, so that the drop from the smallest singular value σ_n to zero becomes an admissible gap.

a) Sample selection: Stanczuk et al. [28] draw the sample $\mathbf{x}_0 \sim p_0$ from the dataset; we instead draw it from the constraint manifold itself. For each constraint family we reuse the ground-truth targets of Appendix C-A. Every target is a constraint $\mathbf{c} = (\mathbf{p}^\star, \mathbf{R}^\star, \mathbf{m})$ together with the set of analytical IK solutions that satisfy it, with near-duplicates removed, i.e. points lying exactly on $\mathcal{M}_\mathbf{c}$. For each target, we draw $K_s = 5$ samples

TABLE VIII: **Per-family models against the jointly-trained model.** Position error (mm) and orientation error (°) for the largest per-family models (N=40,000) and for the single jointly-trained model of Appendix C-C, which conditions on all families. The ID column is the analytical intrinsic dimension $n-k$ of the joint-space solution manifold; a dash marks a metric the constraint does not fix.

			Per-family (N=40k)		Jointly-trained	
	Constraint	ID	Pos. [mm]	Orn. [°]	Pos. [mm]	Orn. [°]
FRANKA	PLANE	6	9.7	–	4.1	–
	LINE	5	18.3	–	8.2	–
	POSITION	4	23.3	–	12.5	–
	ORIENTATION	4	–	5.7	–	2.3
	PLANE + ORIENTATION	3	9.4	6.1	4.7	2.2
	LINE + ORIENTATION	3	21.2	4.6	10.7	2.0
	SE(3)	1	18.1	3.7	12.0	2.4
UR5	PLANE	5	8.9	–	4.7	–
	LINE	4	14.0	–	7.2	–
	POSITION	3	28.9	–	11.7	–
	ORIENTATION	3	–	4.4	–	2.0
	PLANE + ORIENTATION	2	8.8	3.1	4.9	1.7
	LINE + ORIENTATION	2	14.6	4.3	7.8	3.5
	SE(3)	0	12.4	3.4	9.1	2.3

uniformly without replacement from its solution set, encode each in the sine–cosine representation of Appendix C-C, and standardize it with the model's normalizer; the conditioning vector $\mathbf{c}$ is held fixed at the target value. Drawing $\mathbf{x}_0$ from the analytical solver rather than from the model keeps the estimate on the true manifold and isolates the geometry of the learned score from any sampling bias of the generator. We then run Algorithm 1 at each sample with $K = 50$ perturbations at $t = 15$, and report the median of $\hat{d}$ over the K_s samples and over up to 15 targets per family. Here the score is evaluated in the sine–cosine input space, so the ambient dimension n is twice the joint count, namely 12 for the UR5 and 14 for the Franka; because this encoding is a smooth embedding of joint space, $\hat{d}$ recovers the dimension of $\mathcal{M}_{\mathbf{c}}$ unchanged.

b) Example (the discrete UR5 SE(3) *case):* Under a full SE(3) constraint the UR5 reaches a target with at most eight isolated IK branches, so $\mathcal{M}_{\mathbf{c}}$ is 0-dimensional. Its tangent space is zero-dimensional and every ambient direction is normal. The score matrix therefore has no near-zero tangent block, and all $n = 12$ singular values stay comparably large. A representative sample ($t = 15$, $K = 50$) gives the descending spectrum

$$\begin{aligned} \text{SE(3)}: \quad & (149.7,\ 138.0,\ \mathbf{133.2},\ \mathbf{117.6},\ 105.5,\ 102.2,\\ & 99.3,\ 89.4,\ 85.5,\ 71.8,\ 59.5,\ 48.6),\\ \text{LINE}: \quad & (145.8,\ 137.1,\ 128.8,\ 120.6,\ 101.2,\ 93.5,\\ & 85.5,\ \mathbf{79.8},\ \ \mathbf{10.4},\ \ 5.6,\ \ 4.0,\ \ 3.4), \end{aligned}$$

shown alongside a LINE sample (true ID $= 4$) for contrast. For SE(3) the largest interior drop, $\sigma_3 - \sigma_4 = 15.6$, does not mark a tangent–normal boundary, so the standard rule (6) returns $\hat{k} = 3$ and a meaningless $\hat{d} = 9$. Score-SVD† appends $\sigma_{13} = 0$; the drop $\sigma_{12} - \sigma_{13} = 48.6$ now dominates every interior gap, giving $\hat{k} = 12$ and the correct $\hat{d} = 0$. The LINE sample shows that the modification has no effect whenever the manifold is positive-dimensional. Its four tangent directions collapse to a near-zero tail, the interior gap $\sigma_8 - \sigma_9 = 69.4$ is far larger than the appended $\sigma_{12} - \sigma_{13} = 3.4$, and both rules return $\hat{d} = 4$.

