PoseD-Flow: Versatile and Guided Flow Matching Model of Human Pose

Jebastin Nadar Simone Foti Tolga Birdal
Imperial College London
circle-group.github.io/research/PoseD-Flow

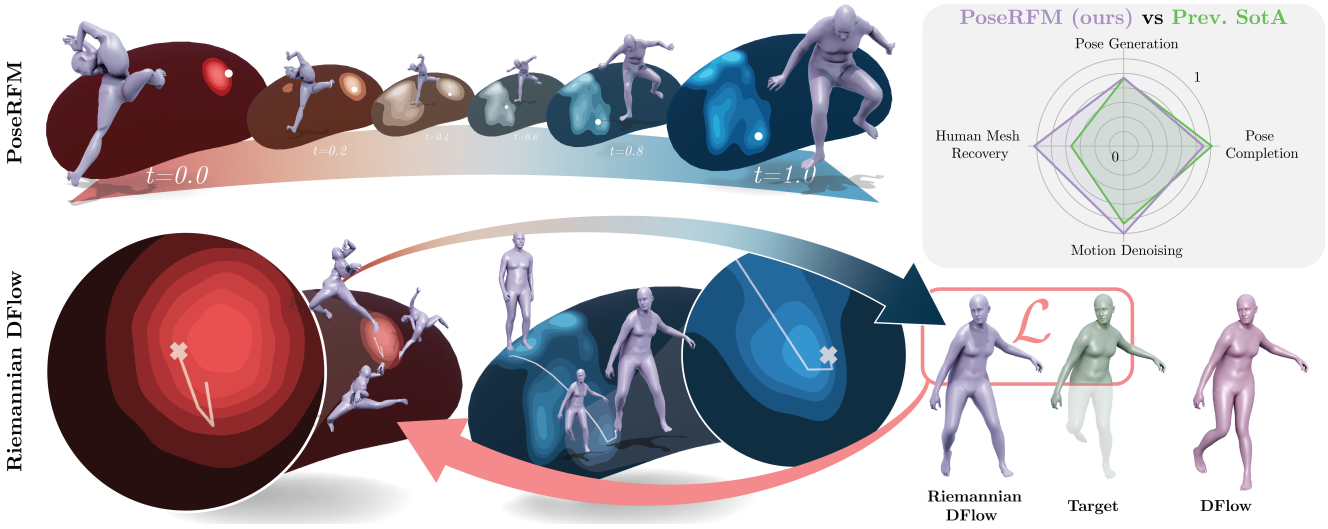


Figure 1. **PoseD-Flow** framework: (**top**) **PoseRFM**, a robust human pose prior defined on the product manifold of joints using Riemannian Flow Matching; (**bottom**): **Riemannian D-Flow**, a flexible, geometry-aware inversion technique for flow models. Together, they provide a novel approach to solving inverse problems in human pose, achieving results competitive with SotA diffusion models.

Abstract

Generative pose priors have recently emerged as a powerful tool for inference under occlusion or noise. Yet today's strongest generative paradigm, flow matching*, remains unused for human pose due to two fundamental barriers: the absence of a pre-trained flow prior and the non-Euclidean nature of articulated poses. We overcome both by introducing PoseD-Flow, a novel framework to unify Riemannian Flow Matching (RFM) with training-free guidance for 3D human pose recovery. PoseD-Flow is composed of two contributions: (i) PoseRFM, the first RFM model of human pose, defined directly on the product manifold of joint rotations, and (ii) Riemannian D-Flow, a principled guidance mechanism that, by differentiating through its ODE sampling dynamics, conditions PoseRFM at inference without any task-specific training. Our theoretical analysis shows that the induced dynamics are shaped by data covariance and manifold curvature, yielding a bias toward realistic poses. Across pose completion, denoising, and inverse kinematics, PoseD-Flow establishes new state of the art, particularly under noise, occlusion, and partial observations.*

1. Introduction

Humans do not merely occupy environments. We co-create them. Our physical configuration is our primary interface with the world, mediating perception, intent, and interaction. This configuration, **human pose**, is the instantaneous, deliberate spatial arrangement of the body in 3D. For machines to operate in human-centric environments, reasoning about pose is not optional but essential. Yet, faithful 3D pose estimation remains profoundly challenging: visual observations are noisy, partial, ambiguous, and frequently occluded, demanding *priors* that understand *what valid human pose actually looks like*.

The rapid evolution of deep generative modeling has opened an enticing direction: learning expressive pose priors that can later be used to infer plausible 3D configurations from incomplete 2D or 3D evidence [14, 24, 43, 50, 56]. Among generative paradigms, diffusion [25, 54] and flow matching [37, 38] have emerged as state of the art for modeling complex distributions. However, using these models as inference engines, *e.g.*, recovering poses that best explain given observations, requires *inverting* a generative process not trained for conditioning. This demands *training-free guidance*, a notoriously delicate prob-

lem [16]. While recent efforts have begun to explore guidance [3, 19, 63, 68], principled inversion of unconditional generative models remains unresolved for human pose.

Several approaches attempt pose recovery by optimizing in latent or geometry spaces at test time: HuMoR [50] inverts conditional VAEs, D-Poser [43] optimizes through diffusion, while PoseNDF [56] and NRDF [24] iteratively project onto neural distance fields [13]. Despite their success, none leverage the most expressive and tractable generative paradigm today, *flow matching* [37], nor do they treat human pose in its true configuration space: *a non-Euclidean product manifold of 3D rotations*.

We bridge this gap by proposing **PoseD-Flow**, the first framework to unite geometric flow-based pose priors with training-free guidance. As illustrated in Fig. 1, our method consists of two key components: (**i**) **PoseRFM**: the first Riemannian Flow Matching model of human pose, which models pose directly on the appropriate manifold of articulated rotations rather than in unconstrained Euclidean space; (**ii**) **Riemannian D-Flow**: a novel, training-free, geometry-respecting guidance mechanism that enables conditional generation by back-propagating through the Riemannian-ODE sampling process, yielding a theoretically grounded and empirically powerful strategy for solving pose recovery as an inverse problem. Unlike classifier-free or heuristic guidance methods, Riemannian D-Flow exposes a deeper inductive mechanism: gradients propagate through the flow in ways governed by data covariance and manifold curvature, creating a natural bias toward realistic, stable pose solutions. The result is a model that not only generates plausible poses, but *inverts* reliably, even under severe occlusions, noise, and partial observations.

Across denoising, pose completion, and inverse kinematics, PoseD-Flow sets a new state of the art, improving both geometric accuracy and perceptual plausibility, while being fully training-free at inference time. Our contributions are:

- **PoseRFM**, the first large-scale, Riemannian Flow Matching (RFM) model of human pose, supporting training-free, geometry-aware inversion on the articulation space;
- **Riemannian D-Flow**, a principled, training-free source-point optimization framework that guides any RFM model via differentiation through Riemannian ODE sampling, preserving geometry and stability;
- **Theoretical insights**, revealing that inference dynamics enjoy an inductive bias shaped by both data covariance and manifold curvature, offering an explanation for robustness against noise and ambiguity;
- **PoseD-Flow**, a versatile framework for human pose estimation, surpassing prior methods on motion denoising, completion, and inverse kinematics, particularly under occlusion and perturbations.

Our implementation is publicly available under github.com/circle-group/PoseDFlow.

2. Related Work

We now review the literature on priors of human pose and recent methods for inversion. Note that, while several works such as MotionVAE [36], HuMoR [50], PhaseMP [53], NRMF [69] or [27, 32, 52, 55, 67], model human motion unconditionally as well as conditionally [15, 26], our scope is human pose. We present an extended review in our supplementary material.

Unconditional human pose priors. Human bodies have been modeled by unconditional priors of various different kinds including Gaussian Processes [65], VAEs [49], normalizing flows [18, 51], neural distance fields (NDFs) [13, 24, 56] and diffusion models [14, 30, 42, 43]. To the best of our knowledge, there is no unconditional flow matching model of human pose, let alone the geometric variant.

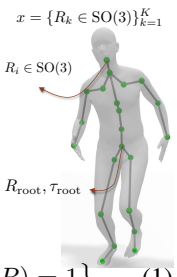
Riemannian flow matching (RFM). The flexible and general toolkit offered by RFM has been leveraged across numerous disciplines: in category level pose estimation by RFMPose [48], molecule generation by FoldFlow [8, 28], protein backbone generation by FrameFlow [66], generating materials by FlowMM [45], protein-ligand docking by FlowDock [46] & MATCHA [20], grasp pose generation by Equigraspflow [34], modeling brain connectivity by BrainFlow [8, 72], and for modeling statistical manifolds [8, 72], metal-organic structure prediction [31] and graph generation [10].

Guided flow matching. The Gaussian probability paths of flow matching have enabled a range of *training-free* methods for controlled generation, allowing a pre-trained model to generate samples that satisfy some target constraints [19, 22, 35, 71]. In particular, FlowGrad [39] inserts learnable control variables u_t at each integration step of the ODE and backpropagates a guidance loss through the trajectory. D-Flow [3] views controlled generation as optimizing the initial noise of a frozen generative model to minimize some terminal cost. Extending D-Flow, [29, 63] optimize the source distribution rather than a single point. OC-Flow [60] guides flow models by formulating flow-based generation as an optimal control problem with convergence guarantees. While OC-Flow has been extended to $\mathrm{SO}(3)$, and TFG-Flow [35] presented an $\mathrm{SO}(3)$-invariant control, our work constitutes the first general training-free guidance for RFM. To this end, we inherit D-Flow, due to its simplicity and the possibility of extension to the geometric domain of human pose we consider here.

3. PoseRFM

We now introduce our *expressive* and *generalizable* Riemannian flow matching model of 3D human pose learned from real human poses. To construct a flow on human poses that connects the target distribution to a source, we must first choose a suitable parametrization of articulation.

Geometry of human poses. The pose of a 3D articulated body $x := \{R_i \in \mathrm{SO}(3)\}_{i=1}^{K}\}$ is composed of K joints, each defined as a rotation $R_i \in \mathrm{SO}(3)$:



Definition 1 (SO(3))**.** *Rotations are elements of the special orthogonal group:*

$$\mathrm{SO}(3) = \left\{R \in \mathbb{R}^{3\times 3} : R^\top R = I, \det(R) = 1\right\}. \quad (1)$$

The power manifold of rotations $\mathcal{M} := \mathrm{SO}(3)^K = \mathrm{SO}(3)\times\cdots\times\mathrm{SO}(3)$, parameterizes all articulated poses. To geometrize these poses, we leverage the distance $d_{\mathcal{M}}$, exponential map Exp, logarithmic map Log and the Riemannian gradient grad on this product space [24, 69]:

$$d_{\mathcal{M}}(x, x') = \|d(R_1, R_1'), d(R_2, R_2'), \ldots, d(R_K, R_K')\|_2$$
$$\mathrm{Log}_x = \left(\mathrm{Log}_{R_1}, \mathrm{Log}_{R_2}, \ldots, \mathrm{Log}_{R_K}\right) \quad (2)$$
$$\mathrm{Exp}_x = \left(\mathrm{Exp}_{R_1}, \mathrm{Exp}_{R_2}, \ldots, \mathrm{Exp}_{R_K}\right) \quad (3)$$
$$\mathrm{grad}_x f(x) = \left(\mathrm{grad}_{R_1} f(x), \ldots, \mathrm{grad}_{R_K} f(x)\right) \quad (4)$$

where $(R_k, R_k') \in x \in \mathrm{SO}(3)^K$, and their operands are inherited from SO(3) as explained in our supplementary material.

We use the differentiable SMPL body model [40], $M(\boldsymbol{\beta}, x) := M(\tau, \boldsymbol{\phi}, x, \boldsymbol{\beta})$, parametrized by $K = 21$ joint rotations x, root translation $\tau \in \mathbb{R}^3$ and orientation $\boldsymbol{\phi} \in \mathrm{SO}(3)$, as well as $\boldsymbol{\beta} \in \mathbb{R}^{16}$ shape parameters. The joint positions $J \in \mathbb{R}^{3\times 22}$ are obtained via forward kinematics, while the vertices of the body mesh $V \in \mathbb{R}^{3\times 6890}$ via M.

PoseRFM: RFM model of human pose. We learn the manifold of plausible human poses through Riemannian flow matching [12] on a large corpus of data $\{x_i\}$, which we now introduce following the necessary definitions:

Definition 2 (Riemannian Flow [12])**.** *A time-dependent **flow** is a one-parameter family of diffeomorphisms $\{\psi_t : \mathcal{M} \to \mathcal{M}\}_{t=0}^{1}$ defined by integrating instantaneous deformations represented by a time-dependent vector field $u_t \in \Gamma(\mathcal{T}\mathcal{M})$ on the tangent space (Riemannian flow-matching field). ψ_t is defined by solving the following Riemannian ordinary differential equation (ODE) on $\mathcal{M}$ over $t \in [0, 1]$:*

$$\frac{d}{dt}\psi_t(x) = u_t\left(\psi_t(x)\right), \quad \psi_0(x) = x. \quad (5)$$

We also denote the flow map at $t = 1$ by $\psi_1 : \mathcal{M} \to \mathcal{M} : \psi_1(x_0) = x(1)$, a smooth cost $\mathcal{L} : \mathcal{M} \to \mathbb{R}_+$, and the source-point objective $\mathcal{L}(x_0) = \mathcal{L}(\psi_1(x_0))$.

Definition 3 (Probability path)**.** *Let $\mathcal{P}(\mathcal{M})$ denote the space of probability distributions on $\mathcal{M}$. A probability path $p_t : [0, 1] \to \mathcal{P}(\mathcal{M})$ interpolates between two distributions $p_0, p_1 \in \mathcal{P}(\mathcal{M})$ indexed by $t \in [0, 1]$. p_t is said to be **generated** by ψ_t if it* pushes forward $p_0 := p(x_0)$ *to $p_1 := p(x_1)$ following u_t,* i.e. *$p_t = [\psi_t]_{\#}(p_0)$. We define a smooth probability path between data p_1 and a reference p_0 as*

$$p_t(x) = \int_{\mathcal{M}} p_t(x \mid x_1)\, p_1(x_1)\, \mathrm{d}V(x_1), \quad (6)$$

where $p_t(x \mid x_1)$ is a geodesic Gaussian kernel with smooth schedulers $\alpha_t, \sigma_t > 0$ with $\alpha_0 = 0$ (see suppl. material).

Remark 1 (Velocity field)**.** *p_t satisfies a continuity (Liouville) equation on the manifold: $\partial_t p_t + \mathrm{div}_g(p_t u_t) = 0$, where u_t is the **velocity field** transporting probability mass along the manifold, where div_g is the divergence on $\mathcal{M}$.*

We are now ready to define Riemannian-FM (RFM).

