Do 2D GANs Know 3D Shape? Unsupervised 3D Shape Reconstruction from 2D Image GANs

Xingang Pan[1], Bo Dai[1], Ziwei Liu[2], Chen Change Loy[2], Ping Luo[3]

[1]The Chinese University of Hong Kong

[2]S-Lab, Nanyang Technological University [3]The University of Hong Kong

ICLR 2021

Do 2D GANs Model 3D Geometry?

Natural images are projections of 3D objects on a 2D image plane.

An ideal 2D image manifold (e.g., GAN) should capture 3D geometric properties.

The following example shows that there is a direction in the GAN image manifold that corresponds to viewpoint variation.

Can we Make Use of such Variations?

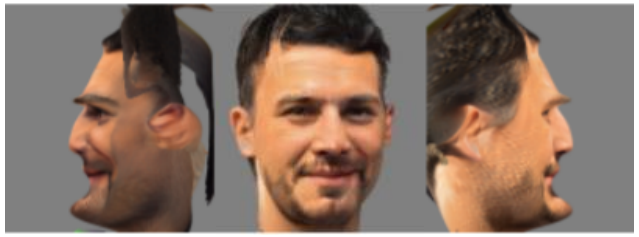
Can we make use of such variations for 3D reconstruction?

If we have multiple viewpoint and lighting variations of the same instance, we can infer its 3D structure.

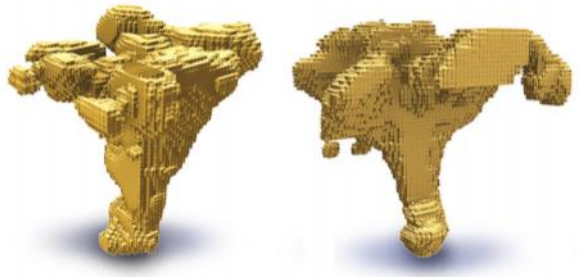
Let's create these variations by exploiting the image manifold captured by 2D GANs!

Prior Work

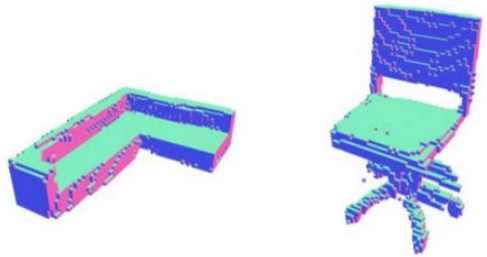
Learning 3D GANs from 2D images: *Need explicit 3D representation for GANs*



Generative3D
(Szabo et al. 2019)

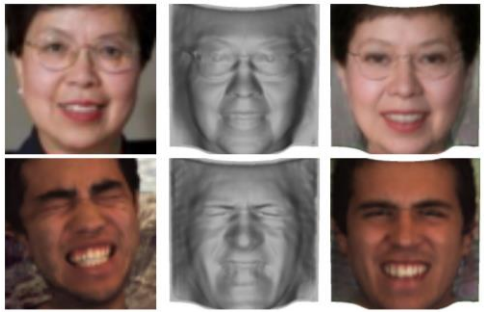


PlatonicGAN
(Henzler et al. ICCV2019)



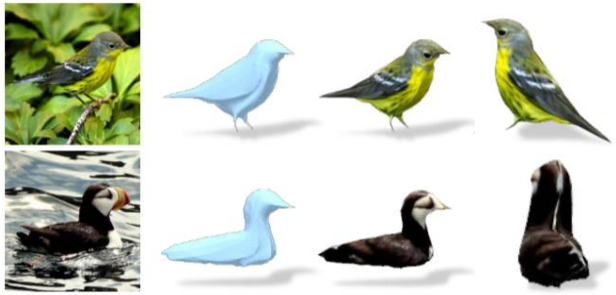
Inverse Graphics GAN
(Lunz et al. 2020)

Unsupervised 3D Shape Learning: *Rely on the symmetry assumption on shapes*



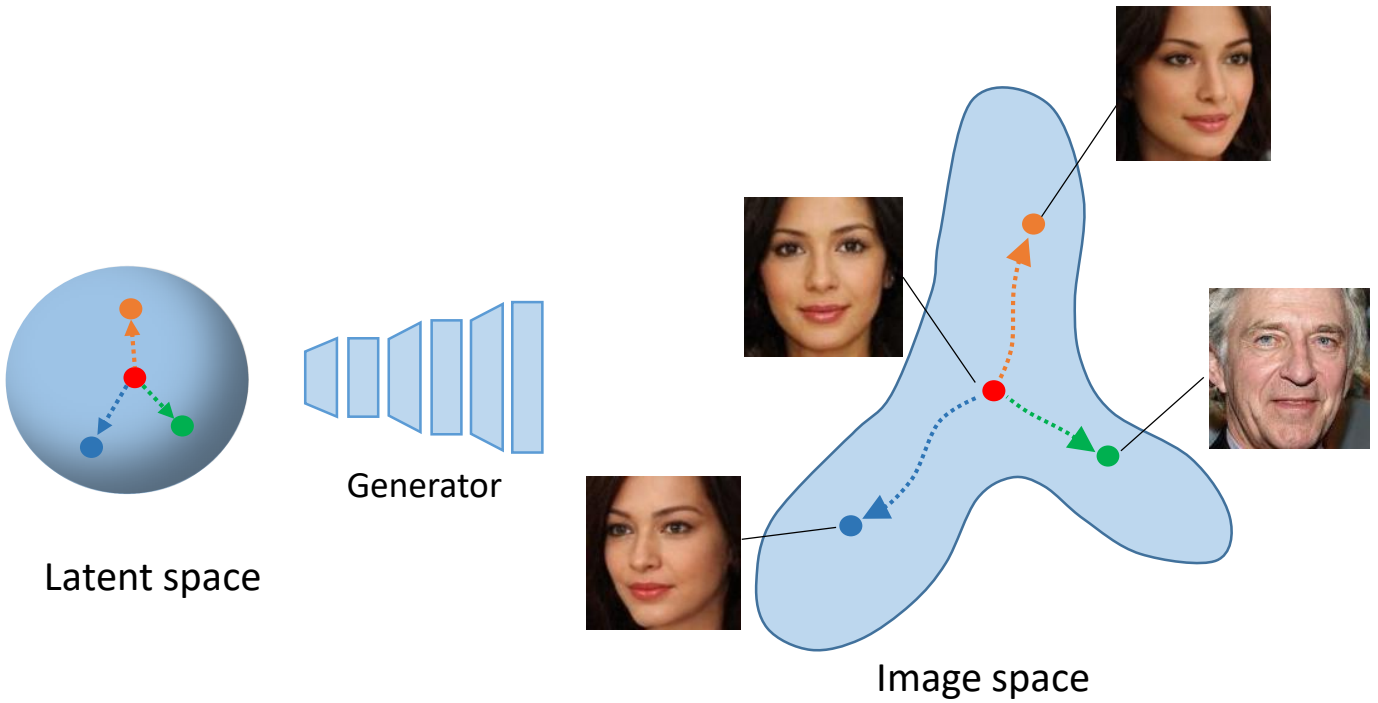
Unsup3d
(Wu et al. CVPR2020)