B. Hyperparameter sensitivity

Algorithm 1 has two free hyperparameters, the noise level t at which the score is evaluated and the number of perturbations K. Every estimate in Section III-A uses $t = 15$ and $K = 50$; here we sweep each in turn, holding the other at its canonical value, and show that both lie inside a wide range over which the recovered dimension is exactly the analytical one. Throughout we report the median of $\hat{d}$ over targets together with its inter-quartile band, following the sample-selection procedure of Appendix D-A, and overlay standard Score-SVD with Score-SVD†. The two estimators coincide on every positive-dimensional condition and separate only on the discrete UR5 SE(3) case, where standard Score-SVD has no spectral gap to detect and only Score-SVD† recovers the analytical $\hat{d} = 0$.

a) Noise level: Figure 6 sweeps t with K fixed at 50. For both robots and all seven constraints the estimate stays at the analytical intrinsic dimension across a wide range of low-to-moderate noise, and the canonical $t = 15$ falls inside it on every one of the fourteen conditions. This range is bounded on both sides. As $t \to 0$ the perturbations are too small to spread the samples across the normal bundle, so the score matrix is near-degenerate and the gap rule becomes unstable. At the smallest noise level it overestimates the highest-codimension conditions (both SE(3) conditions, both LINE + ORIENTATION conditions, and additionally Franka PLANE + ORIENTATION and UR5 POSITION), all of which recover their analytical value by $t = 5$. As t grows large the forward kernel of Ho et al. [11] shrinks the clean-data component, and the noised

samples become progressively isotropic. The tangent–normal split then disappears, and the estimate either collapses towards 0, when the surviving spectrum is dominated by a single direction, or saturates near the ambient dimension n, when the noised spectrum is flat and the largest gap migrates to the leading singular value. The estimator therefore deviates from the analytical dimension above a certain noise level, the same behaviour we exploit in Section III-B. It is reliable precisely in the low-noise region in which the intermediate representation has not yet expanded beyond the constraint manifold.

b) Perturbation count: Figure 7 sweeps K with t fixed at 15. The number of perturbations sets the number of columns of the score matrix $S \in \mathbb{R}^{n\times K}$, so for K below the ambient dimension n the matrix has rank at most K and cannot expose the full tangent–normal split; the estimate is rank-limited and overshoots towards n. As K increases past n the estimate drops onto the analytical intrinsic dimension and saturates, becoming flat for all larger K on both robots. The canonical $K = 50$ lies well inside this saturated range for the $n = 12$ (UR5) and $n = 14$ (Franka) score spaces, and is consistent with the $K = 4n$ rule of thumb of Stanczuk et al. [28], whose values (48 and 56) bracket our choice.

C. Baseline methods

We compare Score-SVD against the three classical neighbourhood-based estimators introduced in Section II, which together form the standard reference family for intrinsic-dimension estimation [15, 8, 22]. All three are agnostic to how the data was generated and infer the dimension from the local geometry of a point cloud, so we run them in the output space, on the valid joint configurations the model produces for each condition, rather than on the learned score.

a) Local PCA: The estimator of Fukunaga and Olsen [10] performs principal-component analysis within each sample's neighbourhood and counts how many leading components are needed to capture a fixed fraction of the local variance, taking that count as the dimension. It is controlled by the neighbourhood size, which sets the scale at which the manifold is treated as locally flat, and by a cumulative-variance threshold, which determines how much of the local variance the leading components must capture. The method is simple and directly geometric, but assumes the neighbourhood is well approximated by a linear subspace, so it is most reliable on weakly curved, well-sampled manifolds.

b) Maximum-likelihood estimation: The estimator of Levina and Bickel [15] treats the neighbours of a point as a Poisson process on the manifold and derives a maximum-likelihood estimate of the dimension from the logarithms of the ratios of successive neighbour distances, aggregated across points in the bias-corrected form of Pope et al. [22]. Its one hyperparameter is the neighbourhood size, which also fixes how many distance ratios enter each local estimate. It returns a smooth non-integer estimate and is the standard estimator for the intrinsic dimension of image datasets [22], though it carries a downward bias that grows with the true dimension and cannot represent the discrete $\hat{d} = 0$ case.

c) TwoNN: The estimator of Facco et al. [8] uses only the two nearest neighbours of each point, exploiting that under a locally uniform density, the ratio of the second to the first neighbour distance follows a distribution whose shape is determined solely by the dimension, which yields a closed-form per-point estimate. It is effectively parameter-free and the most local of the three, which makes it robust to curvature but reliant on the local-uniformity assumption and sensitive to the limited information in two distances.

The baselines measure the spread of the generated configurations in the output space, so they cannot separate the continuous solution manifold from the discrete inverse-kinematics branches that contribute to the same local spread, and none is defined for the 0-dimensional UR5 $\mathrm{SE}(3)$ case. The score-based estimator instead reads the geometry of the learned score on the manifold itself rather than the density of samples around it. Section D-D reports the hyperparameter settings and the full numerical comparison.

D. Full numerical results and score spectra

Table IX reports the recovered intrinsic dimension for every constraint–manipulator condition and every estimator we consider, extending the two-baseline summary of Figure 2. Alongside Score-SVD and Score-SVD†, we evaluate the three neighbourhood-based baselines of Section II, namely the maximum-likelihood estimator of Levina and Bickel [15] at two neighbourhood sizes, local PCA [10] at the 90% and 99% cumulative-variance thresholds, and the parameter-free TwoNN ratio of Facco et al. [8]. The neighbourhood baselines act in the output space, so we run them on the valid joint configurations drawn from the model for each condition, whereas the score-based estimators read the geometry of the learned score on the manifold itself, following the sample-selection procedure of Appendix D-A. Every entry is the median over targets.

