Matérn Noise for Triangulation-Agnostic Flow Matching on Meshes

TIANSHU KUAI, Université de Montréal & Mila, Canada
ARMAN MAESUMI, Brown University, USA
DANIEL RITCHIE, Brown University, USA
NOAM AIGERMAN, Université de Montréal & Mila, Canada

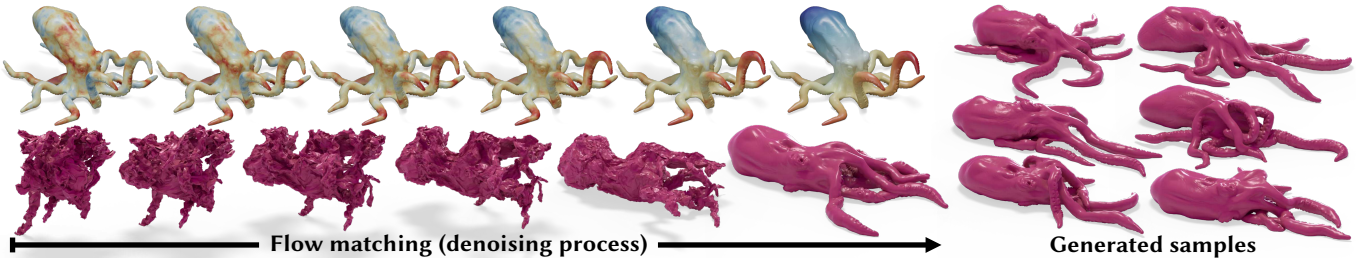


Fig. 1. **Generating different deformations of a high-resolution mesh through triangulation-agnostic flow matching.** Top left: our method generates signals (visualized via colors) on meshes via the flow matching [Lipman et al. 2022] paradigm's denoising process. Bottom left: in this case the signals generated correspond to a deformation of the mesh's vertices. The model was trained on a dataset of physical simulation of elastic equilibrium states, and thus produces various elastic rest poses of the 3D shape (right). The training set comprised of meshes in resolution of *9k* faces, and through the triangulation agnosticism of our method we generate results at test time at a resolution of *100k* faces.

This paper tackles the task of learning to generate signals over triangle meshes in a triangulation-agnostic manner, meaning the trained model can be applied to different meshes and triangulations effectively. Practically, the paper adapts the flow matching (FM) paradigm to a mesh-based, triangulation-agnostic setting. Theoretically, it proposes a specific noise distribution which is triangulation agnostic, to be used inside the FM model's denoising process. While noise distributions are usually trivial to devise for, e.g., images, devising a triangulation-agnostic distribution proves to be a much more difficult task. We formulate a mathematical definition of triangulation agnosticism of distributions, via their spectrum. We then show that a discretization of a specific Gaussian random field called a *Matérn process* holds these desired properties, and provides a simple and efficient sampling algorithm. We use it as our noise model, and adapt FM to the triangulation-agnostic setting by using a state-of-the-art approach for learning signals on meshes in the gradient domain—PoissonNet—as the denoiser. We conduct experiments on elaborate tasks such as sampling elastic rest states, and generating poses of humanoids. Our method is shown to be capable of producing highly realistic results for meshes of over one million triangles, significantly exceeding the state-of-the-art in quality and diversity.

CCS Concepts: • **Computing methodologies** → **Shape analysis**.

Additional Key Words and Phrases: Generative models, mesh deformation

1 Introduction

In recent years, there has been an unprecedented surge in the generative capabilities of machine learning models, enabled by novel techniques such as denoising diffusion models [Ho et al. 2020; Song et al. 2020] and flow matching [Lipman et al. 2022]. These methods have been successfully applied to various visual modalities, e.g., 2D pixel images [Yu et al. 2025], and 3D voxel grids [Xiang et al. 2025].

As of today, triangle meshes stand as a modality that has yet to obtain practical generative capabilities of the same quality. In this context, by "generative capabilities for meshes", we mean treating an *existing* mesh as the domain—analogous to, e.g., the pixel grid of an image—and generating values over its vertices. For instance, in Figure 1, our method generates values representing (x, y, z) coordinates of the vertices of an octopus mesh, resulting in different deformations of the shape.

As meshes are one of the most ubiquitous representations of 2-manifolds—used in, e.g., computer graphics, video games, engineering, and biomedical imaging—introducing such generative capabilities holds potential for impactful applications.

This work tackles one of the main roadblocks to a practical mesh-based generative approach, and devises a generative framework that is *triangulation agnostic* [Aigerman et al. 2022; Maesumi et al. 2025; Sharp et al. 2022], meaning that the method can be applied to arbitrary triangulations of a 3D model and will exhibit near-identical behavior, regardless of the chosen triangulation (as long as it is well-behaved enough).

Indeed, triangulation agnosticism is crucial for real-world applications involving meshes, as it provides two critical benefits:

1) Applicability to heterogeneous data. In contrast to images, meshes are irregular graphs representing arbitrary discretizations of an underlying 3D object, and thus mesh datasets contain models with varying triangulations. It has been shown that naive architectures are sensitive to spurious information in the triangulations of such training data, which leads to invalid results at test time [Sharp et al. 2022] and has prompted the community to prefer architectures that behave agnostically under changes to the triangulation.

2) Efficiency on high-resolution meshes. The computations in triangulation-agnostic methods are less coupled to the triangulation and hence to its resolution, which leads to better performance, and allows triangulation-agnostic methods to benefit from efficient training (at lower resolutions) while still being applicable for inference on high-resolution meshes. Indeed, we demonstrate that

Authors' Contact Information: Tianshu Kuai, Université de Montréal & Mila, Canada; Arman Maesumi, Brown University, USA; Daniel Ritchie, Brown University, USA; Noam Aigerman, Université de Montréal & Mila, Canada.

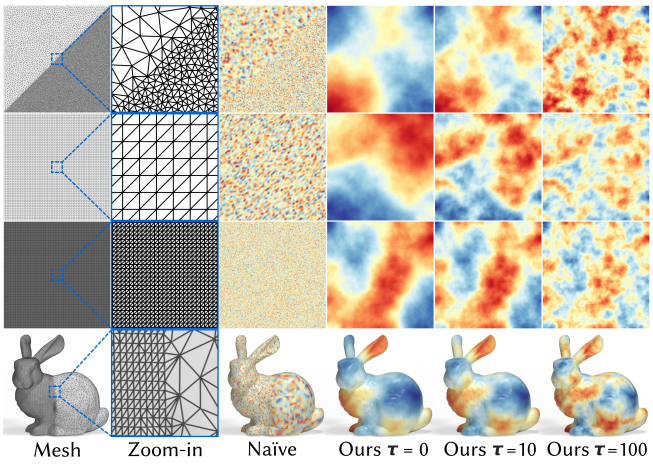


Fig. 2. **Matérn noise on different triangulations.** Our noise exhibits similar structure across differently triangulated regions, as opposed to *Naïve* sampling from iid Gaussians for each vertex.

our method attains state-of-the-art results, at a resolution that far exceeds prior generative methods for meshes.

In designing a triangulation-agnostic *generative framework*, we observe that simply using an existing triangulation-agnostic network architecture is insufficient to obtain agnosticism at sampling time. The primary insight of this paper is that one requires an appropriate *noising scheme* that, like said architectures, is not affected by the specific discretization of the surface. We show that the conventional method of sampling iid Gaussian noise leads to generative models that cannot operate on unseen triangulations of their training data, let alone generalize to out-of-distribution shapes.

This problem setting leads to the main technical challenge of this paper: while devising a noise distribution, e.g., for images, is a trivial task, the same cannot be said for meshes, considering the aforementioned intricacies; see for example bottom of Figure 2. In this setting, the distribution itself needs to be agnostic to triangulation, lest its characteristics change under superficial changes to the mesh, resulting in noise samples that are out-of-distribution for the trained denoiser (see Figure 14).

We formalize a definition of triangulation-agnostic noise distributions through the lens of probabilistic spectral shape analysis, allowing us to compare distributions defined on different triangulations. We then observe that a specific type of *Matérn process* [Matérn 1960; Whittle 1963] has a well-studied linear finite-elements discretization [Lindgren et al. 2011] that exactly describes distributions with the desired properties, and leads to a highly simple and efficient sampling scheme using tools that are customary in geometry processing. In particular, one samples from the discrete Matérn process by simply solving a *screened Poisson equation*, see Algorithm 1.

With this sampling method at hand, we produce a full triangulation-agnostic generative pipeline based on the flow matching [Lipman et al. 2022] paradigm. As a denoising network, we use PoissonNet [Maesumi et al. 2025], an architecture that performs triangulation-agnostic learning in the gradient domain. See Figure 3 for our complete pipeline.

We demonstrate the benefits of our framework on a set of canonical shape editing tasks that are widely studied. In particular, we use our method to generate plausible deformations of humanoid characters as well as resting states of elastic objects in equilibrium. In all cases, our method produces plausible results even on meshes with over one million vertices. By comparison, prior methods in this space are unable to generalize beyond their training meshes (due to sensitivity to triangulation), and in some cases are too inefficient for generation on high-resolution geometry.

To summarize, the main contributions of this work are:

(1) The first generative model over meshes that is triangulation-agnostic and applicable to high-resolution meshes.
(2) A theoretical treatment and definition of triangulation-agnostic distributions of signals over meshes via probabilistic analysis of the spectrum of the signals.
(3) Introduction of Matérn processes [Matérn 1960; Whittle 1963] and their discretization [Lindgren et al. 2011] as an efficient algorithm for sampling from triangulation-agnostic distributions.

Our code, model checkpoints, datasets and data-generation scripts are available at https://github.com/kts707/matern-fm

2 Related Work

Generation through denoising. Our method builds on *denoising-based* generative modeling, where one starts from a simple noise distribution and iteratively transforms it into an empirical data distribution. We adopt *flow matching* (FM) [Lipman et al. 2022], which has been applied to many modalities [Li et al. 2025; Liu et al. 2024]. FM considers a time-varying signal $\mathbf{f}_t$ parametrized by $t \in [0, 1]$, which interpolates between noise ($t = 0$) and data ($t = 1$). Practically, FM predicts a time-dependent vector field, parametrized by a neural network $F_\theta(\mathbf{f}_t, t)$, which aims to predict the velocity of the signal w.r.t time. Then, at inference time, one generates a sample by integrating the corresponding ODE:

$$\frac{\partial}{\partial t}\mathbf{f}_t = F_\theta(\mathbf{f}_t, t), \tag{1}$$

from $t = 0$ to $t = 1$. FM is closely related to *diffusion/score-based* models [Ho et al. 2020; Song et al. 2020], which learn to reverse a noising process, as well as ODE-based methods such as rectified flows [Liu et al. 2022]. We focus on FM and contribute a triangulation-agnostic noising and denoising scheme for signals on meshes.

Generation over existing meshes. Our work focuses on generating signals *on existing meshes, without modifying their underlying structure*. To the best of our knowledge, there are only two prior works that use current state-of-the-art denoising-based generative techniques [Elhag et al. 2024; Wang et al. 2025a]; however, they are not triangulation agnostic. We compare to these methods and show that our method exceeds them in performance. Slightly older methods use variational autoencoders [Tan et al. 2018, 2021; Yuan et al. 2020]; however, this architecture currently does not work sufficiently well for triangulation-agnostic generation. Dupont et al. [2022] focus on learning distributions of functions, via a hypernetwork, and can be applied to meshes.

Deformations are a common application of learning-based methods on meshes [Aigerman et al. 2022; Besnier et al. 2024; Liu et al.

