DGE: Direct Gaussian 3D Editing by Consistent Multi-view Editing

Minghao Chen, Iro Laina, and Andrea Vedaldi

Visual Geometry Group, University of Oxford
{minghao, iro, vedaldi}@robots.ox.ac.uk
silent-chen.github.io/DGE

Gaussian splatting

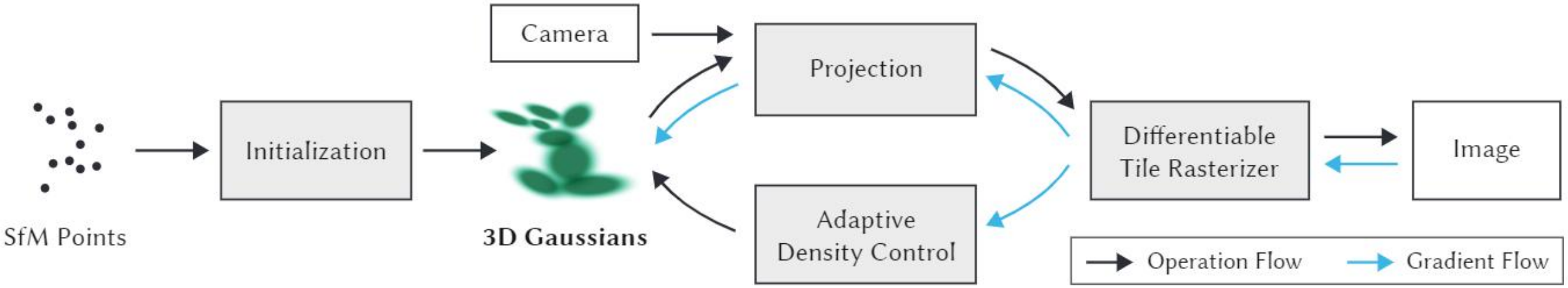


Fig. 2. Optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians. We then optimize and adaptively control the density of this set of Gaussians. During optimization we use our fast tile-based renderer, allowing competitive training times compared to SOTA fast radiance field methods. Once trained, our renderer allows real-time navigation for a wide variety of scenes.

$$I(\boldsymbol{u}) = \int_0^\infty c(\boldsymbol{x}_t, \boldsymbol{\nu})\sigma(\boldsymbol{x}_t)e^{-\int_0^t \sigma(\boldsymbol{x}_\tau)d\tau} dt,$$

