Single Mesh Diffusion Models with Field Latents for Texture Generation

Supplementary Material

8. Equivariance

Here we provide detailed proofs of the claims made in Sections 4 and 5 regarding sufficient conditions for isometry-equivariance.

8.1. Field Latents

Consider the FL variational autoencoder consisting of the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ defined pointwise as in Equations (3) and (4), respectively.

Claim 1. *If for all $\psi \in L^2(M, \mathbb{R}^n)$, isometries $\gamma : M \to N$, and points $p \in M$, the encoded mean and standard deviation satisfy the condition of Equation (7)*

$$d\gamma|_p \cdot \mu_p^\psi = \mu_{\gamma(p)}^{\gamma\psi} \qquad \textit{and} \qquad \sigma_p^\psi = \sigma_{\gamma(p)}^{\gamma\psi},$$

then the FL-VAE commutes with isometries such that

$$\widehat{[\gamma\psi]}_{\gamma(p)} = \gamma\widehat{\psi}_p.$$

Proof. First, we observe that for any isometry $\gamma : M \to N$ and point $p \in M$, the differential $d\gamma|_p$ is an orthogonal transformation taking vectors in T_pM to $T_{\gamma(p)}N$. Thus, within our complexification of the tangent space, $d\gamma|_p$ is an element of $\mathrm{U}(1)$, the group of complex numbers with unit modulus, acting multiplicatively on $\mathbb{C}_p$.

Now, consider any $\psi \in L^2(M, \mathbb{R}^n)$, isometry $\gamma : M \to N$, point $p \in M$, and suppose the encoded mean μ_p^ψ and standard deviation σ_p^ψ satisfy the condition. Then, we can relate latent codes distributed in $T_{\gamma(p)}N$ to those in T_pM with

$$\begin{aligned} z_{\gamma(p)}^{\gamma\psi} &= \mu_{\gamma(p)}^{\gamma\psi} + \sigma_{\gamma(p)}^{\gamma\psi} \odot \epsilon'(\gamma(p)), \quad \epsilon' \sim \mathcal{TN}_N(0, I_d) \\ &\overset{(2)}{=} \mu_{\gamma(p)}^{\gamma\psi} + d\gamma|_p \cdot \sigma_{\gamma(p)}^{\gamma\psi} \odot \epsilon(p), \quad \epsilon \sim \mathcal{TN}_M(0, I_d) \\ &\overset{(7)}{=} d\gamma|_p \cdot \mu_p^\psi + d\gamma|_p \cdot \sigma_p^\psi \odot \epsilon(p) \\ &= d\gamma|_p \cdot z_p^\psi \end{aligned} \tag{15}$$

Using the relationship between the latent codes and the fact that $\log_{\gamma(p)} \gamma(q) = d\gamma|_p \cdot \log_p q$ for $q \in B_p$ as long as γ is an isometry, it follows that the coordinate function in Equation (5) is invariant with

$$\begin{aligned} c_{\gamma(p)\gamma(q)}^{\gamma\psi} &\overset{(5)}{=} \log_{\gamma(p)} \gamma(q) \cdot \overline{z_{\gamma(p)}^{\gamma\psi}} \\ &\overset{(15)}{=} d\gamma|_p \cdot \log_p q \cdot \overline{d\gamma|_p} \cdot \overline{z_p^\psi} \\ &\overset{(5)}{=} c_{pq}^\psi, \end{aligned} \tag{16}$$

with the last equality following from the fact that $d\gamma|_p \cdot \overline{d\gamma|_p} = 1$. Similarly, it follows that

$$\begin{aligned} z_{\gamma(p)}^{\gamma\psi} \left[z_{\gamma(p)}^{\gamma\psi}\right]^* &\overset{(15)}{=} d\gamma|_p \cdot \overline{d\gamma|_p} \cdot z_p^\psi \left[z_p^\psi\right]^* \\ &= z_p^\psi \left[z_p^\psi\right]^*. \end{aligned} \tag{17}$$

