Trivialized Generative Models on Lie Groups

Neil He[1*] **Meenal Jhajharia**[1*] **Qianxi Wu**[2] **Chaoran Cheng**[1] **Arindam Banerjee**[1]

Ge Liu[1]
[1]University of Illinois Urbana-Champaign, [2]Peking University

Abstract

Many problems in diverse fields involve data that naturally live on Lie groups. However, existing Riemannian and Lie group generative models still face several limitations. Riemannian flow and consistency models often require position-dependent vector fields, and expensive geometric terms such as covariant derivatives or Euler-Arnold simulation on Lie groups without a bi-invariant metric. Existing Lie group flow models are usually tied to collinear exponential paths, while momentum-based Lie group diffusion relies on compactness assumptions and does not naturally extend to non-compact groups. To address these limitations, we propose Trivialized Generative Models (TGM), a family of generative models that learns endpoint-constrained paths in the fixed Lie algebra and lifts them to the Lie group. In particular, we propose Trivialized Flow Matching (TFM), Trivialized Consistency Models (TCM), and momentum-based extensions with endpoint correction, enabling flexible path design, simpler few-step objectives, and generation on non-compact Lie groups. We evaluate TGM on compact SO(3) benchmarks, non-compact $\mathrm{Sp}(4,\mathbb{R})$ datasets motivated by continuous quantum systems and paraxial optics, and real-world protein backbone and robot grasp generation. Across these settings, TGM improves sample quality over existing Riemannian and Lie group baselines, and is more efficient on Lie groups without a bi-invariant metric.[1]

1 Introduction

Generative models such as diffusion models [14, 33] and flow matching [18] have achieved incredible success in many fields, such as visual domains [7], texts [26], and the natural sciences [39, 40, 37]. Few-step generative models such as consistency models [32, 21] further enable efficient sampling by removing the need for hundreds or thousands of generation steps. However, in many fields the data lie on a manifold rather than in Euclidean space. A particular set of manifolds of interest are Lie groups, which appear in robotics [12, 36, 30, 28], protein design [40, 39, 37], quantum physics [6], and optics [2]. Applying Euclidean generative models in these fields violates the geometric constraints and leads to suboptimal performance [8]. To address this, prior works have proposed generative models on Riemannian manifolds and Lie groups. RSGM [8] and RDM [15] are early efforts that adapted diffusion models to Riemannian manifolds, but suffered from issues such as needing simulation and exponential time complexity [23]. Later works improved efficiency [20, 23], but are limited to symmetric spaces and require different considerations for different manifolds.

Recent works address some of these issues, but each has its own limitations. RFM [4] is the first to adapt flow matching to Riemannian manifolds as a scalable simulation free method on simple geometry, but it still requires simulating Euler-Arnold dynamics on Lie groups without bi-invariant metrics. RCM [5] extends consistency models to Riemannian manifolds, but learns on vector spaces

[*]Indicates equal contribution.
[1]GitHub link here.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

that change with position and thus requires closed-form solutions for differentiating through the covariant derivative, which adds computational complexity and limits its applicability to general manifolds. On Lie groups, TDM [42] proposes diffusion by evolving second order dynamics on the Lie algebra, but its reference distribution relies on a normalizable Haar measure, which restricts it to compact Lie groups. LFM [29] defines flow matching by the exponential curve of Lie groups, but has limited flexibility in path design. [3] proposes diffusion in the Lie group representation space, but is limited to a chosen Lie group action in Euclidean ambient coordinates.

In light of the above limitations of prior methods, we propose a family of generative models on Lie groups by learning a left-trivialized generative path on the Lie algebra and lifting that path to the Lie group. Trivialization expresses velocities at different group elements in one fixed coordinate system, rather than in different tangent spaces. Using our general framework, we propose trivialized flow matching and consistency models on Lie groups, and extend them to incorporate momentum through second order dynamics. Our contributions are as follows:

(i) We propose a family of trivialized generative models (TGM) on Lie groups, encompassing trivialized flow matching (TFM) and consistency models (TCM). TFM achieves simulation free training on more general Lie groups, without global assumptions such as compactness or a bi-invariant metric, and enables flexible path design. In particular, we propose a preconditioned path (P-TFM) that leads to consistent improvements. TCM fixes a vector space and thereby derives a simpler loss than RCM that avoids geometric terms such as the covariant derivative, which we show leads to better performance and faster convergence.

(ii) We extend TGM to models with second order dynamics and propose TFM with momentum (MoTFM) and TCM with momentum (MoTCM). We propose an endpoint correction mechanism that enables MoTFM to provably connect two distributions when the endpoint correction lies in a logarithm chart. This holds on more general Lie groups, without global assumptions such as compactness or a normalizable Haar measure, addressing prior work's failure on non-compact Lie groups. We further prove stability bounds for MoTFM and show that it achieves better performance than prior momentum-based methods on both compact and non-compact Lie groups.

(iii) We conduct extensive experiments to show that TGM achieves superior performance in four settings: (1) on standard benchmarks for compact Lie groups; (2) on non-compact Lie groups with distributions derived from continuous quantum systems and optics; (3) on protein backbone generation; and (4) on robot grasp generation, where the latter two are real-world tasks.

2 Related Works

Generative Modeling. Diffusion models [7, 33, 14, 1] and flow matching [18, 35] are many-step generative models that connect a source distribution to a target distribution. To reduce the number of generation steps, recent works have proposed few-step generative models, including consistency models [21, 32] that are consistent across time and other few-step or one-step models [38, 9, 41, 22] that often distill a student model from a pre-trained teacher model.

Riemannian and Lie Group Generative Models. Early methods studied normalizing flows on manifolds [13, 27, 24]. Riemannian score-based generative models [8] and Riemannian diffusion models [15] adapt diffusion processes to curved spaces using manifold Brownian motion and Riemannian score estimation, and later works improve their scalability on structured spaces [20, 23]. Riemannian Flow Matching [4] defines probability paths through Riemannian geometry and has been used in downstream geometric generation tasks [39, 16]. For few-step generation, Riemannian Consistency Models [5] learn endpoint maps along manifold probability-flow dynamics, which requires tracking how tangent vector fields change along curved paths through quantities such as covariant derivatives. Free-form flow models [34] directly learn manifold-valued maps with projection-based parameterizations.

Lie groups provide additional algebraic structure and arise in rotations, robotics, protein structures, quantum systems, and optics. Lie group flow matching [29] defines flow matching through exponential curves. TDM [42] introduces momentum-based diffusion by evolving group elements together with left-trivialized momenta in the Lie algebra. Other works model distributions through Lie group representation spaces [3]. Our work also uses the Lie algebra as the space where dynamics are

Table 1: Comparison of TGM with prior works.

Method type	Baseline	Advantage of TGM
Flow matching	RFM [4]	RFM requires Euler-Arnold simulation on Lie groups w/o a bi-invariant metric. TFM (ours) uses endpoint-constrained algebra bridges, making training simulation free and efficient (over 15X speedup on $Sp(4,\mathbb{R})$).
Flow matching	LFM [29]	LFM is restricted to collinear exponential paths. TFM (ours) also supports non-collinear paths, and P-TFM (ours) improves performance by improving the conditioning of endpoint prediction.
Consistency model	RCM [5]	RCM differentiates through position-dependent tangent fields and has covariant derivative terms. TCM (ours) works in the fixed Lie algebra and avoids these terms. RCM also requires Euler-Arnold simulation on Lie groups without a bi-invariant metric, which TCM (ours) avoids.
Momentum model	TDM [42]	TDM's invariant prior distribution relies on a normalizable Haar-based measure, which is unavailable on non-compact Lie groups. MoTFM (ours) and MoTCM (ours) enable momentum generative models on non-compact Lie groups.

learned, but constructs endpoint-constrained algebraic bridges that lift to valid group paths and naturally support flow matching, consistency models, and momentum-based extensions. Table 1 summarizes the differences.

3 Preliminaries

Lie Groups and Lie Algebra. Let G denote a finite-dimensional Lie group, namely a group equipped with a smooth manifold structure for which multiplication and inversion are smooth maps. Its Lie algebra is $\mathfrak{g} = T_eG$, the tangent space at the identity, equipped with the bracket $[\cdot,\cdot]$. The exponential map $\exp : \mathfrak{g} \to G$ is the Lie group exponential. On a logarithm chart where $\exp$ is a diffeomorphism onto its image, we write $\log$ for the inverse map. The purpose of trivialization is to write velocities of curves on G in the fixed vector space $\mathfrak{g}$. For $x \in G$, let $L_x : G \to G$ be left translation, $L_x(h) = xh$. If x_t is a C^1 curve in G, then $\dot{x}_t \in T_{x_t}G$. Pulling this velocity back to the identity by the differential of left translation gives the left-trivialized velocity $\xi_t = (L_{x_t^{-1}})_*\dot{x}_t \in \mathfrak{g}$, equivalently $\dot{x}_t = (L_{x_t})_*\xi_t$. In this form, states remain on the group, while velocities and learned vector fields are represented in $\mathfrak{g}$.

Adjoint Actions. We use the adjoint action to compare algebra elements written in different group frames. For $x \in G$, let $C_x(h) = xhx^{-1}$ be conjugation and define $\mathrm{Ad}_x = (dC_x)_e : \mathfrak{g} \to \mathfrak{g}$. Its infinitesimal form is $\mathrm{ad}_\xi\,\eta = [\xi,\eta]$. The adjoint action appears when a velocity or endpoint correction written in one left-trivialized frame is transported to another. An inner product $\langle\cdot,\cdot\rangle_m$ on $\mathfrak{g}$ induces the norm $\|\cdot\|_m$ and, by left translation, the corresponding left-invariant distance d_m on G.

4 Method

In this section we introduce Trivialized Generative Models (TGM), a framework for generative modeling on Lie groups through their Lie algebras. Let ρ_0, ρ_1 be source and target distributions on a Lie group G with Lie algebra $\mathfrak{g}$. TGM transports samples from ρ_0 to ρ_1 by learning a generative path on $\mathfrak{g}$ and lifting that path to G. We first explain why directly applying Euclidean generative models in a fixed log chart is insufficient. We then introduce endpoint-constrained trivialized bridges and use them to define Trivialized Flow Matching (TFM) for many-step generation and Trivialized Consistency Models (TCM) for few-step generation. Finally, we extend the framework to momentum-based models on $G \times \mathfrak{g}$. See Appendix B for full theoretical results and proofs, as well as the exact training and sampling algorithms for each model.

Naive Model and its Failure. A naive approach would train a Euclidean model between $\log(\rho_0)$ and $\log(\rho_1)$, but these are absolute coordinates that depend on a choice of branch and need not align with the intrinsic displacement between endpoints. TGM avoids this issue by using relative endpoint bridges (see Appendix C).

Algorithm 1 Training Algorithms for TGM Variants

Require: Source/target distributions ρ_0, ρ_1 or μ_0, μ_1; bridge A_t; networks $v_\theta, h_\theta, a_\theta$.

TFM Training

$b_t(y) \leftarrow J_\ell(A_t(y))[\partial_t A_t(y)]$.
while not converged **do**
  Sample $x_0 \sim \rho_0, x_1 \sim \rho_1, t \sim \mathcal{U}[0,1]$.
  $y \leftarrow \log(x_0^{-1}x_1), x_t \leftarrow x_0 \exp(A_t(y))$.
  Update θ by minimizing $\|v_\theta(x_t,t) - b_t(y)\|_m^2$.
end while

TCM Training

$R_t(y) \leftarrow \exp(-A_t(y))\exp(y)$, $f_\theta(x,t) \leftarrow xR_t(h_\theta(x,t))$.
while not converged **do**
  Sample $x_0 \sim \rho_0, x_1 \sim \rho_1, t \sim \mathcal{U}[0,1]$.
  $y \leftarrow \log(x_0^{-1}x_1), x_t \leftarrow x_0 \exp(A_t(y))$.
  $u_t \leftarrow b_t(y)$ for TCT, $u_t \leftarrow \bar{b}_t(x_t)$ for TCD.
  $\hat{y}_t \leftarrow \partial_t h_\theta(x_t,t) + D_x^\ell h_\theta(x_t,t)[u_t]$.
  Update θ by minimizing residual $\|\omega_t\|_m^2$.
end while

MoTFM Training

while not converged **do**
  Sample $(x_0,\xi_0) \sim \mu_0, (x_1,\xi_1) \sim \mu_1, t \sim \mathcal{U}[0,1]$.
  Set $\xi_t^* \leftarrow a(t)\xi_0 + b(t)\xi_1$ and integrate x_t^* from x_0 using ξ_t^*.
  $u \leftarrow \log(x_1(x_1^*)^{-1}), q_t \leftarrow \exp(\eta(t)u)x_t^*$.
  $\xi_t \leftarrow \xi_t^* + \dot{\eta}(t)\mathrm{Ad}(x_t^*)^{-1}u$.
  Compute $v_t = \dot{\xi}_t$ from the corrected bridge.
  Update θ by minimizing $\|a_\theta(q_t,\xi_t,t) - v_t\|_m^2$.
end while

4.1 Endpoint-Constrained Trivialized Flow Matching

Fix an open set $U \subset \mathfrak{g}$ on which exp is a diffeomorphism onto its image. For endpoints satisfying $x_0^{-1}x_1 \in \exp(U)$, define the relative endpoint label $y = \log(x_0^{-1}x_1) \in U$. Throughout, we assume that G is connected and that this relative displacement lies in $\exp(U)$. This is typically much less restrictive than requiring the endpoints to lie in one chart, since ρ_0 and ρ_1 can have support that no single chart covers while their relative displacements remain small. The fundamental object in TFM is an ***algebraic bridge map*** $A : [0,1] \times U \to \mathfrak{g}$, written $A_t(y)$, satisfying $A_0(y) = 0$ and $A_1(y) = y$. It induces the group-valued bridge

$$x_t = x_0 \exp(A_t(y)).$$

By construction, this path satisfies $x_t|_{t=0} = x_0$ and $x_t|_{t=1} = x_1$. Its left-trivialized velocity is

$$b_t(y) = (L_{x_t^{-1}})_* \dot{x}_t = J_\ell(A_t(y))[\partial_t A_t(y)],$$

where $J_\ell(u)[v] = (L_{\exp(u)^{-1}})_* d\exp_u[v] = \frac{1-\exp(-\mathrm{ad}_u)}{\mathrm{ad}_u}[v]$ is the left Jacobian of the exponential map. Thus any choice of A_t determines an endpoint-constrained conditional vector field $b_t(y) \in \mathfrak{g}$, as shown in the following proposition. The proof is given in Appendix B.1.2.

Proposition 4.1. *Let $A : [0,1] \times U \to \mathfrak{g}$, $(t,y) \mapsto A_t(y)$, be C^1 in t, and suppose $A_0(y) = 0, A_1(y) = y, y \in U$. Then, for $y = \log(x_0^{-1}x_1)$, the curve $x_t = x_0 \exp(A_t(y))$ connects x_0 to x_1. Its left-trivialized velocity is $b_t(y) := (L_{x_t^{-1}})_* \dot{x}_t = J_\ell(A_t(y))[\partial_t A_t(y)]$.*

Trivialized Flow Matching. Proposition 4.1 gives the pathwise object used in TFM. Let π be a coupling of ρ_0 and ρ_1, and sample $(x_0,x_1) \sim \pi$. Define

$$y = \log(x_0^{-1}x_1), \qquad x_t = x_0 \exp(A_t(y)).$$

Let ρ_t denote the law of x_t. The corresponding marginal trivialized velocity field is $\bar{b}_t(x) = \mathbb{E}[b_t(y) \mid x_t = x]$. Thus the marginal probability flow is the left-trivialized ODE $\dot{x}_t = (L_{x_t})_* \bar{b}_t(x_t)$. TFM learns $\bar{b}_t$ by regressing an algebra-valued neural vector field $v_\theta : G \times [0,1] \to \mathfrak{g}$ onto the conditional bridge field. With respect to a chosen inner product on $\mathfrak{g}$, the ***TFM objective*** is

$$\mathcal{L}_{\mathrm{TFM}}(\theta) = \mathbb{E}_{t,(x_0,x_1)} \left[\|v_\theta(x_t,t) - b_t(y)\|_m^2 \right].$$

For each fixed t the minimizer satisfies $v_\theta^\star(x,t) = \mathbb{E}[b_t(y) \mid x_t = x] = \bar{b}_t(x)$. Therefore, the learned left-trivialized dynamics agree with the marginal bridge dynamics. The model only parameterizes a vector field in the fixed linear space $\mathfrak{g}$, and the tangent vector in $T_{x_t}G$ is obtained by left translation. Sampling integrates the learned field with exponential updates on the group and never evaluates $\log(x_t)$, so intermediate states do not need to stay in the chart used for the endpoint labels. The training and sampling algorithms are in Algorithms 1 and 2. Next, we discuss concrete choices of A_t.

Collinear Bridges. A simple choice of the bridge is choosing a scalar schedule $s \in C^1([0,1])$ with $s(0) = 0$ and $s(1) = 1$, and setting $A_t(y) = s(t)y$. Then $x_t = x_0 \exp(s(t)y)$, and $b_t(y) = \dot{s}(t)y$. We refer to this path as the collinear path and the corresponding model as Col-TFM, as the path remains in the one-parameter subgroup generated by y and the bridges lie in the span of y. In this case, the generative path is equivalent up to implementation to general Lie group flow matching (LFM) defined with the exponential curve [29] and, when G has a bi-invariant metric, to Riemannian flow matching [4]. Thus, Col-TFM can be seen as a generalization of LFM and RFM on Lie groups. Under a bi-invariant metric, collinear paths are geodesics and locally minimize energy, which makes them a natural choice. However, they confine the group trajectory to a one-parameter subgroup.

Non-Collinear Bridges. Note that in general $A_t(y)$ need not lie in the span of y. In fact, Proposition 4.1 shows that any C^1 map $A_t(y)$ satisfying $A_0(y) = 0$ and $A_1(y) = y$ gives an exact endpoint-constrained bridge. As a result, TFM enables more flexible generative path design than prior Lie group or Riemannian flow matching methods. Paths can be tailored to specific Lie groups or downstream tasks, for example with non-commuting dynamics or by avoiding poorly conditioned regions near branch cuts. In particular, we discuss below P-TFM, a construction of non-collinear paths for general tasks. The exact details are in Appendix B.5, along with other potential non-collinear bridges such as torsion-guided generative paths.

Preconditioned Paths. We introduce **preconditioned TFM (P-TFM)**, a non-collinear variant that makes it easier to reconstruct the endpoint from a predicted bridge label. Let $s \in C^1([0,1])$ satisfy $s(0) = 0, s(1) = 1$, and let $a \in C^1([0,1])$ satisfy $a(0) = a(1) = 0$ and $a(t) > 0$ for $t \in (0,1)$. Let $H : \mathfrak{g} \to \mathfrak{g}$ be self-adjoint. For $\rho \geq 0$, define $P_t^\rho = I + \rho a(t) H$ and

$$A_t^\rho(y) = y - (1 - s(t))P_t^\rho y.$$

Equivalently, $A_t^\rho(y) = s(t)y - \rho(1 - s(t))a(t)Hy$. Then $A_0^\rho(y) = 0$ and $A_1^\rho(y) = y$, so $x_t^\rho = x_0 \exp(A_t^\rho(y))$ is an exact endpoint-constrained bridge. Its conditional bridge field is

$$b_t^\rho(y) = J_\ell(A_t^\rho(y))[\partial_t A_t^\rho(y)], \qquad \partial_t A_t^\rho(y) = \dot{s}(t)P_t^\rho y - (1 - s(t))\partial_t P_t^\rho y.$$

When $\rho > 0$, $A_t^\rho(y)$ does not lie in the one-dimensional subspace spanned by y and is therefore non-collinear. This bridge is chosen to better map y to the remaining endpoint displacement, so that different bridge labels y map to more distinct endpoints. Define $R_t^\rho(y) = \exp(-A_t^\rho(y)) \exp(y)$ to be the residual map, so that along the bridge $(x_t^\rho)^{-1} x_1 = R_t^\rho(y)$. Let $D_t^\rho(y) = (L_{R_t^\rho(y)^{-1}})_* D_y R_t^\rho(y)$ be the left-trivialized differential of the residual map. We only consider times $t \leq 1 - \epsilon$, since $R_1^\rho(y) = e$ is constant and cannot be inverted. Define $\mathcal{H}_\epsilon(\rho) = \int_0^{1-\epsilon} \mathbb{E}\left[\|(D_t^\rho(y))^{-1}\|_{F,m}^2\right] dt$, where $y = \log(x_0^{-1} x_1)$. Since $(D_t^\rho(y))^{-1}$ measures the sensitivity of recovering the bridge label from the residual displacement, $\mathcal{H}_\epsilon(\rho)$ measures the average conditioning of the endpoint residual map, and we have the following proposition:

Proposition 4.2. *Suppose Y lies in a compact subset of the logarithm chart, and assume that for sufficiently small ρ, the corresponding $A_t^\rho(Y)$ and $R_t^\rho(Y)$ remain in compact sets on which the relevant Jacobians are well-defined. If $H = H^{\dagger_m} \succeq 0$ and $\mathrm{tr}_m(H) > 0$, then, on a sufficiently small chart, $\frac{d}{d\rho}\mathcal{H}_\epsilon(\rho)\Big|_{\rho=0} < 0$. Consequently, for all sufficiently small $\rho > 0$, $\mathcal{H}_\epsilon(\rho) < \mathcal{H}_\epsilon(0)$.*

The proof is given with Proposition B.13 in the appendix. Proposition 4.2 shows that P-TFM locally improves the residual endpoint-learning problem. In a small logarithm chart, $D_t^\rho(y) = (1 - s(t))(I + \rho a(t)H) + O(\|y\|_m)$. Thus increasing ρ in a positive direction decreases the norm of $(D_t^\rho(y))^{-1}$ to first order. Equivalently, nearby labels become more separated through the residual map R_t^ρ, so different labels produce more distinguishable remaining motions toward their endpoints.