2021; Maesumi et al. 2023, 2025; Muralikrishnan et al. 2024; Sundararaman et al. 2024]. They can be generated using VAEs [Huang et al. 2021; Muralikrishnan et al. 2022], or by optimizing an objective defined by pre-trained vision transformers [Dinh et al. 2025; Gao et al. 2023; Kim et al. 2025; Michel et al. 2022; Wang et al. 2025b]. Generative techniques can be applied to skeletal rigs, as opposed to the mesh itself [Gat et al. 2025; Shafir et al. 2024; Tevet et al. 2023; Zhong et al. 2025]. We note our approach can be considered a generative method on graphs [Guo and Zhao 2022; Zhang et al. 2023], though its machinery is heavily tailored for meshes.

Non-Euclidean generative modeling. Many works compute flows *over* a manifold in order to sample a *point* on it [Chen and Lipman 2023; De Bortoli et al. 2022; Huang et al. 2022; Mathieu and Nickel 2020]. This is a significantly different setting than ours: we define flows in the *functional space* of the manifold, in order to sample a *function* defined on the manifold.

Generative methods for other 3D modalities. The most common generative task for other 3D modalities is generation from scratch (e.g., from a given text prompt). Methods for this task have been proposed for most shape representations: tetrahedra [Gao et al. 2022; Liu et al. 2023; Shen et al. 2021], voxels [Ren et al. 2024a; Xiang et al. 2025], point clouds [Ren et al. 2024b; Vahdat et al. 2022; Yang et al. 2019], 3D Gaussian Splats [Tang et al. 2024], Neural Radiance Fields [Poole et al. 2022], and hybrid approaches [Yan et al. 2025; Zhang et al. 2024]. Generation of meshes [Gao et al. 2019; Nash et al. 2020] can be achieved via LLM agents [Lu et al. 2025] or GPT architectures [Siddiqui et al. 2023]. Importantly, these works synthesize *shape representations* (geometry and topology) from scratch or from textual/visual prompts. Our problem setting assumes an *existing mesh* and samples from distributions of signals defined on it.

Spectral mesh analysis. We use spectral mesh analysis via eigenvectors of the mesh Laplacian. This is a cornerstone tool in geometry processing [Ben-Chen and Gotsman 2005; Karni and Gotsman 2000; Lévy 2006; Lévy and Zhang 2010; Vallet and Lévy 2008; Zhang et al. 2010], with various applications [Lescoat et al. 2020; Mejia et al. 2017; Song et al. 2014], notably parameterization [Gotsman et al. 2003], shape signatures [Sun et al. 2009], matching [Ovsjanikov et al. 2012; Pierson et al. 2025; Rimon et al. 2025; Zhuravlev et al. 2025], analysis [Fumero et al. 2020; Liu et al. 2017], and learning [Sharp et al. 2022; Smirnov and Solomon 2021]. We use spectral analysis in a probabilistic way to define triangulation-agnostic distributions.

Matérn processes. Our noise distribution is part of a class of Gaussian random fields called Matérn processes [Matérn 1960; Whittle 1963], which are a commonly used statistical tool [Bolin et al. 2023; Stein 1999] applied in various statistical modeling tasks [Coveney et al. 2020; Gillan et al. 2025; Maddix et al. 2021], as well as in machine learning [Williams and Rasmussen 2006]. We use its discretization via linear finite elements, which has been heavily studied [Lindgren et al. 2011] (see [Bolin et al. 2024; Borovitskiy et al. 2021, 2020; Rosa et al. 2023] for other discretizations). All of the above focus on elaborate statistical modeling; we use a specific Matérn process in a simple way: we observe and prove that it is an ideal model for triangulation-agnostic noise.

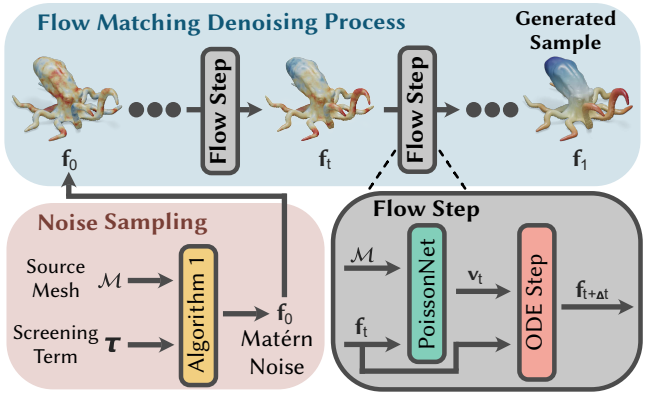


Fig. 3. **Architecture of our triangulation-agnostic generative pipeline.** We first perform *noise sampling* (bottom left) from our proposed triangulation-agnostic Matérn noise distribution via Algorithm 1. We then use that noise as the initial signal $\mathbf{f}_0$ in a *flow matching denoising process* (top), which iteratively denoises the signal into the final sample $\mathbf{f}_1$. In each *flow step* (bottom right) we input the current signal $\mathbf{f}_t$ into PoissonNet [Maesumi et al. 2025], which predicts the current momentary "velocity" $\mathbf{v}_t$. An ODE step time-integrates the velocity, starting from the current signal $\mathbf{f}_t$, to obtain the next signal $\mathbf{f}_{t+\Delta t}$. This continues until reaching $t = 1$.

Non-trivial noise models. While iid Gaussian sampling is the default noise distribution for generative techniques, several works [Huang et al. 2024; Jolicoeur-Martineau et al. 2023; Nachmani et al. 2021] have explored alternative initial noise distributions on toy or image datasets. In graphics, noise is commonly used in procedural generation [Perlin 1985], and point sampling [Ahmed et al. 2022].

3 Triangulation-Agnostic Distributions

Overview. In this section we define triangulation-agnostic distributions (Section 3.3) using a spectral perspective, then we identify that a Matérn process matches that definition in the continuous case (Section 3.4), and finally we describe how to compute samples from its FEM discretization and review its properties (Section 3.5). In Section 4, we then utilize this distribution as the noise scheme in a flow matching [Lipman et al. 2022] pipeline using a triangulation-agnostic denoising network—see Figure 3.

3.1 Preliminaries

Let $\mathcal{M}$ be a manifold triangle mesh discretizing a smooth 2-manifold Ω with lumped mass matrix $\mathbf{M}$ and cotangent Laplacian $\mathbf{L}$ [Pinkall and Polthier 1993] (see the supplemental material, Section F for an elaboration). A *signal* is a scalar function sampled at vertices, represented by $\mathbf{f} \in \mathbb{R}^n$. We use the mass-weighted inner product between two signals, $\langle \mathbf{f}, \mathbf{g} \rangle_{\mathbf{M}} = \mathbf{f}^\top \mathbf{M} \mathbf{g}$. The Laplacian $\mathbf{L}$ yields generalized eigenpairs, $(\boldsymbol{\phi}, \lambda)_i$, satisfying

$$\mathbf{M}^{-1}\mathbf{L}\boldsymbol{\phi}_i = \lambda_i \boldsymbol{\phi}_i, \tag{2}$$

with $\mathbf{M}$-orthonormality $\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle_{\mathbf{M}} = \delta_{ij}$. Let $\boldsymbol{\Phi}$ be the matrix with the eigenvectors as columns.

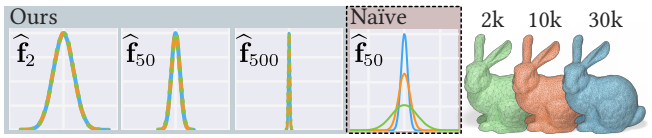


Fig. 4. **Empirical per-frequency statistics of Matérn noise.** For each of three spectral coefficients – $\widehat{\mathbf{f}}_2, \widehat{\mathbf{f}}_{50}, \widehat{\mathbf{f}}_{500}$ – we compute its distribution over 100k sampled noises, visualized as a histogram. We show overlays of the distributions for three mesh resolutions. Our method generates near-identical distributions for all meshes, with decreasing variance for higher coefficients. *Naïve*, iid sampling of Gaussians, yields different distributions for each mesh.

We will heavily use the *spectral* representation of a signal $\mathbf{f}$, which is its representation as the (unique) linear combination of the Laplacian's eigenvectors, $\mathbf{f} = \sum_{i=1}^{n} \widehat{\mathbf{f}}_i \boldsymbol{\phi}_i$, where $\widehat{\mathbf{f}} \in \mathbb{R}^n$ is a vector of coefficients corresponding to each eigenvector. $\widehat{\mathbf{f}}_i$ can be directly computed by projecting $\mathbf{f}$ onto the eigenvector $\boldsymbol{\phi}_i$:

$$\widehat{\mathbf{f}}_i = \langle \mathbf{f}, \boldsymbol{\phi}_i \rangle_{\mathbf{M}} = \mathbf{f}^\top \mathbf{M} \boldsymbol{\phi}_i. \tag{3}$$

The spectral representation is widely used in geometry processing as a generalization of the discrete Fourier transform (DFT) for a 1D signal, with $\boldsymbol{\phi}_i$ analogous to 1D sinusoids, eigenvalues λ_i to frequencies, and the coefficients $\widehat{\mathbf{f}}$ to amplitude of each frequency. The spectrum of the Laplace-Beltrami operator (LBO) Δ of a compact manifold Ω is discrete and infinite, and the spectrum of the cotangent Laplacian $\mathbf{L}$ of a mesh with n vertices is assumed to approximate its n lowest eigenmodes [Wardetzky 2007, 2008].

We will consider multivariate normal distributions $\mathcal{N}(\mathbf{0}, \Sigma)$, with zero mean and Σ covariance. We will make use of the basic fact that if $x \sim \mathcal{N}(\mathbf{0}, \Sigma)$, and $y = Ax$ for some linear transformation A, then y's distribution is a normal distribution equal to:

$$y \sim \mathcal{N}\left(\mathbf{0}, A^T \Sigma A\right). \tag{4}$$

3.2 Problem Statement

Formally, we seek an algorithm that, for a given mesh $\mathcal{M}$, samples from a distribution of signals defined over its vertices, $\mathbf{f} \sim \mathcal{D}^{\mathcal{M}}$. We refer to such a randomly-sampled signal as a *random field*. The resulting distribution from which samples are drawn should be triangulation agnostic, meaning that for two triangulations of the same surface, $\mathcal{M} \approx \Omega$, $\mathcal{N} \approx \Omega$, the sampling procedure should yield similar distributions $\mathcal{D}^{\mathcal{M}} \approx \mathcal{D}^{\mathcal{N}}$.

Since the two meshes have a different number of vertices with no given correspondence between them, a direct comparison between distributions requires notions such as a 1-to-1 map between the meshes and earth mover's distance, which would obstruct our attempt to construct a triangulation-agnostic distribution *a priori*.

Hence, we compare distributions in the spectral domain, where *frequency* provides a canonical one-dimensional coordinate system. Namely, the spectral coefficients $\widehat{\mathbf{f}}$ are a function of $\mathbf{f}$, and hence are also random variables distributed according to an (unknown) distribution $\widehat{\mathcal{D}}^{\mathcal{M}}$. Intuitively, $\widehat{\mathbf{f}}$ is a "dual random field" over the spectrum, and $\widehat{\mathcal{D}}^{\mathcal{M}}$ assigns a probability density for any specific spectral coefficients. Our goal is to construct a distribution $\mathcal{D}^{\mathcal{M}}$ s.t. its spectral distribution $\widehat{\mathcal{D}}^{\mathcal{M}}$ has coefficients that yield consistent statistics across triangulations and stable behavior under mesh refinement (Figure 4). For simplicity, we will henceforth omit $\mathcal{M}$ and refer to the distributions as $\mathcal{D}, \widehat{\mathcal{D}}$ when unambiguous.

3.3 Spectral Properties for Triangulation Agnosticism

We define a triangulation-agnostic distribution $\mathcal{D}$ as satisfying three key spectral properties in terms of the statistical behavior of the spectral coefficients $\widehat{\mathbf{f}}$ of signals sampled from that distribution:

(1) **Per-frequency independence.** The coefficients, $\widehat{\mathbf{f}}_i$, are statistically mutually independent for all i, each sampled from an independent univariate distribution.