U-CMR
(Goel et al. ECCV2020)



UMR
(Li et al. ECCV2020)

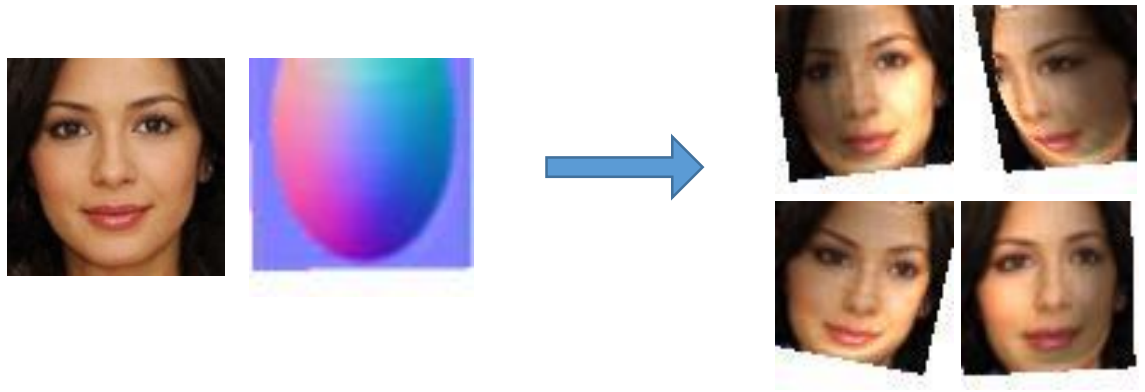
Challenge



It is non-trivial to find well-disentangled latent directions that control *viewpoint* and *lighting* variations in an unsupervised manner.

Our Solution

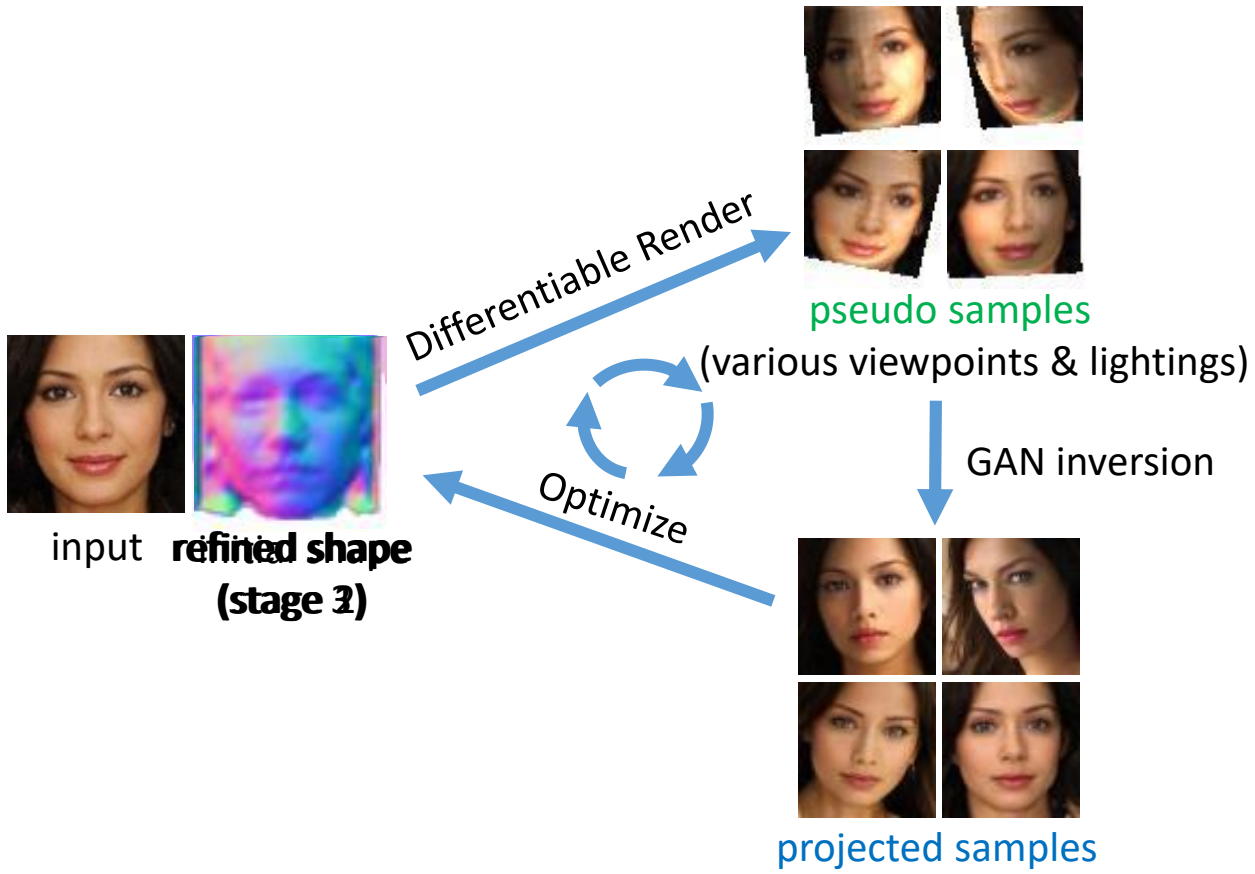
Idea 1: For many objects such as faces and cars, a convex shape prior like ellipsoid could provide a hint on the change of their viewpoints and lighting conditions.