Now, denoting $\mathcal{F}$ as the decoder neural field such that

$$\widehat{\psi}_p(q) \equiv \mathcal{F}\left(c_{pq}^\psi, \mathrm{vec}_{j \geq i}\left(z_p^\psi \left[z_p^\psi\right]^*\right)\right), \tag{18}$$

it follows that for all $q' \in B_{\gamma(p)} \subset N$ we have

$$\begin{aligned} \widehat{[\gamma\psi]}_{\gamma(p)}(q') &\overset{(18)}{=} \mathcal{F}\left(c_{\gamma(p)q'}^{\gamma\psi}, \mathrm{vec}_{j \geq i}\left(z_{\gamma(p)}^{\gamma(\psi)} \left[z_{\gamma(p)}^{\gamma\psi}\right]^*\right)\right) \\ &= \mathcal{F}\left(c_{p\gamma^{-1}(q')}^{\psi}, \mathrm{vec}_{j \geq i}\left(z_p^\psi \left[z_p^\psi\right]^*\right)\right) \\ &\overset{(18)}{=} \gamma\widehat{\psi}_p(q'), \end{aligned}$$

with the equality $c_{\gamma(p)q'}^{\gamma\psi} = c_{p\gamma^{-1}(q')}^{\psi}$ following from Equation (16) and the inveritibility of isometries. Thus we have $\widehat{[\gamma\psi]}_{\gamma(p)} = \gamma\widehat{\psi}_p$ as desired. □

8.2. Field Latent Diffusion Models

Consider the denoising network ε as in Equation (9) and the forward and reverse diffusion processes defined in Equations (8) and (11).

Claim 2. *If for all latent vector fields $Z \in TM^d$, isometries $\gamma : M \to N$, embeddings $\rho \in L^2(M, \mathbb{R}^m)$, and timestep $0 \leq t \leq T$, the denoising network satisfies the condition of Equation (13)*

$$\gamma\varepsilon(Z, t, \rho) = \varepsilon(\gamma Z, t, \gamma\rho),$$

then both the forward and reverse diffusion processes commute with isometries with

$$\gamma Z_t = [\gamma Z]_t \qquad \textit{and} \qquad \gamma\widetilde{Z}_t = \left[\gamma\widetilde{Z}\right]_t.$$

Note: At the risk of abusing notation, we use the subscript t to refer to both the forward and reverse diffusion processes. To disambiguate the two, we denote latent vector fields sampled from the reverse process with a tilde, *e.g.* $\widetilde{Z}_t \in TM^d$.

Proof. Consider any latent vector fields $Z \in TM^d$ and isometry $\gamma : M \to N$. The forward process does not de-

pend on the denoising network and demonstrating equivariance is straight-forward as

$$\begin{aligned}\gamma Z_t &\overset{(8)}{=} \sqrt{\alpha_t}\gamma Z + \sqrt{1-\alpha_t}\gamma\epsilon, \quad \epsilon \sim \mathcal{TN}_M(0, I_d) \\ &\overset{(2)}{=} \sqrt{\alpha_t}\gamma Z + \sqrt{1-\alpha_t}\epsilon', \quad \epsilon' \sim \mathcal{TN}_N(0, I_d) \\ &\overset{(8)}{=} [\gamma Z]_t\end{aligned}$$

Now, suppose the denoising network ε satisfies the condition. Equivariance of the reverse process can be shown through induction, with the base case $t = T$ holding as

$$\begin{aligned}\gamma\widetilde{Z}_T &= \gamma\widetilde{Z}, \quad \widetilde{Z} \sim \mathcal{TN}_M(0, I_d) \\ &\overset{(2)}{=} \gamma\widetilde{Z}, \quad \gamma\widetilde{Z} \sim \mathcal{TN}_N(0, I_d) \\ &= \left[\gamma\widetilde{Z}\right]_T\end{aligned}$$

Then, assuming $\gamma\widetilde{Z}_\tau = \left[\gamma\widetilde{Z}\right]_\tau$ holds at step τ, for step $\tau-1$ we have

$$\begin{aligned}\gamma\widetilde{Z}_{\tau-1} &\overset{(11)}{=} C_1(\alpha_\tau, \alpha_{\tau-1})\,\gamma\widetilde{Z}_\tau \\ &\quad + C_2(\alpha_\tau, \alpha_{\tau-1})\,\gamma\varepsilon(\widetilde{Z}_\tau, \tau, \rho) \\ &\quad + C_3(\alpha_\tau, \alpha_{\tau-1})\,\gamma\epsilon, \quad \epsilon \sim \mathcal{TN}_M(0, I_d) \\ &\overset{(2)}{=} C_1(\alpha_\tau, \alpha_{\tau-1})\,\gamma\widetilde{Z}_\tau \\ &\quad + C_2(\alpha_\tau, \alpha_{\tau-1})\,\gamma\varepsilon(\widetilde{Z}_\tau, \tau, \rho) \\ &\quad + C_3(\alpha_\tau, \alpha_{\tau-1})\,\epsilon', \quad \epsilon' \sim \mathcal{TN}_N(0, I_d) \\ &\overset{(13)}{=} C_1(\alpha_\tau, \alpha_{\tau-1})\,\gamma\widetilde{Z}_\tau \\ &\quad + C_2(\alpha_\tau, \alpha_{\tau-1})\,\varepsilon(\gamma\widetilde{Z}_\tau, \tau, \gamma\rho) \\ &\quad + C_3(\alpha_\tau, \alpha_{\tau-1})\,\epsilon' \\ &= C_1(\alpha_\tau, \alpha_{\tau-1})\left[\gamma\widetilde{Z}\right]_\tau \\ &\quad + C_2(\alpha_\tau, \alpha_{\tau-1})\,\varepsilon(\left[\gamma\widetilde{Z}\right]_\tau, \tau, \gamma\rho) \\ &\quad + C_3(\alpha_\tau, \alpha_{\tau-1})\,\epsilon' \\ &= \left[\gamma\widetilde{Z}\right]_{\tau-1},\end{aligned}$$