The two estimator families separate clearly. The neighbourhood baselines recover the analytical dimension only on the lower-dimensional conditions and drift beyond rounding error as the intrinsic dimension grows. Local MLE and local PCA at the 90% threshold are biased low and match only once the dimension is small enough, while TwoNN is noisier and generally biased high, falling within rounding error on a single condition per robot. Raising the PCA threshold to 99% trades one failure mode for another, recovering many more conditions, including several high-dimensional ones, but overshooting on the lowest-dimensional, highest-codimension conditions, most severely on the two $\mathrm{SE}(3)$ conditions. Score-SVD, in contrast, recovers the analytical dimension exactly on every one of the thirteen positive-dimensional conditions, and Score-SVD† additionally returns the correct $\hat{d} = 0$ on the discrete UR5 $\mathrm{SE}(3)$ case, where standard Score-SVD reports an incorrect $\hat{d} = 3$.

The behaviour of these estimators is visible in the score-matrix spectrum itself. Figure 8 shows the mean descending singular spectrum of S for every condition, pooled over the samples and targets of the procedure of Appendix D-A. On every positive-dimensional condition, the spectrum splits into two parts. A block of large singular values spans the normal

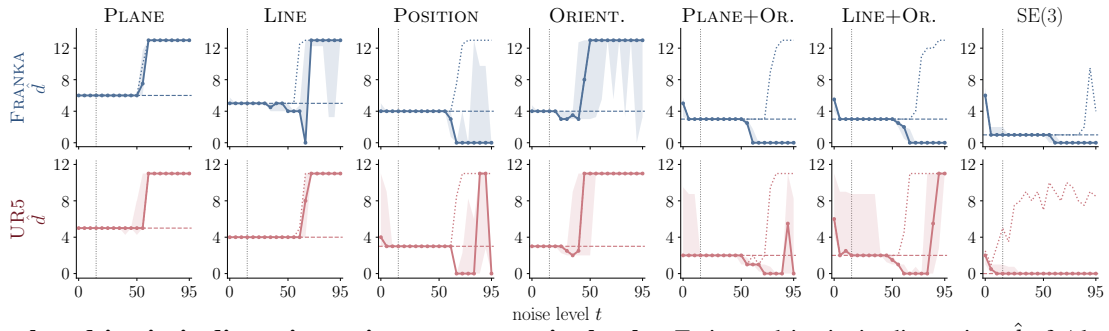

Fig. 6: **Score-based intrinsic-dimension estimate versus noise level t.** Estimated intrinsic dimension $\hat{d}$ of Algorithm 1 as a function of the noise level t, with the perturbation count fixed at $K = 50$; Franka (top) and UR5 (bottom), one constraint per column. ···· / ···· Score-SVD; —— / —— Score-SVD† (median over targets, shaded inter-quartile band); - - - / - - - analytical intrinsic dimension of Table I. The dotted vertical line marks the canonical $t = 15$. The estimate stays at the analytical value across a wide low-to-moderate range that contains $t = 15$ on all fourteen conditions, and degrades only once the noised samples become nearly isotropic.

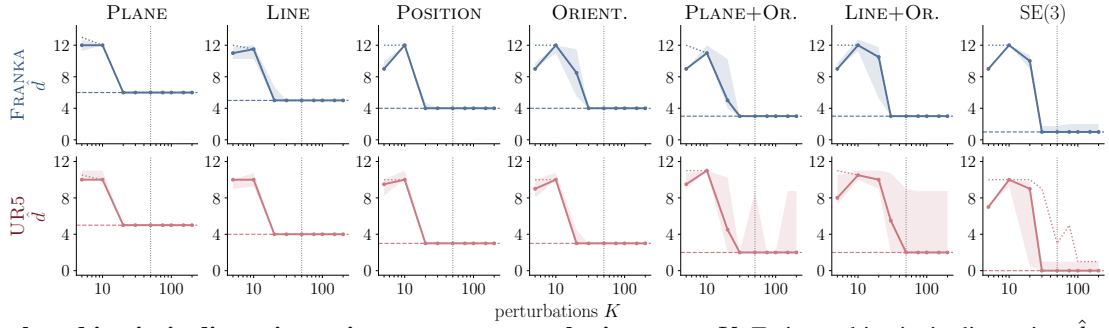

Fig. 7: **Score-based intrinsic-dimension estimate versus perturbation count K.** Estimated intrinsic dimension $\hat{d}$ of Algorithm 1 as a function of the perturbation count K (log axis), with the noise level fixed at $t = 15$; Franka (top) and UR5 (bottom), one constraint per column. Score-SVD (dotted), Score-SVD† (solid, shaded inter-quartile band), the dashed analytical intrinsic dimension, and the canonical $K = 50$ (dotted vertical line) follow Figure 6. Below the ambient dimension n the estimate is rank-limited and overshoots; it saturates at the analytical value around $K \geq 2n$, comfortably before $K = 50$.

directions of the manifold, while a near-zero tail, whose width equals the intrinsic dimension, spans the tangent directions that the score cannot resolve. The gap between these two blocks is precisely what Score-SVD identifies. The markers in the figure are read from this averaged spectrum at index $n - \hat{d}$, and they coincide with the analytical value on all thirteen positive-dimensional conditions. The discrete UR5 SE(3) case is the only exception, and the figure makes its difficulty concrete. Because every ambient direction is normal, no near-zero tail exists and the spectrum decays smoothly without a clear gap, so the standard rule is forced to select whichever interior drop happens to be largest. On this averaged spectrum, the largest drop is the final one, $\sigma_{n-1} \to \sigma_n$, so the standard Score-SVD marker lands at $\hat{d} = 1$, whereas the per-target median reported in Table IX is $\hat{d} = 3$. This two disagree precisely because no spectral gap exists and the largest per-sample drop scatters across the interior, making the estimate sensitive to how it is aggregated. Only Score-SVD†, which places the gap after the final singular value σ_n, recovers the correct $\hat{d} = 0$.