Idea 2: Use GAN inversion constrained by this prior to “find” the latent directions.

Steps

- Initialize the shape with ellipsoid.
- Render '*pseudo samples*' with different viewpoints and lighting conditions.
- GAN inversion is applied to these samples to obtain the '*projected samples*'.
- *'Projected samples'* are used as the ground truth of the rendering process to optimize the 3D shape.
- Iterative training to progressively refine the shape.

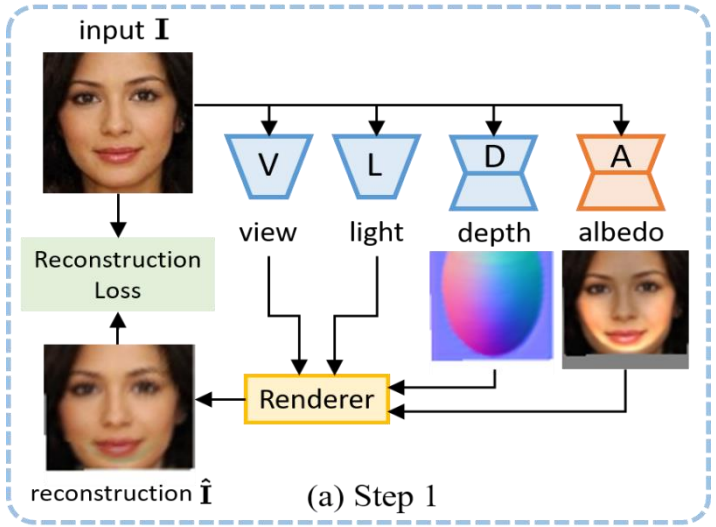


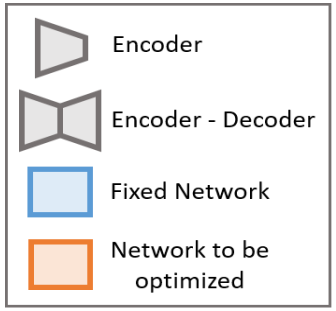
GAN2Shape

Step1:

Initialize shape with ellipsoid.

Optimize albedo network *A*.





$$\boldsymbol{\theta}_A = \arg\min_{\boldsymbol{\theta}_A} \mathcal{L}\Big(\mathbf{I}, \Phi\big(D(\mathbf{I}), A(\mathbf{I}), V(\mathbf{I}), L(\mathbf{I})\big)\Big)$$

$\mathbf{I}$: input image
D: depth network
A: albedo network
V: viewpoint network
L: lighting network
Φ: differentiable render
$\mathcal{L}$: reconstruction loss (L1+perceptual)

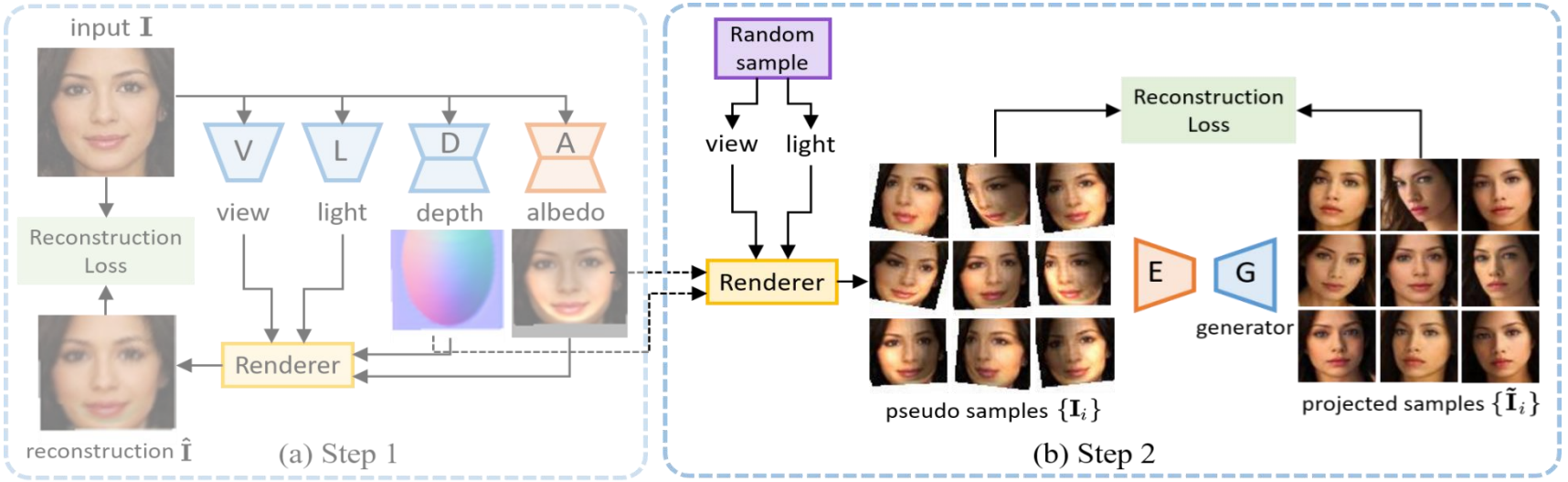
GAN2Shape

Step2:

Render 'pseudo samples' $\{\mathbf{I}_i\}$ with various viewpoints & lightings.

