Supplementary Material:
Surface-Aligned Neural Radiance Fields for Controllable 3D Human Synthesis

A. Algorithms

We present the pseudocodes of our proposed dispersed projection and vertex normal alignment in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Dispersed projection

Input: $\mathbf{x} \in \mathbb{R}^3$, body mesh $\mathcal{M}$
Output: Projected surface point $\mathbf{s}$
Initialize $\mathcal{S} := []$
Find the nearest surface point $\tilde{\mathbf{s}}$ to $\mathbf{x}$.
Find all triangles containing $\tilde{\mathbf{s}}$, denoted as $\mathcal{T}$.
for $T \in \mathcal{T}$ **do**
    vertex normal alignment for T.
    if $\mathbf{x}$ *inside the parallel triangle* T' **then**
        barycentric interpolated projection $\mathbf{x} \to \mathbf{s}_T$.
        $\mathcal{S}$.append($\mathbf{s}_T$)
    end
end
return $\mathbf{s} := \arg\min_{\mathbf{s}_T} \|\mathbf{s}_T - \mathbf{x}\|, \mathbf{s}_T \in \mathcal{S}$

Algorithm 2: Vertex normal alignment

Input: Triangle T
for $i \in \{1, 2, 3\}$ **do**
    Compute two edge directions $\mathbf{e}_1, \mathbf{e}_2$ from $\mathbf{v}_i$.
    Orthogonally project $\mathbf{n}_i$ on plane T at $\mathbf{p}_i$.
    Decomposite $\mathbf{p}_i - \mathbf{v}_i = c_1\mathbf{e}_1 + c_2\mathbf{e}_2$.
    $\tilde{c}_1, \tilde{c}_2 := \max(0, c_1), \max(0, c_2)$. `// only consider that` $\mathbf{p}_i$ `falls within the inward region.`
    $\mathbf{n}_i := \mathbf{n}_i - \tilde{c}_1\mathbf{e}_1 - \tilde{c}_2\mathbf{e}_2$. `// alignment.`
    Normalize the length of $\mathbf{n}_i$ to 1.
end

B. Implementation details

B.1. Network architecture

We present a NeRF network architecture in Fig. 1. We use positional encoding [2] with the frequency $L = 6$ and $L = 4$ for the surface-aligned representation and the view direction, respectively. We use a three-layer, 256-dimensional vanilla graph convolutional network (GCN) [1] with ReLU activation for encoding the skeleton pose $p \in \mathbb{R}^{24\times3}$ to an embedding $\tilde{\mathbf{z}}_p \in \mathbb{R}^{24\times256}$ and then perform a spatial average pooling to obtain a latent code $\mathbf{z}_p \in \mathbb{R}^{256}$.

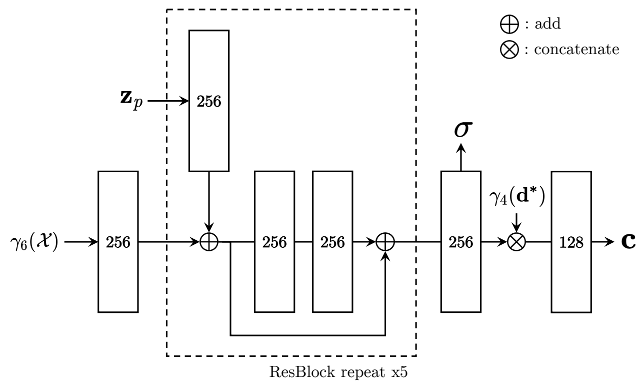

Figure 1. **Network architecture.** The network takes the positional encoding of the surface-aligned representation $\gamma_6(\mathcal{X})$ and the view direction $\gamma_4(\mathbf{d}^*)$ along with the skeleton pose embedding $\mathbf{z}_p$ and outputs the density σ and the RGB color $\mathbf{c}$. The number in each block means the dimension of the input. All linear layers are followed by ReLU activation except the output layers of color and density.

B.2. Training settings

We basically refer to [3] for the training setting. We use the single-level NeRF and sample 64 points along each camera ray. For points that are far from the predicted SMPL surface, we do not feed them into NeRF for faster training. Specifically, for points with signed distance $h > h_0$, our model returns the zero density and color directly. We set $h_0 = 0.2m$ in our experiments. We conduct the training on a single Nvidia V100 GPU. Learning rate decreases exponentially from $5e^{-4}$ to $5e^{-5}$ in training. The training typically converges in about 200k iterations, which takes about 14 hours.