4.2 Trivialized Consistency Models

We next define ***Trivialized Consistency Models (TCM)***, the few-step generative model on G. For a bridge map A_t as above, define the endpoint-residual map $R_t(y) := \exp(-A_t(y)) \exp(y)$. Along

an exact bridge $x_t = x_0 \exp(A_t(y))$, we have $x_t^{-1} x_1 = R_t(y)$. Let $h_\theta : G \times [0,1] \to U$ be a bridge-label predictor. We define the TCM endpoint map by

$$f_\theta(x,t) := x R_t(h_\theta(x,t)) = x \exp(-A_t(h_\theta(x,t))) \exp(h_\theta(x,t)).$$

The boundary condition $A_1(y) = y$ implies $R_1(y) = e$, so $f_\theta(x,1) = x$, which satisfies the consistency boundary condition at $t = 1$. For $t < t + \Delta$, the ***discrete TCM objective*** is

$$\mathcal{L}_{\mathrm{TCM},\Delta}(\theta) = \mathbb{E}\left[d_m(f_\theta(x_t,t), \mathrm{sg}(f_\theta(x_{t+\Delta}, t+\Delta)))^2\right],$$

where $x_t = x_0 \exp(A_t(y))$, $x_{t+\Delta} = x_0 \exp(A_{t+\Delta}(y))$, and sg denotes the stop-gradient operator.

Continuous Consistency Distillation and Training. As the discrete consistency objective typically incurs significant error that is amplified in curved geometry [5], we propose to instead use its continuous-time limit. Let $\bar{b}_t(x) = \mathbb{E}[b_t(y) \mid x_t = x]$ be the marginal trivialized velocity field from TFM, and write $\hat{y}_t = h_\theta(x_t, t)$ and $R_t^\theta = R_t(\hat{y}_t)$. The following proposition gives the ***continuous TCM distillation objective***:

Proposition 4.3. *Assume the preceding objects are C^1 and that the local metric expansion is uniformly integrable along the sampled paths. Then $\lim_{\Delta \to 0} \Delta^{-2} \mathcal{L}_{\mathrm{TCM},\Delta} = \mathbb{E}\left[\|\omega_t\|_m^2\right]$, where*

$$\omega_t = \mathrm{Ad}_{(R_t^\theta)^{-1}} \bar{b}_t(x_t) + \mathrm{Ad}_{\exp(-\hat{y}_t)} J_\ell(-A_t(\hat{y}_t))[-\partial_t A_t(\hat{y}_t) - D_y A_t(\hat{y}_t)[\dot{\hat{y}}_t]] + J_\ell(\hat{y}_t)[\dot{\hat{y}}_t].$$

$$\dot{\hat{y}}_t = \partial_t h_\theta(x_t,t) + D_x^\ell h_\theta(x_t,t)[\bar{b}_t(x_t)].$$

The proof is given in Appendix B.1.4. Thus the ***continuous distillation TCM objective*** is $\mathcal{L}_{\mathrm{TCD}}^{\mathrm{cont}}(\theta) = \mathbb{E}[\|\omega_t\|_m^2]$. For consistency training, let $D_t(y) := (L_{R_t(y)^{-1}})_* D_y R_t(y) : \mathfrak{g} \to \mathfrak{g}$. The following justifies replacing the marginal bridge with the conditional bridge y sampled during training:

Proposition 4.4. *Fix $t < 1$, and suppose $D_t(y)$ is invertible on the labels under consideration. Consider the unscaled stop-gradient consistency update in a fixed local chart on G. Let $\mathcal{L}_\Delta^{\mathrm{marg}}$ denote the loss obtained by evolving the detached branch with the marginal velocity $\bar{b}_t(x_t)$, and let $\mathcal{L}_\Delta^{\mathrm{cond}}$ denote the loss obtained by evolving it with the sampled conditional velocity $b_t(y)$. Then, under the usual local C^2-bounds on f_θ,*

$$\nabla_\theta^{\mathrm{sg}} \mathcal{L}_\Delta^{\mathrm{cond}} = \nabla_\theta^{\mathrm{sg}} \mathcal{L}_\Delta^{\mathrm{marg}} + O(\Delta^2), \qquad \Delta \downarrow 0.$$

Thus, the two constructions have the same first-order population training vector field.

Consequently, the ***continuous consistency training objective*** is

$$\mathcal{L}_{\mathrm{TCT}}^{\mathrm{cont}}(\theta) = \mathbb{E}_{t,(x_0,x_1)}\left[\|\omega_t^{\mathrm{cond}}\|_m^2\right],$$

where $\dot{\hat{y}}_t^{\mathrm{cond}} = \partial_t h_\theta(x_t,t) + D_x^\ell h_\theta(x_t,t)[b_t(y)]$ and

$$\omega_t^{\mathrm{cond}} = \mathrm{Ad}_{(R_t^\theta)^{-1}} b_t(y) + \mathrm{Ad}_{\exp(-\hat{y}_t)} J_\ell(-A_t(\hat{y}_t))[-\partial_t A_t(\hat{y}_t) - D_y A_t(\hat{y}_t)[\dot{\hat{y}}_t^{\mathrm{cond}}]] + J_\ell(\hat{y}_t)[\dot{\hat{y}}_t^{\mathrm{cond}}].$$

The proof is in Appendix B.1.5. The training and sampling algorithms are in Algorithms 1 and 2.

Comparison to RCM. TCM fixes a single vector space $\mathfrak{g}$ instead of modeling a vector field that changes with respect to position. Thus, the TCM loss is simpler in that it does not compute geometric terms that account for differentiating and transporting vectors across position-dependent tangent spaces, such as the covariant derivative and Christoffel symbols. RCM requires these terms, whereas TCM represents the corresponding quantities in the fixed Lie algebra. On Lie groups without a bi-invariant metric, RCM also requires simulation of Euler-Arnold dynamics, which TCM avoids. As an example, consider a planar robot with three body-frame controls in Figure 1: moving straight, turning left, and turning right. Applying each control generates a trajectory of poses in $SE(2)$. In RCM, the velocity at each pose lies in a different tangent space, so the same body-frame control corresponds to different world-frame tangent vectors as the robot rotates. TCM instead left-trivializes these velocities into the Lie algebra, where each body-frame control has a consistent coordinate representation at every pose. We show that this leads to better performance and faster convergence in Section 5.

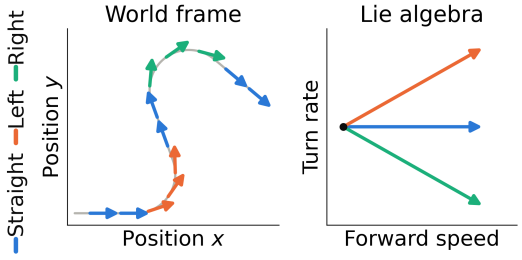


Figure 1: Planar robot on $SE(2)$. The same body-frame control gives different world-frame tangent vectors as the robot rotates (left), but has fixed coordinates in the Lie algebra (right).

4.3 Momentum Extensions of TFM and TCM

TDM [42] established left-trivialized momentum dynamics for diffusion on Lie groups. However, its invariant reference distribution relies on a normalizable Haar measure, which restricts it to compact Lie groups. Here we propose MoTFM and MoTCM, which extend TFM and TCM to allow the Lie algebra path to have true dynamics of its own. To our knowledge, these are the first momentum-based flow matching and consistency models on Lie groups. Instead of an invariant reference distribution, they use an explicit endpoint correction that pulls the generative path toward a reference path, giving a bridge that reaches the target endpoint exactly and preserves the prescribed endpoint momenta, including on non-compact Lie groups.

Momentum Trivialized Flow Matching. Let μ_0, μ_1 be source and target distributions on $G \times \mathfrak{g}$. Given endpoint data $(x_0, \xi_0), (x_1, \xi_1) \in G \times \mathfrak{g}$, choose $a, b \in C^1([0,1])$ with $a(0) = 1, a(1) = 0, b(0) = 0, b(1) = 1$, and let $\xi_t^* = a(t)\xi_0 + b(t)\xi_1$. However, this naive interpolation does not in general connect x_0 to x_1. To overcome this difficulty, let x_t^* solve $\dot{x}_t^* = (L_{x_t^*})_* \xi_t^*$ with $x_0^* = x_0$. Assuming $E = x_1(x_1^*)^{-1}$ lies in the logarithm chart, set $u = \log(E)$. For $\eta \in C^2([0,1])$ satisfying $\eta(0) = 0, \eta(1) = 1, \dot{\eta}(0) = \dot{\eta}(1) = 0$, define $z_t = \exp(\eta(t)u)$ and $q_t = z_t x_t^*$. The final generative path on G is then $x_t = q_t$. The following proposition justifies the validity of this choice:

Proposition 4.5. *The path $q_t = z_t x_t^*$ satisfies $q_0 = x_0$ and $q_1 = x_1$. Its left-trivialized velocity is*

$$\xi_t = \xi_t^* + \dot{\eta}(t)\,\mathrm{Ad}_{(x_t^*)^{-1}}\,u; \qquad \dot{\xi}_t = \dot{\xi}_t^* + \ddot{\eta}(t)\,\mathrm{Ad}_{(x_t^*)^{-1}}\,u - \dot{\eta}(t)[\xi_t^*, \mathrm{Ad}_{(x_t^*)^{-1}}\,u].$$

In particular, $\xi(0) = \xi_0$ and $\xi(1) = \xi_1$.

Thus the corrected bridge gives endpoint-matching dynamics on $G \times \mathfrak{g}$. Since the pose velocity is already given by the state variable ξ_t, MoTFM only needs to learn the acceleration field. Let $a_\theta : G \times \mathfrak{g} \times [0,1] \to \mathfrak{g}$. The ***momentum trivialized flow matching loss*** is then

$$\mathcal{L}_{\mathrm{MoTFM}}(\theta) = \mathbb{E}_{t,((x_0,\xi_0),(x_1,\xi_1))}\left[\|a_\theta(q_t, \xi_t, t) - \dot{\xi}_t\|_m^2\right].$$

The proof and further theoretical guarantees for MoTFM, such as path stability bounds, are given in Appendix B.2. The training and sampling algorithms are in Algorithms 1 and 2.

Momentum Trivialized Consistency Models. For few-step generation, let $f_\theta = (f_\theta^x, f_\theta^\xi) : G \times \mathfrak{g} \times [0,1] \to G \times \mathfrak{g}$. We use the endpoint maps

$$f_\theta^x(x, \xi, t) = x \exp(u_\theta(x, \xi, t)), \qquad u_\theta(x, \xi, t) = r(t) h_\theta^x(x, \xi, t),$$

and $f_\theta^\xi(x, \xi, t) = \xi + r(t) h_\theta^\xi(x, \xi, t)$, where $r(1) = 0$. The derivation for the consistency loss follows the same structure as TCM, and thus we defer the details to Appendix B. The loss can be written as

$$\begin{aligned}\mathcal{L}_{\mathrm{MoTCM}}^{\mathrm{cont}}(\theta) = \mathbb{E}\Big[&\| \mathrm{Ad}_{\exp(-u_t)}\,\xi_t + J_\ell(u_t)[\dot{r}(t)h_\theta^x + r(t)(\partial_t h_\theta^x + D_x^\ell h_\theta^x[\xi_t] + D_\xi h_\theta^x[\dot{\xi}_t])]\|_m^2 \\ &+ \|\dot{\xi}_t + \dot{r}(t)h_\theta^\xi + r(t)(\partial_t h_\theta^\xi + D_x^\ell h_\theta^\xi[\xi_t] + D_\xi h_\theta^\xi[\dot{\xi}_t])\|_m^2\Big]\end{aligned}$$

The derivation is given in Appendix B.4. As training consistency models with second order dynamics is largely unexplored and consistency models are known to exhibit training instabilities [21], we show a proof of concept for MoTCM in Appendix C.5. Exploring the training dynamics of consistency models with momentum in general is outside the scope of this paper.

5 Experiments

We evaluate TGM in 4 settings: (1) on compact Lie groups, which is standard for benchmarking in prior works, where we adopt the $SO(3)$ benchmarks from RCM [5] to test both many-step and few-step generation; (2) on non-compact Lie groups, which is more difficult and rarely tested in prior works; and (3) on protein backbone generation and (4) on robot grasp generation, which test TGM on real-world tasks. We let TCT and TCD denote continuous consistency training and distillation.

5.1 Experiments on $SO(3)$

Table 2: MMD (10^{-2}) comparison of TGM variants against their respective baselines on $SO(3)$ datasets, averaged over 5 runs, where lower is better. Asterisks mark our proposed models, and the suffix on a flow matching model gives its number of sampling steps. The best result in each category is bolded. TGM variants all significantly outperform their respective baseline models.

MMD ↓ /10^{-2}	*Flow matching*				*Consistency training*		*Consistency distillation*		*With momentum*	
	RFM-100	RFM-2	P-TFM-100*	P-TFM-2*	RCT	TCT*	RCD	TCD*	TDM	MoTFM*
Cone	1.63±0.17	13.87±0.64	**0.72**±0.11	12.07±0.57	8.06±0.76	**3.34**±0.31	7.21±0.40	**4.17**±0.18	1.32±0.15	**0.69**±0.07
Fisher	2.08±0.07	12.79±0.40	**1.25**±0.09	10.17±0.55	10.74±0.36	**2.32**±0.17	9.30±0.12	**2.04**±0.30	4.52±0.26	**1.64**±0.05
Line	2.22±0.12	9.56±0.64	**1.31**±0.07	6.72±0.13	6.29±0.68	**1.72**±0.06	5.14±0.42	**1.31**±0.15	2.46±0.18	**0.70**±0.06

Table 4: MMD (10^{-1}) and generation time comparison of TGM variants against baselines on $Sp(4, \mathbb{R})$ datasets, where lower is better. MMD is averaged over 5 runs. Asterisks mark our proposed models. The best result in each category is bolded. The last row gives the time in seconds to generate 10K samples, averaged over 10 iterations, using 100 steps for flow matching and momentum models and 2 steps for consistency models. TGM variants alleviate the limitations of prior works and achieve significantly better results.

MMD ↓ /10^{-1}	*Flow matching*			*Consistency training*		*Consistency distillation*		*With momentum*	
	RFM	Col-TFM*	P-TFM*	RCT	TCT*	RCD	TCD*	TDM	MoTFM*
Braid	5.74±0.21	0.57±0.08	**0.41**±0.04	7.25±0.53	**2.03**±0.23	7.13±0.19	**2.14**±0.34	2.01±0.20	**1.70**±0.20
Quantum	4.84±0.57	0.58±0.23	**0.48**±0.16	5.70±0.33	**1.29**±0.32	5.77±0.56	**1.45**±0.46	1.93±0.38	**1.06**±0.23
Optics	6.89±0.56	0.77±0.17	**0.47**±0.09	7.42±0.55	**1.58**±0.14	8.27±0.79	**2.19**±1.33	5.88±0.44	**0.78**±0.28
Generation Time (s)	131.21±0.14	**1.31**±0.03	**1.30**±0.03	1.39±0.04	**0.10**±0.01	1.39±0.06	**0.10**±0.01	6.24±0.06	**2.33**±0.07

For experiments on $SO(3)$, we follow the setup in [5, 19], where we use synthetic datasets with three different modes (Cone, Fisher, Line). For each dataset, 100K data points are sampled for training and 10K separate data points are used for MMD evaluation. We test each TGM variant against the current state-of-the-art baseline in its category: (1) P-TFM against RFM [4] for flow matching, which tests the advantages of non-collinear paths since on $SO(3)$, RFM, LFM [29], and Col-TFM are equivalent up to implementation differences; (2) TCM against RCM [5] for consistency models; and (3) MoTFM against TDM [42] for models with momentum. The consistency models are evaluated with 2 sampling steps, the flow matching models with 100 and 2 steps, and the momentum models with 100 steps. The results, averaged over 5 runs, are in Table 2. We find that each TGM variant significantly outperforms the baselines in its category. P-TFM's improvement over RFM demonstrates the benefit of non-collinear paths. Moreover, TCM achieves significantly better few-step generation quality than P-TFM at the same 2 steps, demonstrating the advantage of our trivialized consistency models. Momentum can also help, as MoTFM outperforms P-TFM on the Line dataset, while MoTFM is about 33% faster per training iteration than TDM.

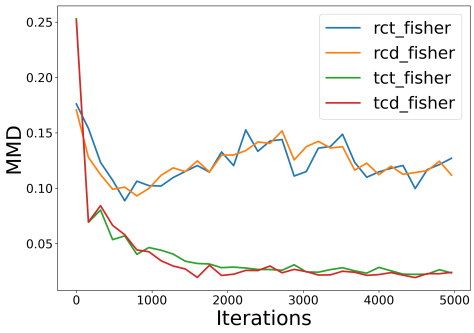


Figure 2: MMD of the Fisher dataset for TCM and RCM models as training progresses. TCM models have much faster convergence.

Consistency Model Convergence. TCM variants not only achieve significantly better performance than RCM variants, but also exhibit much faster convergence. Figure 2 shows that across training stages, TCM models converge in far fewer iterations than RCM and have smoother convergence behavior, suggesting that the simpler loss term for TCM, which results from fixing the vector space $\mathfrak{g}$, is easier to optimize. One training iteration of TCM takes at most about 3% longer than RCM, so the faster convergence carries over to training time.

Table 3: Runtime comparison between Riemannian baselines and TGM for one training iteration. Asterisks mark our proposed models.

Family	Model	Runtime (s) ↓	Speedup
FM	RFM	7.12	-
	TFM*	**0.39**	**18.26**×
CT	RCT	11.26	-
	TCT*	**0.73**	**15.42**×

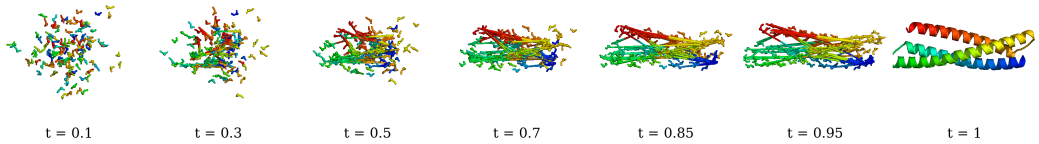


Figure 3: Visualization of the generative trajectory of a protein backbone by P-TFM.

5.2 Experiments on Non-Compact Lie Groups

We test TGM on more difficult non-compact Lie groups, which pose many challenges to prior works. Since they are non-compact, momentum-based models such as TDM face difficulty connecting their source distribution to the target distribution. Additionally, since many of them do not admit a bi-invariant metric, RFM and RCM require simulating Euler-Arnold dynamics [4], which reduces efficiency and accuracy. Many are also *simple Lie groups*, which do not decompose as the direct or semi-direct product of smaller non-trivial Lie groups, and thus prior methods cannot generate separately on the factors, a common workaround on $SE(3)$ [39, 40]. We build 3 datasets on the real symplectic group $Sp(4, \mathbb{R})$: a squeeze-braid dataset, a two-mode Gaussian quantum circuit dataset, and a paraxial optics ray-transfer dataset. The latter two are motivated by symplectic structures in continuous quantum systems and classical optics theory [2, 31].

Results. The results are shown in Table 4. Across all three datasets, TGM variants significantly outperform their corresponding baselines. For flow matching, Col-TFM already improves over RFM by avoiding Euler-Arnold simulation, while P-TFM further improves performance. For consistency models, TCT and TCD reduce MMD substantially compared with RCT and RCD, showing the benefit of fixing the learning problem in the Lie algebra rather than differentiating position-dependent vector fields. The continuous models also improve on their discrete counterparts, as shown in Appendix C.2.1. For momentum models, MoTFM improves over TDM, demonstrating that TGM alleviates TDM's limitations on non-compact Lie groups. TGM variants also generate samples faster. TFM and TCM avoid Euler-Arnold simulation, giving about 100X and 14X speedups over RFM and RCM, and MoTFM is about 2.7X faster than TDM.

Runtime. We compare the runtime for one optimizer step of TGM variants against RFM and RCM, as the latter two require simulating Euler-Arnold dynamics on $Sp(4, \mathbb{R})$ due to the lack of a bi-invariant metric. As shown in Table 3, TFM and TCT avoid this simulation and achieve over 15X speedup compared to the Riemannian models. The speedup is specific to the setting where the group lacks a bi-invariant metric, where RFM and RCM require Euler-Arnold simulation.

5.3 Experiments on Protein Backbone Generation

Experimental Setup. For protein backbone generation, we follow the experimental setup of FrameFlow [39], training on the SCOPe dataset with proteins of length at most 128 and evaluating on unconditional protein monomer generation. Each protein backbone is represented as a sequence of residue frames $T = (x, R) \in SE(3)^N$, with residue translation $x \in \mathbb{R}^3$ and residue orientation $R \in SO(3)$. As in FrameFlow, we use the product metric on $SE(3)^N$, which allows the translation and rotation components to be trained and sampled independently. The translation path is the same as FrameFlow, namely the linear interpolation $x_t =$

Table 5: Unconditional protein backbone generation results. Higher designability and diversity are better, while lower novelty is better. Asterisks mark our proposed model, and the best result in each column is bolded. P-TFM improves significantly over FrameFlow in designability.