(2) **Mesh-invariant frequency statistics.** There exists a family of univariate distributions $\widehat{\mathcal{D}}_x$, parameterized by x, such that each spectral coefficient $\widehat{\mathbf{f}}_i$ with eigenvalue λ_i is distributed as $\widehat{\mathbf{f}}_i \sim \widehat{\mathcal{D}}_{\lambda_i}$, regardless of the mesh. Moreover, $\widehat{\mathcal{D}}_x$ is Lipschitz with respect to x; specifically, for $|\lambda - \lambda'| \leq \varepsilon$,

$$W_2(\widehat{\mathcal{D}}_\lambda, \widehat{\mathcal{D}}_{\lambda'}) = O(\varepsilon),$$

where W_2 is the 2-Wasserstein distance. Note that this should hold between *any two meshes*, even ones that represent different objects or have different genus.

(3) **Bounded high-frequency content.** For any $\varepsilon > 0$, there exists a k such that the tail contribution beyond frequency k is negligible:

$$\mathrm{Var}\left[\sum_{i=k}^{n} \widehat{\mathbf{f}}_i\right] < \varepsilon,$$

up to the usual discretization error in approximating the continuous spectrum.

To give insight into the necessity of these properties:

- Property 1 decouples behavior across frequencies and enables element-wise comparisons between univariate distributions.
- Property 2 enforces that each coefficient's univariate distribution is specified in a mesh-independent manner, purely as a function of frequency, behaving the same across any two arbitrary meshes (see Figure 4). This makes the distribution stable under re-triangulation and spectral perturbations, and prevents resolution changes from altering the distribution, except through predictable high-frequency additions controlled by Property 3.
- Property 3 prevents higher-resolution meshes from injecting unbounded additional high-frequency energy, as the iid gaussian noise does in Figure 2. See also Figure 5 for an illustration of undesired behavior under addition of new vertices (red region) of white noise (unbounded energy) vs. our noise (decaying energy per frequency).

We next identify a distribution that satisfies these properties.

3.4 Matérn Processes

We observe that there exists a specific distribution that satisfies the desired properties from Section 3.3. In the continuous case, it is defined as a distribution over functions $f : \Omega \to \mathbb{R}$ via the stochastic PDE:

$$\Delta f + \tau f = \mathcal{W}, \tag{5}$$

where $\tau > 0$ is a user-chosen parameter, and $\mathcal{W}$ is a sample from a white noise distribution, i.e., sampling iid from a normal distribution for all points on Ω.

In geometry processing terms, f is sampled by first drawing a white noise sample $\mathcal{W}$ (which has iid Gaussian spectrum), and then solving a *screened Poisson equation* with respect to it, which acts as a low-pass filter for the spectrum.

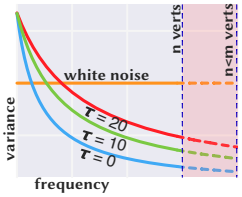


Fig. 5. Spectra of the different noise choices.

For the sake of our application, the screening term τ serves to introduce higher frequency content as desired (see Figures 5 and 2): as τ is raised, the relative variance of the distribution of $\widehat{\mathbf{f}}$ increases; however, asymptotically, as we go up the spectrum, eventually λ_i becomes the dominant term and the variance tends to zero. We use $\tau = 100$ in all experiments.

The above distribution is an instance of a class of Gaussian random fields called *Matérn processes* [Matérn 1960; Whittle 1963]. We will use the straightforward linear FEM discretization of this distribution, which has been extensively studied [Lindgren et al. 2011].

As far as we are aware, Matérn processes and their discretization are yet to be used for sampling noise for generative tasks, nor introduced into graphics and geometry processing. Since we need to prove the discretized version of this distribution satisfies the necessary properties for triangulation agnosticism from Section 3.3, we next provide an intuitive review of the construction of the canonical linear FEM discretization of Equation (5) from a geometry processing perspective. By doing so, we hope to also bring more attention in the computer graphics community to this useful tool.

3.5 Finite-Element Matérn Noise

We next give an intuitive derivation of the FEM discretization of Equation (5) (see Lindgren et al. [2011] for a much deeper discussion). Our derivation follows the discussion above: we derive a discrete equivalent of a white noise distribution $\mathcal{D}_{\text{white}}$, then we analyze the spectrum of the distribution $\mathcal{D}_{\text{Matérn}}$ obtained by applying the screened Poisson equation to samples of that white noise.

For a given mesh $\mathcal{M}$, we consider a multivariate normal distribution over its vertices, $\mathcal{D} \equiv \mathcal{N}(\mathbf{0}, \Sigma)$, with zero mean and Σ covariance. The corresponding spectral distribution is also a normal distribution with zero mean, $\widehat{\mathcal{D}} \equiv \mathcal{N}\left(\mathbf{0}, \widehat{\Sigma}\right)$ per Equation (4), and considering that $\widehat{\mathbf{f}}$ is a linear transformation of $\mathbf{f}$, per Equation (3).

Sampling discrete white noise. We define the white noise distribution as the normal distribution

$$\mathcal{D}_{\text{white}} \triangleq \mathcal{N}\left(\mathbf{0}, \mathbf{M}^{-1}\right), \tag{6}$$

where $\mathbf{M}$ is the lumped mass matrix. To verify this is indeed also a *spectral* white noise distribution, let $\mathbf{w} \sim \mathcal{D}_{\text{white}}$ be a sample, and $\widehat{\mathbf{w}}$ its spectral coefficients. From arithmetic of the variance of Gaussians, we get that the spectral covariance matrix is

$$\begin{aligned}\widehat{\Sigma}_{\text{white}} \equiv \text{Var}\left[\widehat{\mathbf{w}}\right] &\overset{Eq.\ (3)}{=} \text{Var}\left[\mathbf{w}^\top \mathbf{M}\Phi\right] \overset{Eq.\ (4)}{=} \Phi^\top \mathbf{M}\Sigma\mathbf{M}\Phi \\ &\overset{Eq.\ (6)}{=} \Phi^\top \mathbf{M}\mathbf{M}^{-1}\mathbf{M}\Phi = \Phi^\top \mathbf{M}\Phi \overset{orth.}{=} \mathbf{I}.\end{aligned} \tag{7}$$

ALGORITHM 1: Sampling Matérn Noise

Input: Mass matrix $\mathbf{M}$, Laplacian $\mathbf{L}$, screening term τ.

1 Sample an iid Gaussian random field $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

2 Solve $(\mathbf{L} + \tau\mathbf{M})\mathbf{f} = \sqrt{\mathbf{M}}\mathbf{n}$ for $\mathbf{f}$ to obtain the noise.[1]

Thus we have

$$\widehat{\mathcal{D}}_{\text{white}} = \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{8}$$

i.e., the coefficients $\widehat{\mathbf{w}}_i$ are sampled iid from a unit Gaussian, leading to a uniform independent behavior across the spectrum, as desired.

Applying the screened Poisson equation. The linear FEM discretization of the screened Poisson equation from Equation (5) is $(\mathbf{L} + \tau\mathbf{M})\,\mathbf{f} = \mathbf{M}\mathbf{w}$, where $\mathbf{w} \sim \mathcal{D}_{\text{white}}$ is a white noise sample. The solution to this equation is given by

$$\mathbf{f} = (\mathbf{L} + \tau\mathbf{M})^{-1}\,\mathbf{M}\mathbf{w}. \tag{9}$$

$\mathbf{f}$ is thus a random variable, sampled by the above procedure, its distribution being a normal distribution (per Equation (4), and considering it is a linear function of the white noise) denoted $\mathcal{D}_{\text{Matérn}}$.

Next, note that if $\boldsymbol{\phi}_i$ is a generalized eigenvector of $\mathbf{L}$ with eigenvalue λ_i, then it is also a generalized eigenvector of $(\mathbf{L} + \tau\mathbf{M})^{-1}$, with eigenvalue $(\lambda_i + \tau)^{-1}$. Therefore,

$$\begin{aligned}\widehat{\mathbf{f}}_i &= \left\langle \mathbf{f}, \boldsymbol{\phi}_i \right\rangle_{\mathbf{M}} = \left\langle (\mathbf{L} + \tau\mathbf{M})^{-1}\,\mathbf{M}\mathbf{w}, \boldsymbol{\phi}_i \right\rangle_{\mathbf{M}} \\ &= \boldsymbol{\phi}_i^\top\,(\mathbf{L} + \tau\mathbf{M})^{-1}\,\mathbf{M}\mathbf{w} = \frac{\boldsymbol{\phi}_i^\top \mathbf{M}\mathbf{w}}{\lambda_i + \tau} = \frac{\left\langle \mathbf{w}, \boldsymbol{\phi}_i \right\rangle_{\mathbf{M}}}{\lambda_i + \tau} \overset{Eq.\ (3)}{=} \frac{\widehat{\mathbf{w}}}{\lambda_i + \tau}.\end{aligned} \tag{10}$$

Per Equation (8), $\text{Var}\left[\widehat{\mathbf{w}}_i\right] = 1$, thus we get $\text{Var}\left[\widehat{\mathbf{f}}_i\right] = \text{Var}\left[\frac{\widehat{\mathbf{w}}}{\lambda_i+\tau}\right] = \frac{1}{(\lambda_i+\tau)^2}$, finally yielding the desired spectral distribution

$$\widehat{\mathcal{D}}_{\text{Matérn}} \equiv \mathcal{N}\left(\mathbf{0}, (\Lambda + \tau\mathbf{I})^{-2}\right), \tag{11}$$

with Λ being the diagonal matrix of eigenvalues of the Laplacian. In the supplemental material, Appendix A, we show this spectral distribution satisfies the necessary properties for triangulation agnosticism from Section 3.3, entailing that $\mathcal{D}_{\text{Matérn}}$ is a triangulation-agnostic distribution, as desired. On a high level: the covariance matrix is diagonal entailing iid sampling, the variance of each spectral coefficient $\widehat{\mathbf{f}}_i$ is solely a function of its eigenvalue, and decays fast enough w.r.t. frequency.

We summarize the highly simple and efficient process of sampling Matérn noise in Algorithm 1. As the matrix in line (2) is sparse and positive-definite, it can be prefactorized through a Cholesky decomposition for rapid sampling.

Choosing the screening term τ. While any specific value of τ produces consistent noise distributions for different mesh discretizations of the same 3D surface, there is no guarantee of consistent behavior across different shapes, nor between different scaled versions of the same mesh, see Figure 7, left. This is due to the change in the spectrum (i.e., eigenvalues and eigenvectors) between different surfaces and between different scales.

We utilize the Frobenius norm of the discrete Laplace-Beltrami operator, $\Gamma = ||\mathbf{M}^{-1}\mathbf{L}||_F$. It is well-known that $\Gamma^2 = \sum_i \lambda_i^2$, hence

[1] This folds both the white noise and the screened Poisson equation into one equation.

Fig. 6. **Generated reposings**. Our generative model, trained only on the 18k mesh, reposes humans to plausible poses solely via its learned representation.

ALGORITHM 2: Normalized Matérn Noise

Input : Mass matrix $\mathbf{M}$, Laplacian $\mathbf{L}$, user-chosen parameter c.

1 Compute the normalization factor $\Gamma = ||\mathbf{M}^{-1}\mathbf{L}||_F$.
2 Compute the normalized screening term $\tau = \Gamma c$.
3 Compute the signal $\mathbf{f}$ via Algorithm 1, using τ.
4 Return the normalized signal $\sqrt{\Gamma}\mathbf{f}$.