Definition 4 (RFM)**.** *Given a probability path p_t, subject to the boundary conditions $p_0 = p_{\text{source}}$ and $p_1 = p_{\text{target}}$, as well as an associated flow ψ_t, Riemannian flow matching learns a* continuous normalizing flow *by directly regressing u_t through a neural network $v_w(x, t)$ parametrized by w.*

Definition 5 (Riemannian Conditional FM)**.** *The vanilla RFM objective is intractable as we do not have access to the closed-form u_t generating p_t. Instead, we regress v_w against a tractable* conditional vector field *$u_t(x \mid x_1)$, generating a* conditional probability path *$p_t(x \mid x_1)$ which can recover the target unconditional path by marginalization:*

$$u_t(x) = \int_{\mathcal{M}} u_t(x \mid x_1) \frac{p_t(x \mid x_1) p(x_1)}{p_t(x)}\, \mathrm{dV}_{x_1}. \quad (7)$$

Definition 6 (Generating conditional vector field)**.** *RFM defines a vector field $u_t(x \mid x_1)$ that generates $p_t(x \mid x_1)$ through a distance d by enforcing $d(\psi_t(x \mid x_1), x_1) = \kappa(t) d(x, x_1)$. The minimal-norm conditional field is [12]:*

$$u_t(x \mid x_1) = \frac{d}{dt}\log\kappa(t) \frac{d(x, x_1)}{\|\nabla d(x, x_1)\|_g^2} \nabla d(x, x_1). \quad (8)$$

For the geodesic distance $d := d_g$ and $\kappa(t) := 1 - t$, $\|\nabla d_g\|_g = 1$ and $d_g \nabla d_g = \nabla \frac{1}{2} d_g^2 = -\mathrm{Log}_x(x_1)$, giving

$$u_t(x \mid x_1) = \frac{1}{1 - t} \mathrm{Log}_x(x_1) \quad (9)$$

Algorithm 1 : PoseRFM training

Given: base & target distributions: $p(x_0), p(x_1)$
Initialize: parameters w of network $v_w(x, t)$ randomly
while not converged **do**
  sample noise $x_0 \sim p(x_0)$, target $x_1 \sim p(x_1)$
  sample time $t \sim \mathcal{U}(0, 1)$
  $x_t \leftarrow \mathrm{Exp}_{x_0}(t\, \mathrm{Log}_{x_0}(x_1))$
  $\mathcal{L}(w) \leftarrow \|v_w(x, t) - u_t(x_t \mid x_1)\|_g^2$
  $w \leftarrow$ `optimizer_step`$(\mathcal{L}(w))$
end while
Return: v_t

Algorithm 2 : Riemannian D-Flow

Given: Pre-trained flow model $v_w \approx u_t$, loss $\mathcal{L}$
Initialize: $x_0^{(0)} \in \mathcal{M}$ randomly, $v_0 = 0, m_0 = 0 \in T_{x_0}\mathcal{M}$, hyperparameters $\alpha, \beta_1, \beta_2, \varepsilon$
for $i = 1, \ldots, N$ **do**
  $x_1^{(i)} \leftarrow \texttt{solve\_ode}(x_0^{(i)},\ v_w)$
  $\nabla\mathcal{L} \leftarrow \nabla_{x_0^{(i)}}\mathcal{L}(x_1^{(i)})$
  $x_0^{(i+1)} \leftarrow \texttt{RAdam\_step}(x_0^{(i)}, \nabla\mathcal{L})$
end for
Return: x_1^N

With this choice of time scheduling Chen & Lipman [12] then define an explicit Riemannian conditional FM (RCFM) objective for learning as:

$$\mathbb{E}_{t,p(x_1),p(x_0)} \left\| v_w(x_t,t) + d(x_0,x_1)\frac{\text{grad } d(x_t,x_1)}{\|\text{grad } d(x_t,x_1)\|_g^2} \right\|_g^2,$$

whose gradient is the same as that of RFM. We use Eq. (9) to train our PoseRFM, *i.e.*, to obtain the parameters w of v_w. Here, $t \in \mathcal{U}(0,1)$ and $d(\cdot,\cdot)$ is the geodesic distance.

Training algorithm. At each training step, we sample a $x_1 \sim p(x_1)$ from the target distribution and a $t \sim \mathcal{U}(0,1)$ randomly. Using the conditional probability path construction, a sample $x_t \sim p_t(\cdot \mid x_1)$ is obtained. For this pair (x_t, x_1), the target conditional tangent vector field $u_t(x_t \mid x_1) = \dot{x}_t$ is computed in closed form on $\mathcal{T}_x\mathcal{M}$. We also compute the network output $v_w(x,t)$ and evaluate the loss, which is backpropagated to update the network parameters w. The pseudo-code is shown in Alg. 1.

Sampling / generation. Once trained, PoseRFM can sample novel poses by integrating the ODE on the manifold from $t: 0 \to 1$ using a manifold-aware ODE step, *e.g.*, Riemannian-Euler: $x_{t+\eta} \approx \text{Exp}_{x_t}\left(\eta\Pi_{\mathcal{T}_{x_t}\mathcal{M}}\left(v_w(x_t,t)\right)\right)$. Some generation results are shown in Fig. 3.

Implementation details. For all applications, we parameterize the vector field $v_w(x,t)$ as a time-conditional neural network using simple a MLP (4 hidden layers, with a hidden dim size of 512), and Swish activation function. The input is a pose $x_t \in (\mathcal{M} := \text{SO}(3)^{\text{K}})$ with dimension 189 ($21\times3\times3$) concatenated with a time variable $t \in [0,1]$, and returns a vector in the tangent space $\mathcal{T}_{x_t}SO^{\text{K}}$. We train our model for 50,000 steps using AdamW optimizer [41] with a learning rate of 1e-3, weight decay of 1e-4, an exponential weight moving average (EMA) of 0.99, and a batch size of 4096. We use ReduceLROnPlateau scheduler, reducing the learning rate by 0.5 whenever validation loss stops improving for 5 evaluation steps.

Algorithm 3 `RAdam_step`

Given: $x_k \in \mathcal{M}$, $m_k \in \mathcal{T}_{x_k}\mathcal{M}$,$v_k$, $\nabla_{x_k}\mathcal{L}$, $\alpha, \beta_1, \beta_2, \varepsilon$
$g_k \leftarrow \text{grad}_{x_k}\mathcal{L} := \Pi_{T_{x_k}\mathcal{M}}(\nabla_{x_k}\mathcal{L}) \in T_{x_k}\mathcal{M}$
$m_k \leftarrow \text{PT}_{x_{k-1}\to x_k}(m_k)$
$m_{k+1} \leftarrow \beta_1 m_k + (1-\beta_1)\, g_k$
$v_{k+1} \leftarrow \beta_2 v_k + (1-\beta_2)\,\langle g_k, g_k\rangle_{x_k}$
$\hat{m}_k \leftarrow m_{k+1}/(1-\beta_1^t)$
$\hat{v}_k \leftarrow v_{k+1}/(1-\beta_2^t)$
$\Delta x_k \leftarrow -\eta\,\hat{m}_k/(\sqrt{\hat{v}_k} + \varepsilon)$
$x_{k+1} \leftarrow \text{Exp}_{x_k}(\Delta x_k)$

4. Riemannian D-Flow: Guiding PoseRFM

We now develop our algorithm, Riemannian D-Flow, to control the generation process of RFM models, in other words to invert our PoseRFM. We do so by extending D-Flow [3] to Riemannian manifolds following [12].

Given a pre-trained (frozen) PoseRFM model, $v_w(x,t) \approx u_t(x)$, represented as a neural network and some cost function $\mathcal{L}: \mathcal{M} \to \mathbb{R}_+$ and associated regularizers $\mathcal{R}$, our goal is to find likely samples $x \in \mathcal{M}$ that provide low cost $\mathcal{L}(x)$ and are likely under RFM's distribution $p(x_1)$. We formulate this as a *Riemannian source-point optimization problem*:

$$\min_{x_0\in\mathcal{M}} \big(\mathcal{L}(x(1)) := \mathcal{L}_{\text{data}}(x(1)) + \mathcal{R}(x_0,u)\big), \tag{10}$$

where $x(1)$ solves the Riemannian ODE in Eq. (5) with initial condition $x(0) = x_0$.

While our Riemannian D-Flow is a general framework, in this work, we make particular choices for objectives curated for human pose. While $\mathcal{L}_{\text{data}}$ is specified for each task distinctly in Sec. 5, we adopt a *trajectory regularizer* [61] common to all applications we consider:

$$\mathcal{R} := \mathcal{L}_{\text{traj}} = \sum_{i=1}^{K}\sum_{k=1}^{N}\big(3 - \text{tr}(x_{ik})\big), \tag{11}$$

where x_{ik} is the rotation of k^{th} joint at discrete timestep i. $\mathcal{R}$ explicitly forces the angle of rotations to be as small as possible, preventing wild trajectories and penalizing large, physically implausible rotations along the trajectory.

Algorithm and implementation details. Given a pre-trained flow, each iteration of Riemannian D-Flow solves the ODE forward in time via Euler integration, starting from $x_0 := x(0)$, and obtains $x_1 := x(1)$, where the loss $\mathcal{L}(x_1)$ is evaluated. The gradient $\nabla_{x_0}\mathcal{L}(\psi_1(x_0))$ is projected onto the tangent space of x_0 to obtain the Riemannian gradient. We then update the source point x_0 using Riemannian Adam [2]. This algorithm is summarized in Alg. 2. Other task-dependent hyperparameters are discussed in Sec. 5.

We now provide theoretical insights into the implicit biases of our Riemannian D-Flow.

4.1. Theoretical Analysis

We now provide a theoretical analysis that reveals an implicit regularization that comes from the choice of optimizing the cost as a function of the source point, where the gradient updates follow the data distribution $p_1 := p_1(x_1)$ by projecting the Riemannian gradient with the local data covariance matrix while respecting the curvature. Throughout, we assume the data distribution is supported away from the cut locus. All omitted proofs are provided in our supplementary material.

Theorem 1 (Tangent denoiser)**.** *For the data distribution $x_1 \sim p_1$ on $\mathcal{M}$, define at any $x \in \mathcal{M}$ the tangent random variable $\xi_x := \mathrm{Log}_x(x_1) \in \mathcal{T}_x\mathcal{M}$. The tangent denoiser μ at time t given by:*

$$\mu(x) := \mu_{1|t}(x) = \mathbb{E}\left[\xi_x \mid x(t) = x\right], \tag{12}$$

is the unique minimizer of the Riemannian flow matching loss:

$$\mathcal{L}(v) := \mathbb{E}\left[\|\xi - v\|_{g_x}^2 \mid x_t = x\right]. \tag{13}$$

Theorem 2 (Covariant derivative & covariance)**.** *The covariant derivative $(\nabla\mu)_v(x) : \mathcal{T}_x\mathcal{M} \to \mathcal{T}_x\mathcal{M}$ of the tangent denoiser is given by:*

$$(\nabla\mu)_v(x) = A_x[C(x)v] + R_x[v], \tag{14}$$

where $A_x[v]$ is a linear operator, $C(x) := \mathrm{C}_{1|t}(x)$ is a self-adjoint, positive semidefinite linear map $\mathcal{T}_x\mathcal{M} \to \mathcal{T}_x\mathcal{M}$, representing the covariance under the conditional distribution, and $R_x[v]$ is a Riemannian remainder*:*

$$C(x) = \mathbb{E}\left[\left(\xi_x - \mu_{1|t}\right) \otimes \left(\xi_x - \mu_{1|t}\right) \mid x(t) = x\right] \tag{15}$$

$$R_x[v] = \mathbb{E}\left[\nabla_v^{(x)}\xi \mid x\right]. \tag{16}$$

As $t \to 1$, $C(x)$ approaches the local data covariance under variance scheduling of geodesic kernels: $\sigma \to 0$.

Remark 2 (Euclidean space as a special case)**.** *In a Euclidean space with Gaussian densities where $\xi = \log_x(x_1) = x_1 - x$, $\nabla_v^{(x)}\xi = \nabla_v(x_1 - x) = -v$, the remainder becomes:*

$$R_x[v] = \mathbb{E}[-v \mid x] = -v. \tag{17}$$

In the D-Flow derivation this constant $-v$ cancels exactly with the same constant appearing from the score derivative, leaving no leftover curvature term: $R_x = 0$ (disappearing after cancellation). For AGPP in [3], $A_x = \alpha_t/\sigma_t^2 I$.

Corollary 1. *The covariant derivative of the marginal velocity field satisfies ($\circ$ denotes composition):*

$$\nabla u_t(x) = \frac{1}{1-t}\left(C_{1|t}(x) \circ A_x + R_x\right). \tag{18}$$

This is the drift of the adjoint ODE, determining the Riemannian adjoint, as we explain next.

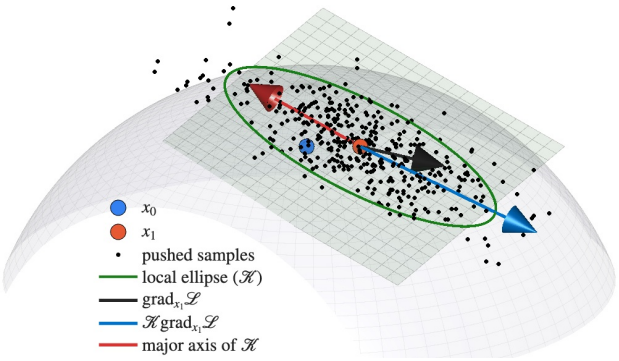


Figure 2. **Implicit bias:** The source point update steers the endpoint gradient towards the reachable subspace induced by the data.

Theorem 3 (Riemannian adjoint)**.** *Let $\Psi : \mathcal{M} \to \mathcal{M}$ denote the flow such that $x_1 = \Psi(x_0)$. Then the Riemannian gradients at the start and end points are related by the* Riemannian adjoint *(pullback map) $D_{x_0}\Psi(x_0)^*$:*

$$\mathrm{grad}_{x_0}\mathcal{L}(x_1) = D\Psi(x_0)^*\left[\mathrm{grad}_{x_1}\mathcal{L}(x_1)\right]. \tag{19}$$

Sketch of the proof. The result can be proven using the standard identities in Riemannian geometry, which we provide in our suppl. material for completeness. □

We finally prove the last bit characterizing the effect of the source-point update on the end-point.

Theorem 4 (Implicit bias in endpoint update)**.** *Consider a small optimization step updating the optimized source variable: $x_0 \longrightarrow \mathrm{Exp}_{x_0}\left(-\eta\,\mathrm{grad}_{x_0}\mathcal{L}(x_1)\right)$ where $\mathrm{grad}_{x_0}\mathcal{L}(x_1)$ is given in Eq. (19). As $\eta \to 0$ (infinitesimal change), the variation of the end-point $x_1 = \Psi(x_0)$, denoted δx_1 reads:*

$$\delta x(1) = -\eta \underbrace{\left(D_{x_0}\Psi(x_0)\, D_{x_0}\Psi(x_0)^*\right)}_{\mathcal{K}\text{ self-adjoint, PSD on }\mathcal{T}_{x_1}\mathcal{M}} \mathrm{grad}_{x(1)}\mathcal{L}(x(1)),$$

where $\mathcal{K} = D_{x_0}\Psi(x_0)\, D_{x_0}\Psi(x_0)^*$ resembles a local covariance on the endpoint gradient, explaining why the update is biased toward directions of high density, see Fig. 2.