Model		Timesteps	Designability (↑)	Diversity (↑)	Novelty (↓)
GENIE	SDE	1000	0.22	0.76	0.54
		750	0.11	**0.79**	**0.51**
		500	0.0	-	-
FrameDiff	SDE	500	0.42	0.36	0.66
		100	0.39	0.32	0.66
FrameDiff	ODE	100	0.26	0.43	0.65
		10	0.16	0.53	0.66
FrameFlow	ODE	500	0.81	0.22	0.69
		100	0.77	0.28	0.67
		10	0.33	0.54	0.63
P-TFM*	ODE	500	**0.88**	0.18	0.72
		100	0.85	0.32	0.67
		10	0.55	0.55	0.66

$(1-t)x_0 + tx_1$. For the rotation component, we replace FrameFlow's geodesic interpolation, which uses RFM, with the preconditioned algebraic bridge of P-TFM.

Evaluation. We follow the setup of FrameFlow and sample backbones for each length between 60 and 128. For each generated backbone, ProteinMPNN designs sequences and ESMFold folds the designed sequences. We report designability, diversity, and novelty. A backbone is considered designable if the minimum self-consistency RMSD between it and the ESMFold-predicted structures is below 2.0 Å. Diversity is computed by clustering designable samples with MaxCluster, and novelty is measured by searching designable samples against PDB structures with FoldSeek and reporting the average maximum TM-score.

Results. The results are shown in Table 5. P-TFM consistently improves designability over FrameFlow across all sampling budgets, by 0.07 to 0.22, at the same sampling cost since only the learning target changes. Notably, the improvement is the largest for generation with fewer steps, where P-TFM achieves 0.55 designability at 10 steps compared with 0.33 for FrameFlow. We treat designability as the main metric following FrameFlow [39], since it measures whether a generated backbone is biologically feasible, while diversity is only meaningful when designability is reasonably strong. FrameFlow is therefore the most relevant comparison, as the baselines with higher diversity have far lower designability. Additionally, compared with FrameFlow, P-TFM increases diversity at 100 and 10 steps and decreases it by 0.04 at 500 steps, while novelty worsens by at most 0.03, indicating that the improvement does not simply come from collapsing to a narrow set of structures. We visualize the generative path of a backbone generated by P-TFM in Figure 3, showing a smooth convergence.

5.4 Experiments on Robot Grasp Generation

Experimental Setup. For robot grasp generation, we follow the experimental setup of EquiGraspFlow [17]. Given the point cloud of an object, the model generates a distribution of valid 6-DoF grasp poses in $SE(3)$ for a parallel-jaw gripper. We use objects from the ACRONYM dataset [10], whose grasps are labeled by physics simulation, and train without rotation augmentation. EquiGraspFlow models rotation and translation separately, so we keep its translation path and replace its rotation path with P-TFM. We evaluate the Earth Mover's Distance (EMD) between generated and ground-truth grasp poses, and the grasp success rate in simulation.

Table 6: Robot grasp generation results without rotation augmentation. Higher success rate and lower EMD are better. Asterisks mark our proposed model, and the best result in each column is bolded. Baseline results are from EquiGraspFlow.

Model	Mug		Laptop	
	Success (%) (↑)	EMD (↓)	Success (%) (↑)	EMD (↓)
6-DOF GraspNet [25]	32.58	0.8285	32.96	0.7990
PoiNt-SE(3)-Dif [36]	19.14	0.7648	13.03	0.7184
EquiGraspFlow [17]	90.56	0.4804	99.72	0.3579
P-TFM*	**91.76**	**0.4096**	**99.76**	**0.3271**

Results. The results are shown in Table 6. P-TFM achieves the best success rate and EMD on both object categories. The improvement is largest on the harder Mug category, where P-TFM reduces EMD from 0.4804 to 0.4096 compared with EquiGraspFlow and raises the success rate from 90.56% to 91.76%. On Laptop, where EquiGraspFlow already reaches a 99.72% success rate, P-TFM still lowers EMD from 0.3579 to 0.3271. Since P-TFM only changes the rotation path, these results suggest that non-collinear paths also benefit conditional generation on $SE(3)$.

6 Conclusion

We introduced TGM, a framework for generative modeling on Lie groups by learning endpoint-constrained paths in the Lie algebra. This gives rise to TFM, TCM, and momentum-based extensions that avoid position-dependent vector fields, support flexible path design, and address prior challenges on non-compact Lie groups. Experiments on $SO(3)$, $Sp(4,\mathbb{R})$, protein backbone and robot grasp generation show that TGM improves sample quality over existing Riemannian and Lie group baselines, and is more efficient on Lie groups without a bi-invariant metric.

Acknowledgments and Disclosure of Funding

Research was supported in part by the DOE Center for Advanced Bioenergy and Bioproducts Innovation (U.S. Department of Energy, Office of Science, Biological and Environmental Research Program under Award Number DESC0018420). Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Energy.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Arvind, B. Dutta, N. Mukunda, and R. Simon. The real symplectic groups in quantum mechanics and optics. *Pramana*, 45:471–497, 1995.

[3] Marco Bertolini, Tuan Le, and Djork-Arné Clevert. Diffusion generative modeling on lie group representations. In *NeurIPS*, 2025.

[4] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *ICLR*, 2024.

[5] Chaoran Cheng, Yusong Wang, Yuxin Chen, Xiangxin Zhou, Nanning Zheng, and Ge Liu. Riemannian consistency model. In *NeurIPS*, 2025.

[6] Kyle Cranmer, Gurtej Kanwar, Sébastien Racanière, Danilo J Rezende, and Phiala E Shanahan. Advances in machine-learning-based sampling motivated by lattice quantum chromodynamics. *Nature Reviews Physics*, 5(9):526–535, 2023.

[7] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *TPAMI*, 2023.

[8] Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

[9] Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

[10] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *ICRA*, 2021.

[11] Jacques Faraut. *Analysis on Lie Groups: An Introduction*. Cambridge University Press, 2008.

[12] Wendelin Feiten, Muriel Lang, and Sandra Hirche. Rigid motion estimation using mixtures of projected gaussians. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1465–1472. IEEE, 2013.

[13] Mevlana C Gemici, Danilo Rezende, and Shakir Mohamed. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

[16] Kacper Kapuśniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Avishek Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *NeurIPS*, 2024.

[17] Byeongdo Lim, Jongmin Kim, Jihwan Kim, Yonghyeon Lee, and Frank C. Park. EquiGraspFlow: SE(3)-equivariant 6-DoF grasp pose generative flows. In *CoRL*, 2024.

[18] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[19] Yulin Liu, Haoran Liu, Yingda Yin, Yang Wang, Baoquan Chen, and He Wang. Delving into discrete normalizing flows on SO(3) manifold for probabilistic rotation modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21264–21273, 2023.

[20] Aaron Lou, Minkai Xu, Adam Farris, and Stefano Ermon. Scaling riemannian diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *ICLR*, 2024.

[22] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

[23] Oren Mangoubi, Neil He, and Nisheeth K. Vishnoi. Efficient diffusion models for symmetric manifolds. In *ICML*, 2025.

[24] E. Mathieu and M. Nickel. Riemannian continuous normalizing flows. In *Advances in Neural Information Processing Systems*, 2020.

[25] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *ICCV*, 2019.

[26] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *NeurIPS*, 2025.

[27] Danilo Jimenez Rezende, George Papamakarios, Sébastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, pages 8083–8092. PMLR, 2020.

[28] Jon M Selig. *Geometrical methods in robotics*. Springer Science & Business Media, 2013.

[29] Finn M. Sherry and Bart M. N. Smets. Flow matching on lie groups. In *Geometric Science of Information*, 2025.

[30] Di Shi, Long Li, Yixin Shao, Wuxiang Zhang, and Xilun Ding. Multi-mode control strategy for robotic rehabilitation on special orthogonal group so (3). *IEEE Transactions on Industrial Electronics*, 2023.

[31] R. Simon and Kurt Bernardo Wolf. Structure of the set of paraxial optical systems. *Journal of the Optical Society of America A*, 17(2):342–355, 2000.

[32] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[34] Peter Sorrenson, Felix Draxler, Armand Rousselot, Sander Hummerich, and Ullrich Köthe. Learning distributions on manifolds with free-form flows. *Advances in Neural Information Processing Systems*, 37:66961–66994, 2024.

[35] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

[36] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *ICRA*, 2023.

[37] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[38] Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching. *arXiv preprint arXiv:2502.15681*, 2025.

[39] Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noé. Fast protein backbone generation with SE(3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

[40] Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *ICML*, 2023.

[41] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[42] Yuchen Zhu, Tianrong Chen, Lingkai Kong, Evangelos A. Theodorou, and Molei Tao. Trivialized momentum facilitates diffusion generative modeling on lie groups. In *ICLR*, 2025.

Algorithm 2 Sampling Algorithms for TGM Variants

Require: Trained networks $v_\theta, h_\theta, a_\theta$; grid $0 = t_0 < \cdots < t_K = 1$.

TFM Sampling

Sample $x_{t_0} \sim \rho_0$.
for $k = 0, \dots, K-1$ **do**
  $\Delta t \leftarrow t_{k+1} - t_k$.
  $x_{t_{k+1}} \leftarrow x_{t_k} \exp(\Delta t\, v_\theta(x_{t_k}, t_k))$.
end for
return x_{t_K}.

TCM Sampling with Renoising

Sample $x_{t_0} \sim \rho_0$.
for $k = 0, \dots, K-1$ **do**
  $\hat{x}_1 \leftarrow f_\theta(x_{t_k}, t_k)$.
  Sample noise $\tilde{x}_0 \sim \rho_0$.
  $\tilde{y}_k \leftarrow \log(\tilde{x}_0^{-1} \hat{x}_1)$.
  $x_{t_{k+1}} \leftarrow \tilde{x}_0 \exp(A_{t_{k+1}}(\tilde{y}_k))$.
end for
return x_{t_K}.

MoTFM Sampling

Sample $(x_{t_0}, \xi_{t_0}) \sim \mu_0$.
for $k = 0, \dots, K-1$ **do**
  $\Delta t \leftarrow t_{k+1} - t_k$.
  $\xi_{k+\frac{1}{2}} \leftarrow \xi_{t_k} + \frac{\Delta t}{2} a_\theta(x_{t_k}, \xi_{t_k}, t_k)$.
  $x_{t_{k+1}} \leftarrow x_{t_k} \exp(\Delta t\, \xi_{k+\frac{1}{2}})$.
  $\xi_{t_{k+1}} \leftarrow \xi_{k+\frac{1}{2}} + \frac{\Delta t}{2} a_\theta(x_{t_{k+1}}, \xi_{k+\frac{1}{2}}, t_{k+1})$.
end for
return x_{t_K}.

A Background on Lie Groups

This section reviews the Lie group concepts used in the paper. See [11] for futher details.

Lie Groups and Lie Algebras. A Lie group G is a group that is also a smooth manifold, such that multiplication and inversion are smooth maps. Many groups in applications are matrix Lie groups, namely closed subgroups of $\mathrm{GL}(n, \mathbb{R})$. Examples include the rotation group $SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^\top R = I,\ \det R = 1\}$, the rigid motion group $SE(3) = \mathbb{R}^3 \rtimes SO(3)$, and the real symplectic group $Sp(2n, \mathbb{R}) = \{S \in \mathrm{GL}(2n, \mathbb{R}) : S^\top J S = J\}$, where $J = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}$. The Lie algebra $\mathfrak{g} = T_e G$ is the tangent space at the identity e. It is a finite-dimensional vector space equipped with a bracket $[\cdot, \cdot]$. For matrix groups, $\mathfrak{g}$ is a space of matrices and the Lie bracaket is given as $[A, B] = AB - BA$. For example, $\mathfrak{so}(3)$ is the space of skew-symmetric 3×3 matrices.

Exponential and Logarithm Maps. The exponential map $\exp : \mathfrak{g} \to G$ sends $\xi \in \mathfrak{g}$ to $\gamma(1)$, where γ is the one-parameter subgroup with $\dot{\gamma}(0) = \xi$. For matrix groups, this is the matrix exponential. Since $\exp$ is a local diffeomorphism near 0, it has a local inverse $\log$ on a chart around the identity. Globally, $\exp$ need not be injective or surjective, and its image lies in the identity component of G. On connected compact Lie groups such as $SO(3)$, $\exp$ is surjective but not injective, since rotating by angles θ and $\theta + 2\pi$ about an axis gives the same element. On $Sp(4, \mathbb{R})$, $\exp$ is neither surjective nor injective. As discussed in the main text, TGM only applies $\log$ to the relative displacement between endpoints.

Left Translation and Trivialization. For $x \in G$, left translation is $L_x(h) = xh$, and its differential $(L_x)_*$ maps $T_h G$ to $T_{xh} G$. The tangent spaces at different group elements are different vector spaces, and thus velocities at different elements cannot be compared without additional structure. Left translation provides this structure by identifying each $T_x G$ with $\mathfrak{g}$. For a C^1 curve x_t, the left-trivialized velocity is $\xi_t = (L_{x_t^{-1}})_* \dot{x}_t \in \mathfrak{g}$, which for matrix groups is $\xi_t = x_t^{-1} \dot{x}_t$. In robotics, this is the body-frame velocity, as in Figure 1.

Adjoint Action and Left Jacobian. The adjoint action $\mathrm{Ad}_x : \mathfrak{g} \to \mathfrak{g}$ is the differential at e of the conjugation $h \mapsto xhx^{-1}$, which for matrix groups is $\mathrm{Ad}_x\, \xi = x \xi x^{-1}$. It converts between left-trivialized frames at different group elements. Its infinitesimal form is $\mathrm{ad}_\xi\, \eta = [\xi, \eta]$. The left

Jacobian $J_\ell(u)[v] = (L_{\exp(u)^{-1}})_* d\exp_u[v] = \frac{1-\exp(-\operatorname{ad}_u)}{\operatorname{ad}_u}[v]$ gives the left-trivialized velocity of a curve $\exp(u_t)$, with $u = u_t$ and $v = \dot{u}_t$. If u and v commute, then $J_\ell(u)[v] = v$.

Metrics and Geodesics. An inner product on $\mathfrak{g}$ extends by left translation to a left-invariant Riemannian metric on G. The metric is bi-invariant if it is also invariant under right translation, which holds when the inner product is invariant under Ad_x. Under a bi-invariant metric, geodesics through x have the closed form $x\exp(t\xi)$. Compact Lie groups admit bi-invariant metrics, but groups such as $SE(3)$ and $Sp(2n,\mathbb{R})$ do not. Under a general left-invariant metric, geodesics satisfy the Euler-Arnold equation $\dot{\xi}_t = \operatorname{ad}^\dagger_{\xi_t} \xi_t$, where $\operatorname{ad}^\dagger$ is the adjoint of ad with respect to the inner product. This equation generally has no closed-form solution and must be simulated numerically, which is the cost that RFM and RCM incur on such groups.

Haar Measure and Compactness. Every Lie group has a left Haar measure, a left-invariant measure that is unique up to scaling. This measure is finite if and only if the group is compact. Thus, only compact Lie groups have a normalizable Haar measure that defines an invariant probability distribution, which is why methods with an invariant reference distribution, such as TDM, are restricted to compact groups.

B Theoretical Results and Proofs

This appendix collects the algebraic identities and endpoint constructions used in the trivialized flow and consistency models. Appendix B.1 treats non-momentum bridges of the form $x_t = x_0\exp(A_t(y))$, derives the residual endpoint map, and proves the continuous TCM limit. Appendix B.2 treats momentum bridges, where the group endpoint constraint is not automatic after interpolating velocities. Appendix B.5.2 then gives a protein-specific path design on $SE(3)^N$ and its rotation-only specialization.

B.1 Endpoint-constrained trivialized bridges

Let $y = \log(x_0^{-1}x_1)$ in a fixed logarithm chart. We use endpoint-constrained algebra paths $A_t(y)$ satisfying $A_0(y) = 0$ and $A_1(y) = y$, and set $x_t = x_0\exp(A_t(y))$. This gives an exact bridge from x_0 to x_1. Its left-trivialized velocity is $b_t(y) = J_\ell(A_t(y))[\partial_t A_t(y)]$. The collinear choice $A_t(y) = s(t)y$ recovers the usual exponential interpolation, while non-collinear choices of A_t bend the path in $\mathfrak{g}$ without changing the endpoint constraint. The appendix gives the corresponding product-exponential identities and the continuous TCM limit.

B.1.1 Notation and left-trivialized differentials

For $g \in G$, let $L_g : G \to G$ denote left multiplication. For a C^1 curve $x_t \in G$, its left logarithmic derivative is

$$\xi_t := T_{x_t}L_{x_t^{-1}}\dot{x}_t \in \mathfrak{g}, \qquad \dot{x}_t = T_e L_{x_t}\xi_t.$$

For matrix groups this is $\xi_t = x_t^{-1}\dot{x}_t$. The left Jacobian of the exponential map is

$$J_\ell(u)[v] := T_{\exp(u)}L_{\exp(u)^{-1}}\, d\exp_u[v] = \frac{1-\exp(-\operatorname{ad}_u)}{\operatorname{ad}_u}[v],$$

where the quotient is understood by its convergent power-series expansion on the chosen logarithm chart. Fix an open set $U \subset \mathfrak{g}$ on which $\exp$ is a diffeomorphism onto its image, restrict to endpoint pairs satisfying $x_0^{-1}x_1 \in \exp(U)$, and write $y = \log(x_0^{-1}x_1) \in U$. For a smooth map $F : G \to V$, where V is a vector space, derivatives in the group variable are left-trivialized by

$$D^\ell_x F(x)[\zeta] := \left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} F(x\exp(\varepsilon\zeta)), \qquad \zeta \in \mathfrak{g}.$$

Lemma B.1 (Product-exponential derivative)**.** *Let $x_t \in G$ have left-trivialized velocity ξ_t, and let $u_t \in \mathfrak{g}$ be C^1. If $z_t = x_t\exp(u_t)$, then*

$$T_{z_t}L_{z_t^{-1}}\dot{z}_t = \mathrm{Ad}_{\exp(-u_t)}\,\xi_t + J_\ell(u_t)[\dot{u}_t].$$

Proof. Differentiate $z_t = x_t \exp(u_t)$ and left-trivialize by $z_t^{-1} = \exp(-u_t)x_t^{-1}$. The contribution from $\dot{x}_t$ is

$$\mathrm{Ad}_{\exp(-u_t)}\, T_{x_t} L_{x_t^{-1}} \dot{x}_t = \mathrm{Ad}_{\exp(-u_t)}\, \xi_t.$$

The contribution from $\dot{u}_t$ is

$$T_{\exp(u_t)} L_{\exp(u_t)^{-1}}\, d\exp_{u_t}[\dot{u}_t] = J_\ell(u_t)[\dot{u}_t].$$

Summing the two terms proves the claim. □

B.1.2 Endpoint-constrained algebra paths

Let $A : [0,1] \times U \to \mathfrak{g}$, $(t, y) \mapsto A_t(y)$, be C^1 in t and satisfy $A_0(y) = 0$ and $A_1(y) = y$. Then $x_t = x_0 \exp(A_t(y))$ satisfies $x_{t=0} = x_0$ and $x_{t=1} = x_1$. Lemma B.1, applied with constant base curve x_0, gives the conditional bridge field

$$b_t(y) := T_{x_t} L_{x_t^{-1}} \dot{x}_t = J_\ell(A_t(y))[\partial_t A_t(y)].$$

Collinear bridges. The collinear bridge is the special case $A_t(y) = s(t)y$, with $s(0) = 0$ and $s(1) = 1$. The schedule used in the main construction is $s(t) = \gamma t + (1-\gamma)t^2$ for $\gamma \in \mathbb{R}$, so that $\dot{s}(t) = \gamma + 2(1-\gamma)t$. Since $A_t(y)$ and $\partial_t A_t(y)$ lie in the same one-dimensional subspace of $\mathfrak{g}$,

$$[A_t(y), \partial_t A_t(y)] = 0, \qquad J_\ell(A_t(y))[\partial_t A_t(y)] = \partial_t A_t(y) = \dot{s}(t)y.$$

Equivalently, the associated Magnus commutators vanish, since the velocity remains in the one-dimensional subalgebra $\mathbb{R}y$. In the collinear case the Jacobian correction is trivial. For non-collinear paths the factor $J_\ell(A_t(y))$ is part of the left logarithmic derivative.

Non-collinear families. The endpoint condition allows several finite-dimensional path families. If $s(0) = 0$, $s(1) = 1$, $w : U \to \mathfrak{g}$ is smooth, and $\psi(0) = \psi(1) = 0$, then $A_t(y) = \mathrm{Ad}_{\exp(\psi(t)w(y))}(s(t)y)$ has the prescribed endpoints. Likewise, for smooth $w_i : U \to \mathfrak{g}$ and scalar schedules ψ_i vanishing at 0 and 1, $A_t(y) = s(t)y + \sum_{i=1}^k \psi_i(t)[w_i(y), y]$ also satisfies $A_0(y) = 0$ and $A_1(y) = y$. Both families preserve the endpoint identities by construction.