APPENDIX E
LATENT INTERPOLATION

This section expands the latent-interpolation study of Section III-B. We first report the full quantitative results across all fourteen constraint–manipulator conditions and compare straight-line against spherical interpolation in the latent space (Appendix E-A), then show task-space illustrations of the decoded paths for every condition (Appendix E-B), and discuss the conditions that push our method to its limits, including the disconnected SE(3) solution set of the UR5 (Appendix E-C). We then give the formal definitions and derivations behind the local-flattening diagnostics of Section III-B (Appendix E-D) and report additional analyses: the dependence on the intermediate noise level and the growth of error with endpoint separation (Appendix E-E).

Unless stated otherwise, all results use the same setup as Section III-B: for each condition we draw 30 task-space targets, and at each target select five endpoint pairs of IK solutions whose joint-space separation spans the empirical distribution, from the tenth percentile to the maximum observed separation. Both endpoints are inverted to an intermediate noise level $\bar{t}$ of the partial DDIM process (5), their latent codes are interpolated,

TABLE IX: **Intrinsic-dimension estimates across all fourteen constraint–manipulator conditions.** Recovered dimension $\hat{d}$ for every estimator, with the analytical intrinsic dimension of the joint-space solution manifold in the ID column. Neighbourhood baselines (local MLE [15] at neighbourhood size NN, local PCA [10] at a cumulative-variance threshold, and TwoNN [8]) run on valid model configurations in joint space; the score-based estimators run on the constraint manifold. Entries are medians over targets; those within rounding error of the analytical dimension are shown in **bold**.

			Local MLE		Local PCA			Score-based	
	Constraint	ID	NN=2	NN=10	90%	99%	TwoNN	SVD	SVD†
Franka	Plane	6	4.2	4.2	4	**6**	3.4	**6**	**6**
	Line	5	4.2	3.7	4	**5**	**5.3**	**5**	**5**
	Position	4	**3.6**	3.2	3	**4**	5.0	**4**	**4**
	Orientation	4	**3.6**	3.2	3	**4**	5.0	**4**	**4**
	Plane + Orientation	3	**2.9**	**2.5**	**3**	4	4.0	**3**	**3**
	Line + Orientation	3	**2.9**	**2.6**	**3**	4	4.0	**3**	**3**
	SE(3)	1	3.7	2.2	2	5	5.1	**1**	**1**
UR5	Plane	5	3.3	3.4	3	**5**	2.4	**5**	**5**
	Line	4	3.2	3.2	3	5	**3.5**	**4**	**4**
	Position	3	**2.9**	**2.6**	**3**	**3**	4.0	**3**	**3**
	Orientation	3	**2.8**	2.3	**3**	**3**	3.9	**3**	**3**
	Plane + Orientation	2	**2.0**	**1.7**	**2**	**2**	2.8	**2**	**2**
	Line + Orientation	2	**2.1**	**1.7**	**2**	3	2.9	**2**	**2**
	SE(3)	0	4.8	4.0	4	6	7.6	3	**0**

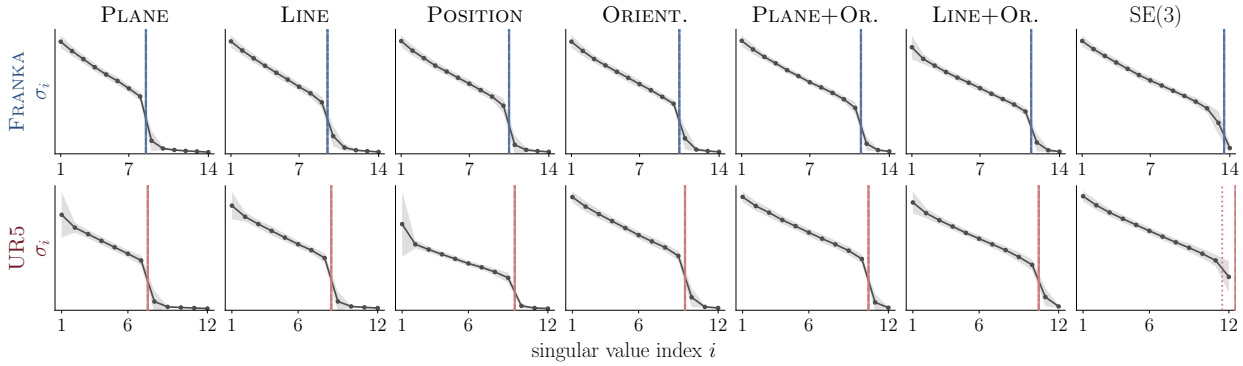

Fig. 8: **Score-matrix singular spectra.** Mean descending singular spectrum of the score matrix S at $t = 15$ and $K = 50$ (grey line, shaded $\pm$ one standard deviation over the pooled samples), for Franka (top) and UR5 (bottom), one constraint per column. Vertical markers give the gap position $n - \hat{d}$ implied by the - - - / - - - analytical intrinsic dimension, · · · · / · · · · Score-SVD, and —— / —— Score-SVD†. The three markers fall at the spectral gap and coincide on every positive-dimensional condition. On the discrete UR5 SE(3) case the spectrum has no clear separation, Score-SVD selects an incorrect interior gap, while Score-SVD† places the gap after σ_n to recover $\hat{d} = 0$.

and the interpolants are decoded back to joint configurations, whose forward-kinematics error to the analytical constraint set $\mathcal{S}_\mathbf{c}$ is then measured.