Perform GAN inversion to the pseudo samples to obtain the 'projected samples' $\{\tilde{\mathbf{I}}_i\}$.

Optimize latent encoder E.

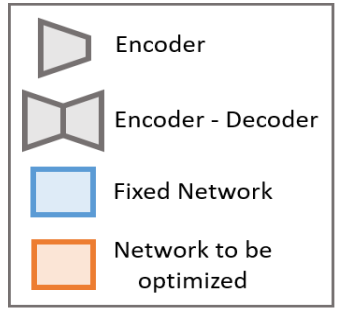


$$\boldsymbol{\theta}_E = \arg\min_{\boldsymbol{\theta}_E} \frac{1}{m} \sum_{i=0}^{m} \mathcal{L}^{'}\left(\mathbf{I}_i, G\big(E(\mathbf{I}_i) + \boldsymbol{w}\big)\right) + \lambda_1 \|E(\mathbf{I}_i)\|_2$$

original latent code

latent offset $\triangle w_i$

L2 regularization

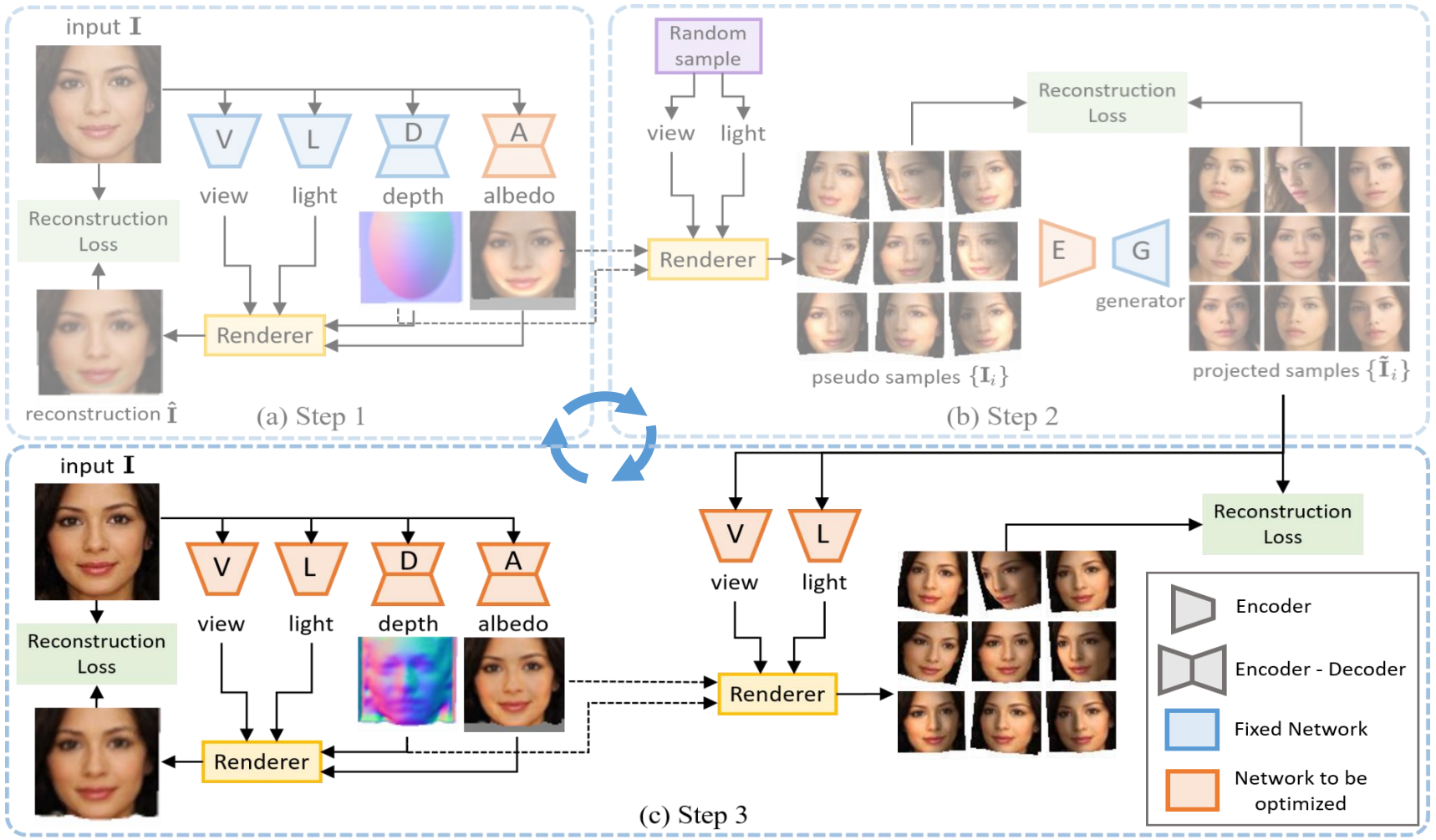


GAN2Shape

Step3:

Reconstruct ‘projected samples’ with shared depth & albedo and independent view & light.