B.1.3 Residual endpoint map and bridge updates

Define

$$\mathcal{R}_t(y) := \exp(-A_t(y)) \exp(y).$$

Along the exact bridge,

$$x_t^{-1} x_1 = \exp(-A_t(y)) \exp(y) = \mathcal{R}_t(y).$$

A bridge-label predictor $h_\theta : G \times [0,1] \to U$ therefore defines the endpoint map

$$f_\theta(x, t) := x \mathcal{R}_t(h_\theta(x,t)) = x \exp(-A_t(h_\theta(x,t))) \exp(h_\theta(x,t)).$$

If $x_t = x_0 \exp(A_t(y))$ and $h_\theta(x_t, t) = y$, then

$$f_\theta(x_t, t) = x_0 \exp(A_t(y)) \exp(-A_t(y)) \exp(y) = x_1.$$

The bridge representation also gives the deterministic update between two times. For $0 \le t < s \le 1$,

$$x_s = x_t \exp(-A_t(y)) \exp(A_s(y)).$$

Now, $x_t = x_0 \exp(A_t(y))$ gives $x_0 = x_t \exp(-A_t(y))$, and substitution into $x_s = x_0 \exp(A_s(y))$ gives the formula. Hence, if $\widehat{x}_1$ is a predicted endpoint and $\widehat{y} = \log(x_0^{-1}\widehat{x}_1)$, then $\widehat{x}_s = x_0 \exp(A_s(\widehat{y}))$.

Equivalently, when the current state lies on the predicted bridge, $\widehat{x}_t = x_0 \exp(A_t(\widehat{y}))$, then

$$\widehat{x}_s = \widehat{x}_t \exp(-A_t(\widehat{y})) \exp(A_s(\widehat{y})).$$

B.1.4 Continuous non-collinear consistency objective

Let $Y = \log(X_0^{-1} X_1)$. The conditional bridge field and its marginal field are

$$b_t(y) = J_\ell(A_t(y))[\partial_t A_t(y)], \qquad \bar{b}_t(x) = \mathbb{E}[b_t(Y) \mid X_t = x].$$

The marginal dynamics are $\dot{x}_t = T_e L_{x_t} \bar{b}_t(x_t)$. Along this flow, set

$$\widehat{y}_t = h_\theta(x_t, t), \quad \mathcal{R}_t^\theta = \mathcal{R}_t(\widehat{y}_t) = \exp(-A_t(\widehat{y}_t)) \exp(\widehat{y}_t), \quad \dot{\widehat{y}}_t = \partial_t h_\theta(x_t, t) + D_x^\ell h_\theta(x_t, t)[\bar{b}_t(x_t)].$$

The discrete non-collinear consistency loss is $L_\Delta^{\mathrm{nc}} = \mathbb{E}\left[d_m \left(f_\theta(X_t, t), \mathrm{sg}(f_\theta(X_{t+\Delta}, t+\Delta)) \right)^2 \right]$. Proposition B.2 shows that $\Delta^{-2} L_\Delta^{\mathrm{nc}} \to \mathbb{E}\|\omega_t\|_m^2$ and identifies the left logarithmic derivative ω_t. The sampled conditional bridge field gives the same first-order population update in the stop-gradient branch. This justifies the conditional continuous training expression used in the main text. The marginal and conditional squared objectives differ by the conditional variance of the bridge field.

Proposition B.2 (Continuous non-collinear consistency objective)**.** *Assume the preceding maps are C^1 and that the rescaled local metric expansion is uniformly integrable along the sampled paths. Define*

$$L_\Delta^{\mathrm{nc}} = \mathbb{E}\left[d_m \left(f_\theta(x_t, t), \mathrm{sg}(f_\theta(x_{t+\Delta}, t+\Delta)) \right)^2 \right].$$

Then

$$\lim_{\Delta \to 0} \Delta^{-2} L_\Delta^{\mathrm{nc}} = \mathbb{E}\|\omega_t\|_m^2,$$

where

$$\begin{aligned} \omega_t = {} & \mathrm{Ad}_{(\mathcal{R}_t^\theta)^{-1}} \bar{b}_t(x_t) \\ & + \mathrm{Ad}_{\exp(-\widehat{y}_t)} J_\ell(-A_t(\widehat{y}_t)) \left[-\partial_t A_t(\widehat{y}_t) - D_y A_t(\widehat{y}_t)[\dot{\widehat{y}}_t] \right] \\ & + J_\ell(\widehat{y}_t)[\dot{\widehat{y}}_t]. \end{aligned}$$

Proof. Set $z_t = f_\theta(x_t, t) = x_t \mathcal{R}_t^\theta$ and $\omega_t := T_{z_t} L_{z_t^{-1}} \dot{z}_t$. The stop-gradient operator affects differentiation in θ, not the value of the loss. Hence the asymptotics of L_Δ^{nc} are obtained from the expansion of $d_m(z_t, z_{t+\Delta})^2$. For a product $z_t = x_t r_t$, the differential of multiplication gives $T_{z_t} L_{z_t^{-1}} \dot{z}_t = \mathrm{Ad}_{r_t^{-1}} T_{x_t} L_{x_t^{-1}} \dot{x}_t + T_{r_t} L_{r_t^{-1}} \dot{r}_t$. Taking $r_t = \mathcal{R}_t^\theta$ and using $T_{x_t} L_{x_t^{-1}} \dot{x}_t = \bar{b}_t(x_t)$ gives

$$\omega_t = \mathrm{Ad}_{(\mathcal{R}_t^\theta)^{-1}} \bar{b}_t(x_t) + T_{\mathcal{R}_t^\theta} L_{(\mathcal{R}_t^\theta)^{-1}} \dot{\mathcal{R}}_t^\theta.$$

Write $B_t = -A_t(\widehat{y}_t)$ and $C_t = \widehat{y}_t$, so that $\mathcal{R}_t^\theta = \exp(B_t)\exp(C_t)$. The chain rule gives $\dot{B}_t = -\partial_t A_t(\widehat{y}_t) - D_y A_t(\widehat{y}_t)[\dot{\widehat{y}}_t]$ and $\dot{C}_t = \dot{\widehat{y}}_t$. Lemma B.1 yields

$$\begin{aligned} T_{\mathcal{R}_t^\theta} L_{(\mathcal{R}_t^\theta)^{-1}} \dot{\mathcal{R}}_t^\theta = {} & \mathrm{Ad}_{\exp(-\widehat{y}_t)} J_\ell(-A_t(\widehat{y}_t)) \left[-\partial_t A_t(\widehat{y}_t) - D_y A_t(\widehat{y}_t)[\dot{\widehat{y}}_t] \right] \\ & + J_\ell(\widehat{y}_t)[\dot{\widehat{y}}_t]. \end{aligned}$$

Substitution gives the expression for ω_t. Since d_m is induced by a smooth left-invariant Riemannian metric, the diagonal expansion of the squared distance gives $d_m(z_t, z_{t+\Delta})^2 = \Delta^2 \|\dot{z}_t\|_{z_t, m}^2 + o(\Delta^2)$. By left invariance, $\|\dot{z}_t\|_{z_t, m} = \|T_{z_t} L_{z_t^{-1}} \dot{z}_t\|_m = \|\omega_t\|_m$, hence $\Delta^{-2} d_m(z_t, z_{t+\Delta})^2 \to \|\omega_t\|_m^2$ pointwise along the sampled paths. The assumed uniform integrability of the rescaled local metric expansion gives convergence in L^1 by Vitali's theorem, and therefore

$$\lim_{\Delta \to 0} \Delta^{-2} L_\Delta^{\mathrm{nc}} = \lim_{\Delta \to 0} \mathbb{E}\left[\Delta^{-2} d_m(z_t, z_{t+\Delta})^2 \right] = \mathbb{E}\|\omega_t\|_m^2.$$

□

B.1.5 Conditional replacement in the detached branch

The marginal bridge field $\bar{b}_t(X_t)$ is the velocity appearing in the population marginal dynamics. In paired training, the endpoint X_1 is observed together with X_t, so the sampled bridge label $Y = \log(X_0^{-1} X_1)$ is available. The statement below shows that replacing $\bar{b}_t(X_t)$ by $b_t(Y)$ inside the

detached branch preserves the first-order population training vector field. Define the left-trivialized label differential of the residual map by

$$\mathcal{D}_t(y) := T_{\mathcal{R}_t(y)} L_{\mathcal{R}_t(y)^{-1}} D_y \mathcal{R}_t(y) : \mathfrak{g} \to \mathfrak{g}.$$

The notation $\mathcal{R}_t^{-1}$ is used only on open sets where $\mathcal{D}_t(y)$ is invertible; in particular, $t = 1$ is excluded because $\mathcal{R}_1(y) = e$ is constant.

Proposition B.3 (Conditional field in the detached branch)**.** *Fix $t < 1$ and suppose $\mathcal{D}_t(y)$ is invertible on the labels under consideration. Let L_Δ^{marg} be the unscaled stop-gradient consistency loss whose detached branch is advanced with velocity $\bar{b}_t(X_t)$, and let L_Δ^{cond} be the corresponding loss with velocity $b_t(Y)$. Assume f_θ has uniform local C^2 bounds on the compact chart under consideration and that the $O(\Delta^2)$ remainders below are uniformly integrable. Then*

$$\nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{cond}} = \nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{marg}} + O(\Delta^2), \qquad \Delta \downarrow 0.$$

Proof. Along an exact bridge, $x_t^{-1} x_1 = \mathcal{R}_t(y)$. By the inverse function theorem, the rank assumption gives the local identity $y = \mathcal{R}_t^{-1}(x_t^{-1} x_1)$. Thus the sampled conditional field can be expressed from the observed pair as

$$b_t(y) = J_\ell \left(A_t \left(\mathcal{R}_t^{-1}(x_t^{-1} x_1) \right) \right) \left[\partial_t A_t \left(\mathcal{R}_t^{-1}(x_t^{-1} x_1) \right) \right].$$

We work in a fixed local chart centered at $p_\theta := f_\theta(X_t, t)$, and let $q_\theta^v(\Delta) := f_\theta(X_{t+\Delta}^v, t+\Delta)$ denote the detached target obtained by advancing the second branch from X_t with velocity $v \in \mathfrak{g}$. The local C^2 bounds give $q_\theta^v(\Delta) = p_\theta + \Delta\, \mathcal{A}_\theta(X_t, t; v) + O(\Delta^2)$, where $\mathcal{A}_\theta(X_t, t; v)$ is affine in v. Expanding the squared Riemannian distance near the diagonal, with the second argument detached, gives

$$\nabla_\theta^{\mathrm{sg}} d_m(p_\theta, \mathrm{sg}\, q_\theta^v(\Delta))^2 = -2\Delta\, (D_\theta p_\theta)^* G_{p_\theta} \mathcal{A}_\theta(X_t, t; v) + O(\Delta^2),$$

where G_{p_θ} is the metric matrix in the chosen chart. Write $\mathcal{A}_\theta(X_t, t; v) = a_\theta(X_t, t) + \mathcal{B}_\theta(X_t, t) v$ and set $M_\theta(X_t, t) := (D_\theta p_\theta)^* G_{p_\theta}$. The first-order expansion gives

$$\begin{aligned}
\nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{cond}} &= -2\Delta\, \mathbb{E}\big[M_\theta(X_t, t) \big(a_\theta(X_t, t) + \mathcal{B}_\theta(X_t, t) b_t(Y) \big) \big] + O(\Delta^2), \\
\nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{marg}} &= -2\Delta\, \mathbb{E}\big[M_\theta(X_t, t) \big(a_\theta(X_t, t) + \mathcal{B}_\theta(X_t, t) \bar{b}_t(X_t) \big) \big] + O(\Delta^2).
\end{aligned}$$

Subtracting cancels the affine offset and gives

$$\begin{aligned}
&\nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{cond}} - \nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{marg}} \\
&\quad = -2\Delta\, \mathbb{E}\big[M_\theta(X_t, t) \mathcal{B}_\theta(X_t, t) \big(b_t(Y) - \bar{b}_t(X_t) \big) \big] + O(\Delta^2).
\end{aligned}$$

Since $M_\theta(X_t, t)$ and $\mathcal{B}_\theta(X_t, t)$ are $\sigma(X_t)$-measurable and $\bar{b}_t(X_t) = \mathbb{E}[b_t(Y) \mid X_t]$, the leading expectation vanishes:

$$\begin{aligned}
&\mathbb{E}\big[M_\theta \mathcal{B}_\theta \big(b_t(Y) - \bar{b}_t(X_t) \big) \big] \\
&\quad = \mathbb{E}\big[M_\theta \mathcal{B}_\theta\, \mathbb{E}\big[b_t(Y) - \bar{b}_t(X_t) \mid X_t \big] \big] = 0.
\end{aligned}$$

Therefore

$$\nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{cond}} = \nabla_\theta^{\mathrm{sg}} L_\Delta^{\mathrm{marg}} + O(\Delta^2).$$

□

This is a statement about the first-order stop-gradient update. In general, $\mathbb{E}\|\omega_\theta(X_t, \bar{b}_t(X_t))\|_m^2 \neq \mathbb{E}\|\omega_\theta(X_t, b_t(Y))\|_m^2$, because the latter contains the conditional variance of the bridge field. Thus the sampled conditional field gives the same first-order population update in the detached branch, while the continuous squared objectives remain distinct.

B.2 Momentum endpoint correction and stability

For momentum models, an affine interpolation $\xi_t^* = a(t)\xi_0 + b(t)\xi_1$ need not integrate to a group path ending at x_1. We correct the simulated reference path by its endpoint defect and then give an intrinsic path-space formulation of the same endpoint constraint. In the momentum setting the group path is obtained by integrating a time-dependent left-trivialized velocity, so interpolating endpoint momenta does not by itself enforce the terminal group element. We first correct the simulated reference path at the group level, then impose the same endpoint condition directly on the momentum path. Throughout this subsection G is a finite-dimensional Lie group with Lie algebra $\mathfrak{g}$. The metric is left-invariant, $\|\cdot\|_m$ denotes the induced norm on $\mathfrak{g}$, and d_m denotes the associated local Riemannian distance on G.

Lemma B.4 (Corrected momentum bridge)**.** *Let endpoint data $(x_0, \xi_0), (x_1, \xi_1) \in G \times \mathfrak{g}$ be given. Let $a, b \in C^1([0,1])$ satisfy $a(0) = 1$, $a(1) = 0$, $b(0) = 0$, and $b(1) = 1$, and set $\xi_t^* = a(t)\xi_0 + b(t)\xi_1$. Let x_t^* solve $\dot{x}_t^* = T_e L_{x_t^*} \xi_t^*$ with $x_0^* = x_0$. Assume that $E = x_1 (x_1^*)^{-1}$ lies in a logarithm chart, and put $u = \log(E)$. If $\eta \in C^2([0,1])$ satisfies $\eta(0) = 0$, $\eta(1) = 1$, and $\dot{\eta}(0) = \dot{\eta}(1) = 0$, then the path $q_t = \exp(\eta(t)u) x_t^*$ satisfies $q_0 = x_0$ and $q_1 = x_1$. Its left-trivialized velocity is*

$$\xi_t = \xi_t^* + \dot{\eta}(t) \operatorname{Ad}_{(x_t^*)^{-1}} u,$$

and its derivative is

$$\dot{\xi}_t = \dot{\xi}_t^* + \ddot{\eta}(t) \operatorname{Ad}_{(x_t^*)^{-1}} u - \dot{\eta}(t)[\xi_t^*, \operatorname{Ad}_{(x_t^*)^{-1}} u].$$

Moreover, $\xi_{t=0} = \xi_0$ and $\xi_{t=1} = \xi_1$.

Proof. The group endpoints follow from the definitions: $q_0 = \exp(0) x_0^* = x_0$ and $q_1 = \exp(u) x_1^* = E x_1^* = x_1$. Differentiating $q_t = z_t x_t^*$, with $z_t = \exp(\eta(t)u)$, and left-trivializing gives

$$T_{q_t} L_{q_t^{-1}} \dot{q}_t = \xi_t^* + \operatorname{Ad}_{(x_t^*)^{-1}} \left(T_{z_t} L_{z_t^{-1}} \dot{z}_t \right).$$

Since z_t remains in the one-parameter subgroup generated by u, one has $T_{z_t} L_{z_t^{-1}} \dot{z}_t = \dot{\eta}(t) u$, which proves the velocity formula. Writing $c_t = \operatorname{Ad}_{(x_t^*)^{-1}} u$, differentiation of the adjoint action along $\dot{x}_t^* = T_e L_{x_t^*} \xi_t^*$ gives $\dot{c}_t = -[\xi_t^*, c_t]$. Differentiating $\xi_t = \xi_t^* + \dot{\eta}(t) c_t$ gives the expression for $\dot{\xi}_t$. The endpoint momentum identities follow from $\dot{\eta}(0) = \dot{\eta}(1) = 0$ and the boundary conditions on a, b. □

Remark B.5 (Pathwise nature of the correction). The correction depends on the endpoint defect $u = \log(x_1 (x_1^*)^{-1})$, and hence on the endpoint data and on the reference path used to compute x_1^*. The lemma constructs endpoint-exact supervised bridges. The resulting velocity depends on the endpoint data through u, rather than only on the current state (x_t, ξ_t).

Zero-endpoint perturbations of the reference interpolation preserve the endpoint momenta. If $w_1, \dots, w_r \in \mathfrak{g}$ are fixed, or fixed after conditioning on the endpoint data, and if $\psi_j \in C^1([0,1])$ satisfy $\psi_j(0) = \psi_j(1) = 0$, then

$$\xi_t^* = a(t)\xi_0 + b(t)\xi_1 + \sum_{j=1}^{r} \psi_j(t) w_j$$

still has $\xi_0^* = \xi_0$ and $\xi_1^* = \xi_1$. Applying Lemma B.4 to this reference path gives the same group endpoint correction and preserves the endpoint momenta. Natural finite-dimensional directions include commutators $[\xi_0, \xi_1]$, $[\xi_0, [\xi_0, \xi_1]]$, $[\xi_1, [\xi_0, \xi_1]]$, and directions involving the endpoint label $y = \log(x_0^{-1} x_1)$, such as $[y, \xi_0]$ and $[y, \xi_1]$. Such perturbations preserve endpoint momenta and modify only the interior of the momentum path.

Proposition B.6 (Stability under small endpoint defect)**.** *Assume x_t^* remains in a compact set $K \subset G$ and $E = x_1 (x_1^*)^{-1}$ lies in a logarithm chart. Put $u = \log(E)$ and $c_t = \operatorname{Ad}_{(x_t^*)^{-1}} u$. If $\sup_t \|\xi_t^*\|_m \le M$, then there is a constant C_K, depending only on K, M, the metric, the bracket, and η, such that*

$$\|\xi - \xi^*\|_{C^0([0,1])} + \left\| \dot{\xi} - \dot{\xi}^* \right\|_{C^0([0,1])} \le C_K \|u\|_m .$$

If, in addition, (x_1, x_1^*) remains in a compact set on which $(g, h) \mapsto \log(gh^{-1})$ is defined, then, after possibly shrinking the chart,

$$\|\xi - \xi^*\|_{C^0} + \left\| \dot{\xi} - \dot{\xi}^* \right\|_{C^0} \le C_K d_m(x_1, x_1^*).$$

Proof. Lemma B.4 gives $\xi_t - \xi_t^* = \dot{\eta}(t) c_t$. Since $x_t^* \in K$ and the adjoint action is continuous, there exists $A_K > 0$ such that $\left\| \operatorname{Ad}_{(x_t^*)^{-1}} v \right\|_m \le A_K \|v\|_m$ for all t and $v \in \mathfrak{g}$. Hence $\|\xi_t - \xi_t^*\|_m \le \|\dot{\eta}\|_{L^\infty} A_K \|u\|_m$. For the derivative, $\dot{\xi}_t - \dot{\xi}_t^* = \ddot{\eta}(t) c_t - \dot{\eta}(t)[\xi_t^*, c_t]$. Finite dimensionality gives a bracket constant C_{br} with $\|[\alpha, \beta]\|_m \le C_{\mathrm{br}} \|\alpha\|_m \|\beta\|_m$. Together with $\sup_t \|\xi_t^*\|_m \le M$, this proves the first estimate. The second follows from the local Lipschitz continuity of $(g, h) \mapsto \log(gh^{-1})$ on the stated compact chart. The compact-chart accounts for the fact that a left-invariant metric need not be right-invariant. □

Proposition B.7 (Local Lipschitz dependence on endpoint data)**.** *Let $p = (x_0, \xi_0, x_1, \xi_1)$ and $\widetilde{p} = (\widetilde{x}_0, \widetilde{\xi}_0, \widetilde{x}_1, \widetilde{\xi}_1)$ be two endpoint data sets. Construct the corrected momentum bridges (q_t, ξ_t) and $(\widetilde{q}_t, \widetilde{\xi}_t)$ by Lemma B.4. Assume the corresponding reference paths remain in a common compact set, the endpoint defects remain in a common logarithm chart, and the endpoint momenta remain bounded. Then there is a constant C, depending on the common compact chart, the metric, the bracket, and the schedules a, b, η, such that*

$$\sup_{t\in[0,1]} d_m(q_t, \widetilde{q}_t) + \left\|\xi - \widetilde{\xi}\right\|_{C^0} + \left\|\dot{\xi} - \dot{\widetilde{\xi}}\right\|_{C^0} \\ \leq C\Big(d_m(x_0, \widetilde{x}_0) + d_m(x_1, \widetilde{x}_1) + \left\|\xi_0 - \widetilde{\xi}_0\right\|_m + \left\|\xi_1 - \widetilde{\xi}_1\right\|_m\Big).$$

Proof. The reference equation for x_t^* is a smooth finite-dimensional ODE with parameters (x_0, ξ_0, ξ_1). On compact sets, its solution depends locally Lipschitzly on these parameters, uniformly for $t \in [0, 1]$. The maps of multiplication, inversion, and logarithm are smooth on the common compact chart, hence $u = \log(x_1(x_1^*)^{-1})$ is locally Lipschitz in the endpoint data. Since $q_t = \exp(\eta(t)u)x_t^*$ and $\xi_t = \xi_t^* + \dot{\eta}(t)\,\mathrm{Ad}_{(x_t^*)^{-1}}\,u$, the stated estimates for q_t and ξ_t follow from smoothness of exp, multiplication, and $(x, u) \mapsto \mathrm{Ad}_{x^{-1}}\,u$ on compact sets. The formula for $\dot{\xi}_t$ in Lemma B.4 is built from the same smooth operations and the continuous Lie bracket, so the same local Lipschitz bound applies to $\dot{\xi} - \dot{\widetilde{\xi}}$. □

B.3 Intrinsic endpoint constraints in momentum path space

The group-level correction depends on the simulated reference endpoint x_1^* through the defect $x_1(x_1^*)^{-1}$. An intrinsic alternative is to impose the endpoint condition directly on the momentum path ξ through the endpoint map $\Phi_{x_0}(\xi) = \log(x_0^{-1}X_1^\xi)$. The appendix proves that the differential of this map is surjective after restricting to perturbations preserving endpoint momenta. This gives controlled endpoint-preserving corrections $h \in H^1([0, 1], \mathfrak{g})$ satisfying $h(0) = h(1) = 0$.