with the second to last equality following from the induction assumption. Thus, by induction we have $\gamma\widetilde{Z}_t = \left[\gamma\widetilde{Z}\right]_t$ for $0 \le t \le T$ as desired. □

9. Implementation Details

Tangent bases At each vertex, the associated tangent space lies in a plane perpendicular to the vertex normal direction. We construct orthonormal bases in the tangent space at a given vertex by simply choosing an arbitrary unit vector perpendicular to the normal as the $x-$axis, then taking its cross product with the normal to get the $y-$axis.

Linearities, Nonlinearities, and Normalization With the exception of the decoder neural field $\mathcal{F}$ as in Equation (18) and a few layers in the encoder, all of our networks are complex — all linear layers and convolutional filters have complex-valued weights, with additive biases omitted to preserve equivariance.

For both FL-VAE and FLDM, we use modified versions of Vector Neurons (**VNs**) [10] and Filter Response Normalization [49] as nonlinear and normalization layers for tangent vector features. Nonlinearities are learnable mappings between TM^C and $TM^{C'}$ applied point-wise. Letting $K, Q \in TM^{C'}$ be the output of two learnable linear layers taking features $X \in TM^C$ as input, the $c'-$th output feature $X'_{c'} \in TM$ of our nonlinear layers is given by

$$X'_{c'}(p) = Q_{c'}(p) + \frac{\text{ReLU}\left(-\text{Re}(\overline{Q_{c'}(p)}K_{c'}(p))\right)}{|K_{c'}(p)|^2 + \delta}K_{c'}(p), \tag{19}$$

where $\delta > 0$ a constant and the argument of the ReLU function is the inner product of $Q_{c'}(p)$ and $K_{c'}(p)$ in T_pM.

Similarly, given tangent vector features $X \in TM^C$, normalization is applied on a per-channel basis independent of the batch size via the mapping

$$X_c \mapsto \frac{a_c X_c}{\sqrt{\mathbb{E}_{p\in M}|X_c(p)|^2 + \delta_c}}, \tag{20}$$

where $a_c \in \mathbb{C}$ and $\delta_c \in \mathbb{R}_{>0}$ are learnable parameters.

It is easy to see that both our normalization and nonlinear layers are equivariant under isometries, as they preserve magnitudes, inner products, and areas (which preserves the computation of means).

9.1. Field Latents

Encoder architecture The encoder operates pointwise, and is carefully constructed so as to satisfy the conditions for equivariance in Equation (7). At each vertex p, a fixed number of points are uniformly sampled in the one ring neighborhood at which the texture is evaluated. The resulting collection of $3-$channel scalar features $\{\psi(q_i)\}$ are lifted to a collection of $C-$channel tangent vector features $\{\zeta^\psi_{pq_i}\}$ by multiplying the logarithms by the output of a linear layer $\mathcal{L}_1 : \mathbb{R}^3 \to \mathbb{C}^C$ such that

$$\zeta^\psi_{pq_i} = \mathcal{L}_1(\psi(q_i)) \cdot \log_p q_i \tag{21}$$

A token feature $\xi^\psi_p \in \mathbb{C}^C_p$ is initialized via a gathering operation with

$$\xi^\psi_p = \sum_i \log_p q_i \cdot \mathcal{L}_2(\psi(q_i)) \odot f(|\log_p q_i|), \tag{22}$$

where $\mathcal{L}_2 : \mathbb{R}^3 \to \mathbb{C}^C$ is a linearity and $f : \mathbb{R} \to \mathbb{C}^C$ is a filter parameterized by a two-layer MLP. It is easy to see

that the lifted features satisfy

$$d\gamma|_p \cdot \zeta^{\psi}_{pq_i} = \zeta^{\gamma\psi}_{\gamma(p)\gamma(q_i)} \quad \text{and} \quad d\gamma|_p \cdot \xi^{\psi}_{p} = \xi^{\gamma\psi}_{\gamma(p)} \tag{23}$$

for any isometry $\gamma : M \to N$ due to the transformation of the logarithm map and the preservation of its magnitude.