We now discuss the importance of Thm. 2 and Thm. 4.

Remark 3 (Learning is **geometry-aware** and **data-adaptive**)**.** *Updating the source point during guidance leads to a* filtered update *on the end point, where the operator $\mathcal{K}$ is a projection onto the* reachable subspace *of the endpoint tangent space. When the flow generates the data distribution, this operator becomes the local covariance of the data manifold. Combined with the decomposition in Eq. (14) into a data-driven covariance term and an additional curvature-induced correction arising from the base-point dependence of the logarithmic map, suggests that the gradient updates of the terminal point are shaped by the* local data covariance *in the tangent space informed by the* curvature*. This is analogous to Euclidean D-Flow, yet mindful of geometry.*

Figure 3. PoseRFM samples from unconditional generation.

5. Experiments

To evaluate its versatility, we evaluate PoseRFM on several inverse problems, covering both linear and non-linear pose problems: unconditional human pose generation, motion denoising from noisy joints, and human mesh recovery from 2D images (non-linear). We then conduct ablation studies to evaluate the contribution of each component in PoseRFM. Detailed hyperparameters, additional ablation studies and evaluations are available in our suppl. material.

Datasets. We use AMASS [44], a large motion capture database, to train our models using the same preprocessing and splits as prior work [24, 49, 50, 56]. We evaluate performance on the held-out AMASS test split, and further assess our method's generalization ability on HPS [23], EHF [49] and 3DPW [59], without additional finetuning.

Baselines. To ensure a fair comparison, we adopt the experimental setup from DPoser [43] for both training and evaluation. To study the impact of our geometric formulation in isolation, we also train a baseline Euclidean FM model, PoseFM, under the exact same setup as PoseRFM.

Metrics. For unconditional generation, we evaluate realism using FID (Fréchet Inception Distance) and d_{NN} (Nearest Neighbor distance) [24], and quantify diversity with APD (Average Pairwise Distance). Where a ground truth pose is available, we use MPJPE (Mean Per Joint Position Error) and MPVPE (Mean Per Vertex Position Error) to measure accuracy. We also report their Procrustes Aligned versions, PA-MPJPE and PA-MPVPE, which are rotation invariant and capture only body shape errors. Finally, PCK@50 (Percentage of Correct Keypoints) evaluates 2D joint localization accuracy, counting a joint as correct if its distance to the ground truth is within 50 mm.

5.1. Evaluations & Results

Pose generation. To generate realistic and diverse poses unconditionally, we randomly sample x_0 and use the midpoint ODE solver with 100 integration steps. Following [24], we sample 20 sets of 500 poses, and report the mean and 95% confidence intervals across runs in Tab. 1.

Our method achieves SotA performance in realism and fidelity, with the lowest FID and d_{NN}, indicating that the generated poses closely match real pose distributions. While not achieving the highest diversity, it remains competitive and outperforms several baselines. Methods achieving the highest APD (Pose-NDF [56] and NRDF [24]), do so at a substantial cost to realism, reflected by their higher FID and d_{NN} scores. Fig. 3 highlights the realism and diversity of our generated poses, suggesting a robust prior, capturing the distribution of plausible body configurations.

Table 1. Results for unconditional pose generation. Last three rows indicate our models.

Method	FID ↓	APD (cm) ↑	d_{NN} (rad) ↓
GMM [6]	$0.435^{\pm.017}$	$21.944^{\pm.102}$	$0.159^{\pm.001}$
VPoser [49]	$0.048^{\pm.002}$	$14.684^{\pm.138}$	$0.074^{\pm.000}$
GAN-S [17]	$0.201^{\pm.030}$	$10.914^{\pm.396}$	$0.098^{\pm.001}$
Pose-NDF [56]	$3.920^{\pm.034}$	$\mathbf{37.813}^{\pm.085}$	$0.838^{\pm.001}$
GFPose-A [14, 24]	$1.246^{\pm.005}$	$13.876^{\pm.116}$	–
GFPose-Q [14, 24]	$1.624^{\pm.002}$	$6.773^{\pm.112}$	$0.159^{\pm.000}$
FM-Dis [24]	$0.346^{\pm.007}$	$6.849^{\pm.199}$	$0.086^{\pm.001}$
NRDF [24]	$0.636^{\pm.007}$	$23.116^{\pm.105}$	$0.177^{\pm.001}$
Lie-HMR [30]	$0.825^{\pm.067}$	$\underline{23.999}^{\pm.602}$	–
DPoser [43]	$0.019^{\pm.001}$	$14.992^{\pm.123}$	$0.073^{\pm.001}$
PoseFM	$0.016^{\pm.001}$	$14.762^{\pm.176}$	$0.079^{\pm.000}$
PoseRFM (N=100)	$\underline{0.014}^{\pm.001}$	$15.481^{\pm.124}$	$\mathbf{0.069}^{\pm.001}$
PoseRFM (N=1000)	$\mathbf{0.013}^{\pm.001}$	$15.544^{\pm.167}$	$\underline{0.070}^{\pm.001}$

Pose completion. We now deploy our learned prior at reconstructing full human poses from partially observed or occluded poses, minimizing the total geodesic distance:

$$\mathcal{L}_{\mathrm{data}}(x) := \sum_{k\in\Omega} \cos^{-1}\left(\frac{\mathrm{tr}\left(x_k^\top x_k^{\mathrm{obs}}\right)-1}{2}\right) \tag{20}$$

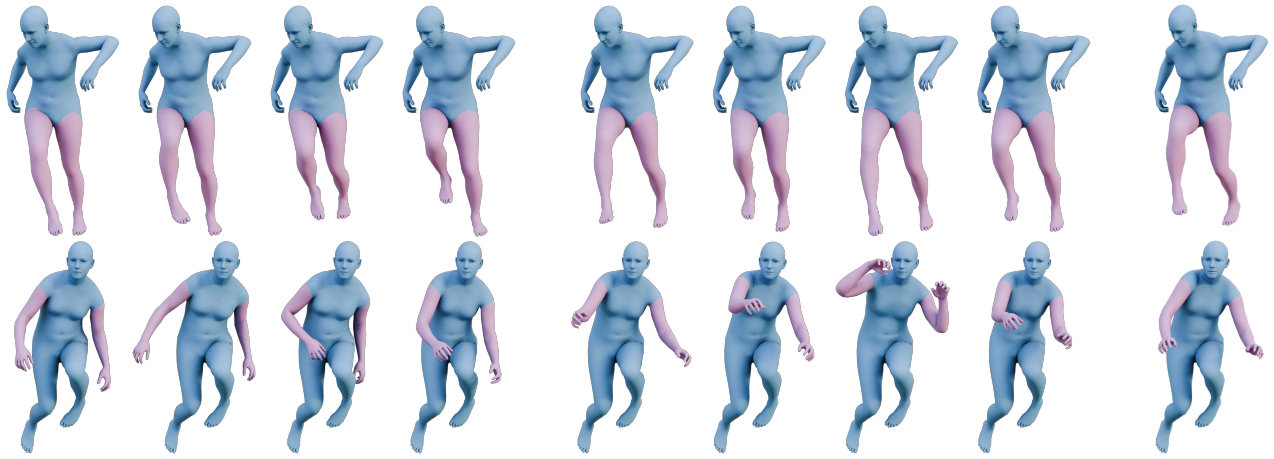
where x_k is the rotation of k^{th} joint of the generated pose. Ω is a known *mask* partially occluding the observation x^{obs}.

We evaluate this on the AMASS test split [44] with a subsampling rate of 10, and synthetically occluding four body parts: *left leg*, *legs*, *arms* and *torso*. For each occluded pose, we generate 10 hypotheses and evaluate their accuracy against the ground truth (GT) using the mean$^{\pm\mathrm{std}}$ of MPVPE across all hypotheses. To assess diversity among generated poses, we compute the APD over the 10 samples.

Our results in Tab. 2 shows competitive performance across all occlusion types, consistently balancing accuracy with diversity, with MPVPE being on par or better than DPoser [43], while also maintaining high diversity scores. Fig. 4 shows a qualitative comparison of completed poses, with occluded joints marked. Both DPoser [43] and PoseRFM produce realistic and diverse completions. This example also exposes a key limitation of the MPVPE metric, that assumes the GT pose as the only valid completion and penalizes other plausible alternatives. This motivates the need for a more suitable metric accounting for the distribution of plausible pose completions, akin to an FID.

Table 2. Pose Completion results under varying occlusion scenarios. We report MPVPE (mm) and APD (cm) metrics.

Method	Occ. left leg		Occ. legs		Occ. arms		Occ. torso	
	MPVPE ↓	APD ↑	MPVPE ↓	APD ↑	MPVPE ↓	APD ↑	MPVPE ↓	APD ↑
VPoser [49]	$200.51^{\pm 12.20}$	2.41	$221.34^{\pm 16.40}$	5.44	$206.72^{\pm 12.91}$	4.08	$58.71^{\pm 7.38}$	1.56
Pose-NDF [56]	$168.45^{\pm 8.66}$	1.95	$169.86^{\pm 6.12}$	1.97	$260.94^{\pm 4.81}$	1.20	$114.97^{\pm 5.43}$	0.93
CVPoser [43]	$128.04^{\pm 10.36}$	1.91	$134.35^{\pm 10.17}$	2.43	$162.82^{\pm 5.58}$	1.08	$51.23^{\pm 4.32}$	0.57
DPoser [43]	$\mathbf{78.31}^{\pm 27.13}$	6.53	$102.46^{\pm 25.39}$	7.75	$\mathbf{104.94}^{\pm 26.44}$	5.69	$44.60^{\pm 14.65}$	2.21
PoseFM	$102.60^{\pm 34.82}$	**8.77**	$129.04^{\pm 30.93}$	**9.94**	$140.88^{\pm 35.63}$	**7.85**	$51.86^{\pm 17.25}$	**2.70**
PoseRFM (ours)	$83.81^{\pm 29.07}$	6.02	$\mathbf{95.00}^{\pm 26.08}$	7.23	$107.36^{\pm 30.41}$	5.72	$\mathbf{39.75}^{\pm 9.12}$	1.21



(a) DPoser [43] (b) PoseRFM (ours) (c) Ground truth (GT)

Figure 4. Completed poses with legs (top row) and arms (bottom row) occluded. Visible and occluded joints.

Table 3. Motion denoising results. We report MPJPE (mm) under different Gaussian noise levels (σ= 4 / 10cm).

Method	AMASS [44]		HPS [23]	
	40 mm	100 mm	40 mm	100 mm
w/o prior	24.19	51.48	23.67	50.87
VPoser [49]	23.42	49.10	22.78	46.69
Pose-NDF [56]	22.13	46.10	21.60	47.50
MVAE [36]	26.80	–	–	–
HuMoR [50]	22.69	–	–	–
DPoser [43]	19.87	**33.18**	20.54	35.32
PoseRFM (ours)	**18.88**	34.68	**19.79**	**32.95**

Table 4. Human Mesh Recovery on EHF [49] dataset. We report PA-MPJPE (mm).

Method	from scratch	CLIFF [33] init.
w/o fitting	108.57	56.62
GMM [6]	58.32	51.02
VPoser [49]	58.08	49.39
GAN-S [17]	57.26	49.58
Pose-NDF [56]	57.87	49.50
NRDF [24]	57.38	49.27
DPoser [43]	56.05	49.05
PoseRFM (ours)	**54.85**	**47.24**

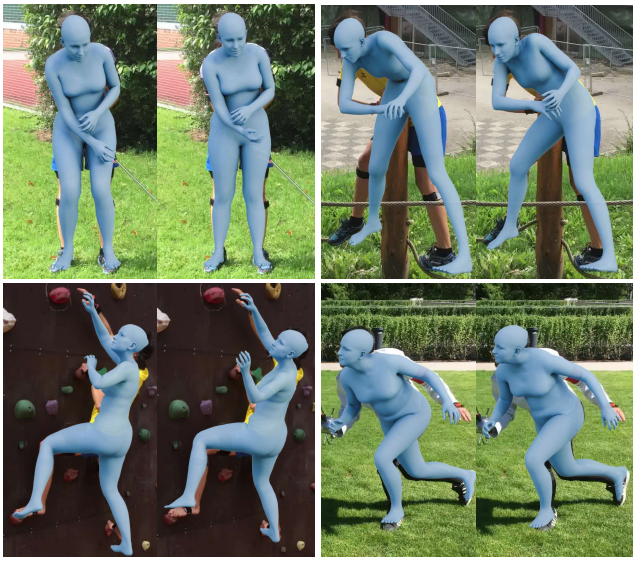
Motion denoising. Next, we consider motion denoising, which aims to recover temporally consistent human motion from noisy joint trajectories. We define the denoising loss:

$$\mathcal{L}_{\text{data}} := \mathcal{L}_{\text{joints}} + \lambda \mathcal{L}_{\text{smooth}} \tag{21}$$
$$= \|J_t - \mathcal{J}(M(\boldsymbol{\beta}, x_t^*))\|_2^2 + \lambda \sum_{k=1}^{K} \left(d_g\left(x_{t-1,k}, x_{t,k}\right)\right)$$

where J_t are some noisy joint inputs, x_t^* the estimated underlying clean pose, $\mathcal{J}$ extracts the joints from SMPL model, and $\mathcal{L}_{\text{smooth}}$ promotes temporal smoothness between consecutive poses.

We test this approach on the HumanEva subset from AMASS [44] and the HPS [23] dataset. All sequences are subsampled at 30 Hz and divided into 60-frame (2s) segments. We add Gaussian noise with $\sigma = 0.04, 0.1$ to the clean 3D joints and apply our conditional generation framework to recover the denoised poses. Our results in Tab. 3 show that PoseRFM consistently achieves superior performance over baselines across datasets and noise levels.

Inverse kinematics. Next, we test our framework on the non-linear human mesh recovery (HMR) task, which aims to estimate SMPL parameters $(\boldsymbol{\beta}^*, x^*)$ from 2D images and



DPoser [43] PoseRFM DPoser [43] PoseRFM

Figure 5. Results of HMR on in-the-wild images from 3DPW [59]. Fitting from scratch (top) and init. using CLIFF [33] (bottom).

observed 2D keypoints J_{obs}:

$$\mathcal{L}_{\text{data}}(x, \boldsymbol{\beta}) = \mathcal{L}_{\text{2D}} + \mathcal{L}_{\alpha} + \mathcal{L}_{\boldsymbol{\beta}} \quad \text{where :} \tag{22}$$

$$\mathcal{L}_{\text{2D}}(x, \boldsymbol{\beta}) = \text{diag}(\boldsymbol{\sigma})\rho(p_k^{\text{obs}} - \Pi(\mathcal{J}(M(\boldsymbol{\beta}, x^*)))) \tag{23}$$

$$\mathcal{L}_{\alpha} = \sum_{i \in (\text{elbow,knees})} \exp(R_i), \quad \mathcal{L}_{\boldsymbol{\beta}} = \|\boldsymbol{\beta}\|^2. \tag{24}$$

$\boldsymbol{\sigma}$ denote detection confidences, Π represents the perspective projection, and ρ the robust Geman-McClure function [6, 21]. If an estimated person segmentation mask is available, it is used to ignore spurious 2D joints.