Optimize network *V, L, D, A*.



$$\boldsymbol{\theta}_D, \boldsymbol{\theta}_A, \boldsymbol{\theta}_V, \boldsymbol{\theta}_L = \underset{\boldsymbol{\theta}_D, \boldsymbol{\theta}_A, \boldsymbol{\theta}_V, \boldsymbol{\theta}_L}{\arg\min} \frac{1}{m} \sum_{i=0}^{m} \mathcal{L}\Big(\tilde{\mathbf{I}}_i, \Phi\big(D(\mathbf{I}), A(\mathbf{I}), V(\tilde{\mathbf{I}}_i), L(\tilde{\mathbf{I}}_i)\big)\Big) + \lambda_2 \mathcal{L}_{smooth}\big(D(\mathbf{I})\big)$$

smoothness term

Implementation

Model: StyleGANv2 (Karras et al., 2020)

Dataset: BFM, CelebA, LSUN Car, LSUN Church, Cat

Renderer: Neural 3D Mesh Renderer (Kato et al., 2018)

Initialization:

We use a scene parsing model to parse the rough object position in the image, which is use to initialize the ellipsoid.

Training:

Joint pre-training for BFM, CelebA, and Cat (optional).

Regularizing the latent offset $\Delta \boldsymbol{w}_i = E(\mathbf{I}_i)$ in the W space of StyleGAN.

Regularizing the latent offset

Original:

$$\Delta \boldsymbol{w}_i = E(\mathbf{I}_i)$$

Regularize with mapping network F of StyleGAN:

$$\begin{aligned}\Delta \boldsymbol{w}_i &= F(\Delta \boldsymbol{z}_i) - F(\mathbf{0}) \\ &= F(E(\mathbf{I}_i)) - F(\mathbf{0})\end{aligned}$$

View F as two consecutive parts $F = F_1 \circ F_2$, regularize with F_1:

$$\Delta \boldsymbol{w}_i = F_1(E(\mathbf{I}_i) + F_2(\mathbf{0})) - F(\mathbf{0})$$

The depth d of F_1 controls the strength of regularization.