The concatenation $\{\zeta^{\psi}_{pq_i}\} \cup \{\xi^{\psi}_{p}\}$ is then passed to eight successive VN-Transformer layers [1]; each layer consists of a normalization, followed by a multi-headed attention block over the sample dimension, a second normalization, and two tangent vector neurons, with residual connections after the attention layer and final nonlinearity. The linear layers in the attention block use complex weights, with the the real part of product between the keys and values passed to the softmax — a construction that is equivariant under both isometries and the ordering of samples.

Afterwards, the token feature is extracted from the output of the VN-Transformer layers $\{\widehat{\zeta}^{\psi}_{pq_i}\} \cup \{\widehat{\xi}^{\psi}_{p}\}$ and used to predict the mean and standard deviation. Specifically, the token feature is passed to a two-layer equivariant MLP $\mathcal{M}_1 : \mathbb{C}^C \to \mathbb{C}^d$ using VN activations to predict the mean with

$$\mu^{\psi}_{p} = \mathcal{M}_1\left(\widehat{\xi}^{\psi}_{p}\right). \tag{24}$$

An invariant scalar feature is constructed from the token with a learnable product [47] and used to predict the log of the standard deviation with

$$\ln \sigma^{\psi}_{p} = \mathcal{M}_2\left(\overline{\widehat{\xi}^{\psi}_{p}} \odot \mathcal{L}_3(\widehat{\xi}^{\psi}_{p})\right), \tag{25}$$

where $\mathcal{L}_3 : \mathbb{C}^C \to \mathbb{C}^C$ is a complex linearity and $\mathcal{M}_2 : \mathbb{C}^C \to \mathbb{R}^d$ is a two-layer real-valued MLP with SiLU activations, taking the stacking of the real and imaginary components of the invariant feature as input.

In practice, we sample 64 uniformly distributed points in each one-ring, and use $C = 128$ channels in all layers of the encoder network.

Decoder architecture The learnable component of the FL decoder is the neural field $\mathcal{F}$ to which the concatenation of the positional and invariant features are passed to predict the texture value as in Equation (18). Specifically, the neural field is a five-layer real-valued MLP with SiLU activations and 512 channels in each layer, taking the stacking of the real and imaginary components of the complex features as input.

Training During training we consider the reconstruction loss over the one-rings

$$L_{\text{R}} = \mathbb{E}_{p\in M, q\in B_p} \|\psi(q) - \widehat{\psi}_p(q)\|_1, \tag{26}$$

in addition to the Kullback-Leibler divergence to regularize the distributions of the latent codes in the tangent space

$$L_{\text{KL}} = \frac{1}{2}\,\mathbb{E}_{p\in M} \sum_{i=1}^{d} \left[1 + \ln [\sigma_p]_i^2 - |[\mu_p]_i|^2 - [\sigma_p]_i^2\right]. \tag{27}$$

The FL-VAE is trained to optimize the sum of the reconstruction and KL losses,

$$L_{\text{FL}} = L_{\text{R}} + \lambda L_{\text{KL}}, \tag{28}$$

with $\lambda > 0$ controlling the weight between the terms.

We train a single FL-VAE for use in all experiments. It is trained with $\lambda = 0.01$. Training is performed for 1.5M iterations with a batch size of 16 and an initial learning rate of 10^{-3}, decaying to 10^{-5} on a cosine schedule. Running on an NVIDIA A100 GPU, convergence usually happens within five hours.

The FL-VAE is trained using OpenImagesV4 [29] which contains *only* images. Planar meshes are randomly generated using farthest point sampling followed by Delaunay triangulation, and are are overlaid on these images. The FL-VAE is trained on the one-rings of the resulting textured 2D meshes. By design, the FL-VAE is agnostic to the 3D embedding of one-rings because it only interfaces with their local flattening, allowing for zero-shot encoding and decoding on arbitrary 3D meshes at inference.

As a general rule, the number and variance of pixel values in one-rings during training should be equal to or greater than what is expected on target 3D assets to ensure robustness (we train on a high-res, diverse image dataset OpenImagesV4 to ensure high-variance). Furthermore the latent dimension should increase proportional to the number of pixels expected in each one-ring, which is dependent on both mesh and texture resolution.