A. *Quantitative results*

Table X reports, for every condition, the median forward-kinematics error of the decoded paths with its 95% bootstrap confidence interval, separately for position- and orientation-constrained coordinates. We compare both straight-line interpolation as described in Section III-B and spherical linear interpolation (slerp), which follows the great-circle arc between the two latent codes on a sphere of fixed radius; each is evaluated at the per-condition noise level that best preserves the endpoints.

Across all conditions the decoded paths stay close to the constraint manifold: the median position error of linear interpolation ranges from 0.40 to 2.41 mm and the median orientation error from 0.23 to 1.18°. Most confidence intervals are narrow relative to their medians, although the hardest position-constrained cases—notably the UR5 under Line + Orientation—have appreciably wider intervals. Both interpolation methods are statistically comparable on most conditions, with overlapping confidence intervals; straight-line interpolation has lower median values on the hardest conditions, especially on the UR5 under SE(3).

TABLE X: **Latent interpolation accuracy across all fourteen constraint–manipulator conditions.** Median forward-kinematics error of the decoded paths for straight-line (lin) and spherical (sph) interpolation, with the 95% bootstrap CI of the median given as the bracketed subscript [low, high]; for each metric the lower of the two medians is set in **bold**. Position error is the Euclidean distance over the constrained axes; orientation error is the full SO(3) geodesic angle (13) for fully orientation-constrained tasks, and the angle between the constrained body-axis directions for LINE + ORIENTATION (Appendix B-A). Dashes mark coordinates left unconstrained.

		Pos. err. [mm]		Orn. err. [°]	
	Constraint	lin	sph	lin	sph
FRANKA	PLANE	$\mathbf{0.40}_{[0.35,0.47]}$	$0.46_{[0.36,0.54]}$	–	–
	LINE	$\mathbf{1.10}_{[0.87,1.42]}$	$1.21_{[0.91,1.47]}$	–	–
	POSITION	$1.61_{[1.33,1.77]}$	$\mathbf{1.57}_{[1.24,1.81]}$	–	–
	ORIENTATION	–	–	$\mathbf{1.18}_{[0.98,1.36]}$	$1.22_{[1.00,1.41]}$
	PLANE + ORIENTATION	$\mathbf{0.43}_{[0.34,0.51]}$	$0.47_{[0.38,0.57]}$	$\mathbf{0.25}_{[0.20,0.30]}$	$0.28_{[0.23,0.34]}$
	LINE + ORIENTATION	$\mathbf{1.29}_{[0.94,1.89]}$	$1.32_{[1.07,1.87]}$	$\mathbf{0.23}_{[0.18,0.29]}$	$0.24_{[0.20,0.39]}$
	SE(3)	$1.66_{[1.44,1.96]}$	$\mathbf{1.64}_{[1.48,2.06]}$	$0.56_{[0.46,0.62]}$	$\mathbf{0.54}_{[0.45,0.61]}$
UR5	PLANE	$0.89_{[0.65,1.23]}$	$\mathbf{0.75}_{[0.62,1.08]}$	–	–
	LINE	$\mathbf{1.60}_{[1.27,2.02]}$	$1.92_{[1.53,2.44]}$	–	–
	POSITION	$\mathbf{1.08}_{[0.84,1.28]}$	$1.20_{[0.88,1.51]}$	–	–
	ORIENTATION	–	–	$\mathbf{0.34}_{[0.28,0.44]}$	$0.35_{[0.29,0.50]}$
	PLANE + ORIENTATION	$\mathbf{0.77}_{[0.59,1.12]}$	$1.08_{[0.79,1.27]}$	$\mathbf{0.45}_{[0.35,0.52]}$	$0.52_{[0.37,0.61]}$
	LINE + ORIENTATION	$\mathbf{2.41}_{[1.98,3.14]}$	$3.24_{[2.41,4.22]}$	$\mathbf{1.00}_{[0.79,1.26]}$	$1.22_{[0.95,1.61]}$
	SE(3)	$\mathbf{1.61}_{[1.42,1.70]}$	$2.30_{[2.08,2.44]}$	$\mathbf{0.35}_{[0.31,0.38]}$	$0.47_{[0.44,0.52]}$

B. Task-space trajectories across all constraints and robots

Figure 9 displays the decoded straight-line interpolation paths in task space for all fourteen conditions, extending the results of Figure 4.