$$G(x) = e^{-\frac{1}{2}(x)^T\Sigma^{-1}(x)}$$

VC-Editor

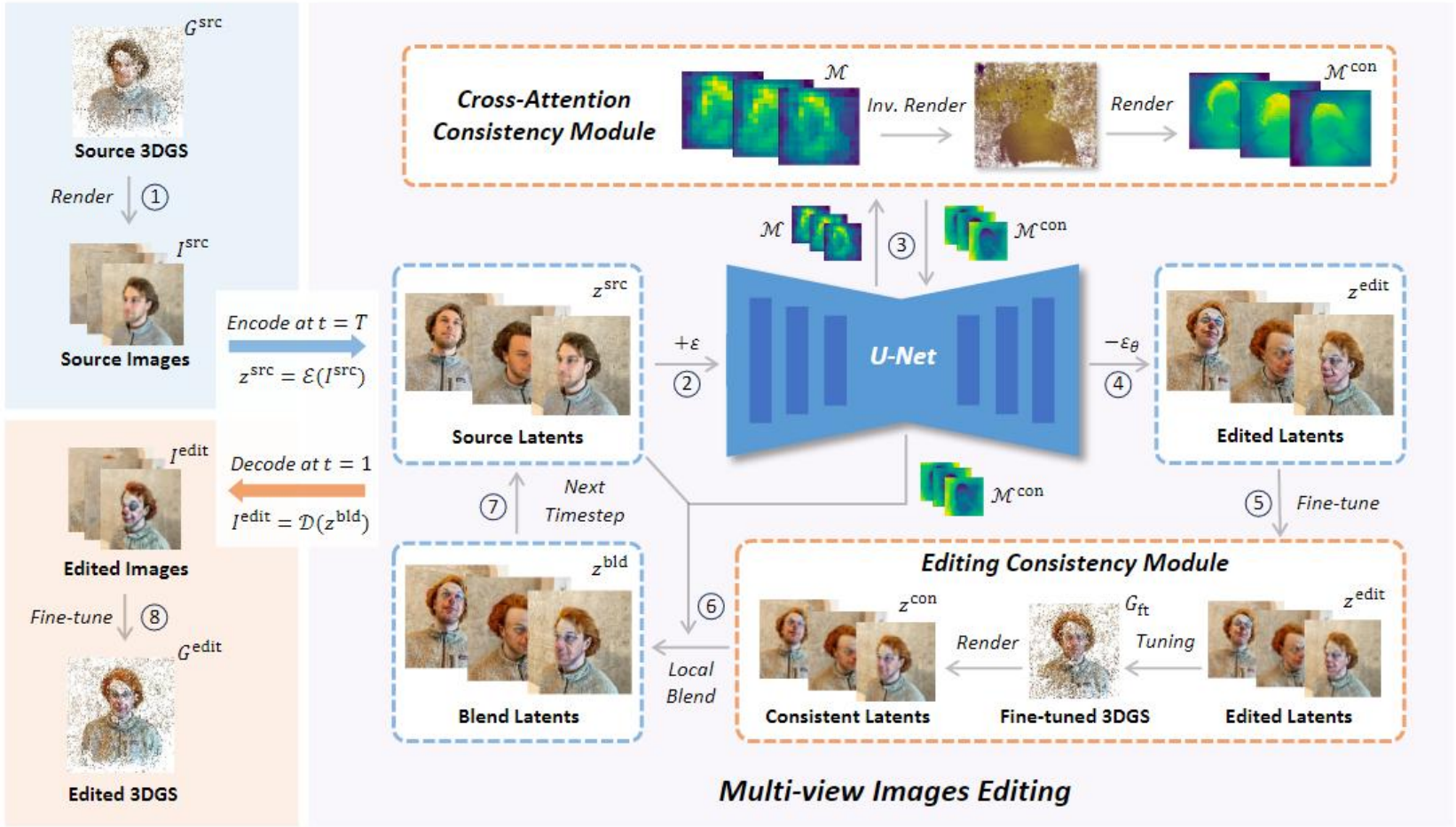


Fig. 3: The pipeline of our VcEdit: VcEdit employs an image-guided editing pipeline. In the image editing stage, the Cross-attention Consistency Module and Editing Consistency Module are employed to ensure the multi-view consistency of edited images. We provide a detailed overview in Sec. 4.1.