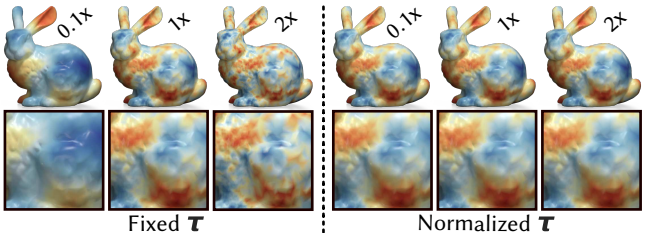


Fig. 7. **Fixed τ vs. normalized τ.** The Matérn noise sampled using a fixed screening term τ is not invariant to the scaling of the mesh (global scaling factors of 0.1 and 2.0); with our proposed normalization scheme, the noise exhibits a consistent, scale-independent spectrum.

linearly proportional to the eigenvalues. Thus, the user chooses the screening term hyperparameter $c \in \mathbb{R}^+$, and we define $\tau = c\Gamma$, yielding a screening term τ proportional to the eigenvalues.

This choice normalizes the screening term w.r.t. the power of the spectrum, leading to more consistent behavior across different meshes, and specifically, to scale invariance; namely , let $\mathbf{M}^*$,$\mathbf{L}^*$,λ^*,Γ^*, $\mathbf{f}^*$ be the mass matrix, cotangent Laplacian, eigenvalue, normalizing factor, and signal of the original, unscaled mesh. Assume the mesh's vertex coordinates are scaled by $k \in \mathbb{R}^+$; the resulting (angle-based) cotangent Laplacian remains the same $\mathbf{L} = \mathbf{L}^*$; the mass matrix scales up by k^2, $\mathbf{M} = k^2\mathbf{M}^*$. Hence, per Equation (2), $\lambda_i = k^{-2}\lambda_i^*$, and hence also $\Gamma = k^{-2}\Gamma^*$. Plugging this into Equation (9) we have $\mathbf{f} = (\mathbf{L} + c\Gamma\mathbf{M})^{-1}\mathbf{M}\mathbf{w} = \left(\mathbf{L}^* + ck^{-2}\Gamma^* k^2\mathbf{M}^*\right)^{-1} k^2\mathbf{M}^*\mathbf{w} = (\mathbf{L}^* + \tau^*\mathbf{M}^*)^{-1} k^2\mathbf{M}^*\mathbf{w}$. Per Equation (6), $\mathbf{w} = k^{-1}\mathbf{w}^*$, thus we get $\mathbf{f} = (\mathbf{L}^* + \tau^*\mathbf{M}^*)^{-1} k\mathbf{M}^*\mathbf{w}^* = k\mathbf{f}^*$. Hence, we normalize the resulting signal by multiplying it with $\sqrt{\Gamma} = k^{-1}\sqrt{\Gamma^*}$, thus yielding complete scale invariance: $\sqrt{\Gamma}\mathbf{f} = \sqrt{\Gamma^*}\mathbf{f}^*$, see Figure 7, right. We summarize this normalization in Algorithm 2.

4 Triangulation-Agnostic Flow Matching

We follow the standard training scheme of rectified flow [Albergo and Vanden-Eijnden 2022; Lipman et al. 2022; Liu et al. 2022]. Given a mesh with a data sample $\mathbf{f}_1$ from the dataset, we construct the linear conditional probability paths between a Matérn noise sample $\mathbf{f}_0$ and the data sample $\mathbf{f}_1$ as $\mathbf{f}_t = (1-t)\cdot\mathbf{f}_0 + t\cdot\mathbf{f}_1$, where $t \in [0,1]$, $\mathbf{f}_1 \sim \mathcal{D}_{\text{Data}}$, and $\mathbf{f}_0 \sim \mathcal{D}_{\text{Matérn}}$ is sampled from our noise distribution.

To complete the flow matching paradigm, we build our denoising network using a state-of-the-art triangulation agnostic architecture, PoissonNet [Maesumi et al. 2025], denoted $F_\theta(\mathbf{f}_t, t) \to (\mathbf{J}_\theta^t, \mathbf{v}_\theta^t)$. It operates by first predicting Jacobian velocity $\mathbf{J}_\theta^t$ in the gradient domain of the signals, which are then translated to vertex velocity $\mathbf{v}_\theta^t$ via the Poisson equation. We apply the standard conditional flow matching loss to both vertex and Jacobian velocities as:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\mathbf{f}_0,\mathbf{f}_1}\left[||\mathbf{M}(\boldsymbol{\delta} - \mathbf{v}_\theta^t)||^2 + ||\boldsymbol{A}(\nabla\boldsymbol{\delta} - \mathbf{J}_\theta^t)||^2\right], \qquad (12)$$

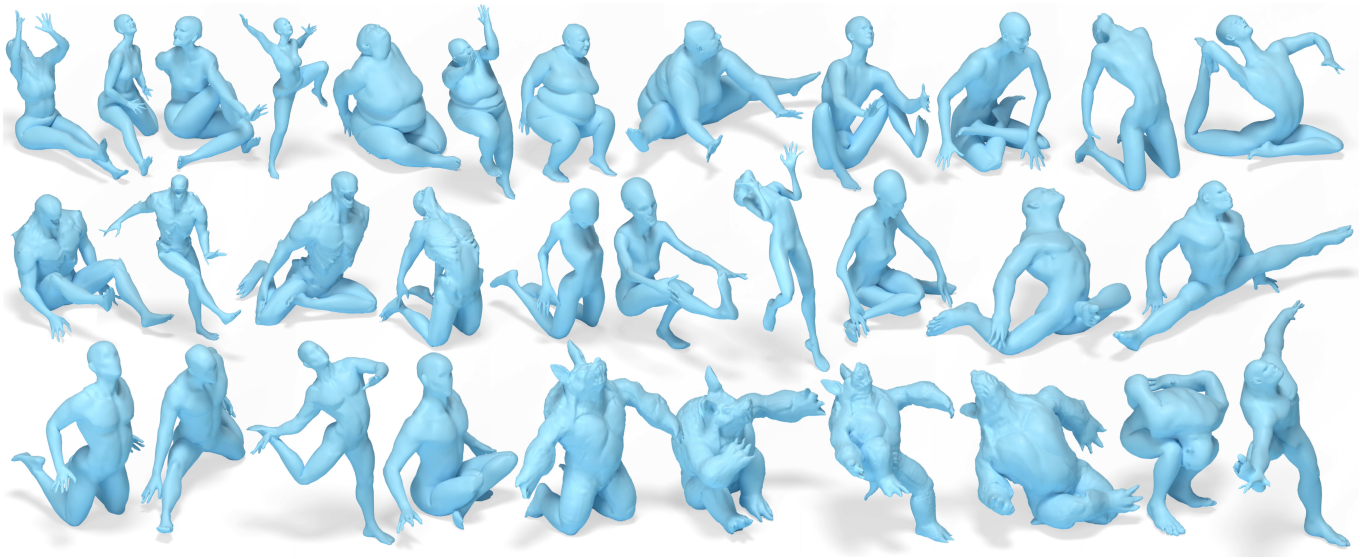


Fig. 8. **Generalization to various 3D shapes**. Our trained model generalizes to various humanoid meshes (never seen during training).

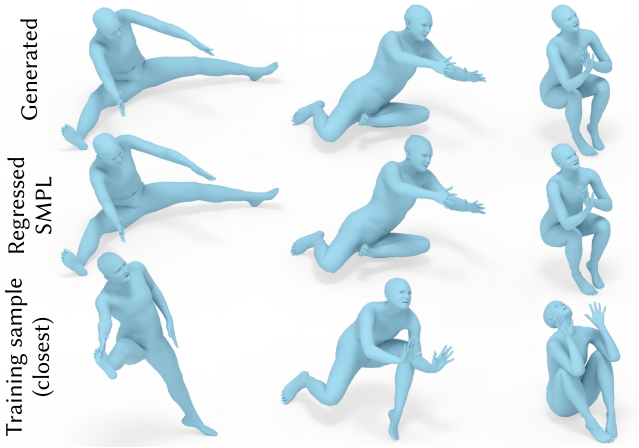


Fig. 9. **Closest training sample and regressed SMPL meshes.** Our generated samples approximate the GT deformation distribution well, and can be matched almost perfectly with the SMPL model, via regression. The closest training sample is quite far, showing our method generalizes.

where $\mathbf{M}$ and $\boldsymbol{A}$ are the diagonal matrices of vertex masses and face areas, respectively, $\boldsymbol{\delta} = \mathbf{f}_1 - \mathbf{f}_0$ is the GT velocity, and ∇ is the *spatial* gradient operator. During the sampling process, we use the vertex velocity $\mathbf{v}_\theta^t$ for numerical integration of the ODE (Equation (1)) via the midpoint ODE solver.

5 Experiments

We next experimentally evaluate the capabilities of our generative framework. In all experiments, we train our model on *one single triangulation*, and validate equivalent performance on various other triangulations, namely of higher-resolution, for which direct training would be infeasible.

Deformation datasets. We choose to focus our experiments on generating deformations, i.e., generating new positions for all vertices of a mesh. This is a central task in learning over meshes [Aigerman et al. 2022; Gao et al. 2023; Maesumi et al. 2023, 2025; Michel et al. 2022; Muralikrishnan et al. 2022] that fits well to our generative framework, as datasets are easy to obtain, and our backbone network, PoissonNet [Maesumi et al. 2025], has been shown to perform well over it (although thus far, only in a non-generative setting).

Evaluation Metrics Aside from an error metric which we devise on a per-experiment basis, we additionally estimate how well generated samples approximate the distribution of the dataset via Minimum Matching Distance (MMD) – the distance to the closest sample in a GT reference set – and Coverage (COV) – percentage of samples in reference set that are the nearest to a generated sample; see [Achlioptas et al. 2018].

5.1 Generating Deformations of Humans

To evaluate our method against a deformation space with a ground truth parameterization, we create a dataset of SMPL [Loper et al. 2015] humans in different yoga poses from the MOYO dataset [Tripathi et al. 2023], all with a resolution of *18k* faces. Previous works conducted similar reposing experiments [Aigerman et al. 2022; Maesumi et al. 2025] by training networks w.r.t. SMPL pose parameters; our method generates deformations without having any awareness of such parameters, making it a much harder task.

5.1.1 Single source generation. We train our model to generate deformations of an SMPL human mesh into various yoga poses. In Figure 6, we show generated samples on different triangulations (up to 1.2M triangles) of the source mesh. Our method consistently generates a diverse set of plausible, elaborate poses.

We check whether the generated deformation is within the SMPL deformation space by regressing the best-fit SMPL parameters to the

Fig. 10. **Elastic equilibrium states of a deflated bunny**. Our method generates diverse samples on the 70k resolution, though trained solely on 9k resolution.

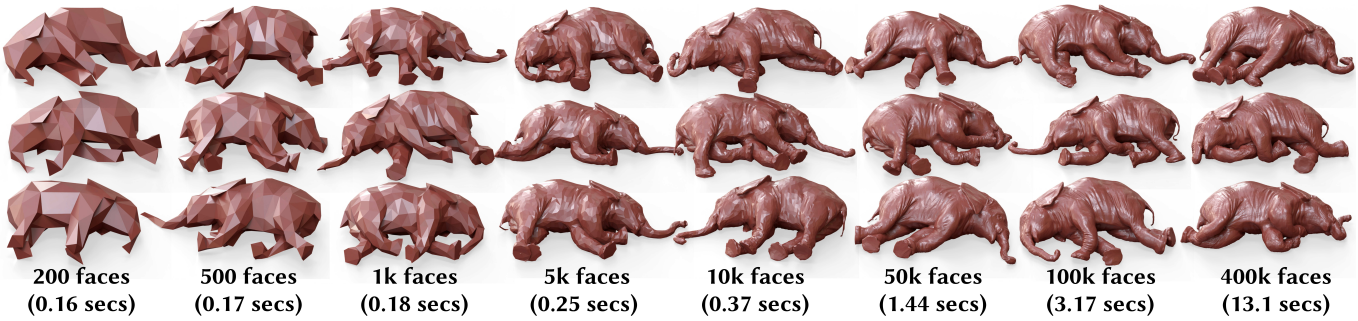


Fig. 11. **Timings for generating elastic rest states over varying resolutions.** Our method produces comparable results on all resolutions, even though it was trained solely on the 10k resolution. Generation time reported underneath each resolution.