C. Disconnected solution sets and branch switching

a) Disconnected solution sets (UR5 under SE(3)*):* For the non-redundant UR5 under a full SE(3) pose constraint, the feasible configurations form a finite set of isolated IK branches, and no continuous pose-preserving path connects two branches in joint space. This is the clearest discrete case in our study: the solution set is zero-dimensional, a finite collection of isolated IK branches. It therefore reveals what a straight latent path decodes to when no continuous on-constraint curve exists to follow. We find that the decoded path does not form a continuous feasible curve between branches (which a finite solution set could not support), but instead stays close to the SE(3) constraint while concentrating the joint-space motion into one or more narrow intervals of α. Its constraint error stays low (Table X, 1.61 mm / 0.35°) and, unlike the continuous conditions, does not grow with endpoint separation but stays flat at the widest pairs (Figure 12), consistent with rapid switching between branches rather than a long infeasible bridge through joint space. Figure 10 makes the switching explicit through the directional sensitivity ρ (Appendix E-D): its median is small and flat along the whole path, while its upper percentile spikes by two to three orders of magnitude near the midpoint, exactly where the decoded configuration crosses from one branch to the next.

b) Hardest conditions: Our approach is weakest on the UR5 LINE + ORIENTATION condition, the tightest continuous constraint we study on the non-redundant arm: it has the largest median errors (2.41 mm / 1.00°) in Table X, and is the one condition whose lower-tail tangent overlap $\tau^{\mathbf{q}}_{\mathrm{p5}}$ falls below the random-direction baseline $\tau^{\mathbf{q}}_{\mathrm{rand}}$ (Table II), indicating a minority of interpolation directions that are no better than random at staying tangent to the constraint manifold. Error also concentrates on the small number of targets whose solution set is not fully covered by the training distribution: the widest endpoint pairs at such targets are where the decoded path drifts farthest from the constraint set, which is consistent with the additional capacity required at high noise levels being unavailable when the target lies at the edge of the trained region. Even there the drift stays within a few centimetres and the decoded motion remains smooth.

D. Definitions of ρ, ν, $\tau^{\mathbf{q}}$, and $\tau^{\mathbf{q}}_{\mathrm{rand}}$

This section gives the full definitions and derivations of the local-flattening diagnostics of Section III-B (Table II). For a fixed constraint $\mathbf{c}$, we invert an endpoint pair to the intermediate noise level $\bar{t}$, giving latent codes $\mathbf{x}_a, \mathbf{x}_b$. We interpolate these along the straight path $\mathbf{x}(\alpha) = (1-\alpha)\,\mathbf{x}_a + \alpha\,\mathbf{x}_b$, with unit direction $\widehat{\mathbf{v}} = (\mathbf{x}_b - \mathbf{x}_a)/\|\mathbf{x}_b - \mathbf{x}_a\|$. The DDIM map $g_{\mathbf{c}}$ (4) from $\bar{t}$ to 0, maps each interpolant to a joint configuration $\mathbf{q}(\alpha) = g_{\mathbf{c}}(\mathbf{x}(\alpha))$. Its Jacobian $Dg_{\mathbf{c}}(\mathbf{x}) \in \mathbb{R}^{n\times 2n}$, where the latent dimension $2n$ (12 for the UR5, 14 for the Franka) is the joint count doubled by the sine–cosine encoding, is taken

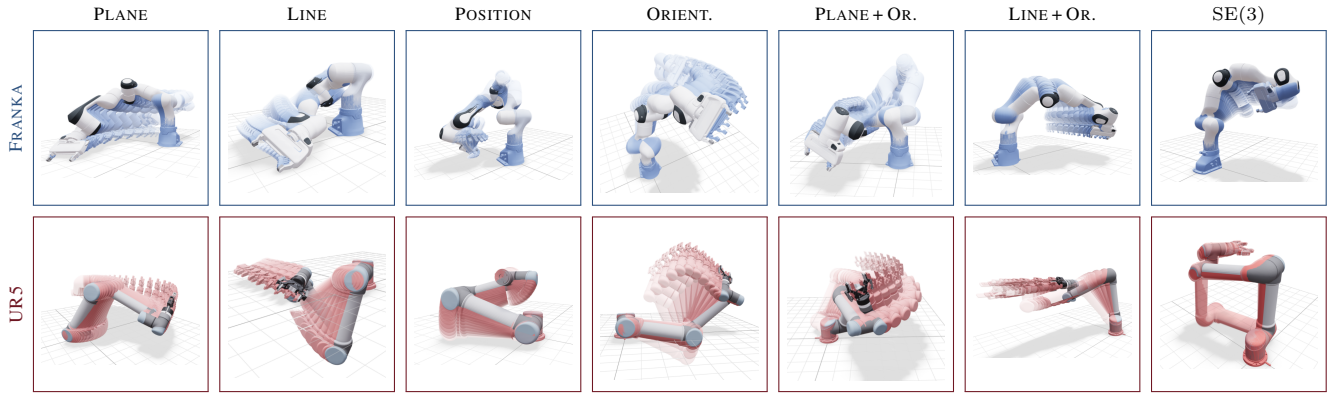

Fig. 9: **Visualization of straight-line latent interpolation for all fourteen conditions.** Each panel plots the end-effector poses of the decoded interpolants in task space; opacity encodes the interpolation parameter, fading from one endpoint (opaque) to the other (transparent). Rows are the Franka and UR5; columns are the seven constraints of Table I.