As the FL-VAE does not see highly asymmetric one-rings at training, to improve robustness at inference we uniformly remesh target 3D assets (though varying the planar mesh quality during training may achieve a similar result).

9.2. Field Latent Diffusion Models

Reverse process The values of the coefficients in the reverse process in Equation (11) are given by [22, 50]:

$$C_1(\alpha_t, \alpha_{t-1}) = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \tag{29}$$

$$C_2(\alpha_t, \alpha_{t-1}) = \sqrt{\frac{\alpha_{t-1}(1-\alpha_t)}{\alpha_t}}\sqrt{\frac{(1-\alpha_{t-1})^2\alpha_t}{\alpha_{t-1}(1-\alpha_t)}} \tag{30}$$

$$C_3(\alpha_t, \alpha_{t-1}) = \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}\left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)} \tag{31}$$

Choice of convolution operator A majority of existing, state-of-the-art surface convolution operators are designed process scalar features, either alone [23, 47] or in tandem with tangent vector features in a multi-stream approach [45, 57]. To our knowledge, field convolutions [35] are the only existing state-of-the-art surface convolution designed specifically to process tangent vector features. The richness of the convolution operators in the multi-stream approaches is due to the intermixing of features, and the tangent vector convolution operators themselves are individually undiscriminating. More generally, field convolutions are part of a larger framework for equivariant convolutions that has demonstrated success across a wide variety of modalities and applications [33, 34, 36, 37].

However, DiffusionNet [47] is relatively fast for a surface network and connectivity agnostic — attractive properties for a denoising network. We experimented with extending it to handle tangent vector features by replacing the Laplacian eigenfunctions with those of the vector Lapalacian and adapting the network with the nonlinearities and normalization in Equations (19) and (20). While we were able to successfully train a denoising model with this architecture, we found that decoded samples had very little diversity and exhibited a "raggedness" similar to the textures reconstructed from INFs [27] as seen in Figure 2. The lack of diversity in the generated samples is likely due to the fact that global support is baked-in to DiffusionNet's "convolutions", making it challenging to prevent it from overfitting to a single example.

Furthermore, DiffusionNet bandlimits features by projecting them to a low-dimensional (vector) Laplacian eigenbasis. This presents several problems when processing vector fields in FL space. Specifically, the vector Laplacian eigenbasis spans *continuous* vector fields. The FL encoder constructs latent vector fields pointwise, which are not guaranteed to be continuous, let alone smooth. Thus, vector fields in FL space are likely to suffer significant deterioration by projecting onto the low-frequency eigenbasis, potentially leading the material degradation of quality observed in reconstructed samples.

Extending field convolutions Field convolutions convolve vector fields $X \in TM^C$ with $C' \times C$ filter banks $\mathbf{f}_{c'c} \in L^2(\mathbb{C}, \mathbb{C})$, returning vector fields $X * \mathbf{f} \in TM^{C'}$. To simplify notation, we convert to polar coordinates: expressing the $c-$th input vector field and the logarithm in the tangent space at a point $p \in M$ as

$$X_c(p) \equiv \varrho_p^c \, e^{i\phi_p^c} \qquad \text{and} \qquad \log_p q \equiv r_{pq} e^{i\theta_{pq}}. \tag{32}$$

Similarly, we denote by φ_{pq} the angle of rotation resulting from the parallel transport $\mathcal{P}_{p \leftarrow q} : T_qM \to T_pM$ along the shortest geodesic from neighboring points q to p on the surface, such that for any $\mathbf{v} \in T_qM$,

$$\mathcal{P}_{p \leftarrow q}(\mathbf{v}) \equiv e^{i\varphi_{pq}} \mathbf{v} \tag{33}$$

Then, the $c'-$th output of the field convolution of X with $\mathbf{f}$ is the vector field [35]

$$[\, X * \mathbf{f} \,]_{c'}(p) = \sum_{c=0}^{C} \int_{B_p} \varrho_q^c \, e^{i(\phi_q^c + \varphi_{pq})} \, \mathbf{f}_{c'c} \left(r_{pq} \, e^{i(\theta_{pq} - \phi_q^c)} \right) dq \tag{34}$$