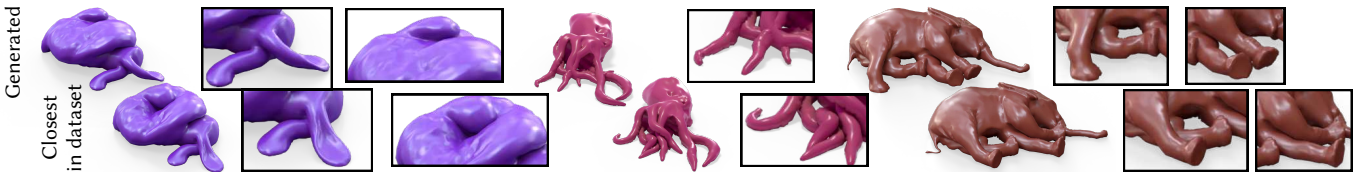


Fig. 12. **Closest sample in the training set to each generated result in the elastic deformation experiment.**

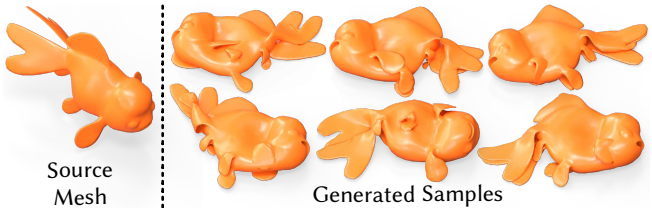


Fig. 13. **Elastic equilibrium of a fish with thin features.** Our method produces plausible deformations even in the thin regions of the fins.

generated mesh. More precisely, we deform the SMPL parametric model by simple gradient descent on the SMPL parameters w.r.t. vertex distance to the generated sample. We visualize the regressed SMPL model in Figure 9, and quantitatively compare the generated sample to it via Mean Squared Error (MSE) over the vertices, in Table 1. Our method produces results that well-conform to the parametric deformation space. Figure 9 further shows that the closest sample from the training set to our generated sample is quite far, entailing generalization.

5.1.2 Generation for arbitrary, unseen 3D human shapes. We use a similar setting as above to produce a model trained to generate deformations for *arbitrary* meshes, representing *different* 3D shapes unseen during training. We create a dataset of random (source, target) pairs of humans and train the model to generate the target vertices as a signal over the source mesh. To enable different source geometries, we use another PoissonNet [Maesumi et al. 2025] to predict a positional encoding over the source mesh, which is input as extra channels in the input to the generative model.

As Figure 8 shows, this single, trained model can be applied successfully to a wide range of human-like meshes of various shapes, attaining similar diversity and quality to that of the model from Section 5.1.1.

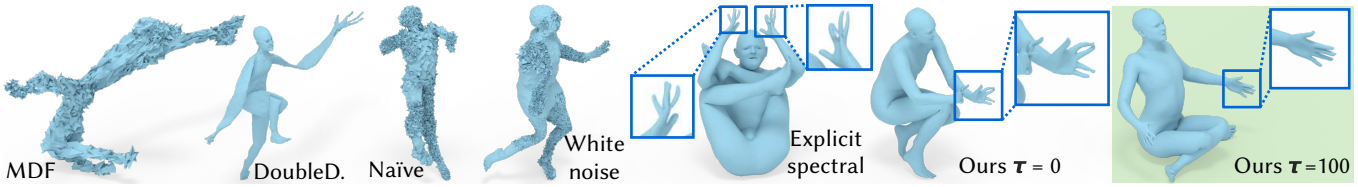


Fig. 14. **Comparisons and ablations**. We use the reposing experiment (Section 5.1.1) to compare to other mesh-based generative methods: MDF [Elhag et al. 2024], DoubleDiffusion [Wang et al. 2025a], the explicit method to compute Matérn noise, and ablations on other types of noise and screening terms.

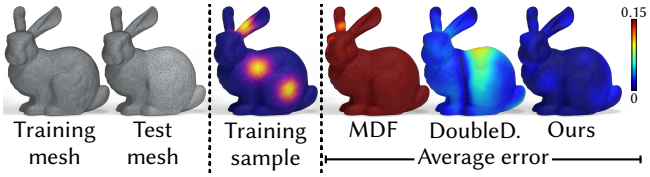


Fig. 15. **Comparisons to MDF [Elhag et al. 2024] and DoubleDiffusion [Wang et al. 2025a] on GMM.** As the competing methods were trained on a regular mesh, MDF and DoubleDiffusion fail to predict correct signals on the irregular test mesh, as shown by the average error between the GT distribution and the generated samples.

Table 1. Quantitative Results on the MOYO [Tripathi et al. 2023] dataset, comparing to DoubleDiffusion [Wang et al. 2025a] and MDF [Elhag et al. 2024], other noise models, and ablations.

Method	**MSE** $\cdot 10^4 \downarrow$	**MMD**$\downarrow$	**COV**$\uparrow$	**# params**
MDF	1411	0.5121	0.051	11.5M
DoubleDiffusion	155	0.1009	0.387	2.2M
Naïve	2.52	0.1365	0.210	1.4M
White noise	2.91	0.1499	0.189	1.4M
Explicit	0.51	0.0744	0.483	1.4M
Ours (no screening)	0.66	0.0897	0.418	1.4M
Ours (full)	**0.27**	**0.0693**	**0.487**	1.4M

Table 2. Comparison to MDF [Elhag et al. 2024] and DoubleDiffusion [Wang et al. 2025a] on the synthetic GMM dataset.

Method	**Test Error** $\downarrow$	**MMD** $(\times 10^{-2}) \downarrow$	**COV** $\uparrow$
MDF	0.5245	44.67	0.012
DoubleDiffusion	0.0325	5.497	0.506
Ours	**0.0071**	**4.420**	**0.601**

5.2 Emulating Elastic Deformations

We evaluate our method on a dataset of elastic deformations of an object. We generate training data by first creating a volumetric tetrahedral version of the object, and then running IPC [Li et al. 2020] to simulate dropping the object at a random initial orientation and angular speed on a ground plane, until it reaches equilibrium. We then extract the boundary triangle mesh from the deformed tetrahedral meshes and train our triangulation-agnostic flow matching framework on the rest states. Figure 1 shows a set of results on the octopus model, attaining elaborate configurations of its tentacles.

Note that triangulation agnosticism is critical for this task: when preparing the training set, the tetrahedral mesh used by IPC cannot accommodate the original, high-resolution boundary surface mesh (e.g., 400k faces for the elephant in Figure 11), hence the training mesh must be of much lower resolution (10k faces).

Figure 11 further explores the triangulation-agnosticism of our method by generating deformations on a wide range of resolutions. We produce plausible results for both extremes of the resolution range (up to the original mesh's resolution). We also report the generation time for each resolution, which is fast enough for real-world applications. In Figure 10, we show diverse generated samples of a hollow Stanford bunny at its original resolution (70k faces) using a model trained on much lower resolution data (9k faces). See also Figure 1 and Figure 13. In Figure 12, we show the closest sample in the training set to a generated sample. For each shape, our model learns and generalizes with plausible variations.

We note that we do not aim to perform neural physical simulation [Daviet et al. 2025], as our generated results have small inaccuracies which sometimes lead to collisions and penetration (see Figure 18). We solely use this dataset due to its very complex and challenging nature for a generative technique.

5.3 Comparisons

As far as we are aware, there are only two previous works that use modern generative approaches (namely, diffusion models) over meshes: Manifold Diffusion Fields (MDF) [Elhag et al. 2024] and DoubleDiffusion [Wang et al. 2025a]. MDF leverages a transformer-based denoising diffusion model [Ho et al. 2020] that operates on inputs positionally encoded via eigenfunctions. DoubleDiffusion uses DiffusionNet [Sharp et al. 2022] as the network backbone and trains a diffusion model using the naive iid Gaussian noise, which our experiments show is not triangulation-agnostic. Being the first works in mesh-based generation, they are not triangulation agnostic due to the inherent problem with naive iid Gaussian noise and to the transformer architecture not being triangulation-agnostic. We next compare to both methods in two experiments. We used DoubleDiffusion's publicly available implementation. MDF's code is not released, hence we used the authors' direct advice to instead use the implementation from [Wang et al. 2023] to run their method.

5.3.1 Generating deformations of human meshes. We train MDF and DoubleDiffusion on the human pose generation task (Section 5.1.1) and evaluate them in the same way as our model. As they are not triangulation agnostic, these methods are significantly less effective in generating diverse and plausible deformations, as Figure 14 and

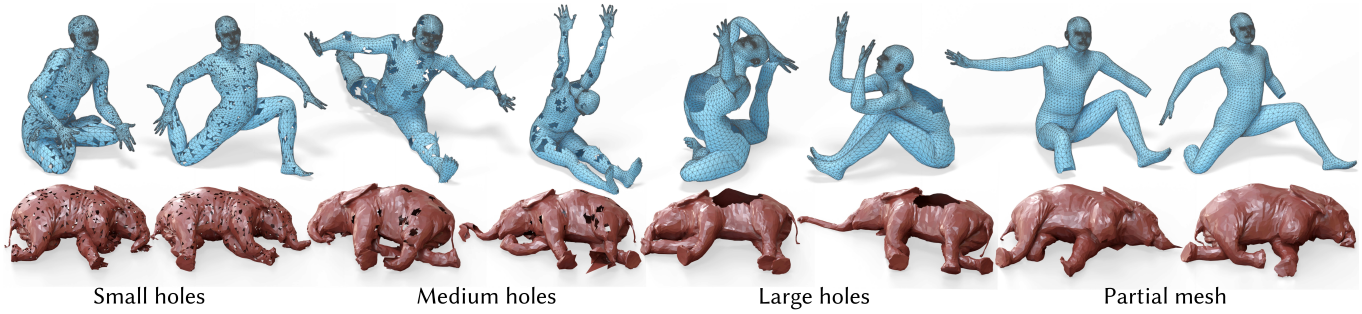


Fig. 16. **Generated results on topologically corrupted source meshes**. Trained solely on the *uncorrupted* source mesh, our method still produces valid results when parts of the source mesh are removed, exhibiting artifacts only near the modified boundaries.

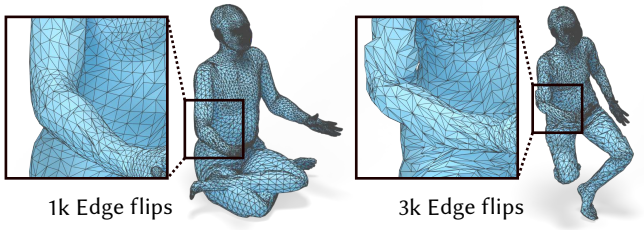


Fig. 17. **Results on source meshes of deteriorated triangulation quality.** We perform 1k and 3k edge flips to create badly triangulated source meshes to test our model, trained solely on *uncorrupted* source meshes.

Table 3. Our model's performance on source meshes of different corruption types and levels shown in Figure 16, evaluated on the MOYO [Tripathi et al. 2023] dataset following the same **MSE** $\cdot 10^4 \downarrow$ metric in Table 1.

Holes			Partial	Deteriorated		Original
small	medium	large	Mesh	mild	severe	Mesh
0.45	1.87	1.35	0.37	0.83	9.92	**0.27**

Table 1 show. Our method exceeds their capabilities both in matching the dataset (MMD, COV) as well as in matching the deformation space (MSE, see Section 5.1.1).