DGE

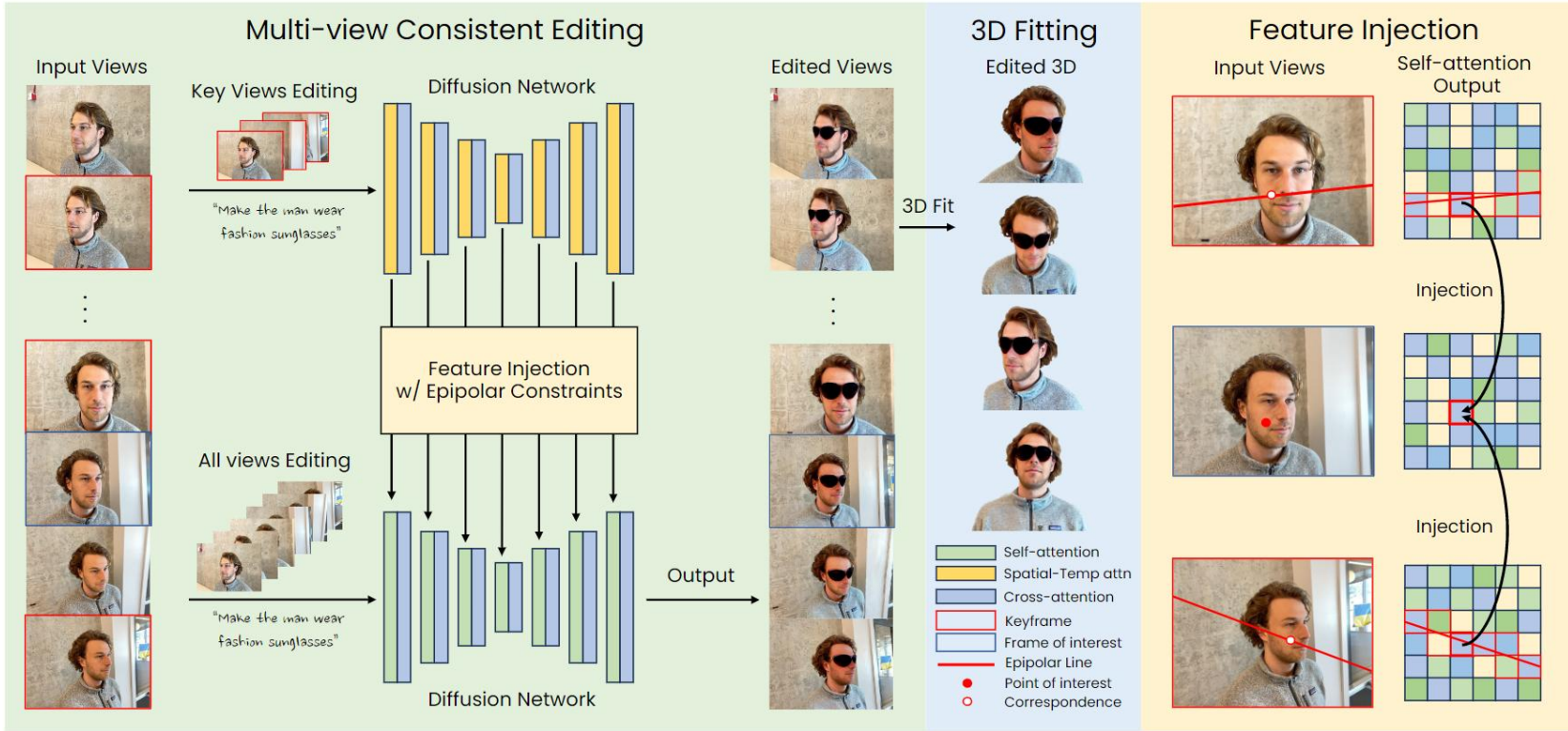


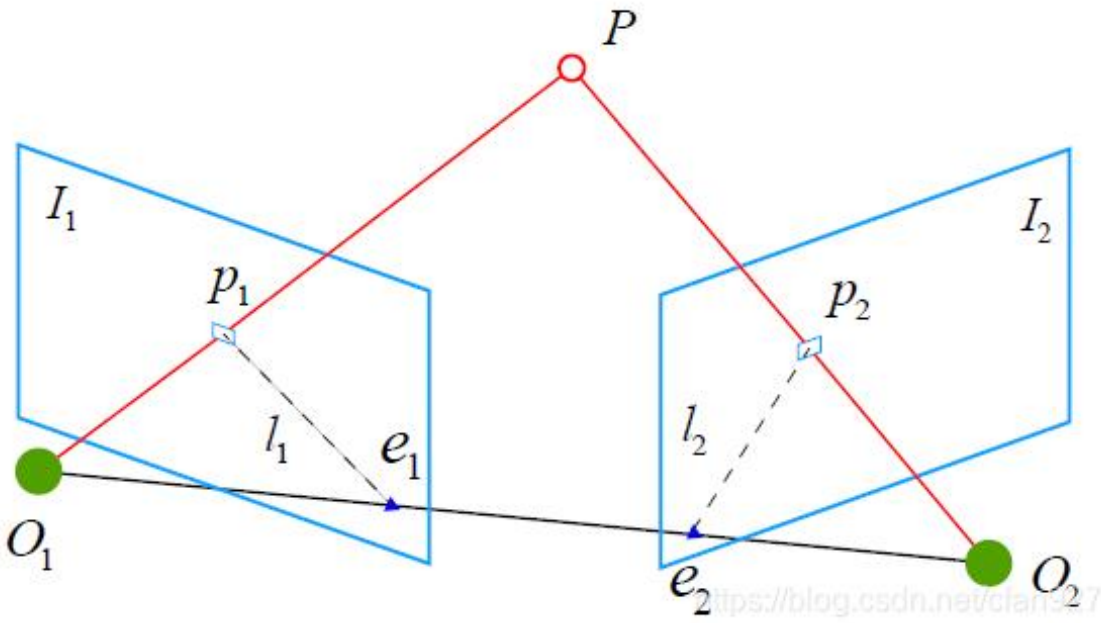
Fig. 1: Overview. As shown on the left, our method is divided into two main parts: multi-view consistent editing with epipolar constraints and direct 3D fitting. In the multi-view editing stage, key views are randomly selected and jointly fed to the editing diffusion network to extract features with the *spatial-temporal attention*. To edit other frames, the features of key views are injected into the diffusion network through correspondence matching on feature maps with epipolar constraints. The detailed feature injection process is shown on the right; only features with a red border (*i.e.*, the points following epipolar constraints) are considered for correspondence matching.