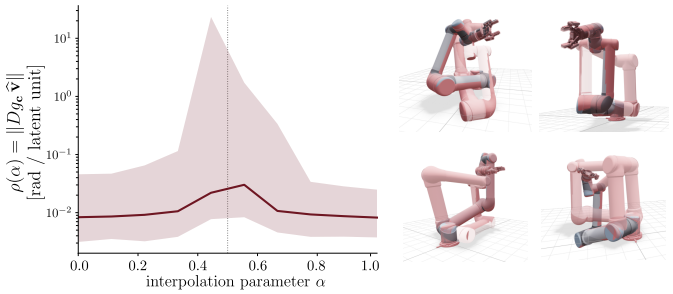

Fig. 10: **Disconnected-branch interpolation on the UR5 under** SE(3)**.** *Left:* directional sensitivity $\rho(\alpha)$ along the interpolation path; the solid line is the median over endpoint pairs, the shaded region the 2.5–97.5 percentile band, and the dotted vertical line marks the path midpoint. The median stays small and flat across the whole path, while the upper percentile spikes by two to three orders of magnitude near the midpoint: each pair switches branches over a narrow interval of α, so at any fixed α only the pairs mid-switch exhibit the large joint-space velocity, yet the decoded poses stay close to the constraint set throughout. *Right:* four interpolations for the widest (maximum-separation, 100th-percentile) endpoint pairs. Each panel shows the generated configurations along the straight-line interpolation; the end-effector approximately holds the fixed SE(3) target while the arm switches between distinct IK branches, crossing from two up to four branches in a single decoded path.

by automatic differentiation at fixed $\mathbf{c}$ and gives the decoded joint-space direction $\mathbf{u}(\alpha) = Dg_{\mathbf{c}}(\mathbf{x}(\alpha))\,\widehat{\mathbf{v}}$.

a) Directional sensitivity and joint-space speed: The directional sensitivity is the norm of the decoded direction, and the joint-space speed rescales it by the endpoint separation,

$$\rho(\alpha) = \|\mathbf{u}(\alpha)\|_2, \qquad \nu(\alpha) = \|\mathbf{x}_b - \mathbf{x}_a\|\,\rho(\alpha), \tag{14}$$

where ρ has units of radians per latent unit and ν units of radians per unit interpolation parameter; a small ρ means a unit latent step produces only a small, smoothly varying change in joint space.

b) Tangent overlap and the random-direction baseline: Let $h_{\mathbf{c}}$ be the constraint-error map, with $\mathcal{M}_{\mathbf{c}} = \{\mathbf{q} : h_{\mathbf{c}}(\mathbf{q}) = \mathbf{0}\}$ and Jacobian $Dh_{\mathbf{c}}(\mathbf{q}) \in \mathbb{R}^{k\times n}$ (Appendix B-A). The tangent overlap is the fraction of the decoded direction lying in the null space of the constraint Jacobian,

$$\tau^{\mathbf{q}}(\alpha) = \frac{\left\|\Pi_{\ker Dh_{\mathbf{c}}(\mathbf{q}(\alpha))}\,\mathbf{u}(\alpha)\right\|_2}{\|\mathbf{u}(\alpha)\|_2}, \tag{15}$$

where $\Pi_{\ker Dh_{\mathbf{c}}(\mathbf{q})}$ is the orthogonal projector (12) onto $\ker Dh_{\mathbf{c}}(\mathbf{q})$, which is the tangent space $\mathcal{T}_{\mathbf{q}(\alpha)}\mathcal{M}_{\mathbf{c}}$ (11) when $\mathbf{q}(\alpha)$ lies on $\mathcal{M}_{\mathbf{c}}$; a value near one then means the motion follows along the manifold and leaves the constraint error unchanged to first order. For comparison, let $\mathbf{r}$ be a random unit vector distributed uniformly on the unit sphere S^{n-1} of joint space; for any fixed rank-$(n-k)$ orthogonal projector Π,

$$\mathbb{E}\left[\|\Pi\,\mathbf{r}\|_2^2\right] = \mathrm{tr}(\Pi)/n = (n-k)/n, \tag{16}$$

which gives the root-mean-square baseline

$$\tau^{\mathbf{q}}_{\mathrm{rand}} = \sqrt{(n-k)/n}. \tag{17}$$

For each condition we report the median over all (pair, α) samples, $\tau^{\mathbf{q}}_{\mathrm{med}}$, the fifth percentile $\tau^{\mathbf{q}}_{\mathrm{p5}}$, and the relative improvement of the median over the baseline,

$$\Delta^{\mathbf{q}}_{\mathrm{rel}} = (\tau^{\mathbf{q}}_{\mathrm{med}} - \tau^{\mathbf{q}}_{\mathrm{rand}})\,/\,\tau^{\mathbf{q}}_{\mathrm{rand}}; \tag{18}$$

the same medians and percentiles define ρ_{med} and ν_{med} in Table II. For the UR5 under SE(3) the constraint is full ($k = n$), so $\ker Dh_{\mathbf{c}}$ is zero-dimensional and both $\tau^{\mathbf{q}}$ and $\tau^{\mathbf{q}}_{\mathrm{rand}}$ are zero by construction; this condition is therefore excluded from the tangent-overlap comparison and treated through the disconnected-topology analysis above (Figure 10).