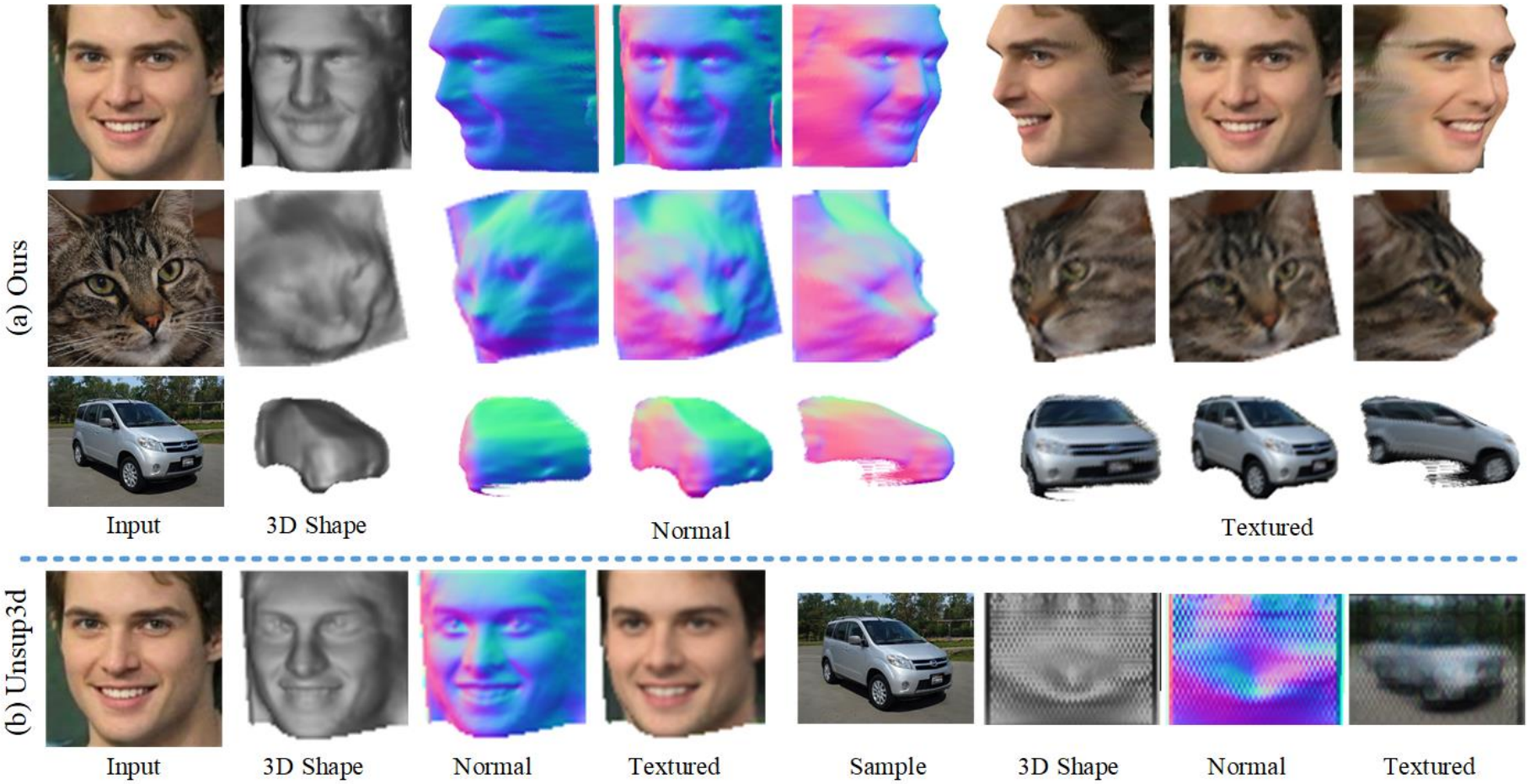
3D Reconstruction Results

Without any 2D keypoint or 3D annotations

Unsupervised 3D shape reconstruction from unconstrained 2D images

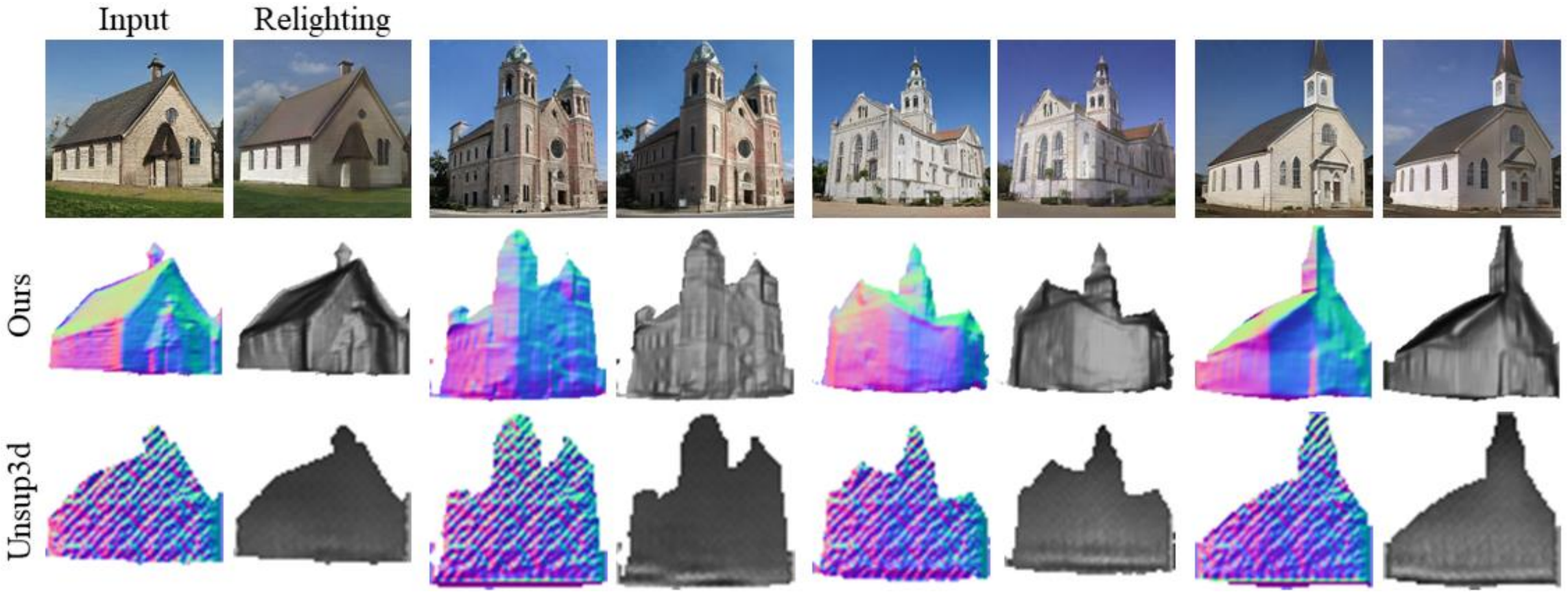
Without symmetry assumption

Work on many object categories such as huma faces, cars, buildings, etc.



Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *CVPR*, 2020

3D Reconstruction Results



Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *CVPR*, 2020

3D Reconstruction Results

Table 1: **Comparisons on the BFM dataset.** We report SIDE and MAD errors. 'Symmetry' indicates whether the symmetry assumption on object shape is used. We outperform others on both metrics.

No.	Method	Symmetry	SIDE ($\times 10^{-2}$)↓	MAD (deg.)↓
(1)	Supervised	N	0.419	10.83
(2)	Const. null depth	/	2.723	43.22
(3)	Average g.t. depth	/	1.978	22.99
(4)	Unsup3d (Wu et al., 2020)	Y	0.807	16.34
(5)	Ours (w/o regularize)	Y	0.925	16.42
(6)	Ours	Y	**0.756**	**14.81**
(7)	Unsup3d (Wu et al., 2020)	N	1.334	33.79
(8)	Ours	N	**1.023**	**17.09**

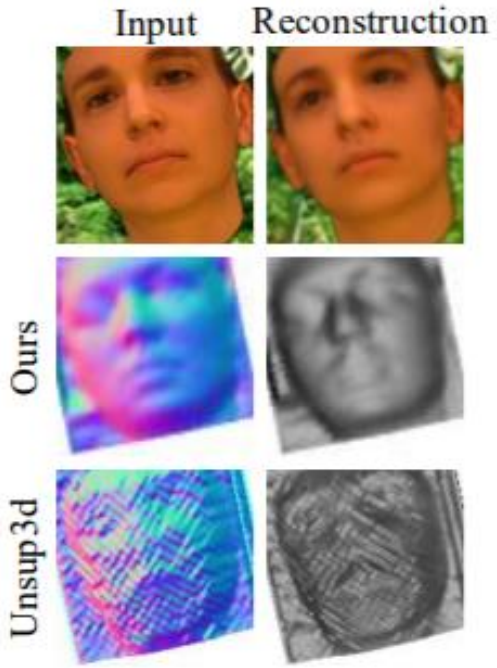
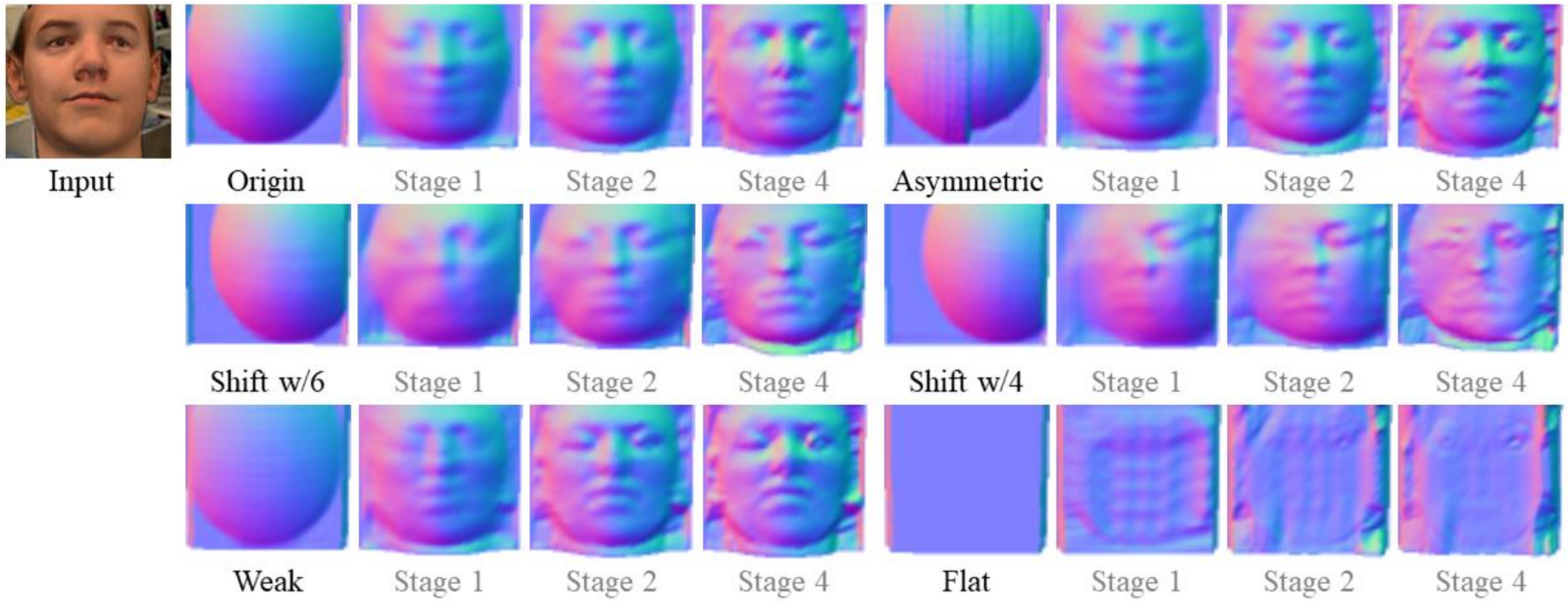