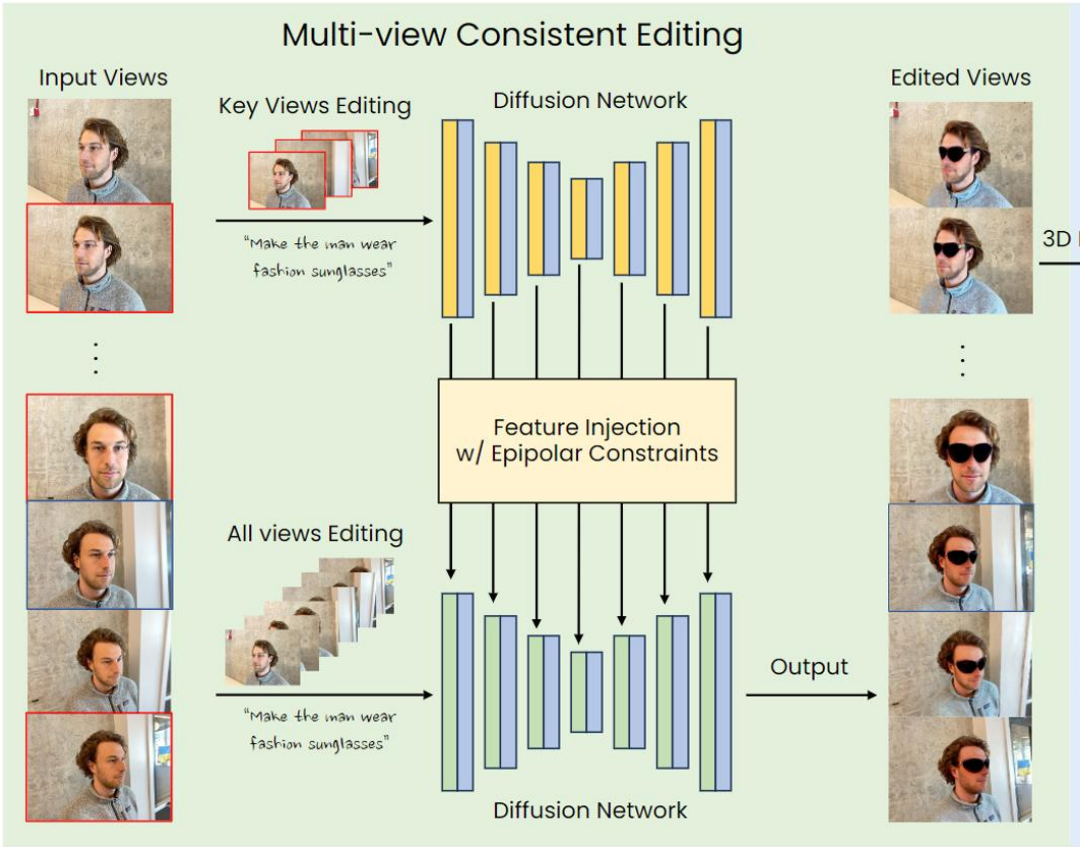
Key-view editing

queries {Qt}t=1~T
keys {Kt}t=1~T
values {Vt}t=1~T
d is the embedding dimension of keys and queries

$$\mathrm{STAttn}(Q, K, t) = \mathrm{Softmax}\left(\frac{Q_t \cdot [K_1, \ldots, K_T]}{\sqrt{d}}\right),$$