C. Proof of injective mapping

We prove that the mapping $\mathbf{x} \rightarrow \mathcal{X}$ using proposed dispersed projection is an injection under certain conditions. Specifically, we prove that, for spatial point $\mathbf{x} \in \mathbb{R}^3 \backslash \mathbb{I}$, the mapping $\mathbf{x} \rightarrow \mathcal{X}$ is an injection, where $\mathbb{I}$ denotes the set of all bilinear surfaces formed by the adjacent vertex normals after vertex normal alignment for all the triangle faces of a mesh. Since the volume of $\mathbb{I}$ is zero, any spatial point is almost surely in $\mathbb{R}^3 \backslash \mathbb{I}$.

The proof is based on the following assumptions:

Assumption 1. *Meshes are watertight and do not have triangle faces with zero area.*

Assumption 2. *The face normal of and the vertex normals of every triangle form an acute angle.*

Assumption 3. *Any spatial point $\mathbf{x} \in \mathbb{R}^3$ can be projected onto the mesh surface through dispersed projection.*

From the definition of dispersed projection, we notice that for $\mathbf{x} \in \mathbb{R}^3 \backslash \mathbb{I}$, it will always be mapped to a surface point $\mathbf{s}$ that is strictly inside a triangle face, *i.e.*, not on edges. In this case, the dispersed projection is reduced to a barycentric interpolated projection based on the aligned vertex normals for the triangle face. We first introduce the following lemma:

Lemma C.1. *If $\mathbf{x} \in \mathbb{R}^3 \backslash \mathbb{I}$ is outside the mesh and mapped to $\mathbf{s}$ through dispersed projection, then $\mathbf{x} - \mathbf{s} = c\mathbf{n_s}$, $c \in \mathbb{R}^+$ is satisfied for all such $\mathbf{x}$. Here $\mathbf{s}$ is a surface point that is strictly inside a triangle face, and $\mathbf{n_s}$ is a unit vector that is invariant to $\mathbf{x}$.*

Proof. Consider a triangle T after vertex normal alignment as in Fig. 3(a) of the main paper. For triangle $T = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and its parallel triangle $T' = (\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3)$, by the definition of barycentric interpolated projection, we can obtain:

$$\begin{cases} \mathbf{x} = \alpha_1 \mathbf{v}'_1 + \alpha_2 \mathbf{v}'_2 + \alpha_3 \mathbf{v}'_3 & (1) \\ \mathbf{s} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 & (2) \end{cases}$$

where $(\alpha_1, \alpha_2, \alpha_3)$ is the barycentric coordinate. Combining the above equations we can obtain:

$$\mathbf{x} - \mathbf{s} = \alpha_1(\mathbf{v}'_1 - \mathbf{v}_1) + \alpha_2(\mathbf{v}'_2 - \mathbf{v}_2) + \alpha_3(\mathbf{v}'_3 - \mathbf{v}_3) \tag{3}$$

$$= \sum_{i=1,2,3} \alpha_i(\mathbf{v}_i - \mathbf{v}'_i) \tag{4}$$

$$\mathbf{v}_i - \mathbf{v}'_i = \|\mathbf{v}_i - \mathbf{v}'_i\| \, \mathbf{n}_{\mathbf{v}_i} = (l \,/\, \langle \mathbf{n}_{\mathbf{v}_i}, \mathbf{n}_T \rangle) \, \mathbf{n}_{\mathbf{v}_i}, \tag{5}$$

where l denotes the distance l between plane T and T', $\mathbf{n}_{\mathbf{v}_i}$ denotes the aligned vertex normal at $\mathbf{v}_i$, and $\mathbf{n}_T$ denotes the surface normal of T. Substituting Eq. (5) for Eq. (4) leads to:

$$\mathbf{x} - \mathbf{s} = \sum_{i=1,2,3} \alpha_i \,(l \,/\, \langle \mathbf{n}_{\mathbf{v}_i}, \mathbf{n}_T \rangle)\, \mathbf{n}_{\mathbf{v}_i} \tag{6}$$

$$= l \left(\sum_{i=1,2,3} (\alpha_i \,/\, \langle \mathbf{n}_{\mathbf{v}_i}, \mathbf{n}_T \rangle)\, \mathbf{n}_{\mathbf{v}_i} \right) \tag{7}$$