We extend field convolutions to interleave scalar embeddings with tangent vector features by convolving vector fields with filters $\mathbf{f}_{c'c} \in L^2(\mathbb{C} \times \mathbb{R}^e, \mathbb{C})$, with e the embedding dimension. Specifically, given a scalar embedding $\pi \in L^2(M, \mathbb{R}^e)$, the $c'-$th output of field convolution of vector fields $X \in TM^C$ and embedding π with $C' \times C$ filter banks $\mathbf{f}_{c'c} \in L^2(\mathbb{C} \times \mathbb{R}^e, \mathbb{C})$ is the vector field

$$[\, \{X, \pi\} * \mathbf{f} \,]_{c'}(p) = \sum_{c=0}^{C} \int_{B_p} \varrho_q^c \, e^{i(\phi_q^c + \varphi_{pq})} \, \mathbf{f}_{c'c} \left(r_{pq} \, e^{i(\theta_{pq} - \phi_q^c)}, \pi(q) \right) dq \tag{35}$$

For any isometry $\gamma : M \to N$, field convolutions have the property [35]

$$\gamma X * \mathbf{f} = \gamma [X * \mathbf{f}], \tag{36}$$

due in part to the invariance of the filter arguments under isometries. Thus, it can be shown that the extension of field convolutions in Equation (35) is equivariant in the sense that

$$\{\gamma X, \gamma \pi\} * \mathbf{f} = \gamma [\, \{X, \pi\} * \mathbf{f} \,]. \tag{37}$$

Here, our implementation of field convolutions differs slightly from that of the original. The filter support is restricted to the immediately adjacent vertices in the one-ring about a point. Additionally, instead of parameterizing filters with Fourier basis functions, we make use of the PointConv trick [16, 59] to efficiently represent filters $\mathbf{f}_{c'c}$ as three-layer MLPs with eight channels in the interior layers.

Denoising network The denoising network is constructed with FCResNet blocks [35], modified to inject scalar embeddings $\pi \in L^2(M, \mathbb{R}^e)$ derived from the diffusion timestep and optional user-input features. Each FCResNet block consists of a three-layer VN MLP, with normalization before each VN, followed by a standard field convolution, a second three-layer VN MLP, and a final field convolution with the embedding.

The architecture of the denoising model is based on the observation that by limiting the size of the network's receptive field, distributions of latent features can be learned without overfitting to the single textural example [28, 54,

58]. Following [54] the denoising network ε takes the form of a shallow two-level U-Net with a single downsampling layer. The input layer consists of two FCResNet Blocks, with four FCResNet blocks in the downsampling and final upsampling layers each, for a total of 10. The receptive field of the network is equivalent to a $28-$ring about each point, which is approximately 6% of the total area of a 30K vertex mesh. Upsampling and downsampling are performed using mean pooling and nearest-neighbor unpooling, respectively, augmented with parallel transport to correctly express tangent vector features in neighboring tangent spaces. In all experiments, our denoising networks use 96 channels per FCResNet block at the finest resolution, increasing to 192 after downsampling.

Scaling input latents In practice, we scale latent vector fields in the diffusion process to have magnitudes proportional to sampled noise in the tangent bundle. From the definition of Gaussian noise in the tangent bundle in Equation (1) as pointwise samples from 2D Gaussians, the expected value of the pointwise magnitude of $X \sim \mathcal{TN}_M(0, I_1)$ follows the chi distribution with

$$\sqrt{\frac{\pi}{2}} = \mathbb{E}_{p \in M} |X(p)|. \tag{38}$$

Thus, we scale input latents $Z \in TM^d$ to the diffusion model per-channel such that

$$Z_i \mapsto \frac{\sqrt{\pi}}{\sqrt{2}\,\mathbb{E}_{p \in M} |Z_i(p)|} \cdot Z_i \tag{39}$$

for $1 \leq i \leq d$ to ensure they are represented at the same scale as the added noise.

Training The denoising network is trained subject to the loss in Equation (10) for 300K iterations, using the Adam [24] optimizer with a batch size of 16 and initial learning rate of 10^{-3}, decaying to 10^{-5} on a cosine schedule. A computation bottleneck is the memory overhead induced by the irregular (one-ring) kernel support for FCs. To scale using a NVIDIA-V100 with only 16GB of memory, we must rematerialize tensors during backprop, resulting in slower training ($\sim$2 days w/8 V100s). This cost would be mitigated with a higher memory GPU. In contrast, Sin3DM [58] reports training in several hours on an A6000 GPU. For inference, sampling new textures is much more efficient and takes just 10 minutes (compared to 3-5 for Sin3DM). While we found other popular surface networks to be unsuitable for FLDMs, FCs are not a fundamental part of our framework, and in the future could be replaced by a more efficient operator.

More generally, we found that training FLDMs on high-res meshes ($>$10K vertices) produced more diverse and higher quality textures than training on low-res meshes. This is possibly because the FLDM sees a larger number of latent codes and each code characterizes a smaller "unit" of texture, affording greater flexibility in generating patterns and smoothing transitions.