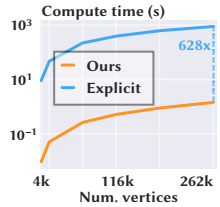
5.3.2 Generating Gaussian Mixtures. To further show the difference in triangulation-agnosticism between the methods, we recreate MDF's GMM experiment that learns to generate Gaussian mixtures, by preparing a dataset of Gaussian mixture signals on a regular triangulation of the Stanford Bunny, on which we train all methods.

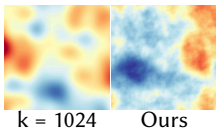
At test time, we generate samples on a non-uniformly triangulated test mesh. We evaluate how accurate the generated signals are w.r.t. the desired GMM by comparing their statistics: we compare the point-wise mean of a large set of generated signals on the mesh vs. the ground truth point-wise mean (evaluated by taking the mean of the training set) – see Table 2 (column *Test Error*). We outperform the two other methods. As shown in Figure 15, our method produces samples with correct statistical parameters. MDF shows a general deviation, while DoubleDiffusion shows clear biases in the generated results, correlating with the density of the triangulation.

We focus our comparisons to other generative methods on meshes, as other modalities solve rather different problems. In Appendix E in the supplemental material, we compare our method to a state-of-the-art point cloud generator [Ren et al. 2024b], and show the differences and advantages of using a mesh-based method.

5.4 Alternatives

Explicit spectral decomposition. After formally defining triangulation agnostic distributions (Section 3.3), we could have deduced a straightforward alternative: *explicitly* compute the first k eigenvectors of the Laplacian and form a random linear combination of them, $\mathbf{f} = \sum_i^k \frac{\chi_i}{\lambda_i + \tau} \boldsymbol{\phi}_i$, $\chi_i \sim \mathcal{N}(0,1)$. This holds several disadvantages: 1) The eigendecomposition is computationally expensive (see inset), especially due to the lack of reliable GPU solvers; 2) an eigenbasis of size 1024 still fails to produce as much high-frequency content as our noise—see inset. This, in turn, leads to less accurate results in Figure 14, particularly in fine-grained regions, e.g. the fingers. Table 1 verifies this quantitatively. We note that one can use more efficient techniques for computing eigenvectors, such as shift-invert spectral transforms that explore the spectrum band-by-band [Vallet and Lévy 2008]—this may enable other explicit constructions of triangulation-agnostic noise.





Latent generation. Other generative methods operate in a *latent space*, e.g., variational autoencoders [Kingma and Welling 2019], and, more recently, latent diffusion models [Rombach et al. 2022]. It is unclear whether latent generation outperforms generation in the signal domain [Li and He 2025; Yu et al. 2025]. Regardless, producing a good latent space in a triangulation-agnostic manner is as of now an unsolved problem, which prevents us from devising a worthy baseline to compare to. To show this, We trained a *non-variational* autoencoder consisting of a DiffusionNet [Sharp et al. 2022] encoder that encodes the input deformation into a 128 dimensional latent feature and a

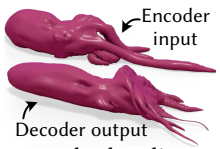


Fig. 18. **Physically inaccurate samples.** Our model generates plausible deformations for the elastic simulation experiment, however can produce inaccuracies that are physically impossible, e.g., the penetration into the ground plane on the right (seen from underneath), as well as self-penetration.

PoissonNet [Maesumi et al. 2025] decoder on our octopus deformation dataset. As the inset shows, the quality of the generated result is too poor to be useful.

5.5 Ablations

Other noise models. We evaluate the importance of our main contribution, the Matérn noise, by repeating the reposing experiment from Section 5.1.1 using both white noise (Equation (6)) and naïve iid Gaussian noise on the vertices. Figure 14 shows the generated deformations over a non-uniform triangulation. The other noise models fail, verifying the necessity of the properties stipulated in Section 3.3. Table 1 verifies this result quantitatively.

Effects of screening term τ. Training our model with Matérn noise without a screening term ($\tau = 0$) leads to deterioration in results on the reposing experiment from Section 5.1.1, as shown in Figure 14 and Table 1. The higher frequency components are essential to the model's success, especially on small parts (e.g. fingers).

5.6 Robustness

We next explore the robustness of the pipeline. Since both the Matérn noise and the backbone, PoissonNet [Maesumi et al. 2025] rely on Poisson's equation, the success of our framework hinges on the deterioration in the accuracy of Poisson's equation.

Topological corruption. After training the generative pipeline, we modify the topology of the source mesh by introducing holes, and evaluate the effect on the quality of the results. As Figure 16 shows, our method continues to generate plausible results (since Poisson's equation is not severely affected), with some deterioration next to the newly-introduced boundaries. Table 3 shows that the results drift from the SMPL parametric model, as expected. Note that Medium holes lead to higher MSE than Large holes, due to a larger total boundary length of the holes on surface (many medium-sized holes spread over the surface vs. four large-sized holes).

Triangulation quality. As the triangulation deteriorates in its condition number, both geometry and Poisson's equation are affected. While Figure 17 shows our method still produces plausible results, Table 3 shows that the accuracy of our generated results drops. We note this happens for extremely badly triangulated meshes (see zoom-in in Figure 17).

6 Conclusion

This work takes a step towards practical generative techniques for triangle meshes, and our experiments validate that it produces highly plausible results and is capable of emulating intricate datasets, making it already a viable option for real-world applications.

Limitations. Our method is still not a catch-all generative technique for meshes. A crucial bottleneck lies in the architecture of the denoiser's backbone, PoissonNet [Maesumi et al. 2025]. As with other triangulation-agnostic architectures [Sharp et al. 2022; Smirnov and Solomon 2021], it can only tackle tasks that do not require extreme high-frequency content, such as segmentation and deformation, thus preventing us from generating, e.g., crisp RGB images on meshes. Additionally, PoissonNet is limited to operating on single connected-component meshes. Lastly, the discussed architectures [Maesumi et al. 2025; Sharp et al. 2022] are limited to 2-manifolds (though our noise can be defined in any dimension). Our Matérn noise can be incorporated into future architectures that eventually emerge to solve these issues.

The Matérn noise distribution itself holds limitations. First, it is sensitive to topological changes (See Figure 16). Second, we assume nice enough approximations of the mesh [Wardetzky 2007, 2008] for our claims to hold—in practice, we observe that all models used in our experiments produce triangulation-agnostic noise (Figure 4).

Future work. In the immediate term, applications of generative deformations include producing variations of existing 3D models from a given template, retargeting existing poses or physical simulations, and adding details to existing mesh models [Liu et al. 2020]. Extensions of the architecture which seem potentially impactful include generating temporal animations of given meshes and extensions to volumetric data with applications in medical imaging.

We believe that Matérn processes have much more to offer to the graphics community, both in pure-graphics applications, e.g., modeling noise [Perlin 1985], as well as in non-meshy generative contexts, such as devising pixel-agnostic image generators.

Acknowledgments

This material is based upon work supported by: NSERC Discovery grant RGPIN-2024-04605, "Practical Neural Geometry Processing"; FRQNT Établissement de la relève professorale 365040, "Calcul rapide et léger des déformations à l'aide de réseaux neuronaux"; FRQNT team grant No. 361570; Digital Research Alliance of Canada, Compute Ontario, and Calcul Québec.

References

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. 2018. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*. PMLR, 40–49.

Abdalla GM Ahmed, Jing Ren, and Peter Wonka. 2022. Gaussian blue noise. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–15.

Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–17.

Michael S Albergo and Eric Vanden-Eijnden. 2022. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571* (2022).

Mirela Ben-Chen and Craig Gotsman. 2005. On the optimality of spectral compression of mesh data. *ACM Transactions on Graphics (TOG)* 24, 1 (2005), 60–80.

Thomas Besnier, Emery Pierson, Sylvain Arguillere, Maks Ovsjanikov, and Mohamed Daoudi. 2024. PaNDaS: Learnable Deformation Modeling with Localized Control. *arXiv preprint arXiv:2412.02306* (2024).

David Bolin, Alexandre Simas, and Jonas Wallin. 2023. Statistical inference for Gaussian Whittle-Mat\'ern fields on metric graphs. *arXiv preprint arXiv:2304.10372* (2023).

David Bolin, Alexandre B Simas, and Jonas Wallin. 2024. Gaussian Whittle–Matérn fields on metric graphs. *Bernoulli* 30, 2 (2024), 1611–1639.

Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. 2021. Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2593–2601.

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, et al. 2020. Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems* 33 (2020), 12426–12437.

Isaac Chavel. 1984. *Eigenvalues in Riemannian geometry*. Vol. 115. Academic press.

Ricky TQ Chen and Yaron Lipman. 2023. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660* (2023).

Sam Coveney, Cesare Corrado, Caroline H Roney, Daniel O'Hare, Steven E Williams, Mark D O'Neill, Steven A Niederer, Richard H Clayton, Jeremy E Oakley, and Richard D Wilkinson. 2020. Gaussian process manifold interpolation for probabilistic atrial activation maps and uncertain conduction velocity. *Philosophical Transactions of the Royal Society A* 378, 2173 (2020), 20190345.

Gilles Daviet, Tianchang Shen, Nicholas Sharp, and David IW Levin. 2025. Neurally integrated finite elements for differentiable elasticity on evolving domains. *ACM Transactions on Graphics* 44, 2 (2025), 1–17.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. 2022. Riemannian score-based generative modelling. *Advances in neural information processing systems* 35 (2022), 2406–2422.

Nam Anh Dinh, Itai Lang, Hyunwoo Kim, Oded Stein, and Rana Hanocka. 2025. Geometry in Style: 3D Stylization via Surface Normal Deformation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 28456–28467.

Emilien Dupont, Yee Whye Teh, and Arnaud Doucet. 2022. Generative Models as Distributions of Functions. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 2989–3015. https://proceedings.mlr.press/v151/dupont22a.html

Ahmed A. A. Elhag, Yuyang Wang, Joshua M. Susskind, and Miguel Ángel Bautista. 2024. Manifold Diffusion Fields. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=BZtEthuXRF

Marco Fumero, Michael Möller, and Emanuele Rodolà. 2020. Nonlinear spectral geometry processing via the TV transform. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–16.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. 2022. GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images. In *Advances In Neural Information Processing Systems*.

Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. 2019. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–15.

William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. 2023. Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 conference proceedings*. 1–11.

Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. 2025. Anytop: Character animation diffusion with any topology. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–10.

Michael Gillan, Stefan Siegert, and Ben Youngman. 2025. Discrete Gaussian Vector Fields On Meshes. *arXiv preprint arXiv:2507.20024* (2025).

Craig Gotsman, Xianfeng Gu, and Alla Sheffer. 2003. Fundamentals of spherical parameterization for 3D meshes. In *ACM SIGGRAPH 2003 Papers*. 358–363.

Xiaojie Guo and Liang Zhao. 2022. A systematic survey on deep generative models for graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (2022), 5370–5390.

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

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics (ToG)* 39, 4 (2020), 117–1.

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

Qixing Huang, Xiangru Huang, Bo Sun, Zaiwei Zhang, Junfeng Jiang, and Chandrajit Bajaj. 2021. Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5815–5825.

Xingchang Huang, Corentin Salaun, Cristina Vasconcelos, Christian Theobalt, Cengiz Oztireli, and Gurprit Singh. 2024. Blue noise for diffusion models. In *ACM SIGGRAPH 2024 conference papers*. 1–11.

Alexia Jolicoeur-Martineau, Kilian Fatras, Ke Li, and Tal Kachman. 2023. Diffusion models with location-scale noise. *arXiv preprint arXiv:2304.05907* (2023).

Zachi Karni and Craig Gotsman. 2000. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 279–286.

Hyunwoo Kim, Itai Lang, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. 2025. Meshup: Multi-target mesh deformation via blended score distillation. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 222–239.