Figure 5: Results without symmetry assumption.

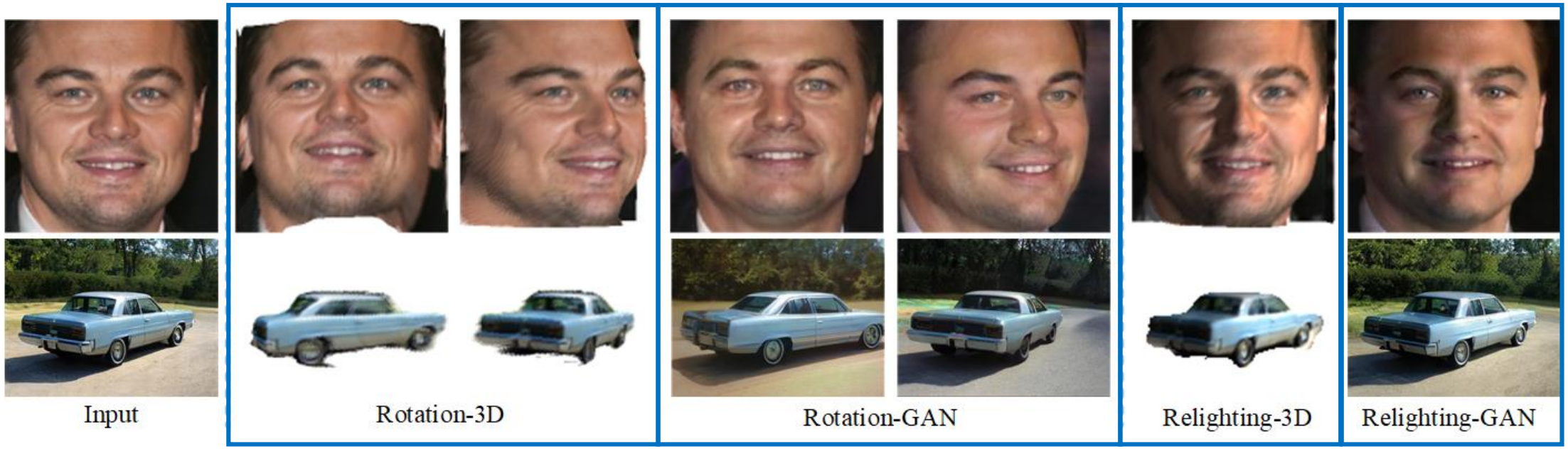
Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *CVPR*, 2020

Effects of Different Shape Prior



Shape prior	Origin	Asymmetric	Shift w/6	Shift w/4	Weak	Flat
SIDE ($\times 10^{-2}$)↓	0.756	0.769	0.767	0.775	0.764	1.021
MAD (deg.)↓	14.81	14.95	14.93	15.07	14.97	20.46