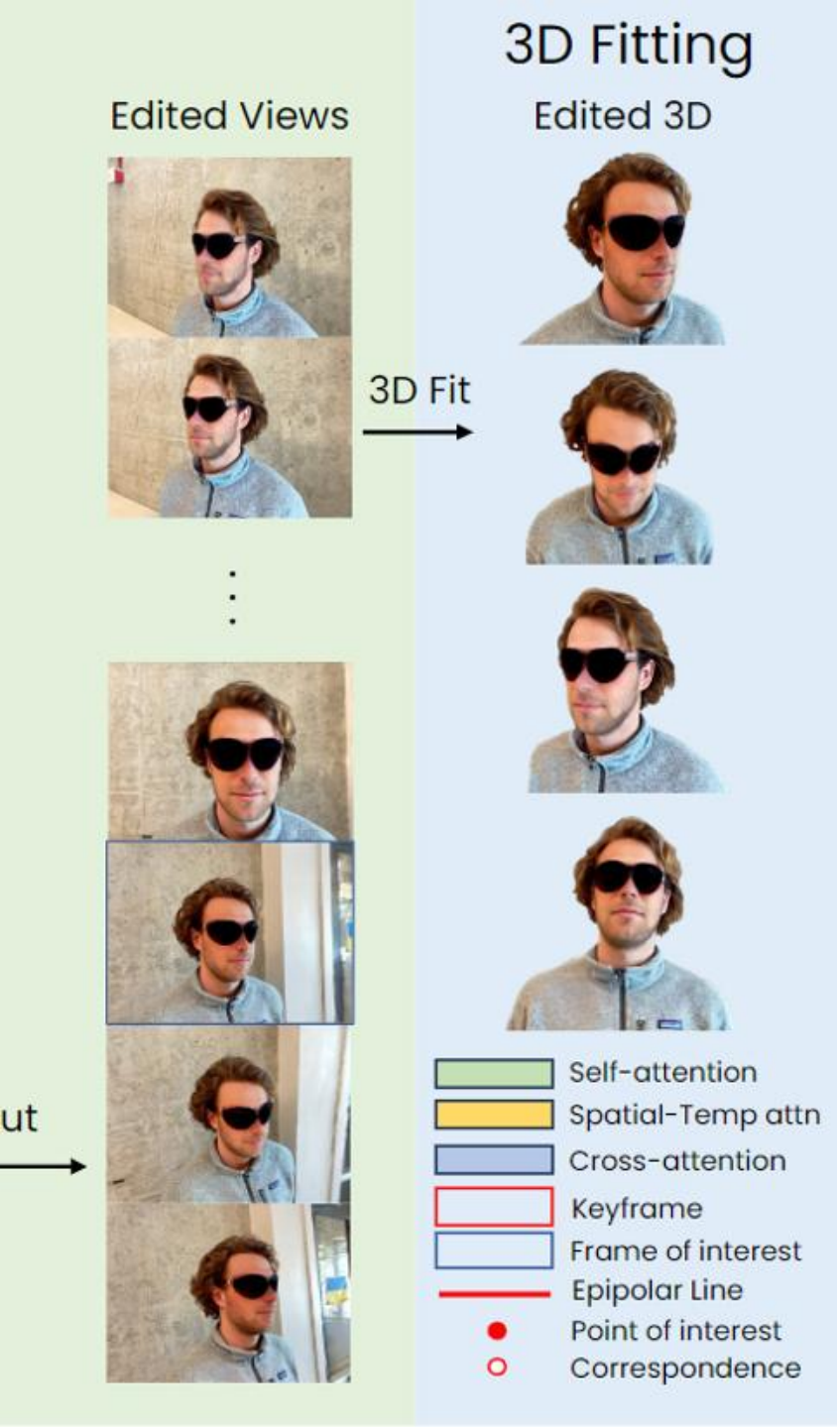
$$\Phi_t = \mathrm{STAttn}(Q, K, t) \cdot [V_1, \ldots, V_T]$$





$$M_{t'}[u] = \underset{v,\, v^\top F u = 0}{\arg\min}\ D\left(\Psi_{t'}[u],\, \Psi_{k^*}[v]\right), \quad \forall\, t' \in \mathcal{T} \backslash \mathcal{K}$$

where D is the cosine distance, u and v index the feature maps spatially, k* is the index of the key view that is the closest to view t′ (in terms of camera viewpoint), and F is the fundamental matrix corresponding to the two views t′and k* . F u is the epipolar line in view k* along which the corresponding point ofv in view k* must lie.



$$\mathcal{G}' = \arg\min_{\mathcal{G}} \sum_{t=1}^{T} \|I'_t - \mathrm{Rend}(\mathcal{G}, \pi_t)\|$$

Method	3D Model	CLIP Similarity	CLIP Directional Similarity	Avg. Editing Time
Instruct-N2N [25]	NeRF	0.215	0.64	$\sim$ 51min
ViCA-NeRF [16]	NeRF	0.204	0.44	$\sim$ 28min
GaussianEditor [13]	GS	0.201	0.60	$\sim$ 7min
IP2P [95] + SDS [70]	GS	0.206	0.61	$\sim$ 6min
Ours	GS	**0.226**	**0.67**	$\sim$ 4min

Table 1: Comparison with other editing methods. Methods based on Gaussian Splatting are much faster than NeRF-based ones. Our DGE achieves the best performance at almost half the time compared to GaussianEditor.