Diederik P. Kingma and Max Welling. 2019. An Introduction to Variational Autoencoders. *Found. Trends Mach. Learn.* 12, 4 (2019), 307–392. doi:10.1561/2200000056

Thibault Lescoat, Hsueh-Ti Derek Liu, Jean-Marc Thiery, Alec Jacobson, Tamy Boubekeur, and Maks Ovsjanikov. 2020. Spectral mesh simplification. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 315–324.

Bruno Lévy. 2006. Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. IEEE, 13–13.

Bruno Lévy and Hao Zhang. 2010. Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses*. 1–312.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49.

Tianhong Li and Kaiming He. 2025. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720* (2025).

Zihao Li, Zhichen Zeng, Xiao Lin, Feihao Fang, Yanru Qu, Zhe Xu, Zhining Liu, Xuying Ning, Tianxin Wei, Ge Liu, et al. 2025. Flow matching meets biology and life science: a survey. *arXiv preprint arXiv:2507.17731* (2025).

Finn Lindgren, Håvard Rue, and Johan Lindström. 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 73, 4 (2011), 423–498.

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

Alexander H. Liu, Matthew Le, Apoorv Vyas, Bowen Shi, Andros Tjandra, and Wei-Ning Hsu. 2024. Generative Pre-training for Speech with Flow Matching. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=KpoQSgxbKH

Hsueh-Ti Derek Liu, Alec Jacobson, and Keenan Crane. 2017. A Dirac operator for extrinsic shape analysis. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 139–149.

Hsueh-Ti Derek Liu, Vladimir G Kim, Siddhartha Chaudhuri, Noam Aigerman, and Alec Jacobson. 2020. Neural subdivision. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 124–1.

Minghua Liu, Minhyuk Sung, Radomir Mech, and Hao Su. 2021. Deepmetahandles: Learning deformation meta-handles of 3d meshes with biharmonic coordinates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12–21.

Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003* (2022).

Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. 2023. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133* (2023).

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

Sining Lu, Guan Chen, Nam Anh Dinh, Itai Lang, Ari Holtzman, and Rana Hanocka. 2025. Ll3m: Large language 3d modelers. *arXiv preprint arXiv:2508.08228* (2025).

Danielle C Maddix, Nadim Saad, and Yuyang Wang. 2021. Modeling Advection on Directed Graphs using Mat\'ern Gaussian Processes for Traffic Flow. *arXiv preprint arXiv:2201.00001* (2021).

Arman Maesumi, Paul Guerrero, Noam Aigerman, Vladimir Kim, Matthew Fisher, Siddhartha Chaudhuri, and Daniel Ritchie. 2023. Explorable mesh deformation subspaces from unstructured 3d generative models. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Arman Maesumi, Tanish Makadia, Thibault Groueix, Vladimir Kim, Daniel Ritchie, and Noam Aigerman. 2025. PoissonNet: A Local-Global Approach for Learning on Surfaces. *ACM Transactions on Graphics (TOG)* 44, 6 (2025), 1–16.

Bertil Matérn. 1960. Spatial variation. Stochastic models and their application to some problems in forest surveys and other sampling investigations. (1960).

Emile Mathieu and Maximilian Nickel. 2020. Riemannian continuous normalizing flows. *Advances in neural information processing systems* 33 (2020), 2503–2515.

Daniel Mejia, Oscar Ruiz-Salguero, and Carlos A Cadavid. 2017. Spectral-based mesh segmentation. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 11, 3 (2017), 503–514.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13492–13502.

Sanjeev Muralikrishnan, Siddhartha Chaudhuri, Noam Aigerman, Vladimir G Kim, Matthew Fisher, and Niloy J Mitra. 2022. Glass: Geometric latent augmentation for

shape spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18552–18561.

Sanjeev Muralikrishnan, Niladri Dutt, Siddhartha Chaudhuri, Noam Aigerman, Vladimir Kim, Matthew Fisher, and Niloy J Mitra. 2024. Temporal residual jacobians for rig-free motion transfer. In *European Conference on Computer Vision*. Springer, 93–109.

Eliya Nachmani, Robin San Roman, and Lior Wolf. 2021. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582* (2021).

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. 2020. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*. PMLR, 7220–7229.

Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–13.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 1–11.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 10975–10985.

Ken Perlin. 1985. An image synthesizer. *ACM Siggraph Computer Graphics* 19, 3 (1985), 287–296.

Emery Pierson, Lei Li, Angela Dai, and Maks Ovsjanikov. 2025. DiffuMatch: Category-Agnostic Spectral Diffusion Priors for Robust Non-rigid Shape Matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5745–5756.

Ulrich Pinkall and Konrad Polthier. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1 (1993), 15–36.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. 2024a. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4209–4219.

Zhiyuan Ren, Minchul Kim, Feng Liu, and Xiaoming Liu. 2024b. TIGER: Time-varying denoising model for 3D point cloud generation with diffusion process. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9462–9471.

Avigail Cohen Rimon, Mirela Ben-Chen, and Or Litany. 2025. FRIDU: Functional Map Refinement with Guided Image Diffusion. In *Computer Graphics Forum*, Vol. 44. Wiley Online Library, e70203.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

Paul Rosa, Slava Borovitskiy, Alexander Terenin, and Judith Rousseau. 2023. Posterior contraction rates for Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems* 36 (2023), 34087–34121.

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. 2024. Human Motion Diffusion as a Generative Prior. In *The Twelfth International Conference on Learning Representations*.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. 2022. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)* 41, 3 (2022), 1–16.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2023. MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers. *arXiv preprint arXiv:2311.15475* (2023).

Dmitriy Smirnov and Justin Solomon. 2021. HodgeNet: Learning spectral geometry on triangle meshes. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–11.

Ran Song, Yonghuai Liu, Ralph R Martin, and Paul L Rosin. 2014. Mesh saliency via spectral processing. *ACM Transactions On Graphics (TOG)* 33, 1 (2014), 1–17.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

Michael L Stein. 1999. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 1383–1392.

Ramana Sundararaman, Nicolas Donati, Simone Melzi, Etienne Corman, and Maks Ovsjanikov. 2024. Deformation recovery: Localized learning for detail-preserving deformations. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–16.

Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. 2018. Variational autoencoders for deforming 3d mesh models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5841–5850.

Qingyang Tan, Ling-Xiao Zhang, Jie Yang, Yu-Kun Lai, and Lin Gao. 2021. Variational autoencoders for localized mesh deformation component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (2021), 6297–6310.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*. Springer, 1–18.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=SJ1kSyO2jwu

Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 2023. 3D human pose estimation via intuitive physics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4713–4725.

Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. 2022. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems* 35 (2022), 10021–10039.

Bruno Vallet and Bruno Lévy. 2008. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 251–260.

Duotun Wang, Hengyu Meng, Zeyu Cai, Zhijing Shao, Qianxi Liu, Lin Wang, Mingming Fan, Xiaohang Zhan, and Zeyu Wang. 2025b. Headevolver: Text to head avatars via expressive and attribute-preserving mesh deformation. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 211–221.

Xuyang Wang, Ziang Cheng, Zhenyu Li, Jiayu Yang, Haorui Ji, Pan Ji, Mehrtash Harandi, Richard Hartley, and Hongdong Li. 2025a. DoubleDiffusion: Combining Heat Diffusion with Denoising Diffusion for Texture Generation on 3D Meshes. *arXiv preprint arXiv:2501.03397* (2025).

Yuyang Wang, Ahmed A Elhag, Navdeep Jaitly, Joshua M Susskind, and Miguel Angel Bautista. 2023. Swallowing the bitter pill: Simplified scalable conformer generation. *arXiv preprint arXiv:2311.17932* (2023).

Max Wardetzky. 2007. *Discrete differential operators on polyhedral surfaces-convergence and approximation*. Ph. D. Dissertation.

Max Wardetzky. 2008. Convergence of the cotangent formula: An overview. *Discrete differential geometry* (2008), 275–286.

Peter Whittle. 1963. Stochastic-processes in several dimensions. *Bulletin of the International Statistical Institute* 40, 2 (1963), 974–994.

Christopher KI Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*. Vol. 2. MIT press Cambridge, MA.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2025. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21469–21480.

Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X Chang. 2025. An object is worth 64× 64 pixels: Generating 3d object via image diffusion. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 123–133.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. 2019. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4541–4550.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. 2025. PixelDiT: Pixel Diffusion Transformers for Image Generation. *arXiv preprint arXiv:2511.20645* (2025).

Yu-Jie Yuan, Yu-Kun Lai, Jie Yang, Qi Duan, Hongbo Fu, and Lin Gao. 2020. Mesh variational autoencoders with edge contraction pooling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 274–275.

Hao Zhang, Oliver Van Kaick, and Ramsay Dyer. 2010. Spectral mesh processing. In *Computer graphics forum*, Vol. 29. Wiley Online Library, 1865–1894.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Mengchun Zhang, Maryam Qamar, Taegoo Kang, Yuna Jung, Chenshuang Zhang, Sung-Ho Bae, and Chaoning Zhang. 2023. A survey on graph diffusion models: Generative ai in science for molecule, protein and material. *arXiv preprint arXiv:2304.01565* (2023).

Lei Zhong, Chuan Guo, Yiming Xie, Jiawei Wang, and Changjian Li. 2025. Sketch2anim: Towards transferring sketch storyboards into 3d animation. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–15.

Aleksei Zhuravlev, Zorah Lähner, and Vladislav Golyanik. 2025. Denoising functional maps: Diffusion models for shape correspondence. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 26899–26909.

A Proof of the Properties from Section 3.3

Property 1. The covariance matrix in Equation (11) is diagonal and hence each coefficient $\widehat{\mathbf{f}}_i$ is sampled independently.

Property 2. Let two meshes $\mathcal{M}, \mathcal{N}$ have coefficients $\widehat{\mathbf{f}}_i^{\mathcal{M}}, \widehat{\mathbf{f}}_j^{\mathcal{N}}$ with corresponding distributions $\widehat{\mathcal{D}}_i^{\mathcal{M}}, \widehat{\mathcal{D}}_j^{\mathcal{N}}$ and corresponding eigenvalues $\lambda_i^{\mathcal{M}}, \lambda_j^{\mathcal{N}}$, respectively, with $\left|\lambda_i^{\mathcal{M}} - \lambda_j^{\mathcal{N}}\right| < \varepsilon$. Then

$$W_2\left(\widehat{\mathcal{D}}_i^{\mathcal{M}}, \widehat{\mathcal{D}}_j^{\mathcal{N}}\right) = \left(\left(\lambda_i^{\mathcal{M}} + \tau\right)^{-1} - \left(\lambda_j^{\mathcal{N}} + \tau\right)^{-1}\right)^2 \leq \frac{\varepsilon^2}{\tau^4} = O\left(\varepsilon^2\right)$$

as ε goes to zero, where we used the trivial definition of the 2-Wasserstein distance W_2 for univariate normal distributions for the first equality.

Property 3. Per the assumed approximation of the continuous spectrum, the eigenvalues obey *Weyl's law* [Chavel 1984], applied to a 2-manifold with area A, satisfying the asymptotic relation $\lambda_j \sim \frac{4\pi}{A} j$. Therefore,

$$\mathrm{Var}\left[\sum_{i=k}^{n} \widehat{\mathbf{f}}_i\right] = \sum_{i=k}^{n} \frac{1}{\left(\lambda_i + \tau\right)^2} \sim \sum_{i=k}^{n} \frac{1}{\left(\frac{4\pi}{A} i + \tau\right)^2} < \sum_{i=k}^{n} \frac{A^2}{i^2},$$

which is a convergent series as n tends to infinity. Hence, there exists k s.t. the above expression is smaller than ε, for any n (i.e., any mesh resolution).