The preceding correction fixes the terminal group element after a reference momentum path has been simulated. The intrinsic formulation treats endpoint matching as a constraint on the momentum path. The unknown is a zero-endpoint perturbation h_t of the left-trivialized velocity. Fix $x_0 \in G$. For $\xi \in H^1([0, 1], \mathfrak{g})$, let X_t^ξ solve $\dot{X}_t^\xi = T_e L_{X_t^\xi}\xi_t$ with $X_0^\xi = x_0$. Define

$$\mathcal{E}_{x_0}(\xi) = X_1^\xi, \qquad \Phi_{x_0}(\xi) = \log\big(x_0^{-1}\mathcal{E}_{x_0}(\xi)\big),$$

where the logarithm is taken on the chosen chart. For a target endpoint x_1, put $y = \log(x_0^{-1}x_1)$. The endpoint constraint is $\Phi_{x_0}(\xi) = y$. The continuous embedding $H^1([0, 1], \mathfrak{g}) \hookrightarrow C^0([0, 1], \mathfrak{g})$ makes the endpoint constraints well-defined. To keep endpoint momenta fixed, work in the affine Hilbert space

$$\mathcal{A}_{\xi_0,\xi_1} = \{\xi \in H^1([0, 1], \mathfrak{g}) : \xi(0) = \xi_0,\ \xi(1) = \xi_1\},$$

whose tangent space is

$$\mathcal{H}_0 = \{h \in H^1([0, 1], \mathfrak{g}) : h(0) = 0,\ h(1) = 0\}.$$

Assumption B.8 (Endpoint chart and smooth dependence)**.** There is an H^1-neighbourhood $\mathcal{O}$ of the reference path ξ such that, for every $\zeta \in \mathcal{O}$, the solution X_t^ζ exists on $[0, 1]$, remains in a fixed compact subset of G, and satisfies $x_0^{-1}X_1^\zeta \in \exp(U)$. Moreover, the endpoint map $\zeta \mapsto X_1^\zeta$ is C^1 from $\mathcal{O} \subset H^1([0, 1], \mathfrak{g})$ into G.

Lemma B.9 (Endpoint differential)**.** *Let $\xi \in \mathcal{A}_{\xi_0,\xi_1}$, and suppose Assumption B.8 holds at ξ. For $h \in \mathcal{H}_0$, let $V_t^h \in \mathfrak{g}$ solve*

$$\dot{V}_t^h = h_t - \mathrm{ad}_{\xi_t} V_t^h, \qquad V_0^h = 0.$$

Then

$$D\Phi_{x_0}(\xi)[h] = J_\ell(\Phi_{x_0}(\xi))^{-1}V_1^h.$$

Moreover, the restricted derivative $L := D\Phi_{x_0}(\xi)|_{\mathcal{H}_0} : \mathcal{H}_0 \to \mathfrak{g}$ is surjective. Hence, with $\mathcal{V} = (\ker L)^\perp$ in $\mathcal{H}_0$, the restriction $L|_\mathcal{V} : \mathcal{V} \to \mathfrak{g}$ is a bounded linear isomorphism.

Proof. For the variation $\xi+\varepsilon h$, write the first variation of X_t^ξ as $\delta X_t = T_e L_{X_t^\xi} V_t^h$. Linearizing $\dot{X}_t^\xi = T_e L_{X_t^\xi}\xi_t$ gives $\dot{V}_t^h = h_t - \operatorname{ad}_{\xi_t} V_t^h$ with $V_0^h = 0$. Since $\delta(x_0^{-1}X_1^\xi) = T_e L_{x_0^{-1}X_1^\xi} V_1^h$ and $x_0^{-1}X_1^\xi = \exp(\Phi_{x_0}(\xi))$, the differential of the logarithm in left-trivialized coordinates is $J_\ell(\Phi_{x_0}(\xi))^{-1}$. To prove surjectivity, fix $r \in \mathfrak{g}$ and set $v_1 = J_\ell(\Phi_{x_0}(\xi))r$. Choose a smooth curve $V : [0,1] \to \mathfrak{g}$ with $V(0) = 0$, $V(1) = v_1$, $\dot{V}(0) = 0$, and $\dot{V}(1) = -\operatorname{ad}_{\xi_1} v_1$. Defining $h_t = \dot{V}_t + \operatorname{ad}_{\xi_t} V_t$ gives $h(0) = h(1) = 0$, so $h \in \mathcal{H}_0$, and the solution of the variational equation is precisely V. Thus $Lh = J_\ell(\Phi_{x_0}(\xi))^{-1}V_1 = r$. Since L is onto and $\mathfrak{g}$ is finite-dimensional, $\ker L$ has a closed orthogonal complement in $\mathcal{H}_0$, and the restriction of L to this complement is a bounded linear isomorphism. □

Theorem B.10 (Local intrinsic momentum correction)**.** *Let $\xi^* \in \mathcal{A}_{\xi_0,\xi_1}$, and suppose Assumption B.8 holds at ξ^*. Set $e = y - \Phi_{x_0}(\xi^*)$. Let $L = D\Phi_{x_0}(\xi^*)|_{\mathcal{H}_0}$ and $\mathcal{V} = (\ker L)^\perp$. For $\|e\|_m$ sufficiently small, there is a unique $h \in \mathcal{V}$, near zero, such that*

$$\Phi_{x_0}(\xi^* + h) = y.$$

The corrected path $\xi = \xi^* + h$ preserves endpoint momenta, $\xi(0) = \xi_0$ and $\xi(1) = \xi_1$, and satisfies

$$\|h\|_{H^1} \le C \|e\|_m$$

for a local constant C.

Proof. Define $F : \mathcal{V} \to \mathfrak{g}$ by $F(h) = \Phi_{x_0}(\xi^*+h) - y$. Then $F(0) = -e$, and $DF(0) = D\Phi_{x_0}(\xi^*)|_{\mathcal{V}}$ is an isomorphism by Lemma B.9. The Banach-space inverse function theorem gives, for $\|e\|_m$ sufficiently small, a unique $h \in \mathcal{V}$ near zero satisfying $F(h) = 0$. Since $\mathcal{V} \subset \mathcal{H}_0$, one has $h(0) = h(1) = 0$, so adding h does not alter endpoint momenta. After possibly shrinking the neighbourhood, the local inverse is Lipschitz, which gives $\|h\|_{H^1} \le C \|e\|_m$. □

Proposition B.11 (Minimum-norm linearized correction)**.** *Let $\xi^* \in \mathcal{A}_{\xi_0,\xi_1}$, and suppose Assumption B.8 holds at ξ^*. Let $L = D\Phi_{x_0}(\xi^*)|_{\mathcal{H}_0} : \mathcal{H}_0 \to \mathfrak{g}$ and $e = y - \Phi_{x_0}(\xi^*)$. Among all $h \in \mathcal{H}_0$ solving $Lh = e$, there is a unique minimizer of the H^1-norm. It is*

$$h_{\mathrm{lin}} = L^*(LL^*)^{-1}e,$$

where the adjoint is taken with respect to the H^1 inner product on $\mathcal{H}_0$ and the metric inner product on $\mathfrak{g}$. In particular, $\|h_{\mathrm{lin}}\|_{H^1} \le C \|e\|_m$.

Proof. By Lemma B.9, L is onto. Hence the affine set $\{h \in \mathcal{H}_0 : Lh = e\}$ is nonempty and closed. The squared H^1-norm is strictly convex and coercive on this affine set, so it has a unique minimizer. The Lagrange multiplier equations give $\lambda \in \mathfrak{g}$ such that $h_{\mathrm{lin}} = L^*\lambda$ and $Lh_{\mathrm{lin}} = e$. Hence $LL^*\lambda = e$. Since L is onto, $LL^* : \mathfrak{g} \to \mathfrak{g}$ is invertible, and therefore $h_{\mathrm{lin}} = L^*(LL^*)^{-1}e$. The norm estimate follows from boundedness of the finite-rank operator $L^*(LL^*)^{-1}$. □

Remark B.12 (Choice of correction gauge). The endpoint map has infinite-dimensional domain and finite-dimensional range, so the correction is non-unique until a gauge is fixed. The preceding results fix this freedom either by restricting to $\mathcal{V} = (\ker L)^\perp$ or by selecting the minimum-H^1-norm solution of the linearized constraint.

B.4 Momentum consistency objective

The corrected momentum bridges supply trajectories (x_t, ξ_t) and accelerations $v_t = \dot{\xi}_t$ for the momentum consistency model $f_\theta = (f_\theta^x, f_\theta^\xi) : G \times \mathfrak{g} \times [0,1] \to G \times \mathfrak{g}$. For the product metric on $G \times \mathfrak{g}$, the continuous limit of the discrete momentum consistency loss is $\mathbb{E}[\|\omega_t\|_m^2 + \left\|\dot{\delta}_t\right\|_m^2]$, where ω_t is the left-trivialized derivative of the group component and δ_t is the momentum component. We now evaluate the consistency objective along the endpoint-corrected momentum paths. The preceding sections produce admissible bridge dynamics $\dot{x}_t = T_e L_{x_t}\xi_t$ and $\dot{\xi}_t = v_t$ with prescribed endpoint data. The continuous objective is the squared norm of the derivative of $f_\theta(x_t, \xi_t, t)$, with the group component expressed in left-trivialized coordinates. Let

$$f_\theta = (f_\theta^x, f_\theta^\xi) : G \times \mathfrak{g} \times [0,1] \to G \times \mathfrak{g},$$

where f_θ^x is group-valued and f_θ^ξ is algebra-valued, and write $y_t = f_\theta^x(x_t, \xi_t, t)$, $\delta_t = f_\theta^\xi(x_t, \xi_t, t)$, and $\omega_t = T_{y_t} L_{y_t^{-1}} \dot{y}_t$. For the product metric on $G \times \mathfrak{g}$, assume the bridge $t \mapsto (x_t, \xi_t)$ and the map $t \mapsto f_\theta(x_t, \xi_t, t)$ are C^1, and that the rescaled local metric expansion is uniformly integrable. Then the discrete momentum consistency loss satisfies

$$\lim_{\Delta \to 0} \Delta^{-2} L_\Delta^{\mathrm{mom}} = \mathbb{E}\left[\|\omega_t\|_m^2 + \left\| \dot{\delta}_t \right\|_m^2 \right].$$

The group component follows from the diagonal expansion of the left-invariant metric, and the momentum component is Euclidean in $\mathfrak{g}$. We use the parameterization $f_\theta^x(x_t, \xi_t, t) = x_t \exp(u_t)$ with $u_t = r(t) h_\theta^x(x_t, \xi_t, t)$, and

$$f_\theta^\xi(x_t, \xi_t, t) = \xi_t + r(t) h_\theta^\xi(x_t, \xi_t, t), \qquad r(1) = 0.$$

If the bridge dynamics are $\dot{x}_t = T_e L_{x_t} \xi_t$ and $\dot{\xi}_t = v_t$, then the chain rule gives

$$\begin{aligned}
\dot{u}_t &= \dot{r}(t) h_\theta^x(x_t, \xi_t, t) \\
&\quad + r(t)\Big(\partial_t h_\theta^x + D_x^\ell h_\theta^x[\xi_t] + D_\xi h_\theta^x[v_t] \Big)(x_t, \xi_t, t), \\
\dot{\delta}_t &= v_t + \dot{r}(t) h_\theta^\xi(x_t, \xi_t, t) \\
&\quad + r(t)\Big(\partial_t h_\theta^\xi + D_x^\ell h_\theta^\xi[\xi_t] + D_\xi h_\theta^\xi[v_t] \Big)(x_t, \xi_t, t).
\end{aligned}$$

By Lemma B.1, applied to $f_\theta^x(x_t, \xi_t, t) = x_t \exp(u_t)$, the left-trivialized derivative of the group component is

$$\omega_t = \mathrm{Ad}_{\exp(-u_t)}\, \xi_t + J_\ell(u_t)[\dot{u}_t].$$

The resulting continuous momentum consistency objective is

$$L_{\mathrm{cont}}^{\mathrm{mom}} = \mathbb{E}\left[\left\| \mathrm{Ad}_{\exp(-u_t)}\, \xi_t + J_\ell(u_t)[\dot{u}_t] \right\|_m^2 + \left\| \dot{\delta}_t \right\|_m^2 \right].$$

For the corrected bridge of Lemma B.4, the velocity is

$$v_t = \dot{\xi}_t = \dot{\xi}_t^* + \ddot{\eta}(t)\, \mathrm{Ad}_{(x_t^*)^{-1}}\, u - \dot{\eta}(t)[\xi_t^*, \mathrm{Ad}_{(x_t^*)^{-1}}\, u].$$

Hence both the corrected affine bridge and its zero-endpoint perturbations provide admissible trajectories for the continuous momentum consistency objective. We describe the momentum consistency model in more detail in Appendix C.5.

B.5 Construction of Non-collinear Paths

In this section we discuss the preconditioned non-collinear path used in Section 5. We also introduce another potential non-collinear path that is guided by protein torsion energy.

B.5.1 Preconditioned Bridge Paths

In this section we discuss the preconditioned bridge paths in more detail. First we prove the following proposition.

Proposition B.13. *Suppose Y lies in a compact subset of the logarithm chart, and assume that for sufficiently small ρ, the corresponding $A_t^\rho(Y)$ and $R_t^\rho(Y)$ remain in compact sets on which the relevant Jacobians are well-defined. If $H = H^{\dagger_m} \succeq 0$ and $\mathrm{tr}_m(H) > 0$, then, on a sufficiently small chart, $\frac{d}{d\rho} \mathcal{H}_\epsilon(\rho)\Big|_{\rho=0} < 0$. Consequently, for all sufficiently small $\rho > 0$, $\mathcal{H}_\epsilon(\rho) < \mathcal{H}_\epsilon(0)$.*

Proof. At $\rho = 0$, $P_t^\rho = I$ and $A_t^\rho(y) = s(t)y$, which is the collinear bridge. For general ρ, define $R_t^\rho(y) = \exp(-A_t^\rho(y)) \exp(y)$ and $D_t^\rho(y) = T_{R_t^\rho(y)} L_{R_t^\rho(y)^{-1}} D_y R_t^\rho(y)$. Since $A_t^\rho(y) = y - (1 - s(t)) P_t^\rho y$ with $P_t^\rho = I + \rho a(t) H$, we have $y - A_t^\rho(y) = (1 - s(t))(I + \rho a(t) H) y$. In a sufficiently small logarithm chart, the product expansion gives $R_t^\rho(y) = \exp((1 - s(t))(I + \rho a(t) H) y + O(\|y\|_m^2))$. Hence

$$D_t^\rho(y) = (1 - s(t))(I + \rho a(t) H) + O(\|y\|_m).$$

We first compute the leading-order contribution:

$$(D_t^\rho(y))^{-1} = \frac{1}{1-s(t)}(I+\rho a(t)H)^{-1}.$$

Since $H = H^{\dagger_m}$, differentiating the squared Frobenius norm induced by $\langle\cdot,\cdot\rangle_m$ gives

$$\frac{d}{d\rho}\|(D_t^\rho(y))^{-1}\|_{F,m}^2\Big|_{\rho=0} = -\frac{2a(t)}{(1-s(t))^2}\,\mathrm{tr}_m(H).$$

Therefore,

$$\frac{d}{d\rho}\mathcal{H}_\epsilon(\rho)\Big|_{\rho=0} = -2\,\mathrm{tr}_m(H)\int_0^{1-\epsilon}\frac{a(t)}{(1-s(t))^2}dt + \mathrm{Err}.$$

The leading term is strictly negative because $a(t) > 0$ on $(0,1)$, $1-s(t) > 0$ on $[0, 1-\epsilon]$, and $\mathrm{tr}_m(H) > 0$. It remains to control the error term. By the compact-chart assumption, the product map $(u,v) \mapsto \exp(-u)\exp(v)$ and its left-trivialized differential are smooth and uniformly bounded on the chart containing $A_t^\rho(y)$ and y. After possibly shrinking the chart, the $O(\|y\|_m)$ contribution and its ρ-derivative are uniformly smaller than the absolute value of the negative leading term. Thus $\frac{d}{d\rho}\mathcal{H}_\epsilon(\rho)\Big|_{\rho=0} < 0$ for the exact differential $D_t^\rho(y)$. By continuity of $\mathcal{H}_\epsilon(\rho)$ in ρ, the strict inequality implies $\mathcal{H}_\epsilon(\rho) < \mathcal{H}_\epsilon(0)$ for all sufficiently small $\rho > 0$. □

Bridge Construction. Next we discuss how we construct the path in practice. Given training pairs (x_0^i, x_1^i), define the empirical bridge labels $y_i = \log((x_0^i)^{-1}x_1^i)$. Let $\Sigma = \frac{1}{n}\sum_{i=1}^n y_i y_i^\top$ be the empirical covariance matrix in metric coordinates, and let $\Sigma_\delta = \Sigma + \delta I$ for a small $\delta > 0$. We first consider the Euclidean surrogate problem

$$\min_{C=C^\top\succ 0}\mathrm{tr}(C^{-2}) + \beta\,\mathrm{tr}(C\Sigma_\delta C).$$

The first term is the leading-order residual conditioning cost, while the second term penalizes expanding the path energy under the label distribution. If $\Sigma_\delta = U\,\mathrm{diag}(\sigma_1,\dots,\sigma_d)U^\top$, the minimizer is $C_\beta = \beta^{-1/4}\Sigma_\delta^{-1/4}$. Thus directions of small label variance are expanded more, since they are harder to recover from the remaining endpoint displacement. For stability, we use this matrix only to define a positive preconditioning direction. Let $C_0 = \Sigma_\delta^{-1/4}$, and write $C_0 = U\,\mathrm{diag}(c_1,\dots,c_d)U^\top$. We set $H = C_0 - \lambda_{\min}(C_0)I$, so $H \succeq 0$. If C_0 is not a scalar multiple of the identity, then $H \neq 0$ and $\mathrm{tr}_m(H) > 0$. In experiments, the strength of the preconditioner is controlled by the scalar hyperparameter ρ, and we define $\widetilde{C} = I + \rho H$. Then we have $C_t = (1-a(t))I + a(t)\widetilde{C} = I + \rho a(t)H$. The P-TFM bridge used in practice is then

$$A_t^{\mathrm{P}}(y) = y - (1-s(t))C_t y.$$

Equivalently, $A_t^{\mathrm{P}}(y) = s(t)y - \rho(1-s(t))a(t)Hy$. It satisfies $A_0^{\mathrm{P}}(y) = 0$ and $A_1^{\mathrm{P}}(y) = y$, so $x_t = x_0\exp(A_t^{\mathrm{P}}(y))$ is endpoint-constrained. Its conditional bridge field is $b_t^{\mathrm{P}}(y) = J_\ell(A_t^{\mathrm{P}}(y))[\partial_t A_t^{\mathrm{P}}(y)]$, with $\partial_t A_t^{\mathrm{P}}(y) = \dot{s}(t)C_t y - (1-s(t))\rho\dot{a}(t)Hy$.

Proposition B.13 gives the local improvement of the residual map for small positive ρ. In practice, we choose ρ as a hyperparameter or by grid search so that $A_t^{\mathrm{P}}(y_i)$ remains inside the chosen logarithm chart.