We evaluate our method on the EHF [49] and 3DPW [59] datasets, by following the SMPLify [6] optimization pipeline. We begin with 2D keypoints predicted by ViTPose [64], initialize the pipeline with the mean pose, and optimize for camera parameters. We then optimize for the SMPL parameters $(\boldsymbol{\beta}, x)$ using Riemannian D-Flow. Additionally, we test an alternative initialization where the SMPL and camera parameters are first predicted by CLIFF [33] and used to initialize Riemannian D-Flow. We report the predicted joints accuracy using PA-MPJPE in Tab. 4. We advance the SotA under both initialization schemes, demonstrating the effectiveness of PoseD-Flow in solving non-linear inverse pose problems. The qualitative results in Fig. 5 highlight our method's ability to recover accurate poses in complex in-the-wild scenarios, often containing out-of-distribution and rare pose configurations. While PoseRFM generally fits better than DPoser [43], both methods occasionally struggle with unnatural hand and foot orientations. We provide an extended comparison with the current SotA in Tab. 5, including additional metrics for a more comprehensive evaluation.

Table 5. Detailed HMR evaluation with additional metrics.

Init.	Method	PA-MPJPE ↓	PA-MPVPE ↓	PCK@50 ↑
from scratch	DPoser [43]	56.05	53.67	61.60
	PoseRFM	**54.85**	**53.16**	**65.29**
CLIFF [33]	DPoser [43]	49.05	52.92	66.62
	PoseRFM	**47.24**	**48.67**	**71.40**

Table 6. Ablation study on the effect of geometric components.

Method	Geo. Loss	Traj. Loss	Occ. left leg		Occ. legs	
			MPVPE ↓	APD ↑	MPVPE ↓	APD ↑
PoseFM	✗	✗	102.60	8.77	129.04	9.94
	✓	✗	215.25	20.11	273.96	25.94
	✗	✓	141.66	5.92	148.77	6.29
	✓	✓	180.51	16.81	212.90	20.52
PoseFM + geometry	✗	✗	99.79	7.45	119.92	8.56
	✓	✗	91.00	6.34	115.58	7.45
	✗	✓	129.55	4.86	138.05	4.78
	✓	✓	90.46	5.79	115.67	6.53
PoseRFM	✗	✗	107.32	9.56	120.26	11.26
	✓	✗	91.80	8.07	107.28	10.02
	✗	✓	109.35	1.81	129.44	1.50
	✓	✓	83.81	6.02	95.00	7.23

Ablation study. Finally, we ablate on the contribution of each Riemannian component of our model on the pose completion task, including the model type (FM vs. RFM), the data loss (MSE vs. geodesic), and the trajectory loss. The results are summarized in Tab. 6. While the Euclidean baseline (PoseFM) is competitive, its accuracy is limited, reflected by its higher MPVPE. Adding geometric losses and regularization to this model further degrades performance for PoseFM. This is expected since PoseFM does not model the underlying manifold. In contrast, PoseRFM yields similar initial results but achieves significantly higher accuracy once geodesic and trajectory losses are incorporated, while still preserving output diversity.

6. Conclusion

We introduced PoseD-Flow, the first framework to unify Riemannian Flow Matching with training-free geometric guidance for human pose understanding. By learning an expressive pose prior (PoseRFM) directly on the pose manifold and developing a principled mechanism (Riemannian D-Flow) to invert it, we set a new state of the art across various challenging tasks, where robustness matters most. We theoretically show that the induced guidance dynamics inherit structure from data covariance and manifold curvature, yielding a natural inductive bias toward realistic poses. Empirically, PoseD-Flow sets a new state of the art across various challenging tasks.

Limitations & future work. Due to the backpropagation all the way to the source, our method is not real-time. Future work can take inspiration from FlowGrad [39] and OC-Flow [60] which provide ways to reduce memory usage. We plan to increase diversity by injecting stochasticity during inversion. Lastly, we will extend our work to motion.

Acknowledgments

T. Birdal and S. Foti acknowledges support from the Engineering and Physical Sciences Research Council Project GNOMON [grant EP/X011364/1]. T. Birdal is supported by a UKRI Future Leaders Fellowship [grant number MR/Y018818/1]. The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR), funded by the DSIT, STFC and UKRI. S. Foti was supported by the Turing AI Fellowship MAGAL (EP/Z534699/1).

References

[1] P-A Absil and Kyle A Gallivan. Accelerated line-search and trust-region methods. *SIAM Journal on Numerical Analysis*, 47(2):997–1018, 2009. 12

[2] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations (ICLR 2019)*, pages 6384–6399. Curran, 2023. 4

[3] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: differentiating through flows for controlled generation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 3462–3483, 2024. 2, 4, 5, 16, 19

[4] Tolga Birdal and Umut Simsekli. Probabilistic permutation synchronization using the riemannian structure of the birkhoff polytope. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11105–11116, 2019. 12

[5] Tolga Birdal, Umut Simsekli, Mustafa Onur Eken, and Slobodan Ilic. Bayesian pose graph optimization via bingham distributions and tempered geodesic mcmc. *Advances in Neural Information Processing Systems*, 31, 2018. 12

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision*, pages 561–578. Springer, 2016. 6, 7, 8

[7] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9): 2217–2229, 2013. 12

[8] Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian FATRAS, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael M Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[9] Nicolas Boumal. An introduction to optimization on smooth manifolds. *Available online, May*, 2020. 12

[10] Tianci Bu, Chuanrui Wang, Hao Ma, Haoren Zheng, Xin Lu, and Tailin Wu. Ggball: Graph generative model on poincar\'e ball. *arXiv preprint arXiv:2506.07198*, 2025. 2

[11] Jiayi Chen, Yingda Yin, Tolga Birdal, Baoquan Chen, Leonidas J Guibas, and He Wang. Projective manifold gradient layer for deep rotation regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6655, 2022. 12

[12] Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 4, 13, 19

[13] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020. 2

[14] Hai Ci, Mingdong Wu, Wentao Zhu, Xiaoxuan Ma, Hao Dong, Fangwei Zhong, and Yizhou Wang. Gfpose: Learning 3d human pose prior with gradient fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4800–4810, 2023. 1, 2, 6

[15] Manolo Canales Cuba, Vinícius do Carmo Melício, and João Paulo Gois. Flowmotion: Target-predictive conditional flow matching for jitter-reduced text-driven human motion generation. *Computers & Graphics*, page 104374, 2025. 2

[16] Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024. 2

[17] Andrey Davydov, Anastasia Remizova, Victor Constantin, Sina Honari, Mathieu Salzmann, and Pascal Fua. Adversarial parametric pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10997–11005, 2022. 6, 7

[18] Olaf Dünkel, Tim Salzmann, and Florian Pfaff. Normalizing flows on the product space of so (3) manifolds for probabilistic human pose modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, 2024. 2

[19] Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. *arXiv preprint arXiv:2502.02150*, 2025. 2

[20] Daria Frolova, Talgat Daulbaev, Egor Sevryugov, Sergei A Nikolenko, Dmitry N Ivankov, Ivan Oseledets, and Marina A Pak. Matcha: Multi-stage riemannian flow matching for accurate and physically valid molecular docking. *arXiv preprint arXiv:2510.14586*, 2025. 2

[21] Donald Geman and Stuart Geman. Bayesian image analysis. In *Disordered systems and biological organization*, pages 301–319. Springer, 1986. 8

[22] Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. *arXiv preprint arXiv:2502.11420*, 2025. 2

[23] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 6, 7, 16

[24] Yannan He, Garvita Tiwari, Tolga Birdal, Jan Eric Lenssen, and Gerard Pons-Moll. Nrdf: Neural riemannian distance fields for learning articulated pose priors. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1661–1671, 2024. 1, 2, 3, 6, 7, 12, 16

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[26] Vincent Tao Hu, Wenzhe Yin, Pingchuan Ma, Yunlu Chen, Basura Fernando, Yuki M Asano, Efstratios Gavves, Pascal Mettes, Bjorn Ommer, and Cees GM Snoek. Motion flow matching for human motion synthesis and editing. *arXiv preprint arXiv:2312.08895*, 2023. 2

[27] Yiheng Huang, Hui Yang, Chuanchen Luo, Yuxi Wang, Shibiao Xu, Zhaoxiang Zhang, Man Zhang, and Junran Peng. Stablemofusion: Towards robust and efficient diffusion-based motion generation framework. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 224–232, 2024. 2

[28] Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Chenghao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein generation. *Advances in neural information processing systems*, 37:33007–33036, 2024. 2

[29] Adhithyan Kalaivanan, Zheng Zhao, Jens Sjölund, and Fredrik Lindsten. Ess-flow: Training-free guidance of flow-based models as inference in source space. *arXiv preprint arXiv:2510.05849*, 2025. 2

[30] Donghwan Kim and Tae-Kyun Kim. Liehmr: Autoregressive human mesh recovery with $so(3)$ diffusion. *arXiv preprint arXiv:2509.25739*, 2025. 2, 6

[31] Nayoung Kim, Seongsu Kim, Minsu Kim, Jinkyoo Park, and Sungsoo Ahn. Mofflow: Flow matching for structure prediction of metal-organic frameworks. *arXiv preprint arXiv:2410.17270*, 2024. 2

[32] Nilesh Kulkarni, Davis Rempe, Kyle Genova, Abhijit Kundu, Justin Johnson, David Fouhey, and Leonidas Guibas. Nifty: Neural object interaction fields for guided human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 947–957, 2024. 2

[33] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *European Conference on Computer Vision*, pages 590–606. Springer, 2022. 7, 8, 16, 21

[34] Byeongdo Lim, Jongmin Kim, Jihwan Kim, Yonghyeon Lee, and Frank C Park. Equigraspflow: Se (3)-equivariant 6-dof grasp pose generative flows. In *8th Annual Conference on Robot Learning*, 2024. 2

[35] Haowei Lin, Shanda Li, Haotian Ye, Yiming Yang, Stefano Ermon, Yitao Liang, and Jianzhu Ma. Tfg-flow: Training-free guidance in multimodal generative flow. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[36] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*. ACM, 2020. 2, 7

[37] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2

[38] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 1

[39] Xingchao Liu, Lemeng Wu, Shujian Zhang, Chengyue Gong, Wei Ping, and Qiang Liu. Flowgrad: Controlling the output of generative odes with gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24335–24344, 2023. 2, 8

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6), 2015. 3

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 4

[42] Junzhe Lu, Jing Lin, Hongkun Dou, Ailing Zeng, Yue Deng, Yulun Zhang, and Haoqian Wang. Dposer: Diffusion model as robust 3d human pose prior. *arXiv preprint arXiv:2312.05541*, 2023. 2

[43] Junzhe Lu, Jing Lin, Hongkun Dou, Ailing Zeng, Yue Deng, Xian Liu, Zhongang Cai, Lei Yang, Yulun Zhang, Haoqian Wang, et al. Dposer-x: Diffusion model as robust 3d whole-body human pose prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9988–9997, 2025. 1, 2, 6, 7, 8, 17, 18, 19, 20, 21

[44] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 6, 7, 15

[45] Benjamin Kurt Miller, Ricky TQ Chen, Anuroop Sriram, and Brandon M Wood. Flowmm: Generating materials with riemannian flow matching. In *International Conference on Machine Learning*, pages 35664–35686. PMLR, 2024. 2

[46] Alex Morehead and Jianlin Cheng. Flowdock: Geometric flow matching for generative protein-ligand docking and affinity prediction. In *Intelligent Systems for Molecular Biology (ISMB)*, 2025. 2

[47] Julien Munier. Steepest descent method on a riemannian manifold: the convex case. *Balkan Journal of Geometry & Its Applications*, 12(2), 2007. 12

[48] Wenzhe Ouyang, Qi Ye, Jinghua Wang, Zenglin Xu, and Jiming Chen. Rfmpose: Generative category-level object pose estimation via riemannian flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2

[49] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 2, 6, 7, 8, 16

[50] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11488–11499, 2021. 1, 2, 6, 7

[51] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Humaniflow: Ancestor-conditioned normalising flows on so (3) manifolds for human pose and shape distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4779–4789, 2023. 2

[52] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[53] Mingyi Shi, Sebastian Starke, Yuting Ye, Taku Komura, and Jungdam Won. Phasemp: Robust 3d pose estimation via phase-conditioned human motion prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14725–14737, 2023. 2

[54] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020. 1

[55] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[56] Garvita Tiwari, Dimitrije Antic, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *European Conference on Computer Vision*, pages 572–589, 2022. 1, 2, 6, 7

[57] J. Townsend, N. Koep, and S. Weichwald. PyManopt: a Python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5, 2016. 12

[58] Nilesh Tripuraneni, Nicolas Flammarion, Francis Bach, and Michael I Jordan. Averaging stochastic gradient descent on riemannian manifolds. In *Conference On Learning Theory*, pages 650–687. PMLR, 2018. 12

[59] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018. 6, 8, 21

[60] Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. Training free guided flow-matching with optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 8

[61] Xi Wang, Xiaoyi Wang, and Victor Solo. Stochastic kinematic optimal control on so (3). *arXiv preprint arXiv:2412.08124*, 2024. 4

[62] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021. 16

[63] Zifan Wang, Alice Harting, Matthieu Barreau, Michael M Zavlanos, and Karl H Johansson. Source-guided flow matching. *arXiv preprint arXiv:2508.14807*, 2025. 2

[64] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022. 8, 16

[65] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019. 2

[66] Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023. 2

[67] Hua Yu, Weiming Liu, Jiapeng Bai, Xu Gui, Yaqing Hou, YewSoon Ong, and Qiang Zhang. Towards efficient and diverse generative model for unconditional human motion synthesis. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2535–2544, 2024. 2

[68] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023. 2

[69] Zhengdi Yu, Simone Foti, Linguang Zhang, Amy Zhao, Cem Keskin, Stefanos Zafeiriou, and Tolga Birdal. Geometric neural distance fields for learning human motion priors. *arXiv preprint arXiv:2509.09667*, 2025. 2, 3

[70] Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638. PMLR, 2016. 12

[71] Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Ying Nian Wu, and Oscar Leong. Flow priors for linear inverse problems via iterative corrupted trajectory matching. *Advances in Neural Information Processing Systems*, 37:57389–57417, 2024. 2

[72] Zhixuan Zhou, Tingting Dan, and Guorong Wu. Brainflow: A holistic pathway of dynamic neural system on manifold. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2

PoseD-Flow: Versatile and Guided Flow Matching Model of Human Pose

Supplementary Material

A. Intuition

We believe that an intuitive understanding of our contributions is essential for appreciating their impact and value. A human skeleton / pose is best explained not by a set of real numbers, but by subset of numbers in a # joints $\times$ dim(parameterization)-dimensional space. This subset exactly corresponds to the *product manifold of rotations*, parameterized as #joints amount of rotations. This is not a mere technicality, but a fundamental way to incorporate the structure in the problem into the solution / method. In our case, the inclusion of geometry transforms the model from a purely statistical generator into a structure-aware dynamical system. By operating on the manifold of articulated rotations, the flow respects the intrinsic constraints of human pose, ensuring that both learning and inference unfold along physically and mathematically meaningful trajectories (through curvature and covariance). This is the implicit bias toward realistic, stable, and data-consistent poses. Riemannian geometry is the key enabler of this, which we will review next.