The term inside the parentheses of Eq. (7) is invariant to $\mathbf{x}$, and we describe its direction by a unit vector $\mathbf{n_s}$, which gives:

$$\mathbf{x} - \mathbf{s} = c\mathbf{n_s}, \; c \in \mathbb{R}^+ \tag{8}$$

This concludes the proof. □

We call $\mathbf{n_s}$ in Eq. (8) an *interpolated normal at* $\mathbf{s}$, which only depends on the barycentric coordinates of $\mathbf{s}$ and the mesh with aligned vertex normals. We then introduce the following lemma:

Lemma C.2. *For $\mathbf{x} \in \mathbb{R}^3 \backslash \mathbb{I}$, the mapping $\mathbf{x} \rightarrow (\mathbf{s}, h)$ is a injection, where $h = \|\mathbf{x} - \mathbf{s}\|$ (when $\mathbf{x}$ is outside the mesh) or $h = -\|\mathbf{x} - \mathbf{s}\|$ (when $\mathbf{x}$ is inside the mesh).*

Proof. We first prove the case of when $\mathbf{x}$ is outside the mesh, *i.e.*, $h = \|\mathbf{x} - \mathbf{s}\|$. From the definition of h, it is exactly c in Eq. (8), which gives:

$$\mathbf{x} - \mathbf{s} = h\mathbf{n_s} \tag{9}$$

Consider the following:

$$\begin{cases} \mathbf{x}_1 - \mathbf{s}_1 = h_1 \mathbf{n}_{\mathbf{s}_1} & (10) \\ \mathbf{x}_2 - \mathbf{s}_2 = h_2 \mathbf{n}_{\mathbf{s}_2} & (11) \end{cases}$$

where $\mathbf{s}_1 = \mathbf{s}_2$, $h_1 = h_2$. For the same surface point $\mathbf{s}_1 = \mathbf{s}_2$, they have the same *interpolated normal*, that is, $\mathbf{n}_{\mathbf{s}_1} = \mathbf{n}_{\mathbf{s}_2}$. Therefore, from Eq. (10) and Eq. (11), we can obtain:

$$\mathbf{x}_1 = \mathbf{x}_2 \tag{12}$$

Above equations indicate that:

$$(\mathbf{s}_1, h_1) = (\mathbf{s}_2, h_2) \Rightarrow \mathbf{x}_1 = \mathbf{x}_2 \tag{13}$$

which concludes that the mapping $\mathbf{x} \rightarrow (\mathbf{s}, h)$ is a injection. The case of when $\mathbf{x}$ is inside the mesh is proved similarly by inverting the face and vertex normals of a triangle face and applying Lemma C.1. □

We finally introduce the following lemma:

Lemma C.3. *The mapping $\mathbf{s} \rightarrow \mathbf{s}_c$ is a injection, where $\mathbf{s}_c$ is the corresponding surface point on the T-pose mesh.*

Proof. From Assumption 2, the shared T-pose mesh is watertight and thus does not have self-intersection. From the definition of $\mathbf{s}_c$, that is, $\mathbf{s}_c$ is inside the same triangle with the same barycentric coordinates as $\mathbf{s}$, the proof is trivial. $\square$

From Lemma C.2 and Lemma C.3, we prove that for $\mathbf{x} \in \mathbb{R}^3 \backslash \mathbb{I}$, the mapping $\mathbf{x} \rightarrow \mathcal{X}$ is an injection.

Proof. Lemma C.3 indicates that

$$\mathbf{s}_{c1} = \mathbf{s}_{c2} \Rightarrow \mathbf{s}_1 = \mathbf{s}_2. \tag{14}$$

Considering $h_1 = h_2 (h_1, h_2 \in \mathbb{R})$, it immediately follows that

$$(\mathbf{s}_{c1}, h_1) = (\mathbf{s}_{c2}, h_2) \Rightarrow (\mathbf{s}_1, h_1) = (\mathbf{s}_2, h_2). \tag{15}$$

Combining Eq. (13) and Eq. (15), we can obtain

$$(\mathbf{s}_{c1}, h_1) = (\mathbf{s}_{c2}, h_2) \Rightarrow \mathbf{x}_1 = \mathbf{x}_2, \tag{16}$$

which is exactly

$$\mathcal{X}_1 = \mathcal{X}_2 \Rightarrow \mathbf{x}_1 = \mathbf{x}_2. \tag{17}$$

This concludes the proof. $\square$

References

[1] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 1

[2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[3] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 1