B.5.2 Torsion-guided endpoint-exact bridges on $SE(3)^N$

For protein backbones, internal coordinates provide local geometric information while the state remains a product of residue frames. We define a protein-specific non-collinear bridge by deforming the endpoint interpolation in the direction of the left-trivialized gradient of an internal-coordinate energy. If $Y = \mathrm{Log}(X_0^{-1}X_1)$ and $X_t^0 = X_0\mathrm{Exp}(s(t)Y)$, define $\Gamma_t = \Gamma(X_t^0)$ and $X_t^\varepsilon = X_0\mathrm{Exp}(s(t)Y)\mathrm{Exp}(-\varepsilon q(t)\Gamma_t)$, where $q(0) = q(1) = 0$. For every ε for which the expression is defined, the endpoint constraint is exact; at each interior time, the first variation is the negative Riemannian gradient direction of the chosen internal-coordinate energy.

We now give the corresponding $SE(3)^N$ construction. Given $Y = \mathrm{Log}(X_0^{-1}X_1)$, the reference curve $X_t^0 = X_0\mathrm{Exp}(s(t)Y)$ is deformed in the left-trivialized gradient direction of $\mathcal{E}_{\mathrm{int}}$, with an endpoint-vanishing schedule. Internal-coordinate information enters through the bridge deformation; the state

space and endpoint constraint remain group-valued. Let $G_N := SE(3)^N, \qquad \mathfrak{g}_N := \mathfrak{se}(3)^N$. All exponentials, logarithms, adjoint actions, and left Jacobians are taken componentwise on G_N. We write $\mathrm{Exp} : \mathfrak{g}_N \to G_N$ for the product exponential and $\mathrm{Log} : G_N \supset \mathcal{V} \to \mathfrak{g}_N$ for a fixed local logarithm chart. Let m be the chosen left-invariant product metric on G_N, with induced inner product $\langle \cdot, \cdot \rangle_m$ on $\mathfrak{g}_N$.

Internal-coordinate energy. Let $\Pi : G_N \to \mathcal{C}$ be the smooth reconstruction map from residue frames to the atomistic configuration space on the configurations used below. The internal coordinates are smooth functions of $\Pi(X)$; to lighten notation we write $\tau(X) := \tau_{\mathrm{atom}}(\Pi(X))$, $\ell(X) := \ell_{\mathrm{atom}}(\Pi(X))$, and $\alpha(X) := \alpha_{\mathrm{atom}}(\Pi(X))$.

Assumption B.14 (Nondegenerate internal-coordinate chart)**.** There is an open set $\mathcal{U} \subset G_N$ of nondegenerate configurations on which all bond vectors used below are nonzero, all relevant bond angles avoid their singular values, and the selected torsion angles admit smooth local representatives. All paths here are assumed to remain in $\mathcal{U}$.

On $\mathcal{U}$, let $\tau : \mathcal{U} \to \mathbb{T}^M$ denote a collection of backbone and/or side-chain torsion angles. Let $\ell : \mathcal{U} \to \mathbb{R}^B$ and $\alpha : \mathcal{U} \to \mathbb{R}^A$ denote bond-length and bond-angle coordinates included in the energy. For smooth periodic potentials $V_j \in C^\infty(\mathbb{T})$, reference values $\ell^\star, \alpha^\star$, and weights $\lambda_\ell, \lambda_\alpha \geq 0$,

$$\mathcal{E}_{\mathrm{int}}(X) = \frac{\lambda_\ell}{2} \|\ell(X) - \ell^\star\|^2 + \frac{\lambda_\alpha}{2} \|\alpha(X) - \alpha^\star\|^2 + \sum_{j=1}^{M} V_j(\tau_j(X)).$$

The purely torsional case is obtained by taking $\lambda_\ell = \lambda_\alpha = 0$. Let $\nabla^m \mathcal{E}_{\mathrm{int}}(X) \in T_X G_N$ be the Riemannian gradient with respect to m, and define its left-trivialization by

$$\Gamma(X) := T_X L_{X^{-1}} \nabla^m \mathcal{E}_{\mathrm{int}}(X) \in \mathfrak{g}_N.$$

Equivalently, for every $Z \in \mathfrak{g}_N$, $d\mathcal{E}_{\mathrm{int}}(X)[T_e L_X Z] = \langle \Gamma(X), Z \rangle_m$.

Construction. Let $s, q \in C^2([0,1])$ satisfy $s(0) = 0$, $s(1) = 1$, $q(0) = q(1) = 0$, and $q(t) \geq 0$. Given $X_0, X_1 \in G_N$ with $X_0^{-1} X_1 \in \mathcal{V}$, set

$$Y := \mathrm{Log}(X_0^{-1} X_1), \qquad X_1 = X_0 \mathrm{Exp}(Y). \tag{1}$$

The reference bridge is

$$X_t^0 := X_0 \mathrm{Exp}(S_t), \qquad S_t := s(t) Y. \tag{2}$$

Assume $X_t^0 \in \mathcal{U}$ for all $t \in [0,1]$, and define

$$\Gamma_t(X_0, Y) := \Gamma(X_t^0) = \Gamma(X_0 \mathrm{Exp}(s(t) Y)). \tag{3}$$

For $\varepsilon \geq 0$, put

$$C_t^\varepsilon := -\varepsilon q(t) \Gamma_t(X_0, Y), \tag{4}$$

and define the torsion-guided bridge by

$$X_t^\varepsilon(X_0, Y) := X_0 \mathrm{Exp}(S_t) \mathrm{Exp}(C_t^\varepsilon). \tag{5}$$

Whenever $\mathrm{Exp}(S_t)\mathrm{Exp}(C_t^\varepsilon)$ lies in the chosen product logarithm chart, define the corresponding algebra path

$$A_t^\varepsilon(X_0, Y) := \mathrm{Log}(\mathrm{Exp}(S_t)\mathrm{Exp}(C_t^\varepsilon)), \qquad X_t^\varepsilon(X_0, Y) = X_0 \mathrm{Exp}(A_t^\varepsilon(X_0, Y)). \tag{6}$$

The formulas below are used on intervals on which (2) remains in $\mathcal{U}$ and (6) is defined.

Proposition B.15 (Endpoint exactness and trivialized velocity)**.** *Consider the bridge X_t^ε and algebra path A_t^ε defined by* (1)–(6)*. Then the following hold:*

(i) $X_0^\varepsilon = X_0$ *and* $X_1^\varepsilon = X_1$*. Equivalently,* $A_0^\varepsilon(X_0, Y) = 0$ *and* $A_1^\varepsilon(X_0, Y) = Y$*.*

(ii) *The left-trivialized velocity* $v_t^\varepsilon(X_0, Y) := T_{X_t^\varepsilon} L_{(X_t^\varepsilon)^{-1}} \dot{X}_t^\varepsilon$ *is*

$$v_t^\varepsilon(X_0, Y) = \mathrm{Ad}_{\mathrm{Exp}(-C_t^\varepsilon)}\big(J_\ell(S_t)[\dot{S}_t]\big) + J_\ell(C_t^\varepsilon)[\dot{C}_t^\varepsilon],$$

where

$$\dot{C}_t^\varepsilon = -\varepsilon \dot{q}(t) \Gamma_t(X_0, Y) - \varepsilon q(t) \dot{\Gamma}_t(X_0, Y), \qquad \dot{\Gamma}_t = D\Gamma(X_t^0)\big[T_e L_{X_t^0} J_\ell(S_t)[\dot{S}_t]\big].$$

(iii) *Since* $S_t = s(t)Y$*, one has* $J_\ell(S_t)[\dot{S}_t] = \dot{s}(t) Y$*. Equivalently,* $v_t^\varepsilon = J_\ell(A_t^\varepsilon)[\partial_t A_t^\varepsilon]$*.*

Proof. For *(i)*, use $S_t = s(t)Y$ and $C_t^\varepsilon = -\varepsilon q(t)\Gamma_t(X_0, Y)$. Since $s(0) = q(0) = 0$, one has $S_0 = 0$ and $C_0^\varepsilon = 0$, hence $X_0^\varepsilon = X_0\mathrm{Exp}(0)\mathrm{Exp}(0) = X_0$. Since $s(1) = 1$ and $q(1) = 0$, one has $S_1 = Y$ and $C_1^\varepsilon = 0$, hence $X_1^\varepsilon = X_0\mathrm{Exp}(Y)\mathrm{Exp}(0) = X_0\mathrm{Exp}(Y) = X_1$ by (1). Moreover $\mathrm{Exp}(A_t^\varepsilon) = \mathrm{Exp}(S_t)\mathrm{Exp}(C_t^\varepsilon)$ by (6). Therefore $\mathrm{Exp}(A_0^\varepsilon) = e$ and $\mathrm{Exp}(A_1^\varepsilon) = \mathrm{Exp}(Y)$. Since A_t^ε is defined in the fixed logarithm chart, this gives $A_0^\varepsilon = 0$ and $A_1^\varepsilon = Y$.

For *(ii)*, set $P_t := \mathrm{Exp}(S_t)$ and $Q_t := \mathrm{Exp}(C_t^\varepsilon)$. Since $X_t^\varepsilon = X_0P_tQ_t$ and X_0 is constant, left multiplication by X_0 does not affect the left-trivialized velocity; hence it suffices to compute the left-trivialized velocity of $Z_t := P_tQ_t$. Differentiating $Z_t = P_tQ_t$ gives $\dot{Z}_t = \dot{P}_tQ_t + P_t\dot{Q}_t$. Thus

$$\begin{aligned} T_{Z_t}L_{Z_t^{-1}}\dot{Z}_t &= T_{P_tQ_t}L_{Q_t^{-1}P_t^{-1}}(\dot{P}_tQ_t) + T_{P_tQ_t}L_{Q_t^{-1}P_t^{-1}}(P_t\dot{Q}_t) \\ &= \mathrm{Ad}_{Q_t^{-1}}\big(T_{P_t}L_{P_t^{-1}}\dot{P}_t\big) + T_{Q_t}L_{Q_t^{-1}}\dot{Q}_t. \end{aligned}$$

By the definition of the left Jacobian, $T_{P_t}L_{P_t^{-1}}\dot{P}_t = J_\ell(S_t)[\dot{S}_t]$ and $T_{Q_t}L_{Q_t^{-1}}\dot{Q}_t = J_\ell(C_t^\varepsilon)[\dot{C}_t^\varepsilon]$. Since $Q_t^{-1} = \mathrm{Exp}(-C_t^\varepsilon)$, this gives

$$v_t^\varepsilon = \mathrm{Ad}_{\mathrm{Exp}(-C_t^\varepsilon)}\big(J_\ell(S_t)[\dot{S}_t]\big) + J_\ell(C_t^\varepsilon)[\dot{C}_t^\varepsilon].$$

It remains to differentiate C_t^ε. From $C_t^\varepsilon = -\varepsilon q(t)\Gamma_t(X_0, Y)$ we get $\dot{C}_t^\varepsilon = -\varepsilon\dot{q}(t)\Gamma_t(X_0, Y) - \varepsilon q(t)\dot{\Gamma}_t(X_0, Y)$. Also, $X_t^0 = X_0\mathrm{Exp}(S_t)$ has left-trivialized velocity $J_\ell(S_t)[\dot{S}_t]$, so $\dot{X}_t^0 = T_eL_{X_t^0}J_\ell(S_t)[\dot{S}_t]$. Applying the chain rule to $\Gamma_t = \Gamma(X_t^0)$ gives $\dot{\Gamma}_t = D\Gamma(X_t^0)\big[T_eL_{X_t^0}J_\ell(S_t)[\dot{S}_t]\big]$.

For *(iii)*, since $S_t = s(t)Y$ and $\dot{S}_t = \dot{s}(t)Y$, the two vectors commute: $[S_t, \dot{S}_t] = 0$. Every higher commutator $\mathrm{ad}_{S_t}^k(\dot{S}_t)$ therefore vanishes for $k \geq 1$. The power-series definition $J_\ell(S_t) = \frac{1-\exp(-\mathrm{ad}_{S_t})}{\mathrm{ad}_{S_t}}$ then gives $J_\ell(S_t)[\dot{S}_t] = \dot{S}_t = \dot{s}(t)Y$. Since $X_t^\varepsilon = X_0\mathrm{Exp}(A_t^\varepsilon)$, the definition of J_ℓ applied to the curve A_t^ε gives $v_t^\varepsilon = T_{X_t^\varepsilon}L_{(X_t^\varepsilon)^{-1}}\dot{X}_t^\varepsilon = J_\ell(A_t^\varepsilon)[\partial_tA_t^\varepsilon]$. □

Theorem B.16 (First-order internal-energy descent)**.** *Fix $t \in (0, 1)$ and suppose $q(t) > 0$ and $\Gamma_t(X_0, Y) \neq 0$. Then the following hold:*

(i) *The first variation satisfies*

$$\frac{d}{d\varepsilon}\mathcal{E}_{\mathrm{int}}(X_t^\varepsilon(X_0, Y))\Big|_{\varepsilon=0} = -q(t)\|\Gamma_t(X_0, Y)\|_m^2 < 0.$$

(ii) *If, for some $H_t \geq 0$ and all $0 \leq r \leq \varepsilon$, the curve $\phi_t(r) = \mathcal{E}_{\mathrm{int}}(X_t^0\mathrm{Exp}(-rq(t)\Gamma_t(X_0, Y)))$ satisfies $|\phi_t''(r)| \leq H_tq(t)^2\|\Gamma_t(X_0, Y)\|_m^2$, then*

$$\mathcal{E}_{\mathrm{int}}(X_t^\varepsilon(X_0, Y)) \leq \mathcal{E}_{\mathrm{int}}(X_t^0) - \varepsilon q(t)\|\Gamma_t(X_0, Y)\|_m^2 + \frac{H_t}{2}\varepsilon^2q(t)^2\|\Gamma_t(X_0, Y)\|_m^2.$$

Consequently, if $H_t > 0$, then every $0 < \varepsilon < 2/(H_tq(t))$ for which the hypotheses hold gives $\mathcal{E}_{\mathrm{int}}(X_t^\varepsilon(X_0, Y)) < \mathcal{E}_{\mathrm{int}}(X_t^0)$. If $H_t = 0$, the same strict inequality holds for every positive ε in the interval where the hypotheses hold.

Proof. We first prove *(i)*. Fix t and write $\Gamma_t = \Gamma_t(X_0, Y)$. At this time, $X_t^\varepsilon = X_t^0\mathrm{Exp}(-\varepsilon q(t)\Gamma_t)$. Therefore

$$\frac{\partial}{\partial\varepsilon}X_t^\varepsilon\Big|_{\varepsilon=0} = T_eL_{X_t^0}\big[-q(t)\Gamma_t\big].$$

By definition of the left-trivialized Riemannian gradient, for every $Z \in \mathfrak{g}_N$, $d\mathcal{E}_{\mathrm{int}}(X_t^0)[T_eL_{X_t^0}Z] = \langle\Gamma_t, Z\rangle_m$. Taking $Z = -q(t)\Gamma_t$ gives

$$\begin{aligned} \frac{d}{d\varepsilon}\mathcal{E}_{\mathrm{int}}(X_t^\varepsilon)\Big|_{\varepsilon=0} &= d\mathcal{E}_{\mathrm{int}}(X_t^0)\big[T_eL_{X_t^0}(-q(t)\Gamma_t)\big] \\ &= \langle\Gamma_t, -q(t)\Gamma_t\rangle_m = -q(t)\|\Gamma_t\|_m^2. \end{aligned}$$

The strict inequality follows from $q(t) > 0$ and $\Gamma_t \neq 0$.

For *(ii)*, the curve $\phi_t(r) = \mathcal{E}_{\mathrm{int}}(X_t^0 \mathrm{Exp}(-rq(t)\Gamma_t))$ satisfies $\phi_t(0) = \mathcal{E}_{\mathrm{int}}(X_t^0)$ and $\phi_t(\varepsilon) = \mathcal{E}_{\mathrm{int}}(X_t^\varepsilon)$. Part *(i)* gives $\phi_t'(0) = -q(t)\|\Gamma_t\|_m^2$. Taylor's theorem gives, for some $r_\varepsilon \in [0, \varepsilon]$,

$$\phi_t(\varepsilon) = \phi_t(0) + \varepsilon\phi_t'(0) + \frac{\varepsilon^2}{2}\phi_t''(r_\varepsilon).$$

Using the assumed bound on ϕ_t'' gives

$$\begin{aligned}\mathcal{E}_{\mathrm{int}}(X_t^\varepsilon) &\le \mathcal{E}_{\mathrm{int}}(X_t^0) - \varepsilon q(t)\|\Gamma_t\|_m^2 + \frac{\varepsilon^2}{2}H_t q(t)^2\|\Gamma_t\|_m^2 \\ &= \mathcal{E}_{\mathrm{int}}(X_t^0) + \left(-\varepsilon q(t) + \frac{H_t}{2}\varepsilon^2 q(t)^2\right)\|\Gamma_t\|_m^2.\end{aligned}$$

If $H_t > 0$, the coefficient in parentheses is negative precisely for $0 < \varepsilon < 2/(H_t q(t))$. Since $\|\Gamma_t\|_m^2 > 0$, this proves strict descent in that range. If $H_t = 0$, the coefficient is $-\varepsilon q(t) < 0$ for every $\varepsilon > 0$, so strict descent again follows on the interval where the hypotheses hold. □

Linearized algebra path. A first-order implementation may avoid evaluating the full BCH map. The exact algebra path $A_t^\varepsilon(X_0, Y) = \mathrm{BCH}(s(t)Y, -\varepsilon q(t)\Gamma_t)$ admits the first-order approximation

$$A_{t,\mathrm{lin}}^\varepsilon(X_0, Y) := s(t)Y - \varepsilon q(t) J_\ell(s(t)Y)^{-1}\Gamma_t, \qquad \Gamma_t = \Gamma(X_0\mathrm{Exp}(s(t)Y)).$$

The endpoint identities are preserved because $q(0) = q(1) = 0$, and the infinitesimal group deformation agrees with that of the exact product bridge: $J_\ell(s(t)Y)[-q(t)J_\ell(s(t)Y)^{-1}\Gamma_t] = -q(t)\Gamma_t$. The corresponding flow-matching target is obtained from $X_{t,\mathrm{lin}}^\varepsilon = X_0\mathrm{Exp}(A_{t,\mathrm{lin}}^\varepsilon)$, namely

$$v_{t,\mathrm{lin}}^\varepsilon = J_\ell(A_{t,\mathrm{lin}}^\varepsilon)[\partial_t A_{t,\mathrm{lin}}^\varepsilon],$$

where

$$\begin{aligned}\partial_t A_{t,\mathrm{lin}}^\varepsilon = \dot{s}(t)Y &- \varepsilon\dot{q}(t)J_\ell(s(t)Y)^{-1}\Gamma_t \\ &- \varepsilon q(t)\partial_t\big(J_\ell(s(t)Y)^{-1}\big)\,\Gamma_t - \varepsilon q(t)J_\ell(s(t)Y)^{-1}\dot{\Gamma}_t,\end{aligned}$$

with $\dot{\Gamma}_t = D\Gamma(X_t^0)[T_e L_{X_t^0} J_\ell(s(t)Y)[\dot{s}(t)Y]]$.

Remark B.17 (Conditioned bridge rule). The deformation field used in the torsion-guided bridge is evaluated along the anchored reference curve, $X_t^0 = X_0\mathrm{Exp}(s(t)Y), \qquad \Gamma_t(X_0, Y) = \Gamma(X_t^0)$. Accordingly, the bridge rule is a map of (X_0, Y, t): $A_t^\varepsilon = A_t^\varepsilon(X_0, Y)$. The endpoint predictor and sampling update use the same conditioning. Accordingly, the label predictor is taken in the conditioned form $h_\theta(X, X_0, t)$, and the endpoint map in Lemma B.18 is written as $f_\theta(X, t; X_0)$. In the experiments, the initial/noise frame is supplied to the network together with the current state and time.

Lemma B.18 (Endpoint consistency of the conditioned map)**.** *Let $h_\theta : G_N \times G_N \times [0,1] \to \mathfrak{g}_N$ be a label predictor conditioned on X_0, and define*

$$f_\theta(X, t; X_0) = X\mathrm{Exp}\big(-A_t^\varepsilon(X_0, h_\theta(X, X_0, t))\big)\mathrm{Exp}\big(h_\theta(X, X_0, t)\big).$$

If $X = X_t^\varepsilon(X_0, Y)$ and $h_\theta(X_t^\varepsilon(X_0, Y), X_0, t) = Y$, then $f_\theta(X_t^\varepsilon(X_0, Y), t; X_0) = X_1$.

Proof. The bridge identity gives $X_t^\varepsilon(X_0, Y) = X_0\mathrm{Exp}(A_t^\varepsilon(X_0, Y))$. Substituting this identity and the condition $h_\theta(X_t^\varepsilon(X_0, Y), X_0, t) = Y$ into the definition of f_θ yields

$$\begin{aligned}f_\theta(X_t^\varepsilon(X_0, Y), t; X_0) &= X_0\mathrm{Exp}(A_t^\varepsilon(X_0, Y))\mathrm{Exp}(-A_t^\varepsilon(X_0, Y))\mathrm{Exp}(Y) \\ &= X_0\mathrm{Exp}(Y) = X_1,\end{aligned}$$

using $X_1 = X_0\mathrm{Exp}(Y)$. □

Conditioned flow-matching target. The torsion-guided bridge determines a supervised velocity label $v_t^\varepsilon(X_0, Y)$ along each sampled pair (X_0, X_1), with $Y = \mathrm{Log}(X_0^{-1}X_1)$. Since the bridge rule depends on the anchored curve $X_0\mathrm{Exp}(s(t)Y)$, the corresponding regression problem is conditioned on X_0. Thus the TFM loss is taken over vector fields $v_\theta : G_N \times G_N \times [0,1] \to \mathfrak{g}_N$ of the form $v_\theta(X, X_0, t)$. The population minimizer is the corresponding conditional mean.