B. Geometry of Human Poses

We follow NRDF [24] as well as [4, 5, 11] to provide a detailed treatise regarding the product Riemannian manifold on which an articulated human pose lives.

Riemannian geometry. We define an m-dimensional *Riemannian manifold*, embedded in an ambient Euclidean space $\mathcal{X} = \mathbb{R}^d$ and endowed with a *Riemannian metric* $G \triangleq (G_x)_{x\in\mathcal{M}}$ to be a smooth curved space $(\mathcal{M}, G)$. A vector $v \in \mathcal{X}$ is said to be *tangent* to $\mathcal{M}$ at x *iff* there exists a smooth curve $\gamma : [0, 1] \to \mathcal{M}$ s.t. $\gamma(0) = x$ and $\dot\gamma(0) = v$. The velocities of all such curves through x form the *tangent space* $\mathcal{T}_x\mathcal{M} = \{\dot\gamma(0) \,|\, \gamma : \mathbb{R} \to \mathcal{M}$ is smooth around 0 and $\gamma(0) = x\}$, whose union is called the *tangent bundle*: $\mathcal{TM} = \bigcup_x \mathcal{T}_x\mathcal{M} = \{(x, v) \mid x \in \mathcal{M}, v \in \mathcal{T}_x\mathcal{M}\}$. The Riemannian metric $G(\cdot)$ equips each point x with an inner product in the tangent space $\mathcal{T}_x\mathcal{M}$, $\langle \mathbf{u}, v\rangle_x = \mathbf{u}^T G_x v$. We will also work with a product of K manifolds, $\mathcal{M}_{1:K} := \mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_K$, For identical manifolds, *i.e.* $\mathcal{M}_i \equiv \mathcal{M}_j$, we recover the *power manifold*, $\mathcal{M}^K := \mathcal{M}_{1:K}$, whose tangent bundle admits the *natural isomorphism*, $\mathcal{TM}^K \simeq (\mathcal{TM} \times \cdots \times \mathcal{TM})$. We now define the operators required for our algorithm.

Definition 7 (Riemannian Gradient)**.** *For a smooth function $f : \mathcal{M} \to \mathbb{R}$ and $\forall (x, v) \in \mathcal{TM}$, we define the* Riemannian gradient *of f as the unique vector field* grad f *satisfying [9]:*

$$\mathrm{D}f(x)[v] = \langle v, \operatorname{grad} f(x)\rangle_x \tag{B.1}$$

where $\mathrm{D}f(x)[v]$ is the derivation of f by v. It can further be shown that an expression for grad f *can be obtained through the projection of the* Euclidean *gradient orthogonally onto the tangent space*

$$\operatorname{grad} f(x) = \nabla f(x)_{\|} = \Pi_x\big(\nabla f(x)\big). \tag{B.2}$$

where $\Pi_x : \mathcal{X} \to \mathcal{T}_x\mathcal{M} \subseteq \mathcal{X}$ is an orthogonal projector with respect to $\langle\cdot,\cdot\rangle_x$.

In most packages such as ManOpt [57], Eq. (B.2) is known as the *egrad2rgrad*.

Definition 8 (Riemannian Optimization)**.** *We consider gradient descent to solve the problems of* $\min_{x\in\mathcal{M}} f(x)$*. For a local minimizer or a* stationary point $x^\star$ *of f, the Riemannian gradient vanishes* grad $f(x^\star) = 0$ *enabling a simple algorithm,* Riemannian Gradient Descent *(RGD):*

$$x_{k+1} = R_{x_k}(-\tau_k \operatorname{grad} f(x_k)) \tag{B.3}$$

where τ_k is the step size at iteration k and R_{x_k} is the retraction *usually chosen related to the exponential map. Note that both RGD and its stochastic variant [7] are practically convergent [7, 9, 47, 58, 70]. Though, only in rare cases is τ_k analytically computable. Therefore, most minimizers use either Armijo or Wolfe line-search [1].*

SO(3). We now explaing the space of a single joint. A rotation R is an element of the SO(3) group:

Definition 9 (SO(3))**.** *Rotations are elements of the special orthogonal group defined as:*

$$\mathrm{SO}(3) = \left\{R \in \mathbb{R}^{3\times3} : R^\top R = \mathbf{I}, \det(R) = 1\right\}. \tag{B.4}$$

Definition 10 ($\mathcal{T}$SO(3))**.** *Differentiating the constraint gives the tangent space at identity (Lie algebra):*

$$\mathcal{T}\mathrm{SO}(3) := \mathfrak{so}(3) = \left\{\Omega \in \mathbb{R}^{3\times3} \mid \Omega^\top = -\Omega\right\}. \tag{B.5}$$

Every $\Omega \in \mathfrak{so}(3)$ can be written using the hat operator:

$$\Omega = \hat\omega = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad \omega \in \mathbb{R}^3 \tag{B.6}$$

The inverse map (vee) satisfies $(\hat\omega)^\vee = \omega$. Standard choice (from embedding in $\mathbb{R}^{3\times3}$) gives the notion of an inner product:

Definition 11 (Bi-invariant Riemannian metric & distance)**.** SO(3) *is a compact Lie group with bi-invariant metric*

$$\langle\Omega_1, \Omega_2\rangle_R = \frac{1}{2}\operatorname{Tr}\left(\Omega_1^\top \Omega_2\right) = \omega_1^\top \omega_2. \tag{B.7}$$

Because the metric is bi-invariant:

$$d(R_1, R_2) = \left\| \mathrm{Log}_{R_1}(R_2)^\vee \right\|_2 = \cos^{-1}\left(\frac{\mathrm{Tr}\left(R_1^\top . R_2\right) - 1}{2} \right).$$

This equals the rotation angle between them.

Definition 12 (Exp / Log maps)**.** *At identity the exponential maps is given by the Rodrigues formula and the logarithmic map follows from its inverse:*

$$\mathrm{Exp}_I(\hat{\omega}) = \exp(\hat{\omega}) = I + \frac{\sin\theta}{\theta}\hat{\omega} + \frac{1-\cos\theta}{\theta^2}\hat{\omega}^2 \quad \text{(B.8)}$$

$$\mathrm{Log}_I(R) = \frac{\theta}{2\sin\theta}\left(R - R^\top\right),\ \theta = \cos^{-1}\left(\frac{\mathrm{Tr}(R)-1}{2}\right),$$

where $\theta = \|\omega\|$. At a general point R :

$$\mathrm{Exp}_R(\hat{\omega}) = R\exp(\hat{\omega}) \quad \text{(B.9)}$$

$$\mathrm{Log}_R(Q) = \mathrm{Log}_I\left(R^\top Q\right) = \log\left(R^\top Q\right) \quad \text{(B.10)}$$

Manifold of human poses ($\mathcal{M} := \mathrm{SO}(3)^K$). We parameterize the pose of a 3D articulated body composed of K joints, $x := \{R_i \in \mathrm{SO}(3)\}_{i=1}^K$, on the power manifold of rotations $\mathcal{M} := \mathrm{SO}(3)^K = \mathrm{SO}(3) \times \cdots \times \mathrm{SO}(3)$.

Definition 13 (Geometry of 3D articulated poses)**.** $\mathrm{SO}(3)^K$ *turns into a Riemannian manifold* $(\mathrm{SO}(3)^K, G^K)$ *when endowed with the* L_p product metric $d_{\mathrm{SO}(3)^K} : \mathrm{SO}(3)^K \times \mathrm{SO}(3)^K \to \mathbb{R}$*:*

$$d_{\mathcal{M}}(x, x') = \|d(R_1, R_1'), d(R_2, R_2'), \ldots, d(R_K, R_K')\|_p,$$

where $R \in x \in \mathrm{SO}(3)^K$ and $R' \in x' \in \mathrm{SO}(3)^K$. In this work, we use $p = 1$. The natural isomorphism further allows us to write its exponential map $\mathrm{Exp}_x : \mathcal{T}\mathrm{SO}(3)^K \to \mathrm{SO}(3)^K$ component-wise: $\mathrm{Exp}_x = \left(\mathrm{Exp}_{R_1}, \mathrm{Exp}_{R_2}, \ldots, \mathrm{Exp}_{R_K}\right)$. Akin to this, is the logarithmic map, Log_x. Since the tangent spaces and therefore Π_x are replicas, the gradient of a smooth function $f : \mathrm{SO}(3)^K \to \mathbb{R}$ w.r.t. x is also the Cartesian product of the individual gradients:

$$\mathrm{grad}_x f(x) = \left(\mathrm{grad}_{R_1} f(x), \ldots, \mathrm{grad}_{R_K} f(x)\right). \quad \text{(B.11)}$$

C. Omitted Proofs

We now provide the proofs that are excluded from the main paper. Whenever necessary, we will recall the definitions / theorems from the main paper for the sake of a self-contained exposition.

Proof that $\mathcal{L}_{\mathrm{traj}}$ promotes small rotations. Let $R \in \mathrm{SO}(3)$ represent a rotation by angle θ about some axis. By Rodrigues' rotation formula:

$$\mathrm{tr}(R) = 1 + 2\cos\theta \quad \text{(C.1)}$$

$$\min\ 3 - \mathrm{tr}(R) = \min\ 2(1-\cos\theta) \quad \text{(C.2)}$$

$$\arg\min_\theta\ 2(1-\cos\theta) = 0 \quad \text{(C.3)}$$

Hence minimizing $\mathcal{L}_{\mathrm{traj}}$ prefers small angle solutions. □

Theorem 5 (Tangent denoiser)**.** *For the data distribution $x_1 \sim p_1$ on $\mathcal{M}$, define at any $x \in \mathcal{M}$ the tangent random variable $\xi_x := \mathrm{Log}_x(x_1) \in \mathcal{T}_x\mathcal{M}$. The tangent denoiser μ at time t given by:*

$$\mu(x) := \mu_{1|t}(x) = \mathbb{E}\left[\xi_x \mid x(t) = x\right], \quad \text{(C.4)}$$

is the unique minimizer of the Riemannian flow matching loss:

$$\mathcal{L}(v) := \mathbb{E}\left[\|\xi - v\|_{g_x}^2 \mid x_t = x\right]. \quad \text{(C.5)}$$

Proof of Thm. 5. The proof follows from the observation that in an inner-product space the best single-vector predictor (in mean squared error) of a random vector is its conditional expectation. This idea has been key to developing RFM [12]. We start by expanding the RFM loss (in the tangent space) as:

$$\mathcal{L}(v) = \mathbb{E}\left[\|\xi - v\|^2 \mid x\right] = \mathbb{E}[\langle \xi - v, \xi - v\rangle \mid x] \quad \text{(C.6)}$$

$$= \mathbb{E}\left[\|\xi\|^2 \mid x\right] - 2\langle \mathbb{E}[\xi \mid x], v\rangle + \|v\|^2. \quad \text{(C.7)}$$

Observing that the first term does not depend on v, we write:

$$\arg\min_v \mathcal{L}(v) = \arg\min_v \|v\|^2 - 2\langle \mu, v\rangle = \|v - \mu\|^2. \quad \text{(C.8)}$$

The unique minimum is obtained at $v = \mu$. □

Theorem 6 (Covariant derivative & covariance)**.** *The covariant derivative $(\nabla\mu)_v(x) : \mathcal{T}_x\mathcal{M} \to \mathcal{T}_x\mathcal{M}$ of the tangent denoiser is given by:*

$$(\nabla\mu)_v(x) = A_x[C(x)v] + R_x[v], \quad \text{(C.9)}$$

where $A_x[v]$ is a linear operator, $C(x) := \mathrm{C}_{1|t}(x)$ is a self-adjoint, positive semidefinite linear map $\mathcal{T}_x\mathcal{M} \to \mathcal{T}_x\mathcal{M}$, representing the covariance under the conditional distribution, and $R_x[v]$ is a Riemannian remainder*:*

$$C(x) = \mathbb{E}\left[\left(\xi_x - \mu_{1|t}\right) \otimes \left(\xi_x - \mu_{1|t}\right) \mid x(t) = x\right] \quad \text{(C.10)}$$

$$R_x[v] = \mathbb{E}\left[\nabla_v^{(x)}\xi \mid x\right]. \quad \text{(C.11)}$$

As $t \to 1$, $C(x)$ approaches the local data covariance under variance scheduling of geodesic kernels: $\sigma \to 0$.