Directional control We expect that user-specified vector fields could also be passed to FLDMs to provide directional control, with embeddings formed by taking dot products with the network features. For example, the user may choose to condition the FLDMs in Fig. 6 with a vector field pointing about the whorl of the shell. The same field could be used during sampling (or, alternatively, a rotated or otherwise manipulated vector field) to control the orientation of synthesized features in the desired way.

10. Additional Results

See the last pages of this supplementary material document for additional visualizations and comparisons.

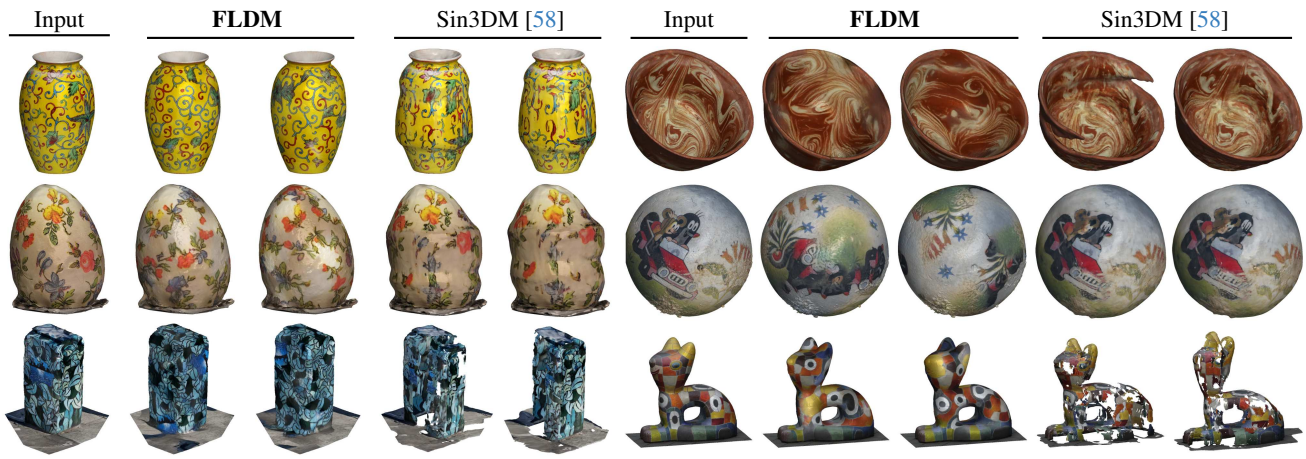

Figure 8. Additional samples generated from FLDMs and Sin3DM [58] in the unconditional paradigm described in section 6.2. The electrical box and cat sculpture on the bottom row are not included in our evaluations, but present significant failure cases for Sin3DM, which can struggle to synthesize samples from thin, shell-like models of the kind often arising from real-world 3D scans. *Zoom in to compare.*

Figure 9. Additional examples of label-guided texture synthesis with FLDMs. Many meshes contain interesting textural details only in specific regions and are otherwise uniformly colored. Label guidance can be used to ensure generated textures generally reflect the distribution of content on the training mesh. *Zoom in to view.*

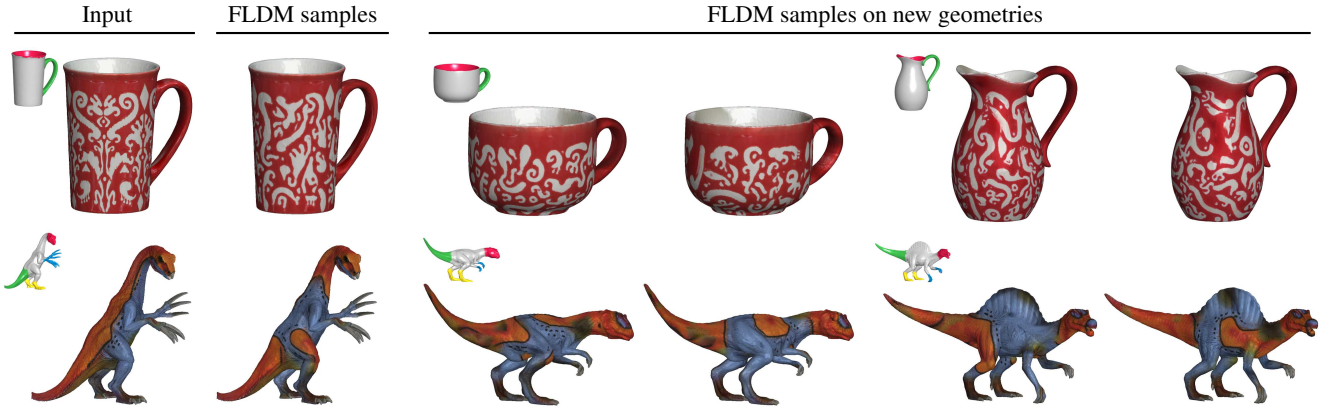

Figure 10. Additional examples of generative texture transfer with FLDMs. The FL-VAE and FLDMs are in fact equivariant under *local* isometries, and are thus able to replicate textural details on new geometries that are only *locally* similar to the training mesh. *Zoom in to view.*