Lemma B.19 (Population field for conditioned TFM)**.** *Let (X_0, X_1) be sampled from the endpoint coupling, set $Y = \mathrm{Log}(X_0^{-1}X_1)$, and let $X_t^\varepsilon = X_t^\varepsilon(X_0, Y)$ be the torsion-guided bridge. For the loss*

$$\mathcal{L}_{\mathrm{TFM}}^\varepsilon(\theta) = \mathbb{E}\left[\|v_\theta(X_t^\varepsilon, X_0, t) - v_t^\varepsilon(X_0, Y)\|_m^2\right],$$

the population minimizer is, for almost every (x, x_0, t),

$$v^\star(x, x_0, t) = \mathbb{E}[v_t^\varepsilon(X_0, Y) \mid X_t^\varepsilon = x,\ X_0 = x_0].$$

Equivalently, $v^\star(\cdot, x_0, t)$ is the left-trivialized velocity field of the bridge law conditioned on $X_0 = x_0$.

Proof. Let $\mathcal{Z} = (X_t^\varepsilon, X_0, t)$ and $V = v_t^\varepsilon(X_0, Y)$. The loss has the form $\mathbb{E}\|v_\theta(\mathcal{Z}) - V\|_m^2$ in the finite-dimensional Hilbert space $(\mathfrak{g}_N, \langle\cdot,\cdot\rangle_m)$. For any measurable vector field v, decompose

$$V - v(\mathcal{Z}) = \big(V - \mathbb{E}[V \mid \mathcal{Z}]\big) + \big(\mathbb{E}[V \mid \mathcal{Z}] - v(\mathcal{Z})\big).$$

The two terms are orthogonal after taking conditional expectation, since $\mathbb{E}[V - \mathbb{E}[V \mid \mathcal{Z}] \mid \mathcal{Z}] = 0$. Hence

$$\mathbb{E}\|V - v(\mathcal{Z})\|_m^2 = \mathbb{E}\|V - \mathbb{E}[V \mid \mathcal{Z}]\|_m^2 + \mathbb{E}\|\mathbb{E}[V \mid \mathcal{Z}] - v(\mathcal{Z})\|_m^2.$$

The first term is independent of v, and the second is minimized uniquely almost surely by $v(\mathcal{Z}) = \mathbb{E}[V \mid \mathcal{Z}]$. Substituting $\mathcal{Z} = (X_t^\varepsilon, X_0, t)$ and $V = v_t^\varepsilon(X_0, Y)$ gives the formula for $v^\star$.

Fix x_0 and write $X_t = X_t^\varepsilon(X_0, Y)$ and $V_t = v_t^\varepsilon(X_0, Y)$. Let $\mu_t^{x_0}$ be the law of X_t conditional on $X_0 = x_0$. For every smooth compactly supported test function φ in the chosen chart, the pathwise identity $\dot{X}_t = T_e L_{X_t} V_t$ gives

$$\frac{d}{dt}\mathbb{E}[\varphi(X_t) \mid X_0 = x_0] = \mathbb{E}[d\varphi(X_t)[T_e L_{X_t} V_t] \mid X_0 = x_0].$$

Since the map $Z \mapsto d\varphi(X_t)[T_e L_{X_t} Z]$ is linear in Z, conditioning on X_t yields

$$\begin{aligned}&\mathbb{E}[d\varphi(X_t)[T_e L_{X_t} V_t] \mid X_0 = x_0] \\ &\quad = \mathbb{E}[d\varphi(X_t)[T_e L_{X_t} \mathbb{E}[V_t \mid X_t, X_0 = x_0]] \mid X_0 = x_0].\end{aligned}$$

By the conditional-mean formula already proved, $\mathbb{E}[V_t \mid X_t = x, X_0 = x_0] = v^\star(x, x_0, t)$. Therefore

$$\frac{d}{dt}\int \varphi\, d\mu_t^{x_0} = \int d\varphi(x)[T_e L_x v^\star(x, x_0, t)]\, d\mu_t^{x_0}(x).$$

This is the weak continuity equation for the bridge law conditioned on $X_0 = x_0$, with left-trivialized velocity field $v^\star(\cdot, x_0, t)$. □

Implementation of the torsion-guided bridge. The following routines specify the supervised TFM target and the anchored sampler. Given a training pair (X_0, X_1), the bridge state X_t^ε is sampled from the endpoint-exact torsion-guided path, and the regression label is its left-trivialized velocity v_t^ε. The same algebra path A_t^ε is used in the conditioned endpoint map of Lemma B.18.

Algorithm 3 Torsion-guided bridge target on $SE(3)^N$

Require: Endpoints $X_0, X_1 \in G_N$; time $t \in (0,1)$; schedules s, q and derivatives $\dot{s}, \dot{q}$; strength $\varepsilon \geq 0$; product logarithm Log; gradient oracle Γ; JVP oracle $D\Gamma(X)[\cdot]$. Assume all logarithms, exponentials, and internal-coordinate evaluations below lie in their corresponding charts.

Ensure: Bridge state X_t^ε, endpoint label Y, algebra path A_t^ε, and left-trivialized velocity v_t^ε.

$Y \leftarrow \mathrm{Log}(X_0^{-1}X_1)$.
$S_t \leftarrow s(t)Y$ and $\dot{S}_t \leftarrow \dot{s}(t)Y$.
$X_t^0 \leftarrow X_0\mathrm{Exp}(S_t)$.
$\Gamma_t \leftarrow \Gamma(X_t^0)$.
$C_t \leftarrow -\varepsilon q(t)\Gamma_t$.
$\dot{\Gamma}_t \leftarrow D\Gamma(X_t^0)\Big[T_e L_{X_t^0}\big(J_\ell(S_t)[\dot{S}_t]\big)\Big]$.
$\dot{C}_t \leftarrow -\varepsilon\dot{q}(t)\Gamma_t - \varepsilon q(t)\dot{\Gamma}_t$.
$X_t^\varepsilon \leftarrow X_0\mathrm{Exp}(S_t)\mathrm{Exp}(C_t)$.
$A_t^\varepsilon \leftarrow \mathrm{Log}(\mathrm{Exp}(S_t)\mathrm{Exp}(C_t))$.
$v_t^\varepsilon \leftarrow \mathrm{Ad}_{\mathrm{Exp}(-C_t)}\Big(J_\ell(S_t)[\dot{S}_t]\Big) + J_\ell(C_t)[\dot{C}_t]$.
return $(X_t^\varepsilon, Y, A_t^\varepsilon, v_t^\varepsilon)$.

The Jacobian-vector product in line 6 is the derivative of the left-trivialized gradient field Γ in the tangent direction $T_eL_{X_t^0}(J_\ell(S_t)[\dot{S}_t])$. Since $S_t = s(t)Y$ and $\dot{S}_t = \dot{s}(t)Y$, this direction may be written as $T_eL_{X_t^0}(\dot{s}(t)Y)$. For a rotation-only implementation, the same routine is applied with $(G_N, \mathfrak{g}_N, X, Y, \Gamma)$ replaced by $(K_N, \mathfrak{k}_N, R, \Omega, \Gamma_R)$.

Conditioned endpoint map and sampling. For the consistency formulation, the label predictor is conditioned on the initial frame, $h_\theta : G_N \times G_N \times [0,1] \to \mathfrak{g}_N$, and the endpoint map is

$$f_\theta(X, t; X_0) = X\mathrm{Exp}(-A_t^\varepsilon(X_0, h_\theta(X, X_0, t)))\,\mathrm{Exp}(h_\theta(X, X_0, t))\,.$$

Sampling uses the same conditioned bridge rule. At each step the model predicts a label $\widehat{Y}_k$ from the current state, the fixed initial/noise frame, and the current time; the next state is evaluated on the bridge determined by that label.

Algorithm 4 Few-step sampling with a torsion-guided conditioned bridge

Require: Initial/noise group element $X_0 \in G_N$; trained label predictor h_θ; grid $0 = t_0 < t_1 < \cdots < t_K = 1$; schedules s, q; strength ε; gradient oracle Γ; logarithm/exponential routines.

Ensure: Final sample $\widehat{X}_{t_K}$.

1: $\widehat{X}_{t_0} \leftarrow X_0$.
2: **for** $k = 0, \ldots, K-1$ **do**
3: $\quad\widehat{Y}_k \leftarrow h_\theta(\widehat{X}_{t_k}, X_0, t_k)$.
4: $\quad\widehat{S}_{k+1} \leftarrow s(t_{k+1})\widehat{Y}_k$.
5: $\quad\widehat{X}^0_{k+1} \leftarrow X_0\mathrm{Exp}(\widehat{S}_{k+1})$.
6: $\quad\widehat{\Gamma}_{k+1} \leftarrow \Gamma(\widehat{X}^0_{k+1})$.
7: $\quad\widehat{C}_{k+1} \leftarrow -\varepsilon q(t_{k+1})\widehat{\Gamma}_{k+1}$.
8: $\quad\widehat{X}_{t_{k+1}} \leftarrow X_0\mathrm{Exp}(\widehat{S}_{k+1})\mathrm{Exp}(\widehat{C}_{k+1})$.
9: **end for**
10: **return** $\widehat{X}_{t_K}$.

Since $q(1) = 0$, the last update gives $\widehat{X}_{t_K} = X_0\mathrm{Exp}(\widehat{Y}_{K-1})$ when $t_K = 1$. Thus the final state is determined by the last predicted endpoint label. If the current iterate lies on the bridge defined by $\widehat{Y}_k$, the anchored update can equivalently be written in relative form as

$$\widehat{X}_{t_{k+1}} = \widehat{X}_{t_k}\mathrm{Exp}\Big(-A^\varepsilon_{t_k}(X_0, \widehat{Y}_k)\Big)\,\mathrm{Exp}\Big(A^\varepsilon_{t_{k+1}}(X_0, \widehat{Y}_k)\Big)\,.$$

The anchored form is the one used in Algorithm 4.

Remark B.20 (Scope of the construction). The torsion-guided bridge has several features relevant for experiments or implementation. It is endpoint exact because the deformation is multiplied by q with $q(0) = q(1) = 0$; its left-trivialized velocity is explicit, giving the flow-matching target in Algorithm 3; and, at each fixed interior time, Theorem B.16 identifies the deformation as a first-order descent direction for the chosen internal-coordinate energy. The statements are local on the chosen logarithm chart and on the nondegenerate internal-coordinate chart.

Rotation-only specialization. The construction specializes directly to the rotational component of a frame model. Let $K_N := SO(3)^N$, $\mathfrak{k}_N := \mathfrak{so}(3)^N$, and let $R_0, R_1 \in K_N$ satisfy $R_0^{-1}R_1 \in \mathcal{V}_R$ for a fixed product logarithm chart. Set $\Omega := \mathrm{Log}(R_0^{-1}R_1)$. Assume that the internal-coordinate energy is a smooth function on an open nondegenerate set $\mathcal{U}_R \subset K_N$, either directly or through a smooth reconstruction map $\Pi_R : K_N \to \mathcal{C}$, and define its left-trivialized gradient by $\Gamma_R(R) := T_RL_{R^{-1}}\nabla^m\mathcal{E}_{\mathrm{int}}(R) \in \mathfrak{k}_N$. For the same schedules s, q, put $S_t = s(t)\Omega$, $R_t^0 = R_0\mathrm{Exp}(S_t)$, $\Gamma_{R,t} = \Gamma_R(R_t^0)$, $C_t = -\varepsilon q(t)\Gamma_{R,t}$. Define the rotational torsion-guided bridge by

$$R_t^\varepsilon = R_0\mathrm{Exp}(S_t)\mathrm{Exp}(C_t).$$

It is endpoint exact, $R_0^\varepsilon = R_0$ and $R_1^\varepsilon = R_1$, and its left-trivialized velocity is

$$v^\varepsilon_{R,t} = \mathrm{Ad}_{\mathrm{Exp}(-C_t)}\big(J_\ell(S_t)[\dot{S}_t]\big) + J_\ell(C_t)[\dot{C}_t],$$

with

$$\dot{C}_t = -\varepsilon\dot{q}(t)\Gamma_{R,t} - \varepsilon q(t) D\Gamma_R(R_t^0)[T_e L_{R_t^0} J_\ell(S_t)[\dot{S}_t]].$$

At each fixed interior time for which $q(t) > 0$ and $\Gamma_{R,t} \neq 0$, the same first-variation computation gives

$$\frac{d}{d\varepsilon}\mathcal{E}_{\mathrm{int}}(R_t^\varepsilon)\Big|_{\varepsilon=0} = -q(t)\|\Gamma_{R,t}\|_m^2.$$

Thus the rotational version is obtained by replacing $(G_N, \mathfrak{g}_N, X, Y, \Gamma)$ with $(K_N, \mathfrak{k}_N, R, \Omega, \Gamma_R)$ throughout. The remaining modelling input is the choice of a smooth rotational reconstruction map or a smooth rotational internal-coordinate energy.

Corollary B.21 (Rotation-only torsion-guided TFM)**.** *Let $R_t^\varepsilon = R_t^\varepsilon(R_0, \Omega)$ be the rotational bridge defined above. For a conditioned model $v_\theta^R : K_N \times K_N \times [0,1] \to \mathfrak{k}_N$, the population minimizer of*

$$\mathcal{L}_{\mathrm{rot}}^\varepsilon(\theta) = \mathbb{E}\left[\|v_\theta^R(R_t^\varepsilon, R_0, t) - v_{R,t}^\varepsilon(R_0, \Omega)\|_m^2\right]$$

is

$$(v_\theta^R)^\star(R, R_0, t) = \mathbb{E}\big[v_{R,t}^\varepsilon(R_0, \Omega) \mid R_t^\varepsilon = R,\ R_0\big].$$

Proof. The proof is the conditioned least-squares projection in the Hilbert space $(\mathfrak{k}_N, \langle\cdot,\cdot\rangle_m)$, exactly as in Lemma B.19, with $(G_N, \mathfrak{g}_N, X, Y)$ replaced by $(K_N, \mathfrak{k}_N, R, \Omega)$. □

C Experimental Details

Next we discuss details regarding our experimental setup and datasets. All of our models were trained on NVIDIA RTX A6000 GPUs.

C.1 Model Architecture Detail For $SO(3)$ and $Sp(4,\mathbb{R})$ Experiments

The model used in these experiments is the same as the one used by RCM [5]. Specifically, we use a simple magnitude-preserving multilayer perceptron following the EDM2 design. The input group element is flattened into its ambient matrix representation and concatenated with a time embedding, while the output is an algebra-valued vector field or endpoint label depending on the model. Time is encoded using magnitude-preserving Fourier features, and the network is built from magnitude-preserving fully-connected layers, SiLU activations, concatenations, sums, and residual blocks. We use an encoder-decoder MLP structure with skip connections between corresponding layers, similar to a small UNet, so that low-level information from the noisy group element is preserved while deeper layers incorporate the time embedding.

Hyperparameters. Both experiments use the same hyperparameters. The MLP consists of four hidden blocks with widths 256, 512, 512, and 256, and the Fourier time embedding has dimension 256. We train all models with learning rate 1e-3, including both TGM models and all baselines. For all datasets, we use batch size of 5000 as we find that both TGM and the baselines generally benefit from larger batch sizes. Each model is trained for a total budget of 30 million sampled training examples. For P-TFM, we use $\sigma = 1e-4$ and $\rho = 0.25$.

As we find models to be unstable on $Sp(4,\mathbb{R})$, we stabilize TGM models by clipping the norm of Lie algebra elements before applying the $\exp$ map and after applying the $\log$ map, so that the norms are bounded and do not explode.

C.2 $Sp(4,\mathbb{R})$ Dataset Details

Here we discuss the exact datasets we use for $Sp(4,\mathbb{R})$. We use the phase-space ordering $z = (q_1, q_2, p_1, p_2)$ and the symplectic form $J = \begin{pmatrix} 0 & I_2 \\ -I_2 & 0 \end{pmatrix}$, so that $Sp(4,\mathbb{R}) = \{S \in GL(4,\mathbb{R}) : S^\top J S = J\}$. We parameterize $\mathfrak{sp}(4,\mathbb{R})$ by vectors $\xi \in \mathbb{R}^{10}$ through $\hat{\xi} = \begin{pmatrix} A & B \\ C & -A^\top \end{pmatrix}$, where $A \in \mathbb{R}^{2\times 2}$ is arbitrary and $B, C \in \mathbb{R}^{2\times 2}$ are symmetric.

Squeeze-braid dataset. Let $u \sim \mathrm{Unif}[0, 2\pi]$ and $c \sim \mathrm{Unif}\{0, \dots, 7\}$, $\alpha_1(u) = 0.9 + 0.25\cos(3u) + 0.15\sin(5u)$ and $\alpha_2(u) = 0.45 + 0.2\sin(4u) + 0.12\cos(7u)$. The branch index

c determines whether the two squeeze strengths are swapped and also determines their signs. Specifically, let $s(c) = c \bmod 2$, $\epsilon_1(c) = 1$ if $\lfloor c/2 \rfloor$ is even and -1 otherwise, and $\epsilon_2(c) = 1$ if $\lfloor c/4 \rfloor$ is even and -1 otherwise. We set $\lambda_1 = \epsilon_1(c)((1-s(c))\alpha_1 + s(c)\alpha_2)$ and $\lambda_2 = \epsilon_2(c)((1-s(c))\alpha_2 + s(c)\alpha_1)$, and define $d(u, c) = (\lambda_1, 0, 0, \lambda_2, 0, 0, 0, 0, 0, 0) \in \mathbb{R}^{10}$. To define the left and right braiding terms, let $\kappa(a, b_{11}, b_{12}, b_{22}) = (0, -a, a, 0, b_{11}, b_{12}, b_{22}, -b_{11}, -b_{12}, -b_{22})$. We then set $L(u) = \kappa(1.7u, 0.5\sin(2u), 0.4\cos(3u), 0.3\sin(5u))$ and $R(u) = \kappa(-2.3u, 0.4\cos(4u), 0.3\sin(3u), 0.5\cos(u))$. The final sample is $S = \exp(\widehat{L(u)}) \exp(\widehat{d(u, c)}) \exp(\widehat{R(u)}) \exp(\widehat{\sigma Z})$, where $Z \sim N(0, I_{10})$ and $\sigma = 0.01$.

Gaussian quantum dataset. This dataset considers two-mode continuous-variable quantum systems. The dataset is constructed as a simple two-mode Gaussian circuit containing passive transformations and single-mode squeezing. For phases ϕ_1, ϕ_2 and angle θ, let $U(\phi_1, \phi_2, \theta) = \operatorname{diag}(e^{i\phi_1}, e^{i\phi_2}) \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ and its real symplectic representation $K(\phi_1, \phi_2, \theta) = \begin{pmatrix} \operatorname{Re} U & -\operatorname{Im} U \\ \operatorname{Im} U & \operatorname{Re} U \end{pmatrix}$. For r_1, r_2, let $D(r_1, r_2) = \operatorname{diag}(e^{r_1}, e^{r_2}, e^{-r_1}, e^{-r_2})$. We then sample $u \sim \operatorname{Unif}[0, 2\pi]$ and $c \sim \operatorname{Unif}\{0, 1, 2\}$, and set $r_1 = 0.35 + 0.12\cos(2u + 2\pi c/3)$, $r_2 = -0.25 + 0.10\sin(3u + 2\pi c/3)$, $\theta_L = \pi/4 + 0.18\sin(2u) + 0.08c$, $\theta_R = 0.35\cos(3u) + \pi c/6$, $\phi_{L,1} = u$, $\phi_{L,2} = 2u + 0.4\sin u$, $\phi_{R,1} = -u + 0.3\cos(2u)$, and $\phi_{R,2} = 0.5u$. The final sample is $S = K(\phi_{L,1}, \phi_{L,2}, \theta_L) D(r_1, r_2) K(\phi_{R,1}, \phi_{R,2}, \theta_R) \exp(\widehat{\sigma Z})$, where $Z \sim N(0, I_{10})$ and $\sigma = 0.008$.

Paraxial optics dataset. In first-order paraxial optics, a ray with two transverse coordinates is represented by (x, y, p_x, p_y), and free-space propagation, thin lenses, and rotations of the transverse axes are all symplectic transformations. Thus the dataset consists of physically valid ray-transfer matrices in $Sp(4, \mathbb{R})$. For a rotation angle a, let $R(a) = \begin{pmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{pmatrix}$ and $Q(a) = \begin{pmatrix} R(a) & 0 \\ 0 & R(a) \end{pmatrix}$. For length ℓ, we have the drift matrix $T(\ell) = \begin{pmatrix} I_2 & \ell I_2 \\ 0 & I_2 \end{pmatrix}$. For a symmetric lens-power matrix F, define the thin-lens matrix $L(F) = \begin{pmatrix} I_2 & 0 \\ -F & I_2 \end{pmatrix}$. We parameterize astigmatic lenses by $F(f_x, f_y, a) = R(a)\operatorname{diag}(f_x, f_y) R(a)^\top$. We sample $u \sim \operatorname{Unif}[0, 2\pi]$ and $c \sim \operatorname{Unif}\{0, 1, 2, 3\}$, and set $\ell_1 = 0.8 + 0.15\sin u$, $\ell_2 = 0.6 + 0.12\cos(2u)$, $f_{1x} = 0.35 + 0.08\sin(3u + 2\pi c/4)$, $f_{1y} = -0.25 + 0.06\cos(2u)$, $f_{2x} = -0.30 + 0.08\cos(4u)$, $f_{2y} = 0.20 + 0.05\sin(u + 2\pi c/4)$, $a_0 = 0.25\sin u$, $a_1 = 0.3\sin(2u)$, $a_2 = 0.4\cos u + \pi c/8$, and $\beta = \pi c/8$. The final sample is $S = Q(\beta) L(F(f_{2x}, f_{2y}, a_2)) T(\ell_2) L(F(f_{1x}, f_{1y}, a_1)) T(\ell_1) Q(a_0) \exp(\widehat{\sigma Z})$, where $Z \sim N(0, I_{10})$ and $\sigma = 0.006$.