Proof of Thm. 6. By definition of the denoiser, we have:

$$\mu(x) = \int_{\mathcal{T}_x\mathcal{M}} \xi p(\xi \mid x) d\xi. \tag{C.12}$$

We then take the covariant derivative:

$$(\nabla_v \mu)(x) = \int \nabla_v^{(x)}(\xi) p(\xi \mid x) d\xi + \int \xi \nabla_v^{(x)} p(\xi \mid x) d\xi.$$

The gradient in the first term is the *base-point derivative* whereas the second gradient measures how the posterior weight changes when we move x. Using the score identity $\nabla_v p = p \nabla_v \log p$, we re-write the second integral as:

$$\int \xi \nabla_v p(\xi \mid x) d\xi = \int \xi p(\xi \mid x) \nabla_v \log p(\xi \mid x) d\xi \tag{C.13}$$

$$= \mathbb{E}\left[\xi \nabla_v \log p(\xi \mid x) \mid x\right]. \tag{C.14}$$

By the definition of expectation, we have:

$$\mathbb{E}\left[\nabla_v^{(x)} \xi \mid x\right] := \int \nabla_v^{(x)}(\xi) p(\xi \mid x) d\xi. \tag{C.15}$$

Combining this with Eq. (C.14) and plugging into the definition of the covariant derivative, we write:

$$(\nabla_v \mu)(x) = \mathbb{E}\left[\nabla_v^{(x)} \xi \mid x\right] + \mathbb{E}\left[\xi \nabla_v \log p(\xi \mid x) \mid x\right].$$

We now make the substitution $\xi = \mu(x) + (\xi - \mu(x))$ into Eq. (C.14) and write:

$$\mathbb{E}\left[\xi \nabla_v \log p(\xi \mid x) \mid x\right] = \mathbb{E}\left[(\xi - \mu) \nabla_v \log p \mid x\right] \tag{C.16}$$

since $\mathbb{E}\left[\nabla_v \log p(\xi \mid x) \mid x\right] = 0$ (score is 0-mean). This yields:

$$(\nabla_v \mu)(x) = \mathbb{E}\left[\nabla_v^{(x)} \xi \mid x\right] + \mathbb{E}\left[(\xi - \mu) \nabla_v \log p \mid x\right].$$

Next, we decompose the score function into linear (best-fit) and residual, non-linear (orthogonal) terms as follows:

$$\nabla_v \log p(\xi \mid x) = \langle A_x[v], (\xi - \mu) \rangle + s(\xi, x, v), \tag{C.17}$$

where $A_x : \mathcal{T}_x\mathcal{M} \to \mathcal{T}_x\mathcal{M}$ is a linear map [1], $A_x[v]$ a tangent vector, and $\mathbb{E}[(\xi - \mu) s(\xi, x, v) \mid x] = 0$ from orthogonality. This leads to:

$$\mathbb{E}\left[(\xi - \mu) \nabla_v \log p \mid x\right] = \mathbb{E}\left[(\xi - \mu) \langle A_x[v], (\xi - \mu) \rangle \mid x\right] + \underbrace{\mathbb{E}\left[(\xi - \mu) s(\xi, x, v) \mid x\right]}_{=0} \tag{C.18}$$

$$= (A_x \circ C(x))[v]. \tag{C.19}$$

[1] best-fit linear coefficient in the least-squares sense

The last equality follows from the definition of Riemannian covariance $C(x)$ in Eq. (C.10). We now collect the terms and re-write the covariant derivative:

$$(\nabla_v \mu)(x) = \underbrace{\mathbb{E}\left[\nabla_v^{(x)} \xi \mid x\right]}_{(Eq.\ (C.15))} + \underbrace{(A_x \circ C(x))[v]}_{(Eq.\ (C.19))}. \tag{C.20}$$

Calling the first term $R_x[v] := \mathbb{E}\left[\nabla_v^{(x)} \xi \mid x\right]$:

$$(\nabla \mu)(x)[v] = A_x[v] \circ C(x) + R_x[v]. \tag{C.21}$$

We can also write this in operator-style revealing:

$$(\nabla \mu)(x) = A_x \circ C(x) + R_x. \tag{C.22}$$

□

Corollary 2. *The covariant derivative of the marginal velocity field satisfies ($\circ$ denotes composition):*

$$\nabla u_t(x) = \frac{1}{1-t}\left(C_{1|t}(x) \circ A_x + R_x\right). \tag{C.23}$$

This is the drift of the adjoint ODE, determining the Riemannian adjoint, as we explain next.

Proof of Corollary 2. Pointwise, the velocity of the geodesic interpolation equals the tangent vector from the current position to the final endpoint, scaled by the inverse remaining time:

$$u_t(x) := \mathbb{E}\left[\dot{X}_t \mid x_t = x\right] \tag{C.24}$$

$$= \mathbb{E}\left[\frac{1}{1-t} \mathrm{Log}_x(x_1) \,\middle|\, x_t = x\right] \tag{C.25}$$

$$= \frac{1}{1-t} \mathbb{E}\left[\xi \mid x_t = x\right] \tag{C.26}$$

$$= \frac{1}{1-t} \mu(x). \tag{C.27}$$

Plugging Eq. (C.9) in Thm. 6 into Eq. (C.27):

$$\nabla u_t(x) = \frac{1}{1-t}(\nabla \mu)(x) = \frac{1}{1-t}\left(C_{1|t}(x) \circ A_x + R_x\right).$$

□

While the next theorem is a standard result in Riemannian geometry, we nevertheless prove it, showing backpropagation through the continuous Riemannian flow is pulling the gradient at $t = 1$ back to $t = 0$ via the Riemannian adjoint of the flow map.

Theorem 7 (Riemannian adjoint)**.** *Let $\Psi : \mathcal{M} \to \mathcal{M}$ denote the flow such that $x_1 = \Psi(x_0)$. Then the Riemannian gradients at the start and end points are related by the* Riemannian adjoint *(pullback map) $D_{x_0}\Psi(x_0)^*$:*

$$\mathrm{grad}_{x_0} \mathcal{L}(x_1) = D\Psi(x_0)^* \left[\mathrm{grad}_{x_1} \mathcal{L}(x_1)\right]. \tag{C.28}$$

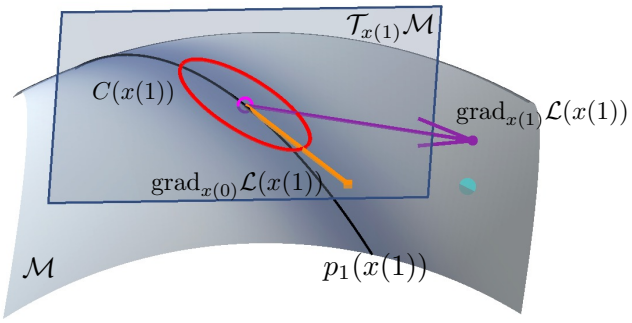


Figure C.1. Additional illustration of the implicit bias in differentiating through the solver. A infinitesimally small change in the source point causes the projection the intrinsic gradient at the end point onto the flow directions up to a curvature term.

Proof of Thm. 7: Relating Riemannian gradients. Let $\Psi : \mathcal{M} \to \mathcal{M}$ denote the flow such that $x_1 = \Psi(x_0)$. Let us write the composite objective as $F(x_0) := \mathcal{L} \circ \Psi(x_0) = \mathcal{L}(\Psi(x_0))$. By the defining property of the Riemannian gradient, for all $\xi_0 \in T_{x_0}\mathcal{M}$, we have

$$dF_{x_0}(\xi_0) = \left\langle \text{grad}_{x_0} F,\ \xi_0 \right\rangle_{g_{x_0}}, \tag{C.29}$$

where g is the Riemannian metric. Applying the chain rule to F yields:

$$dF_{x_0} = d\mathcal{L}_{x_1} \circ D\Psi(x_0), \tag{C.30}$$

where $x_1 = \Psi(x_0)$. Evaluating this on ξ_0 yields

$$\begin{aligned} dF_{x_0}(\xi_0) &= d\mathcal{L}_{x_1}(D\Psi(x_0)[\xi_0]) && \text{(C.31)} \\ &= \left\langle \text{grad}_{x_1} \mathcal{L},\ D\Psi(x_0)[\xi_0] \right\rangle_{g_{x_1}}, && \text{(C.32)} \end{aligned}$$

where the last equality again uses the definition of the gradient. The Riemannian adjoint $D\Psi(x_0)^*$ is defined implicitly by

$$\langle v_1,\ D\Psi(x_0)[\xi_0] \rangle_{g_{x_1}} = \langle D\Psi(x_0)^*[v_1],\ \xi_0 \rangle_{g_{x_0}}, \tag{C.33}$$

$\forall\, v_1 \in T_{x_1}\mathcal{M},\ \xi_0 \in T_{x_0}\mathcal{M}$. Applying this definition with $v_1 = \text{grad}_{x_1} \mathcal{L}$ converts Eq. (C.31) into

$$dF_{x_0}(\xi_0) = \left\langle D\Psi(x_0)^*[\text{grad}_{x_1} \mathcal{L}],\ \xi_0 \right\rangle_{g_{x_0}}. \tag{C.34}$$

Comparing this with Eq. (C.29), and using the non-degeneracy of g_{x_0}, we identify the unique tangent vector that reproduces the linear functional dF_{x_0}, namely

$$\text{grad}_{x_0} F = D\Psi(x_0)^*[\text{grad}_{x_1} \mathcal{L}(x_1)], \tag{C.35}$$

which is precisely Eq. (C.28). This completes the proof. □

Theorem 8 (Implicit bias in endpoint update)**.** *Consider a small optimization step updating the optimized source variable:* $x_0 \longrightarrow \text{Exp}_{x_0}\left(-\eta\, \text{grad}_{x_0} \mathcal{L}\left(x_1\right)\right)$ *where* $\text{grad}_{x_0} \mathcal{L}\left(x_1\right)$ *is given in Eq.* (C.28)*. As* $\eta \to 0$ *(infinitesimal change), the variation of the end-point* $x_1 = \Psi(x_0)$*, denoted* δx_1 *reads:*

$$\delta x(1) = -\eta \underbrace{\left(D_{x_0}\Psi\left(x_0\right) D_{x_0}\Psi\left(x_0\right)^*\right)}_{\mathcal{K} \text{ self-adjoint, PSD on } \mathcal{T}_{x_1}\mathcal{M}} \text{grad}_{x(1)} \mathcal{L}(x(1)),$$

where $\mathcal{K} = D_{x_0}\Psi\left(x_0\right) D_{x_0}\Psi\left(x_0\right)^*$ *resembles a local covariance on the endpoint gradient, explaining why the update is biased toward directions of high density.*

Proof of Thm. 8. Assume a small gradient step:

$$x_0 \longrightarrow x_0' = \text{Exp}_{x_0}\left(-\eta\, \text{grad}_{x_0} \mathcal{L}\left(x_1\right)\right) \tag{C.36}$$

For infinitesimal step size $\eta \to 0$, the first-order variation is

$$\begin{aligned} \delta x_0 &= -\eta\, \text{grad}_{x_0} \mathcal{L}\left(x_1\right) \in \mathcal{T}_{x_0}\mathcal{M} && \text{(C.37)} \\ &= -\eta d\Psi\left(x_0\right)^* \left[\text{grad}_{x_1} \mathcal{L}\left(x_1\right)\right], && \text{(C.38)} \end{aligned}$$

where the last equality follows from plugging Eq. (C.28). The end-point is affected through the flow as: $x_1 = \Psi(x_0)$ and $x_1' = \Psi(x_0 + \delta x_0)$. Assuming an infinitesimally small change, this yields:

$$\delta x_1 := x_1' - x_1 = D\Psi\left(x_0\right)\left[\delta x_0\right] \in \mathcal{T}_{x_1}\mathcal{M}. \tag{C.39}$$

Plugging Eq. (C.37) into Eq. (C.39) (substituting δx_0), gives the tangent vector at x_1 describing how the endpoint moves in response to the gradient step at x_0:

$$\delta x(1) = -\eta \underbrace{\left(D_{x_0}\Psi\left(x_0\right) D_{x_0}\Psi\left(x_0\right)^*\right)}_{\mathcal{K} \text{ self-adjoint, PSD on } \mathcal{T}_{x_1}\mathcal{M}} \text{grad}_{x(1)} \mathcal{L}(x(1)),$$

where $\mathcal{K}$ is a projection onto the reachable subspace of the endpoint tangent space. When the flow generates the data distribution, this operator becomes the local covariance of the data manifold. This completes the proof. □

D. Implementation Details

In this section, we detail the implementation setup used in our experiments.

Training. We train our models using the training split of AMASS [44]. We preprocess the dataset by trimming the first and last 10% of all sequences and sampling them at 30 Hz, resulting in approximately 18 million poses. We train our model for 50000 steps, with a runtime of 8 hours on a single NVIDIA A30 GPU.

Inverse problems. Here, we expand on the implementation details of our optimization algorithm. For initialization,

Table D.1. Hyperparameters for various inverse tasks

Task	LR	Solver	NFE	Num. of Iters	Blending α	Loss weights
Pose Completion	0.1	Euler	100	200 / 300	0.25	$\lambda_{\text{data}} = 1,\ \lambda_{\text{traj}} = 1e^{-3}$
Motion Denoising	0.25	Euler	100	300 / 400	0.75	$\lambda_{\text{data}} = 1,\ \lambda_{\text{temp}} = 1e^{-1}/1e^{-2},\ \lambda_{\text{traj}} = 1e^{-5}$
Human Mesh Recovery	0.1	Euler	100	400	0.75	$\lambda_{\text{data}} = 1,\ \lambda_{\alpha} = \lambda_{\beta} = \lambda_{\text{traj}} = 50$

we use the linear blend strategy from D-Flow [3] for better convergence. Specifically, we initialize x_0 with a blend of a sample from the source distribution p and the backward ODE solution of x^{obs} from $t = 1$ to $t = 0$:

$$x_0 = \sqrt{\alpha} \cdot x_0^{\text{obs}} + \sqrt{1-\alpha} \cdot x_0' \quad x_0' \sim p, \tag{D.1}$$

where

$$x_0^{\text{obs}} = \int_1^0 v_w(x^{\text{obs}}, t)\, dt \tag{D.2}$$

x^{obs} varies by task. For pose completion, it corresponds to the partially observed pose, with the occluded joints filled with the mean pose. For human mesh recovery, x^{obs} is given by the output of CLIFF [33]. Using this initialization, the optimization proceeds as described in Alg. 2. The hyperparameters used in all experiments are presented in Tab. D.1.

For pose completion, we use 300 iterations for the arms occluded case, and 200 iterations for all other cases. For motion denoising, we run 400 iterations when denoising the HPS [23] dataset and when the noise standard deviation is set to 0.1, along with setting $\lambda_{\text{temp}} = 1e^{-1}$ for the latter.

For pose completion, we evaluate on the AMASS test split with a sampling rate of 10. We generate 10 hypotheses for each partial ground truth. For motion denoising, we use the HumanEva split from AMASS and the HPS [23] dataset. Both datasets undergo the same preprocessing: we sample motions at 30 Hz and segment them into 60-frame chunks. This yields 190 sequences from HumanEva and 6,359 from HPS. To limit HPS size, we randomly select 50 sequences per subject with a seed of 42, resulting in a final set of 350 sequences. For human mesh recovery, we use the EHF [49] dataset, which contains 100 images. We extract 2D keypoints using ViTPose-H [64]. When initializing with CLIFF [33], we predict the poses using the "hr48-PA53.7_MJE91.4_MVE110.0_agora_val.pt" checkpoint.

Additional Details on the Teaser Figure. Fig. 1 depicts how the noise distribution $p(x_0)$ flows towards the data distribution $p(x_1)$ with PoseRFM. We project the distributions onto a 2D surface embedded in 3D space. While the extrinsic shape of this surface is arbitrarily chosen for illustrative purposes, the distributions mapped onto it are rigorously obtained by dimensionally reducing real flow trajectories. In particular, we randomly sample $10,000$ poses on $\mathcal{M} = \text{SO}(3)^K$: $\{x_0^{(i)}\}_{i=1}^{10\text{k}} \sim p(x_0)$ and propagate them using the learned PoseRFM, $v_w(x, t)$. We record samples positions on the manifold $\{x_t^{(i)}\}_{i=1}^{10\text{k}}$ for multiple time-steps t. Then we collect all these samples and learn a transformation to a common two-dimensional manifold using PaCMAP [62] with its default hyperparameter selection, but replacing the available distances with the correct geodesic distance, $d_{\text{SO}(3)^K}$. The learned transformation is used to project the samples $(\{x_t^{(i)}\}_{i=1}^{10\text{k}})$ at $t = [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$, which are then used to estimate the distributions via kernel density estimation. The same transformation is also used to draw the optimization trajectories of Riemannian D-Flow on $p(x_0)$ and $p(x_1)$.

Further details. **(i) How is β obtained?** For IK from images, we use CLIFF [33] to predict human pose and shape β, and then refine these predictions via optimization. For all other tasks, we fix $\beta = 0$ and optimize for the pose. **(ii) FID for pose.** As in NRDF [24], we compute the FID using the Frechet distance between the 3D positions of the generated and real body joints, both obtained through SMPL model. **(iii) Dimension of u_t.** Our flow field lives in the tangent space of K articulated joints. Hence, $\dim(u_t) = K \times 3$.