E. Extended analysis

a) Dependence on the intermediate noise level: Figure 11 extends the error-versus-noise panels of Figure 3 to all seven constraints. The error is large at low noise and largest for the widest pairs, but every separation falls onto a common low floor as the noise level increases. This is consistent with our observation in Section III-B, Figure 3 that the interpolation

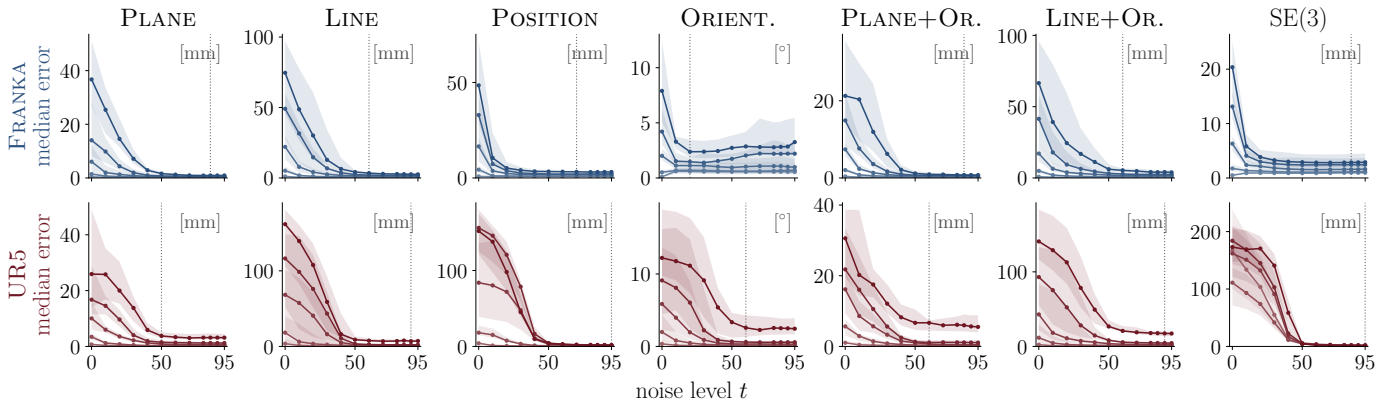

Fig. 11: **Constraint error versus intermediate noise level.** Constraint error of the decoded straight-line paths as a function of the noise level $\bar{t}$ the endpoints are inverted to, extending the bottom rows of Figure 3 to all seven constraints; Franka (top) and UR5 (bottom). Within each panel the five curves are endpoint pairs at increasing joint-space separation (lighter to darker: 10th to 100th separation percentile), each the median over targets with a shaded 95% bootstrap CI. Panels report position error in millimetres, except the orientation-only ORIENTATION column, which reports the geodesic angle in degrees (°). The dotted vertical line marks the $\bar{t}$ used for Table X. The error is largest for the widest pairs and at low noise, and all separations converge to a low floor as the noise level increases. For the combined position + orientation constraints these panels report position error only; the corresponding orientation residuals are summarized in Table X rather than plotted.

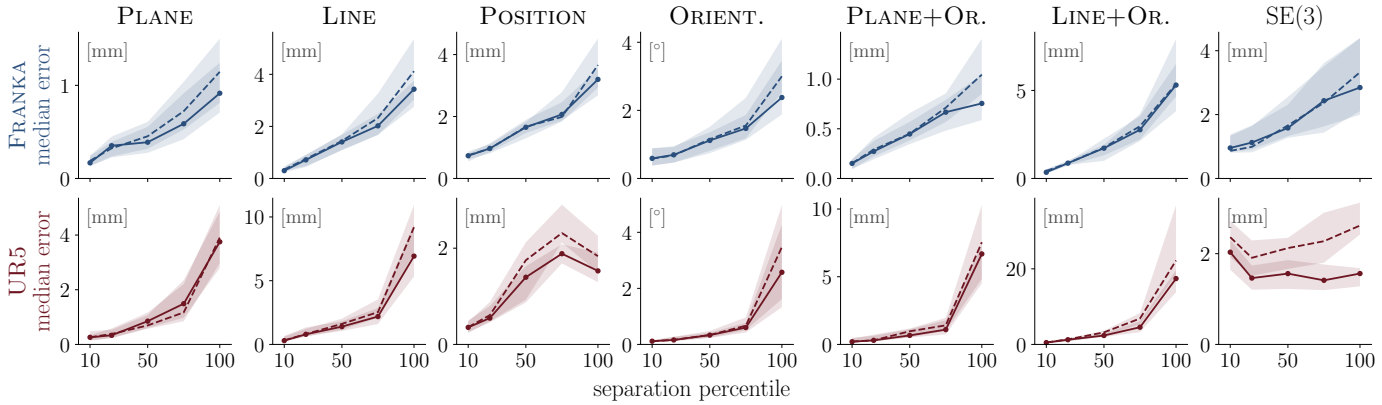

Fig. 12: **Median constraint error versus endpoint separation.** Median error of the decoded path, with a shaded 95% bootstrap CI of the median, as a function of the joint-space separation percentile between the endpoints, for straight-line (solid) and spherical (dashed) interpolation, each at its per-condition noise level $\bar{t}$; Franka (top) and UR5 (bottom). Panels report position error in millimetres, except the orientation-only ORIENTATION column, which reports the geodesic angle in degrees (°); for the combined position + orientation constraints the orientation residuals are summarized in Table X rather than plotted. For the continuous conditions the error generally increases with endpoint separation; the UR5 under SE(3) is the exception, staying nearly flat because the decoded path switches branches while holding close to the fixed pose.

error collapses only once the intermediate representation has expanded beyond the analytical intrinsic dimension. The one clear exception is the orientation-only ORIENTATION condition on the Franka, whose error is already small at low noise and rises slightly as more noise is injected into an already-near-feasible range; this is why we use a comparatively low noise level for this specific constraint.

b) Growth of error with endpoint separation: Figure 12 reports the median constraint error as a function of the joint-space separation between the two endpoints, expressed as a percentile of the per-target solution-distance distribution.