B Implementation Details

B.1 Shape Deformation Network

Our shape deformation network consists of five PoissonNet [Maesumi et al. 2025] blocks. Each block has hidden dimension of 128 and VectorMLPs with three layers. Our network takes the noisy mesh's xyz as input, and condition the source mesh xyz and the timestep t at each block. We follow PoissonNet [Maesumi et al. 2025] to use the same NJF head [Aigerman et al. 2022] at the end of the network to predict the flow velocity in Jacobian space, which can be transformed to velocity in vertex space via solving Poisson's equation. We set the batch size to be 32, and use equal weights for conditional flow matching losses in vertex and Jacobian space. During training, we clip the norm of the gradient to be below 1. We set the screening term $\tau = 100$ for all the experiments. The timings we report in all the experiments were conducted on a single NVIDIA RTX 4090 GPU.

Single source generation and elastic deformation models. We train the network for 160k iterations with a learning rate of 0.0005 for the first 40k iterations, and a learning rate of 0.0001 for rest of the iterations. All trainings were conducted on a single NVIDIA H100 GPU. The total training time of a model is sensitive to the resolution of the source mesh. It takes around 42 hours to train the single source generation (source mesh at resolution of 18k faces) and around 36 hours to train the elastic deformation models (source mesh at resolution of 9k/10k faces).

Arbitrary 3D human shapes model . We train the network for 240k iterations with an initial learning rate of 0.0005, and multi-step decay of rate 0.2 at 40k and 160k iterations. Besides the source mesh xyz and timestep t, we provide the predictions from another pretrained eigenvector predictor as additional conditioning signals to each PoissonNet block. They serve as intrinsic positional encoding on the given source mesh. The eigenvector predictor is trained on SMPL human meshes of random poses and shapes with the same training target (i.e. the normalized eigenvectors of the neutral T-pose SMPL human mesh's Laplacian that correspond to the smallest 64 eigenvalues). This eigenvector predictor can provide consistent positional encoding on arbitrary 3D human-like meshes, including both SMPL meshes in arbitrary poses and shapes, and general humanoid meshes, which plays an important role in the success of this model. We cache Cholespy solvers [Nicolet et al. 2021] for all the source meshes, and move them to the GPU when needed instead of creating them on-the-fly to accelerate the training. We train the model on a single NVIDIA H100 GPU for 80 hours. We apply random global scaling and shifts to the source meshes as data augmentations when training both the eigenvector predictor and the FM model to further help the models generalize on arbitrary sources.

B.2 Deformation Autoencoder

We made the first attempt to explore the possibility of encoding deformations via an autoencoder in Sec. 5.4. Our encoder has five DiffusionNet [Sharp et al. 2022] blocks using spectral size of 128. It encodes the input deformation into a 128-dimensional latent code by applying a global mean at the end of the encoder. The latent code is served as the conditioning signal to a PoissonNet [Maesumi et al. 2025] decoder, consisting of five blocks as well. The encoder and decoder's hidden dimensions are 128 and 256, respectively. We train the autoencoder on our Octopus dataset with a learning rate of 0.001 and batch size of 16 for 40k iterations.

C Datasets

C.1 SMPL datasets

We create datasets of deformed SMPL human meshes in yoga poses from the MOYO dataset [Tripathi et al. 2023]. We generate a dataset of 64k deformed human meshes for the single source generation model and the eigenvector predictor used in arbitrary human source model, and 32k pairs of deformed human meshes as source-target pairs for the arbitrary human source model. We follow [Maesumi et al. 2025] to sample diverse deformed poses from MOYO dataset's motion captures via a greedy farthest point sampling scheme and generate human bodies using the SMPL-X [Pavlakos et al. 2019] parametric model. We keep the hands to always be in neutral poses in our datasets. The deformed shapes used in single source generation model are kept to be in a neutral shape, while in arbitrary human source model the body shape parameters are randomly sampled from a Gaussian distribution with a standard deviation of 5 to obtain diverse human bodies. All the meshes in the dataset are at resolution of 18k faces.

C.2 Elastic deformations datasets

We apply a physical simulation method, IPC [Li et al. 2020], to simulate the resting states of elastic object on the floor. We downsample our high-resolution mesh assets and convert them to tetrahedral meshes via fTetWild [Hu et al. 2020]. We simulate object dropping

with random initial orientation, as well as initial angular speed randomly sampled between π and 3π. All objects have a Young's modulus, Poisson's ratio, and density of 1e4, 0.4, and 1e3, respectively. The simulation runs till the object reaches an equilibrium state on the floor and we collect it as one deformed sample in the training dataset. We convert the resting states back to triangle meshes. To remove the global rotation of the object around the y-axis (up-direction in our coordinate system), we apply PCA on each shape's x and z coordinates, and rotate the mesh such that its largest principal component aligns with the largest component of the source mesh (the canonical template mesh used in simulation). We create datasets for four objects: Octopus, Elephant, Fish, and the Stanford Bunny, and for each object we sample 20k diverse resting states (via farthest point sampling) from 30k post-PCA simulation results. The original Octopus, Elephant, and Stanford Bunny meshes have 100k, 400k, and 70k faces, respectively. The low resolution meshes used to train our models for Octopus, Elephant, and Stanford Bunny have 9k, 10k, and 9k faces, respectively. We use Fish in original resolution (7k faces) to train our model.

C.3 GMM datasets

We follow the setting from MDF [Elhag et al. 2024] to create datasets of random Gaussian Mixture Models (GMM) on a uniformly triangulated Stanford Bunny of 70k faces. We randomly sample GMM of three non-overlapping centers on the mesh. Note that the probability of sampling a vertex as Gaussian center is weighted by its vertex mass, to ensure uniform distribution of Gaussians on the surface. We create a dataset of 32k random GMMs as the training dataset. We also create a test mesh by downsampling the left half of the bunny, resulting in a mesh of non-uniform triangulation (40k faces). We generate 1k random GMMs on the test mesh using the same sampling strategy, yielding a reference set for evaluation.

D Evaluation Metrics

MMD and COV. As initially proposed by [Achlioptas et al. 2018], MMD reports the average distance between generated samples and their closest samples in the reference set, providing a measure of fidelity. On the other hand, COV computes the percentage of the samples in the reference set being closest sample to the generated data. These two metrics are complementary and together they represent the quality of the generation. We use average $l2$ distance weighted by the corresponding vertex mass as the distance metrics for MMD and COV.

SMPL deformation metrics. To measure how realistic the generated deformations are, we introduce a new metric for quantitatively evaluating the generated SMPL shapes, by taking advantage of the differentiable SMPL-X [Pavlakos et al. 2019] parametric model. We regress the SMPL pose parameters that best match the generated shape by gradient descent. We then use the SMPL-X model and the obtained pose parameters to generate a deformed mesh. We measure the mean squared error (MSE) weighted by the vertex mass as our metric. This metric is more accurate than the MMD in terms of measuring how close the generated samples are to the real deformation space of human.

Table 4. Comparison with baseline methods.

Method	MMD ($\times 10^{-2}$) ↓		COV ↑		1-NNA ↓	
	CD	EMD	CD	EMD	CD	EMD
TIGER	**0.56**	**2.53**	0.456	0.456	**0.595**	**0.635**
Ours	0.89	3.05	**0.490**	**0.510**	0.694	0.669

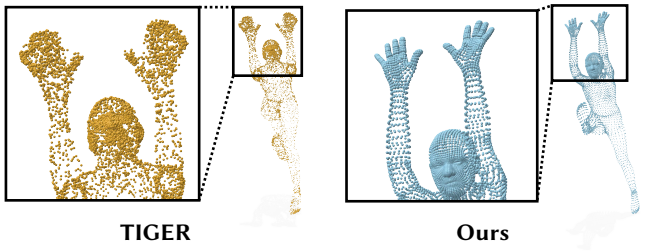


Fig. 19. **Comparison to state of the art in point cloud generation, TIGER [Ren et al. 2024b].** By using a triangulation-agnostic intrinsic network over a mesh, we are able to preserve and produce minute details such as the face and individual fingers; these are completely gone from the point cloud generated by TIGER.

E Comparison to Point Cloud Generation

Different 3D representations serve different goals (e.g., voxels are better fit for volumetric data), and hence generative methods on meshes cannot be compared directly to other representations, as their goal is to produce details on the vertices of an existing mesh. To further display this point, we evaluate the differences between our mesh-based generative method and a point cloud generative method, TIGER [Ren et al. 2024b], which is currently the state-of-the-art diffusion-based method that applies noising and denoising on point clouds directly. We compare the two on the experiment from Section 5.1.1. We use the vertices of the SMPL human deformation dataset as point clouds, and train TIGER from scratch on our dataset to generate deformed SMPL human as point clouds.

We then compare to our method by treating its generated samples as point clouds as well. We use the commonly used metrics for point cloud generation (MMD, COV, and 1-NNA) under two different distance metrics, Chamfer distance (CD) and Earth Mover's Distance (EMD) (see TIGER [Ren et al. 2024b]). Table 4 shows that TIGER slightly outperforms us on most of these metrics, implying better adherence to the target distribution. These results make sense as each method aims for a different task; namely, our method needs to also account for the mesh connectivity and not just the points.

The gain from using a mesh as opposed to a point cloud is critical, and can be appreciated in Figure 19: while TIGER [Ren et al. 2024b] generates a point cloud that well-minimizes the chamfer distance to a valid SMPL human, the lack of connectivity information leads it to completely omit small geometric details (which chamfer distance is insensitive to and hence does not quantify): separate fingers with gaps are non-existent, and the model has a head but no face. This is in contrast to the point cloud from our mesh-based network, which clearly exhibits fingers and a face, due to our method utilizing

a triangulation-agnostic framework which operates on the mesh intrinsically and thereby preserves minute detail.

F FEM Discretization

Given a signal $\mathbf{f}$, we have values $\mathbf{f}_i$ on the vertices; we interpret these values in standard linear finite-elements fashion, as representing a piecewise-linear function $\mathbf{f}(p)$ defined over the triangles as the unique function that is linear on each triangle and interpolates the values $\mathbf{f}_i$ on the vertices. Given another signal $\mathbf{g}$, the inner product between the two piecewise-linear functions is defined as the integral over the domain Ω:

$$\langle f, g \rangle_{L^2} = \int_\Omega f(p) g(p)\, dA. \tag{13}$$

Substituting the interpolants $f(p) = \sum_i f_i \phi_i(p)$ and $g(p) = \sum_j g_j \phi_j(p)$, where $\{\phi_i\}$ are the standard Lagrange "hat" basis functions associated with each vertex, the inner product is expressed in matrix form as $\mathbf{f}^\top \mathbf{M} \mathbf{g}$. $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the (consistent) *mass matrix*, which is a symmetric matrix with entries:

$$\mathbf{M}_{ij} = \int_\Omega \phi_i(p) \phi_j(p)\, dA. \tag{14}$$

To simplify the metric and diagonalize the system, the *lumped* mass matrix $\mathbf{M}$ is frequently employed. This diagonal matrix is defined by summing the rows of the consistent mass matrix:

$$(\mathbf{M})_{ii} = \int_\Omega \phi_i(p)\, dA. \tag{15}$$

Geometrically, $(\mathbf{M})_{ii}$ corresponds to the area of the dual cell associated with vertex i. In practice, this is implemented as the barycentric area (one-third of the area of all incident triangles).

The cotangent Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$ is often referred to as the stiffness matrix, and represents the discretization of the Dirichlet energy $\frac{1}{2} \int_\Omega \|\nabla f\|^2\, dA$. Its entries are given by:

$$L_{ij} = \int_\Omega \nabla \phi_i(p) \cdot \nabla \phi_j(p)\, dA. \tag{16}$$

For a triangle mesh, this yields the standard cotangent weight formula for $i \neq j$:

$$L_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}), \tag{17}$$

where α_{ij} and β_{ij} are the angles opposite the edge (i, j). The diagonal entries are defined such that rows sum to zero: $L_{ii} = -\sum_{j \in \mathcal{N}(i)} L_{ij}$, where $\mathcal{N}(i)$ is the neighborhood of i (excluding i itself).