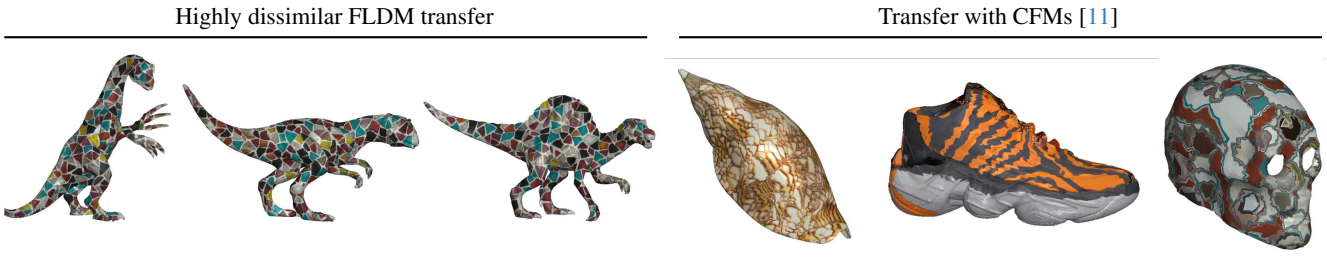

Figure 11. *Left:* While we believe the most practical use of generative texture transfer is intra-category, FL-VAEs and FLDMs can in fact transfer textures between between highly dissimilar shapes such as from the *skull* to *dinosaur* meshes. *Right:* Complex Functional Maps (**CFMs**) [11] transfer textures based on estimated dense correspondences. Despite SoTA performance in correspondence tasks, CFMs struggle in the presence of topological changes, as in the skull, and pointwise mappings don't explicitly promote smooth transfers.

ID	Visualization
TROCHILUS_BOOST	—
Schleich_Therizinosaurus_ln9cruulPqc	—
SAPPHIRE_R7_260X_OC	—
Tieks_Ballet_Flats_Electric_Snake	—
Reebok_FS_HI_INT_R12	Figure 2 (Top)
Snack_Catcher_Snack_Dispenser	—
Polar_Herring_Fillets_Smoked_Peppered_705_oz_total	—
Olive_Kids_Mermaids_Pack_n_Snack_Backpack	—
Olive_Kids_Trains_Planes_Trucks_Bogo_Backpack	—
Olive_Kids_Dinosaur_Land_Munch_n_Lunch	—
Fruity_Friends	—
Horse_Dreams_Pencil_Case	Figure 2 (Bottom)
Horses_in_Pink_Pencil_Case	—
LEGO_City_Advent_Calendar	—
Digital_Camo_Double_Decker_Lunch_Bag	—
ASICS_GELDirt_Dog_4_SunFlameBlack	—

Table 3. List of Google Scanned Objects [13] assets used in the texture compression and reconstruction evaluations in section 6.1.

ID	Source	Visualization
3969fe35c6444293af27b976dca085a4	Objaverse	Figure 8 (Top left)
656f83a4d9224c29abc82ade2207d2ce	Objaverse	—
e7caba92073d4adba3477c21aa25e91f	Objaverse	Figure 3 (Top left)
ed0afde32a194337ba1a18b1018f2d2e	Objaverse	Figure 3 (Top right)
e5712bffe9714a6aaa148b6b262b748d	Objaverse	Figure 8 (Center right)
f0e3c872f1984cf7a467645d9e0d3abd	Objaverse	Figure 3 (Bottom left)
2df58d08f5604c058d43e00677d325b6	Objaverse	Figure 8 (Top right)
4a6d4fa8eb83401ca9ac0446bad83c0a	Objaverse	Figure 3 (Bottom right)
190211a19360444699dccec9eda105e6	Objaverse	Figure 8 (Center left)
Now_Designs_Bowl_Akita_Black	Google Scanned Objects	—

Table 4. List of Objaverse [8] and Scanned Objects [13] assets used in the unconditional generation evaluations in section 6.2.