E. Additional Results

Ablation studies. First, we conduct an ablation study on the sampling strategy for unconditional pose generation. We evaluate the impact of different ODE solvers, retraction methods, and number of function evaluations (NFE), aiming to balance generation quality, diversity, and runtime. The results are summarized in Tab. E.1, with the best configuration highlighted.

Using the midpoint ODE solver increases sample diversity with only a small runtime overhead, while integrating for 1000 steps offers no meaningful gains. For retraction, projecting back to the manifold after every step is prohibitively slow and provides no benefit, making it impractical. Retraction only at the final step or using the Exp map are both viable; with the former being faster, while the latter yielding higher diversity.

We perform the same analysis for the pose completion inverse task. Guided by the previous findings, we exclude per-step retraction and the 1000 NFE setting. We evaluate accuracy, diversity, and runtime for the remaining choices, as shown in Tab. E.2.

In this setting, using the midpoint solver becomes more

Table E.1. Ablation results on sampling strategies for pose generation

Solver	Retraction	100 steps				1000 steps			
		FID ↓	APD ↑	d_{NN} ↓	Time ↓	FID ↓	APD ↑	d_{NN} ↓	Time ↓
Euler	Last step	0.013	14.984	0.066	0.614	0.013	15.401	0.069	2.028
	Every step	0.014	15.256	0.068	28.406	0.014	15.431	0.069	284.602
	Exp map	0.014	15.252	0.068	1.294	0.014	15.431	0.069	10.734
Midpoint	Last step	0.014	15.447	0.070	0.773	0.014	15.451	0.070	3.570
	Every step	0.014	15.448	0.070	28.401	0.014	15.451	0.070	284.082
	Exp map	0.015	15.786	0.071	1.383	0.014	15.483	0.070	12.350

Table E.2. Ablation results on sampling strategies for pose completion.

Solver	Retraction	Occ. left leg			Occ. legs		
		MPVPE ↓	APD ↑	Time ↓	MPVPE ↓	APD ↑	Time ↓
Euler	Last step	83.81	6.02	100.71	95.00	7.23	106.10
	Exp map	82.89	5.94	483.72	94.20	7.18	475.08
Midpoint	Last step	83.85	6.12	140.55	95.82	7.37	146.80
	Exp map	84.01	6.12	523.92	95.90	7.38	517.94

Table E.3. Runtime comparison across various tasks. Timings are measured in seconds.

Task	Batch Size	DPoser [43]	PoseFM	PoseRFM
Pose Generation	500	1.055	0.118	0.773
Pose Completion	500	0.82	20.91	100.71
Motion Denoising	60	5.07	32.49	80.85
HMR (IK)	100	17.05	47.90	120.13

expensive as it requires two passes through the model per step, unlike Euler's single pass. Likewise, using the Exp map significantly slows optimization with only modest accuracy gains. We adopt the highlighted configuration for this and all other inverse problems.

Runtime comparison. We benchmark the runtime for each task over 10 runs and report the median in Tab. E.3. All measurements were performed on a NVIDIA A100 GPU.

The ODE formulation of flow models enables fast pose generation relative to SotA diffusion methods. Unfortunately, the benefits stop there: PoseRFM remains orders of magnitude slower than diffusion-based approaches in downstream tasks. Each iteration of Riemannian D-Flow requires backpropagating all the way to the source point, which is substantially slower than the single-step denoising used in DPoser [43]. In addition, Riemannian D-Flow requires differentiating through geometric components such as geodesics and Exp map, introducing further computational overhead. Developing an optimized version of our algorithm is left for future work.

Convergence analysis. Thus far, we have focused on the final output of our inverse algorithm. In this section, we will examine the behavior of x_1 throughout optimization.

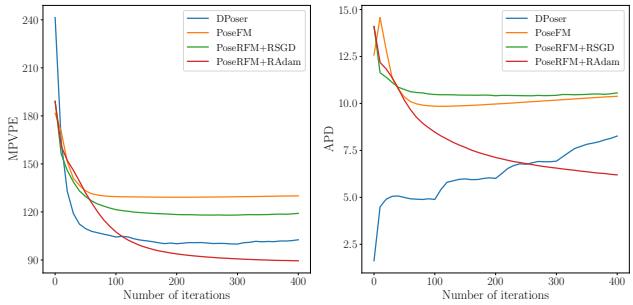
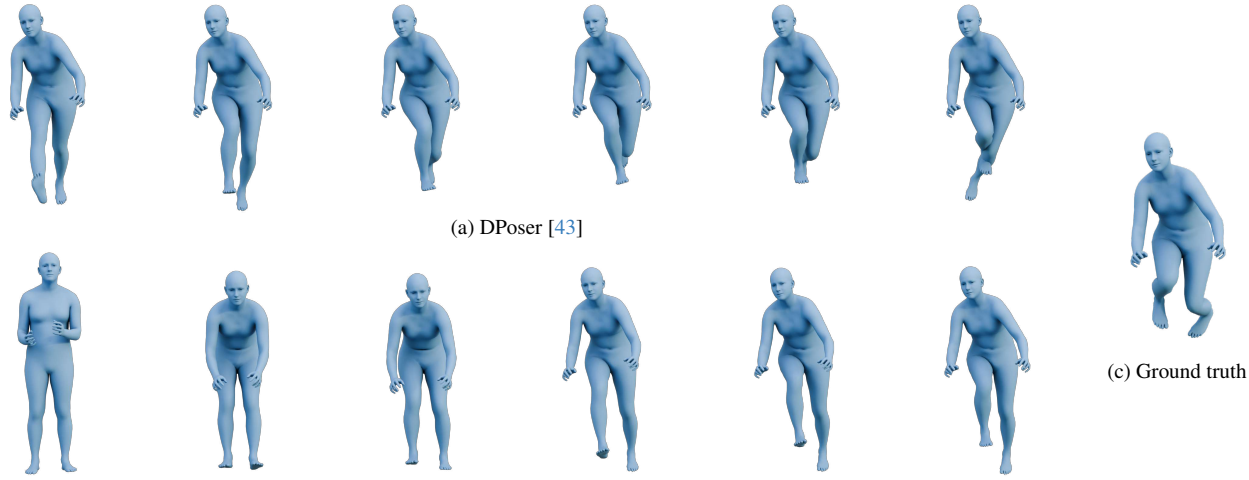


Figure E.1. Convergence comparison for various methods on pose completion with legs occluded.

We first plot the convergence curves in Fig. E.1, showing how accuracy and diversity evolve over the optimization steps. For this analysis, we use the pose completion task with legs occluded, and compare four methods: the diffusion baseline DPoser [43], PoseFM, PoseRFM with RiemannianSGD, and PoseRFM with RiemannianAdam.

For MPVPE, both PoseFM and PoseRFM with RSGD stop converging after a few iterations, leading to their poor accuracy. Meanwhile, PoseRFM with RAdam continues to improve up to 400 iterations. APD follows a similar pattern, except that flow and diffusion models converge differently. DPoser [43] begins with similar poses and gradually introduces diversity, while PoseRFM starts from diverse poses and progressively converges toward poses resembling the ground truth. To further illustrate this behavior, we plot intermediate poses at various iterations K for the same inverse problem in Fig. E.2. A similar visualization for human mesh recovery is shown in Fig. E.3. In this case, both DPoser [43] and PoseRFM converge rapidly, with later iterations introducing only minor refinements.

Metrics for unoccluded regions. Since unoccluded joints are *observed* as ground truth in the data, their fitting error is 0 by construction. To better understand the fitting, we remove this replacement step and report results for both the visible regions and the full pose in Tab. E.4. Although the geodesic loss is geometrically accurate, it is harder to optimize than simple MSE. Nevertheless, our method general-

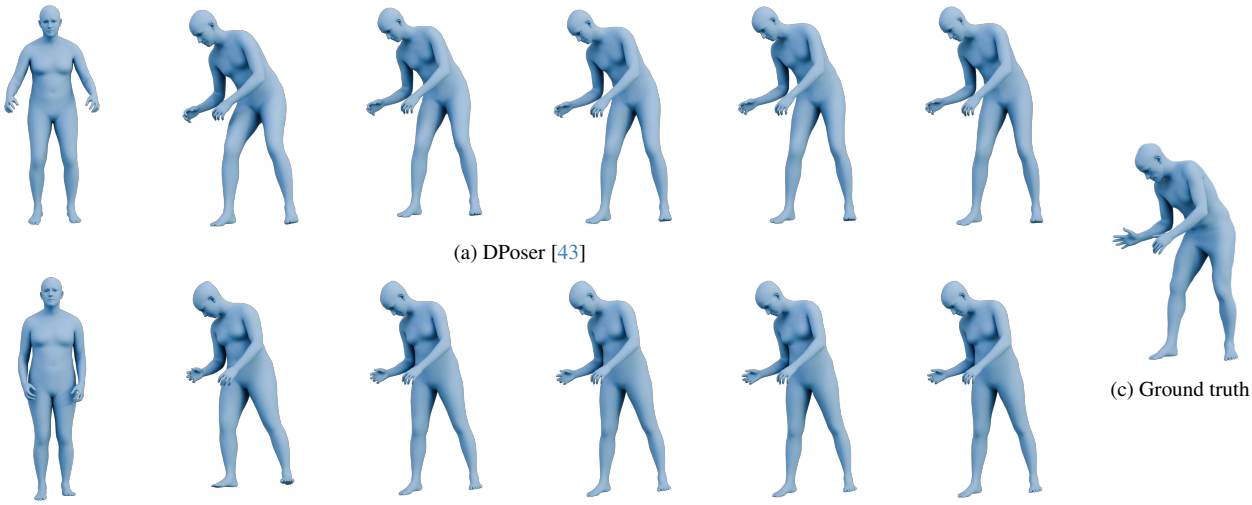


(a) DPoser [43]

(b) PoseRFM

(c) Ground truth

Figure E.2. Intermediate results at iterations K for pose completion with occluded legs. Left to right: $(K = 0, 40, 80, 120, 160, 200)$.



(a) DPoser [43]

(b) PoseRFM

(c) Ground truth

Figure E.3. Intermediate results at iterations K for inverse kinematics. Left to right: $(K = 0, 100, 200, 300, 400, 500)$.

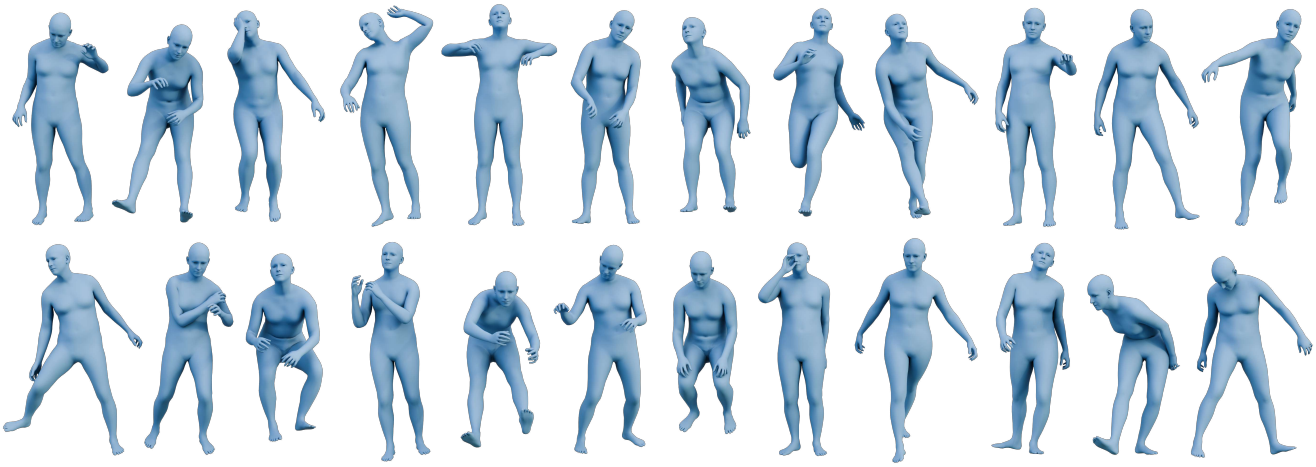


Figure E.4. More samples from unconditional generation using PoseRFM.

Table E.4. Additional Pose Completion metrics

Method	Occ. part	MPVPE ↓			APD ↑		
		Vis.	Occ.	Full	Vis.	Occ.	Full
DPoser [43]	Left leg	0.38	78.31	6.59	0.01	6.53	1.22
Ours	Left leg	10.02	83.81	16.03	0.81	6.02	1.76
DPoser [43]	Legs	0.60	102.46	17.26	0.01	7.75	2.80
Ours	Legs	8.57	95.00	22.66	0.74	7.23	3.10
DPoser [43]	Arms	19.19	104.94	28.54	0.01	5.69	2.09
Ours	Arms	26.56	107.36	36.46	0.60	5.72	2.55

izes well to occluded regions despite potential underfitting.

Qualitative comparison. We provide additional qualitative examples for pose generation (Fig. E.4), pose completion (Fig. E.5), motion denoising (Fig. E.6), and human mesh recovery (Fig. E.7). Comments on each comparison are included in the corresponding figure captions.

Beyond human pose. Our versatile Riemannian D-Flow framework can be used across different data domains and tasks. To illustrate, we now provide an application in **Earth science**. In this context, we no longer operate on the power manifold of rotations, but on the surface of the Earth, which is approximated by a sphere $\mathcal{S}^2$ embedded in the ambient 3D Euclidean space $\mathbb{R}^3$. Samples on this manifold are now points on $\mathcal{S}^2$.