C.2.1 Comparison of Discrete and Continuous TCM

We compare the performance of continuous and discrete variants of TCM on the $Sp(4, \mathbb{R})$ datasets. The results are shown in Table 7, where TCT and TCD denote the trivialized continuous consistency training and consistency distillation models respectively, and TDT and TDD denote trivialized discrete training and consistency distillation models respectively. Comparing training with training and distillation with distillation, the continuous trivialized consistency models consistently outperform the discrete models, demonstrating that discretization results in error accumulation for the trivialized consistency models and degrades performance.

Table 7: MMD (10^{-1}) comparison of discrete and continuous TCM models on $Sp(4, \mathbb{R})$ datasets, where lower is better.

Dataset	TCT	TCD	TDT	TDD
Braid	**2.03**	2.14	2.16	2.57
Quantum	**1.29**	1.45	2.57	3.11
Optics	**1.58**	2.19	1.99	2.53

C.3 Experiments on Protein Backbone Generation Implementation Details

We follow FrameFlow's setup and train the $\mathbb{R}^3$ factor and $SO(3)$ factor separately. We handle the translation factor in the exact same way as FrameFlow, where the translations are kept as Center of Mass (COM) vectors and interpolated toward the target distribution. For the rotation factor we apply P-TFM with $\sigma = 1e-4$ and $\rho = 1500$. The large value of ρ is chosen as the signal from C is small. We find that in general larger ρ worked better for the model in this case. We use largely the same backbone as FrameFlow [39]. The output for the rotation is unchanged. For rotation, we add a linear layer that projects the rigid outputs to a $3D$ vector, which predicts the bridge $y = \log(R_0^{-1}R_1)$. For sampling, following FrameFlow, we use a linear time scheduler on the translation factor and an exponential time scheduler on the rotation factor. All other hyperparameters are the same as FrameFlow. We train our model on 4 GPUs for 12 hours to keep consistent with FrameFlow.

C.4 Discussion of Naive Approach

In this section we discuss why the naive approach of applying any Euclidean generative models on $\mathfrak{g}$ by treating $\log(\rho_0), \log(\rho_1)$ as Euclidean distributions is not ideal.

Proposition C.1. *Let $G = SO(2)$ and identify $SO(2)$ with angles modulo 2π. Let $\log$ be the principal logarithm with values in $(-\pi, \pi]$. For a C^1 path $\gamma_t \in SO(2)$, define $\mathcal{E}(\gamma) = \int_0^1 \|(L_{\gamma_t^{-1}})_* \dot{\gamma}_t\|^2 dt$. Then for every $\epsilon \in (0, \pi)$, there exist endpoints $x_0, x_1 \in SO(2)$ such that the naive log-linear path $\gamma_t^{\text{naive}} = \exp((1-t)\log(x_0) + t\log(x_1))$ has energy $4(\pi-\epsilon)^2$, while the relative-log path $\gamma_t^{\text{rel}} = x_0 \exp(t \log(x_0^{-1}x_1))$ has energy $4\epsilon^2$. Hence $\mathcal{E}(\gamma^{\text{naive}})/\mathcal{E}(\gamma^{\text{rel}}) = ((\pi-\epsilon)/\epsilon)^2$, which diverges as $\epsilon \to 0$.*

Proof. Let R_θ denote rotation by angle θ, and choose $x_0 = R_{-\pi+\epsilon}$ and $x_1 = R_{\pi-\epsilon}$. Then $\log(x_0) = -\pi + \epsilon$ and $\log(x_1) = \pi - \epsilon$. Thus the naive path moves with constant algebra velocity $2(\pi-\epsilon)$, and therefore $\mathcal{E}(\gamma^{\text{naive}}) = 4(\pi-\epsilon)^2$. On the other hand, $x_0^{-1}x_1 = R_{2\pi-2\epsilon} = R_{-2\epsilon}$, so $\log(x_0^{-1}x_1) = -2\epsilon$. The relative-log path therefore moves with constant algebra velocity -2ϵ, giving $\mathcal{E}(\gamma^{\text{rel}}) = 4\epsilon^2$. Taking the ratio proves the claim.

□

The above proposition shows that even in the simple case of an abelian compact Lie group $SO(2)$, the naive model can incur large energy in its generative paths. The issue is that absolute logarithmic coordinates depend on a branch choice. Two endpoints can be close on the group while their principal logarithms lie on opposite sides of the branch cut. In this case, a Euclidean model trained between $\log(x_0)$ and $\log(x_1)$ learns the long displacement $2\pi - 2\epsilon$, whereas the intrinsic relative displacement is -2ϵ. Thus the naive model incurs an arbitrarily larger squared velocity target and path energy. TGM avoids this pathology by using relative endpoint labels $y = \log(x_0^{-1}x_1)$ and endpoint-constrained bridges $x_t = x_0 \exp(A_t(y))$, so the learned velocity is expressed in the fixed Lie algebra while still respecting the group displacement.

C.5 Proof of Concept for MoTCM

We describe the momentum consistency model used in our proof-of-concept experiments. The construction follows the notation of Section 4.3. The state is now a pair $(x, \xi) \in G \times \mathfrak{g}$, where ξ is the left-trivialized velocity. The model predicts a terminal pair rather than only a terminal group element. The main difficulty compared with TCM is that the data provide a pose endpoint x_1, but do not provide a canonical terminal momentum label ξ_1. For this reason, our MoTCM uses a trained MoTFM teacher to define the terminal pair used for consistency distillation.

Endpoint map. Let

$$f_\theta = (f_\theta^x, f_\theta^\xi) : G \times \mathfrak{g} \times [0,1] \to G \times \mathfrak{g}.$$

We parameterize the endpoint map by two algebra-valued heads

$$h_\theta = (h_\theta^x, h_\theta^\xi) : G \times \mathfrak{g} \times [0,1] \to \mathfrak{g} \times \mathfrak{g}$$

and a scalar schedule r with $r(1) = 0$:

$$f_\theta^x(x, \xi, t) = x \exp(u_\theta(x, \xi, t)), \qquad u_\theta(x, \xi, t) = r(t) h_\theta^x(x, \xi, t),$$

$$f_\theta^\xi(x,\xi,t) = \xi + r(t) h_\theta^\xi(x,\xi,t).$$

Thus $f_\theta(x,\xi,1) = (x,\xi)$, matching the terminal boundary condition of TCM on the product space $G \times \mathfrak{g}$.

Corrected momentum bridge. Given endpoint pairs

$$(x_0,\xi_0), \qquad (x_1,\xi_1) \in G \times \mathfrak{g},$$

choose $a, b \in C^1([0,1])$ with $a(0) = 1$, $a(1) = 0$, $b(0) = 0$, and $b(1) = 1$, and set

$$\xi_t^* = a(t)\xi_0 + b(t)\xi_1.$$

Let x_t^* solve

$$\dot{x}_t^* = (L_{x_t^*})_* \xi_t^*, \qquad x_0^* = x_0.$$

Since x_1^* need not equal x_1, assume $E = x_1 (x_1^*)^{-1}$ lies in the logarithm chart and set

$$u = \log(E).$$

For $\eta \in C^2([0,1])$ satisfying

$$\eta(0) = 0, \qquad \eta(1) = 1, \qquad \dot{\eta}(0) = \dot{\eta}(1) = 0,$$

define

$$z_t = \exp(\eta(t)u), \qquad q_t = z_t x_t^*.$$

Proposition C.2. *The corrected path $q_t = z_t x_t^*$ satisfies $q_0 = x_0$ and $q_1 = x_1$. Its left-trivialized velocity is*

$$\xi_t = \xi_t^* + \dot{\eta}(t)\,\mathrm{Ad}_{(x_t^*)^{-1}}\, u.$$

In particular, $\xi(0) = \xi_0$ and $\xi(1) = \xi_1$.

Proof. The endpoint identities follow from

$$q_0 = \exp(0) x_0^* = x_0, \qquad q_1 = \exp(u) x_1^* = E x_1^* = x_1.$$

Differentiating $q_t = z_t x_t^*$ and left-trivializing by q_t^{-1} gives

$$q_t^{-1}\dot{q}_t = (x_t^*)^{-1} z_t^{-1} \dot{z}_t x_t^* + (x_t^*)^{-1}\dot{x}_t^*.$$

Since $z_t = \exp(\eta(t)u)$ remains in the one-dimensional subalgebra generated by u,

$$z_t^{-1}\dot{z}_t = \dot{\eta}(t) u.$$

Also $(x_t^*)^{-1}\dot{x}_t^* = \xi_t^*$. Hence

$$q_t^{-1}\dot{q}_t = \xi_t^* + \dot{\eta}(t)\,\mathrm{Ad}_{(x_t^*)^{-1}}\, u.$$

The velocity endpoints follow from $\dot{\eta}(0) = \dot{\eta}(1) = 0$ and the endpoint conditions on a and b. □

In the implementation the same formulas are evaluated on a grid ending at $1 - \delta$. The bridge is used only at nonterminal sampling times; on the last step the sampler returns the predicted terminal pose.

Teacher terminal map. Let the trained MoTFM teacher define the left-trivialized second-order dynamics

$$\dot{x}_t = (L_{x_t})_* \xi_t, \qquad \dot{\xi}_t = F_T(x_t,\xi_t,t) - \lambda(t)\xi_t,$$

where F_T is the teacher network output and $\lambda(t)$ is the damping used in training. Let

$$\Phi_T^{1-\varepsilon \leftarrow t}(x_t,\xi_t)$$

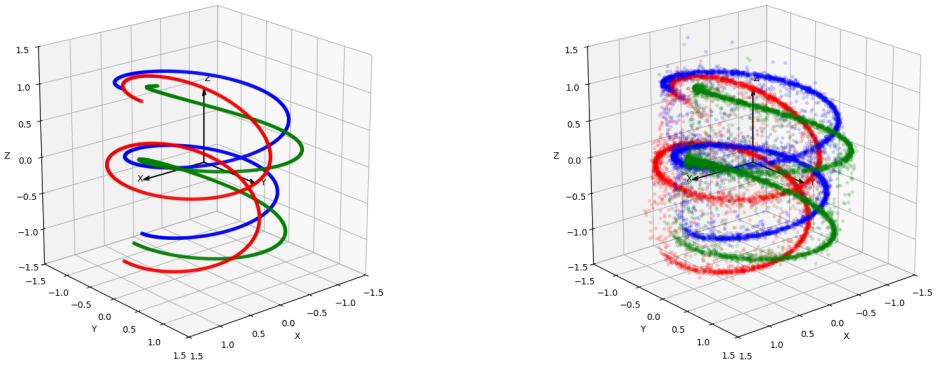
denote teacher integration from time t to $1-\varepsilon$, and define the teacher terminal map

$$Y_T(x_t,\xi_t,t) = \Phi_T^{1-\varepsilon\leftarrow t}(x_t,\xi_t) = (x_1^T, \xi_1^T).$$

This map supplies the missing terminal momentum label.

Proposition C.3. *The exact teacher terminal map is constant along teacher characteristics. If $(x_s,\xi_s) = \Phi_T^{s\leftarrow t}(x_t,\xi_t)$, then*

$$Y_T(x_s,\xi_s,s) = Y_T(x_t,\xi_t,t).$$



(a) Ground-truth SE(3) spiral dataset.

(b) Samples generated by MoTCM.

Figure 4: Visualization of the SE(3) spiral dataset and generated samples from the momentum trivialized consistency model. The dataset consists of structured spiral trajectories in the group, while MoTCM learns to generate samples that recover the main spiral modes and their spatial organization.

Proof. This is the semigroup property of the teacher flow:

$$\Phi_T^{1-\varepsilon\leftarrow s} \circ \Phi_T^{s\leftarrow t} = \Phi_T^{1-\varepsilon\leftarrow t}.$$

Applying both sides to (x_t, ξ_t) gives the result. □

Therefore the MoTCM student is trained to approximate the teacher terminal map

$$f_\theta(x_t, \xi_t, t) \approx Y_T(x_t, \xi_t, t),$$

where the right-hand side is time-consistent along teacher trajectories. This is the distillation form of the MoTCM objective used in the proof-of-concept experiments.

Fixed teacher cache. To avoid computing a teacher rollout for every optimization step, we precompute a fixed cache. Sample prior pairs

$$(x_0^i, \xi_0^i) \sim \mu_0, \qquad i = 1, \dots, M,$$

and roll the teacher to the final grid point $t_K = 1 - \delta$. The cache stores

$$\{(x_{t_k}^{T,i}, \xi_{t_k}^{T,i}, t_k)\}_{k=0}^K, \qquad (x_1^{T,i}, \xi_1^{T,i}) = (x_{t_K}^{T,i}, \xi_{t_K}^{T,i}).$$

Optionally, the terminal momentum label can be rescaled or perturbed:

$$\widetilde{\xi}_1^{T,i} = s_\xi \xi_1^{T,i} + \sigma_\xi \epsilon, \qquad \epsilon \sim N(0, I).$$

In our baseline, $s_\xi = 1$ and $\sigma_\xi = 0$.

Each training batch samples a trajectory index i and a nonterminal time index $k < K$. The fixed-cache loss is

$$L_{\text{fixed}} = w_x \left\| f_\theta^x(x_{t_k}^{T,i}, \xi_{t_k}^{T,i}, t_k) - x_1^{T,i} \right\|_F^2 + w_\xi \left\| f_\theta^\xi(x_{t_k}^{T,i}, \xi_{t_k}^{T,i}, t_k) - \widetilde{\xi}_1^{T,i} \right\|_2^2.$$

In the current implementation, the pose term uses the Frobenius norm of the matrix representation of G.

Bridge cache. The fixed teacher cache labels states on teacher trajectories from the prior. Few-step generation, however, repeatedly predicts a terminal pair and moves to an intermediate state of the corrected bridge. We therefore add an on-policy bridge cache. Periodically, using the current student, we sample prior pairs, run the same N-step sampler used for generation up to intermediate times, and collect bridge states

$$(\bar{x}_{t_j}, \bar{\xi}_{t_j}, t_j), \qquad j = 1, \dots, N-1.$$

The teacher is then rolled from each collected state to $1-\varepsilon$ to obtain labels $(\bar{x}_1^T, \bar{\xi}_1^T)$. The bridge-cache loss is

$$L_{\text{bridge}} = w_x \left\| f_\theta^x(\bar{x}_{t_j}, \bar{\xi}_{t_j}, t_j) - \bar{x}_1^T \right\|_F^2 + w_\xi \left\| f_\theta^\xi(\bar{x}_{t_j}, \bar{\xi}_{t_j}, t_j) - \overset{\sim}{\xi}_1^T \right\|_2^2,$$

where $\overset{\sim}{\xi}_1^T$ uses the same optional momentum rescaling and noise as the fixed teacher cache.

The final training objective for the bridge-cached proof-of-concept model is

$$L_{\text{MoTCM}} = (1 - w_b) L_{\text{fixed}} + w_b L_{\text{bridge}}.$$

We keep the optional local continuous residual disabled in this proof-of-concept. The reason is that the local equation

$$\frac{\mathrm{d}}{\mathrm{d}t} f_\theta(x_t, \xi_t, t) = 0$$

does not by itself choose the terminal pair. In MoTCM the missing terminal momentum makes this anchor ambiguous, while the teacher terminal map gives a concrete dynamically consistent target.

Sampling. Let

$$0 = t_0 < t_1 < \cdots < t_N = 1 - \varepsilon$$

be the sampling grid. Starting from $(x_{t_0}, \xi_{t_0}) = (x_0, \xi_0) \sim \mu_0$, repeat:

$$(\widehat{x}_1, \widehat{\xi}_1) = f_\theta(x_{t_k}, \xi_{t_k}, t_k).$$

If $k = N - 1$, return $\widehat{x}_1$. Otherwise, rebuild the corrected momentum bridge from a prior pair to $(\widehat{x}_1, \widehat{\xi}_1)$ and evaluate it at t_{k+1} to obtain $(x_{t_{k+1}}, \xi_{t_{k+1}})$. In our baseline, we use $N = 3$ steps. We also use a prior refresh mixture between bridge updates; with refresh parameter ρ, the momentum prior is updated as

$$\xi_0^{\text{new}} = (1 - \rho)\xi_0^{\text{base}} + \rho \xi_0^{\text{fresh}},$$

with the analogous interpolation in algebra coordinates for the group prior.

Proof-of-concept setting. Our baseline uses $M = 5000$ fixed teacher trajectories, a bridge cache built from 5000 prior samples, $w_b = 0.5$, 200 teacher integration steps, $w_x = 1$, $w_\xi = 0.3$, $N = 3$ sampling steps, and refresh mixture $\rho = 0.5$. With $N = 3$, the bridge cache stores two intermediate-time rows per sampled prior. The proof of concept plot showing that MoTCM can learn distribution on $SE(3)$ is shown at Figure 4.

D Scope and Limitations

Chart Assumption. We assume that the relative displacement $x_0^{-1} x_1$, and for the momentum models the endpoint correction $x_1 (x_1^*)^{-1}$, lies in one logarithm chart. This does not require the endpoints or the marginal supports to lie in one chart. For instance, consider ρ_0 on $SO(3)$ with support on the entire group, and construct $x_1 = x_0 \exp(\delta)$ for sampled small noise δ. Neither marginal support lies in a single bijective exponential chart, while every relative displacement equals $\exp(\delta)$ and lies in a small chart around the identity. Furthermore, the exponential map is surjective on compact connected Lie groups, which is the typical testing scenario in prior works such as benchmarking on $SO(3)$. On $Sp(4, \mathbb{R})$, the exponential map is neither surjective nor injective, and TGM outperforms all baselines in our experiments, demonstrating that TGM remains practically effective in this setting.

Disconnected Lie Groups. We follow the convention of prior works [42] and assume that G has a single connected component. An ODE trajectory is continuous and thus cannot leave the component in which it originates. Moreover, the exponential map sends $\mathfrak{g}$ into the identity component G_e, since $\exp$ is continuous and $\exp(0) = e$. Consequently, a bridge of the form $x_t = x_0 \exp(A_t)$ remains in

the component of x_0 and reaches x_1 only if $x_0^{-1}x_1 \in G_e$. The continuous dynamics alone therefore cannot produce transitions between components. TGM could be extended to disconnected groups with an additional discrete model. Fix coset representatives g_c for $c \in G/G_e$, write $x = g_c h$ with $h \in G_e$, and sample the component label c before applying TGM to h within the identity component. Under this construction, the continuous model governs transport within a component and the discrete model governs the selection of the component. For example, the Lorentz group $O(1,3)$ has four components, indexed by $\det(\Lambda) = \pm 1$ and $\mathrm{sgn}(\Lambda^0{}_0)$, with G_e equal to the proper orthochronous component.

Limitations and Societal Impact. Our method relies on a chosen logarithm chart and could face numerical instability, although we prove stability bounds for MoTFM. Stability measures such as norm clipping were employed during parts of the experiments. This work is primarily methodological and aims to improve generative modeling on geometrically constrained spaces. Potential positive impacts include better modeling tools for robotics, physical sciences, optics, quantum systems, and protein structure generation. At the same time, improved generative models for scientific objects could be misused if applied to sensitive domains without appropriate validation or expert oversight.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[N/A]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We carefully and clearly state our contributions in the abstract and introduction, which are then substantiated by our subsequent theoretical and empirical results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Appendix D.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide full statements and comprehensive proofs in our main text and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all details for the datasets we construct in the Appendix, along with references that point to other papers we used. We further provide all model details and hyperparameters for experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: For the datasets we construct, we provide detailed instructions in the Appendix on how to construct them. We also reference all data we used that are not created by us and give detailed algorithmic instructions. Finally, our code is available here.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details, including model architecture, hyperparameter and dataset details are shown in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: To reflect the significance of our results and minimize the randomness of generative models, we draw a large number of samples for each generative model and compute the evaluation metric. For the $SO(3)$ and $Sp(4, \mathbb{R})$ experiments, we report the mean and standard deviation of MMD over 5 runs and of generation time over 10 iterations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss our compute setup in Appendix C, including which GPU and how many were used. We report runtimes for the $SO(3)$ and $Sp(4, \mathbb{R})$ experiments in the main text.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not crowdsource or use patient data as we propose a general machine learning method. We comply completely with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss societal impacts in Appendix D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: We do not release any data or model that can be used for malicious purposes.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite all baselines, datasets, and other works we base on in both our main text and our appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: We propose a general machine learning method, thus we do not release any new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: We do not conduct any crowdsourcing experiments or experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: We did not work with any human subjects or conduct experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM is only used for formatting and editing the paper for grammar or word choices and formatting plots for better presentation. We comply with all LLM policies of the conference.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.