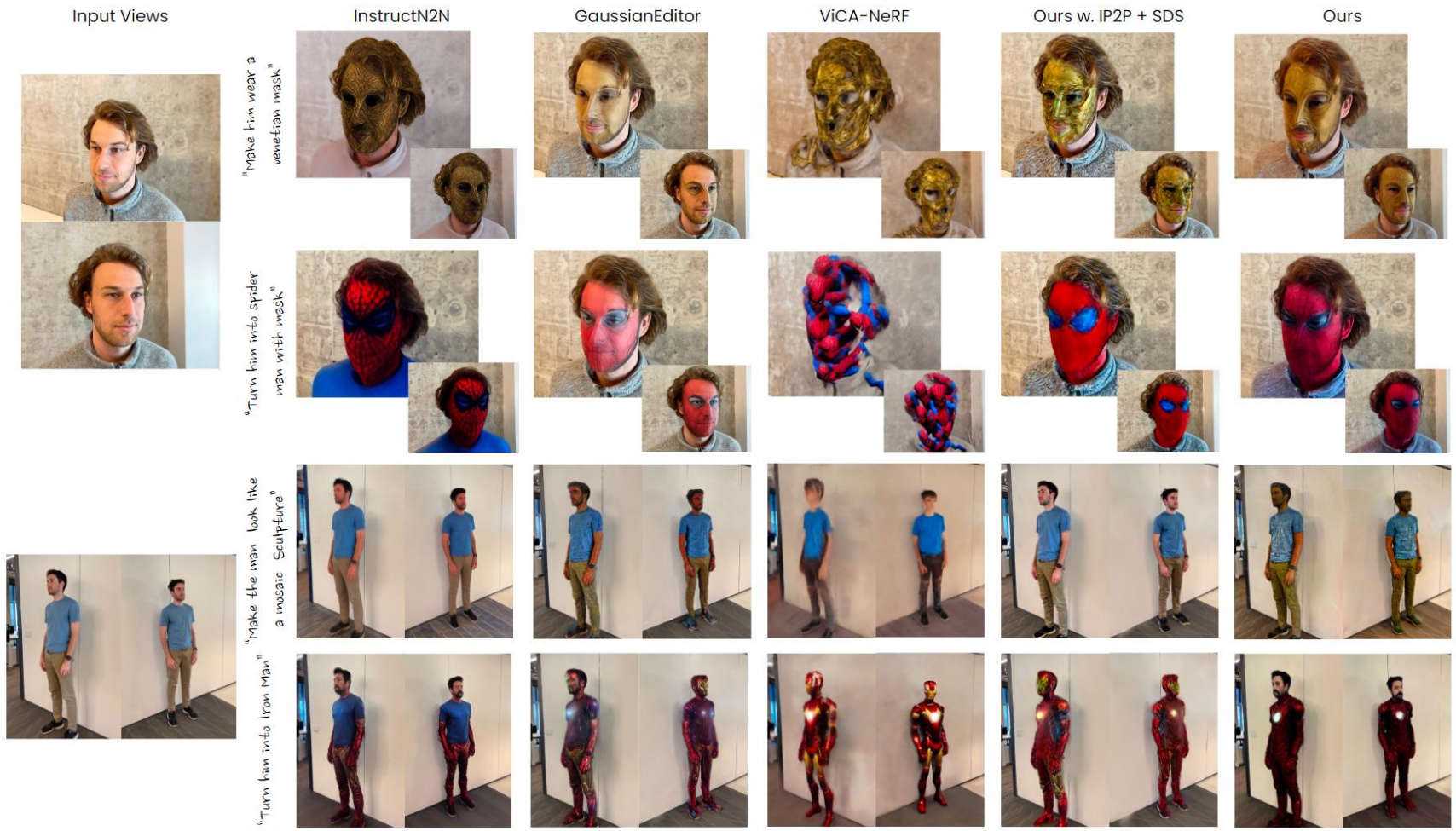


Fig. 2: Comparison with other methods. Our method can provide fast and detailed editing effects, such as the textures on the Venetian mask and mosaic sculpture. Other methods, such as InstructN2N and IP2P+SD, fail to get the mosaic effects because they average over inconsistent editing.

Fig. 5: Comparison between our DGE and GaussianEditor [13] in terms of the number of iterations. Our method achieves realistic editing results with much fewer iterations. With more iterations, our method also gradually refines the details.