We leverage the pre-trained RFM models of [12] as priors over the distributions of: volcanic activity (since $4,360$ BC), earthquakes (since $2,150$ BC), major floods (since 1985), and recent wildfires. Each data point represents the spatial occurrence of an event, without currently factoring in temporal or auxiliary meteorological data (e.g., temperature, wind, pressure, etc.). With each of these models we formulate an inpainting problem on a portion of the Earth's surface and use our Riemannian D-Flow framework to infer the x_0 producing the most accurate results outside the region to be in-painted. Guidance is still provided in the form of Riemannian source-point optimization following Eq. (10):

$$\min_{x_0 \in \mathcal{S}^2} \big(\mathcal{L}(x(1)) := \mathcal{L}_{\text{data}}(x(1)) + \mathcal{R}(x_0, u)\big).$$

$\mathcal{L}_{\text{data}}(x(1))$ is conceptually similar to Eq. (20), but uses the geodesic distance on $\mathcal{S}^2$

$$d_{\mathcal{S}^2}(x_a, x_b) = r \ \arccos\left(\frac{x_a \cdot x_b}{r^2}\right),$$

and the mask is applied on a subset $\Omega \subset \mathcal{S}^2$ of $\mathcal{S}^2$ rather than on the state itself. Assuming to operate on a unit-sphere, the data loss can be written as:

$$\mathcal{L}_{\text{data}}(x_1) = \sum_{i \in (\mathcal{S}^2 \setminus \Omega)} \arccos\left(x_1^{(i)} \cdot x_{\text{obs}}^{(i)}\right),$$

Table E.5. Dataset specifications for the climate experiments

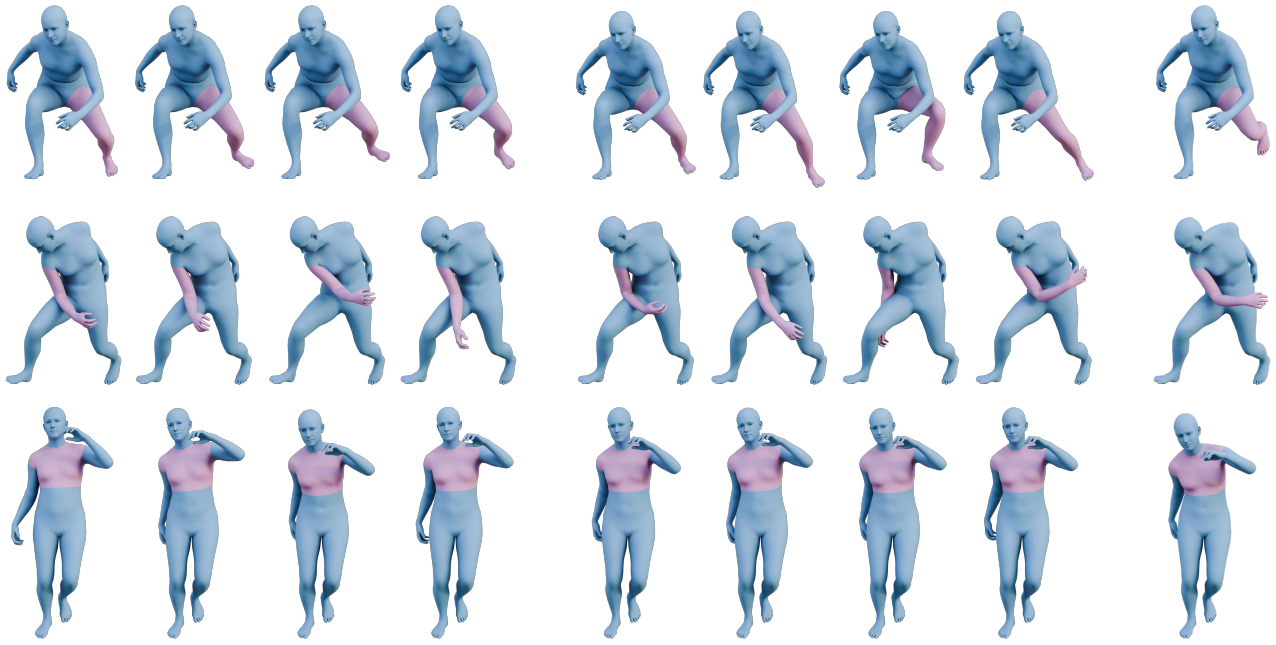
Event	Masked region	GT size	Total Test size
Volcano	Japan & nearby	16	82
Earthquake	Central & South America	104	612
Flood	South & South-East Asia	182	487
Fire	Africa	499	1280

Table E.6. Negative log-likelihood (NLL) measured on the inpainted region on Earth Climate dataset.

Method	Volcano	Earthquake	Flood	Fire
D-Flow [3]	-4.669	0.445	-0.171	-0.472
Riemannian D-Flow	**-6.337**	**-0.641**	**-0.451**	**-1.174**

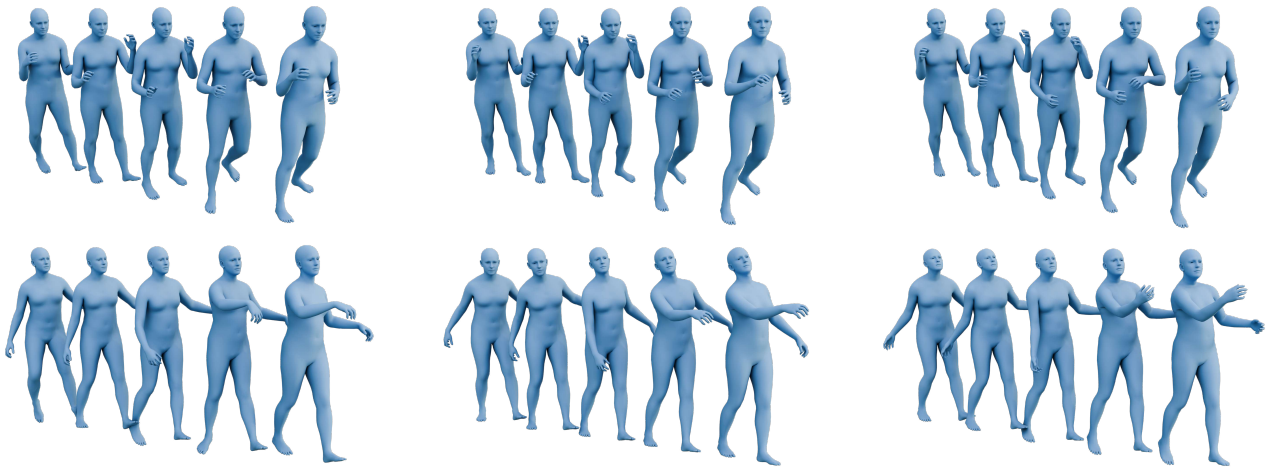
where $x_{\text{obs}}^{(i)}$ is the i-th observed data point (ground truth) outside the masked region. By minimizing the full objective using this data term, the prior allows us to infer plausible points also within the masked region Ω.

We use the test split of each dataset as our ground truth and select regions across the Earth where the corresponding natural event has commonly occurred. Details of the selected regions are provided in Tab. E.5. We compare 2 methods, D-Flow [3] and Riemannian D-Flow, for recovering the missing points in the masked regions. We evaluate performance using the negative log-likelihood (NLL) of the predicted points within the masked area, where lower values indicate a higher likelihood of the event occurring at that point. Results for all four datasets are presented in Tab. E.6. Riemannian D-Flow achieves the best performance across every event, indicating its generalizability to different manifolds. To visualize the results, we plot the masked regions, along with the ground truth and predicted points in Fig. E.8.



(a) DPoser [43] (b) PoseRFM (ours) (c) Ground truth (GT)

Figure E.5. Completed poses with left leg (top), right arm (middle) and torso (bottom) occluded. We show both visible and occluded joints. PoseRFM continues to generate realistic and diverse completions, except in the torso-occlusion case, where the limited diversity is clearly noticeable.



(a) DPoser [43] (b) PoseRFM (ours) (c) Ground truth (GT)

Figure E.6. Motion denoising with standard deviations of $40\mathrm{mm}$ (top row) and $100\mathrm{mm}$ (bottom row).

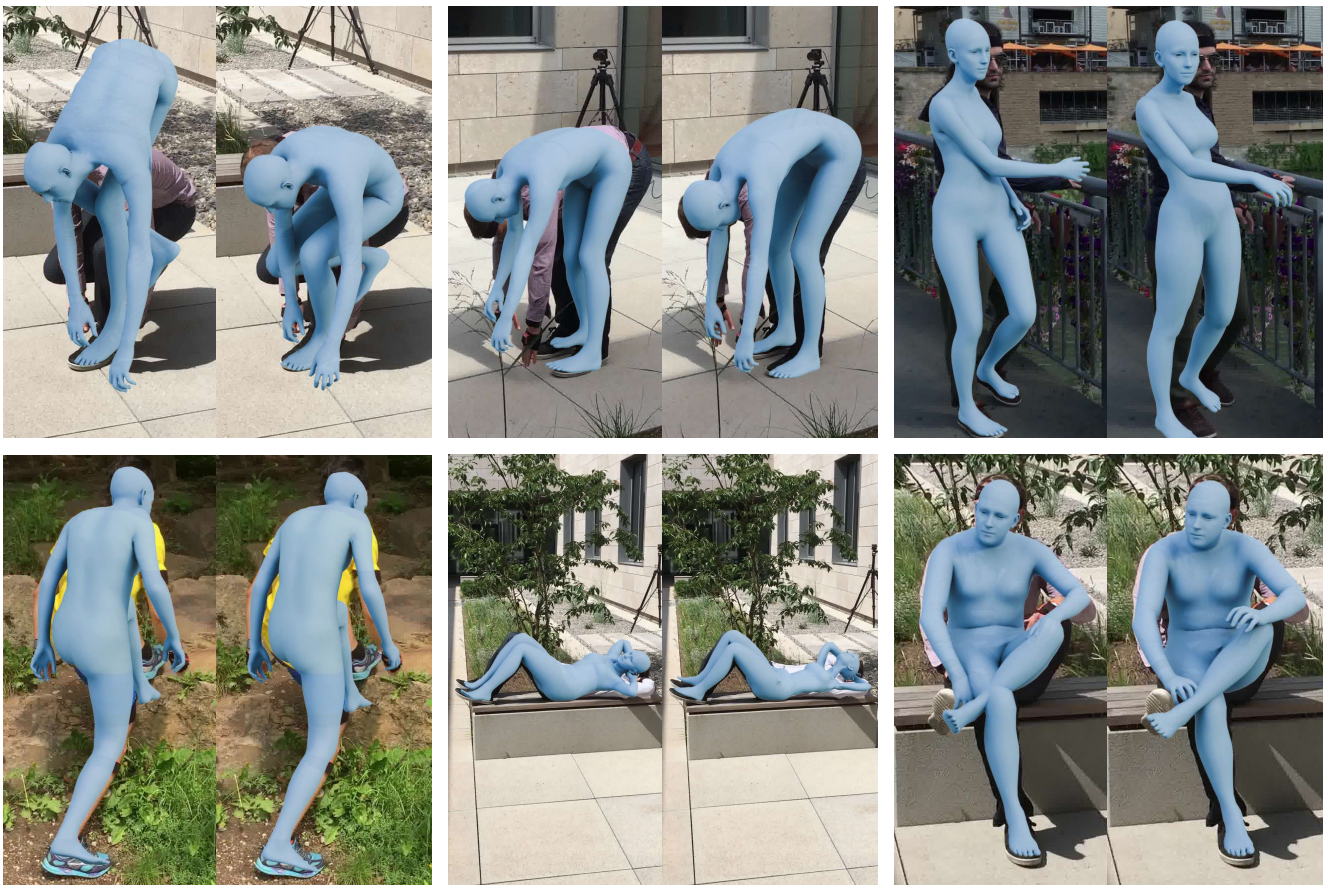


Figure E.7. Additional results of HMR on in-the-wild images from 3DPW [59]. Fitting from scratch (top) and initialization using CLIFF [33] (bottom). These results highlight the strength of PoseRFM on a challenging non-linear problem. DPoser [43] fails to optimize in two cases, and predicts poses with self-intersections.

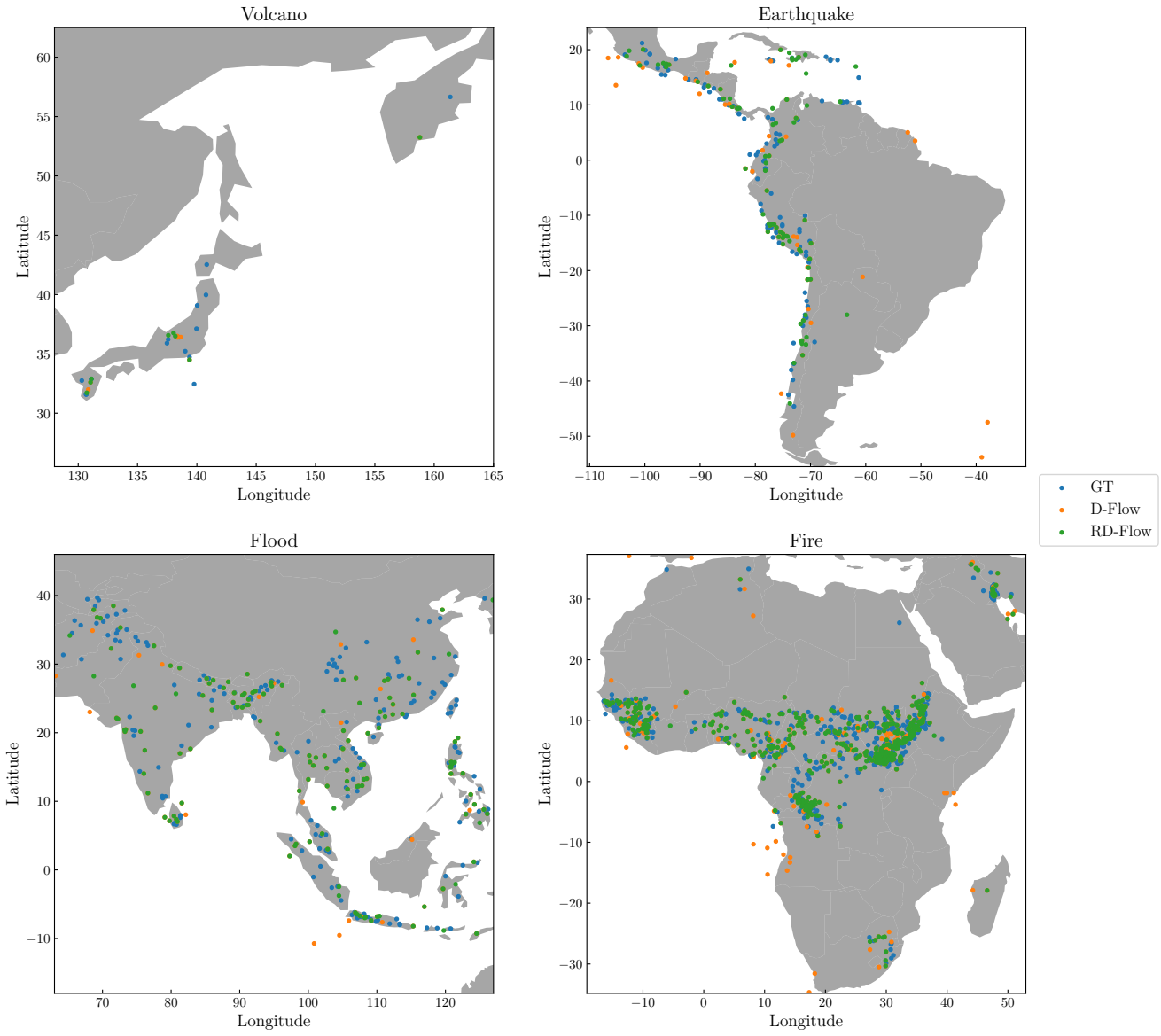


Figure E.8. Results on the climate dataset. Each visualization depicts the prediction over the region to be inpainted Ω. Riemannian D-Flow generates geographically consistent and likely points across diverse hotspots. In contrast, D-Flow often produces unlikely predictions, such as wildfires appearing in the ocean, leading to its poor NLL.