3D-aware Image Manipulation

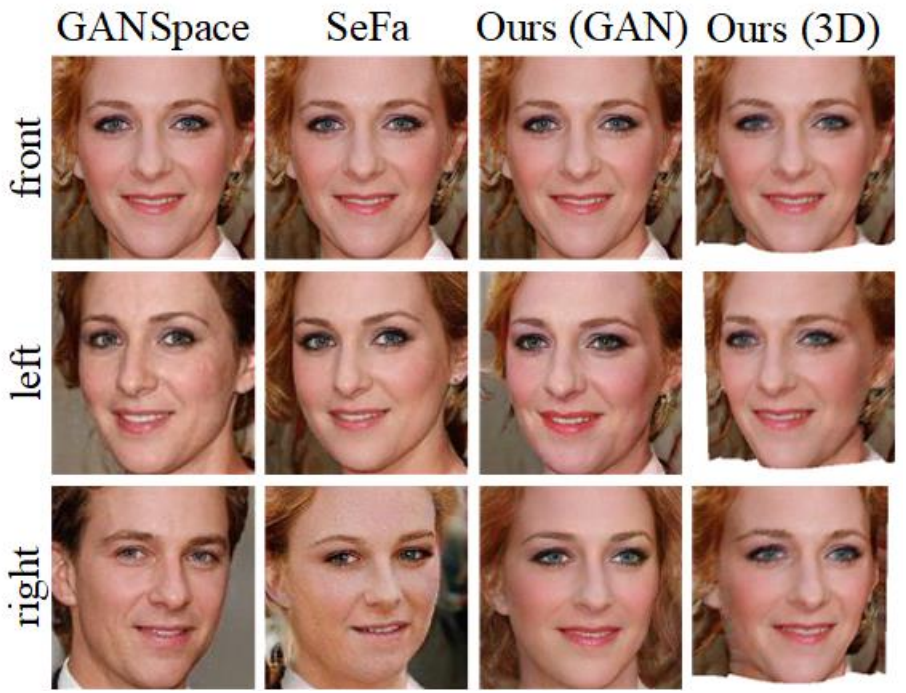


- Effect-3D: Rendered using the reconstructed 3D shape and albedo.
- Effect-GAN: project Effect-3D on the GAN image manifold using the trained encoder *E*.

3D-aware Image Manipulation

Table 2: **Identity-preserving face rotation.** We compare with HoloGAN, GANSpace, and SeFa. The metrics are identity distances measured as angles in the ArcFace feature embeddings.

Method	error_mean (deg.)↓	error_max (deg.)↓
HoloGAN	47.38	69.24
GANSpace	41.17	58.93
SeFa	41.79	60.73
Ours (3D)	**28.93**	**43.02**
Ours (GAN)	39.85	57.21



Nguyen-Phuoc, Thu, et al. "Hologan: Unsupervised learning of 3d representations from natural images." *ICCV* 2019.

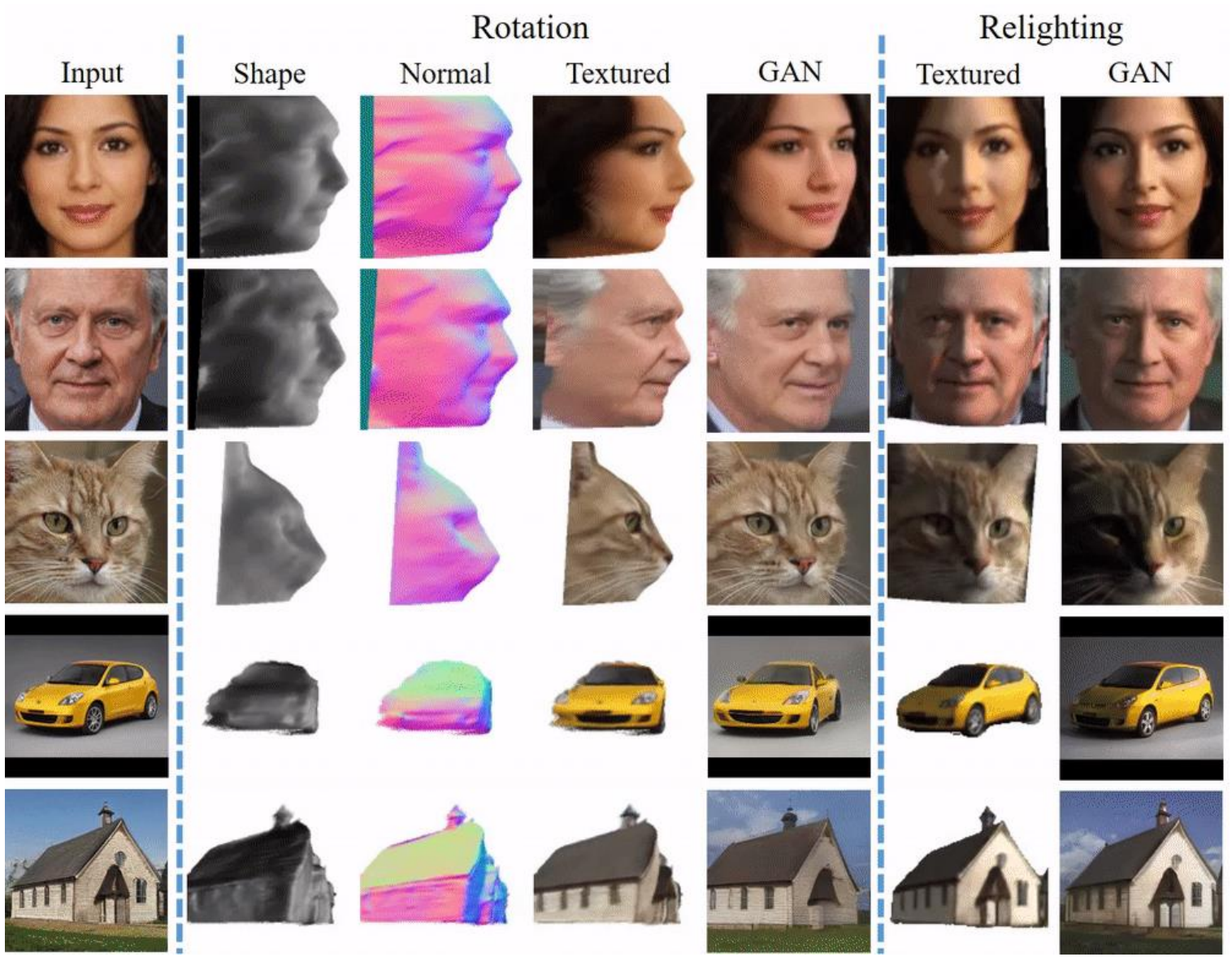
Härkönen, Erik, et al. "GANSpace: Discovering Interpretable GAN Controls." *NIPS* 2020.

Shen, Yujun, and Bolei Zhou. "Closed-form factorization of latent semantics in gans." *CVPR* 2020.

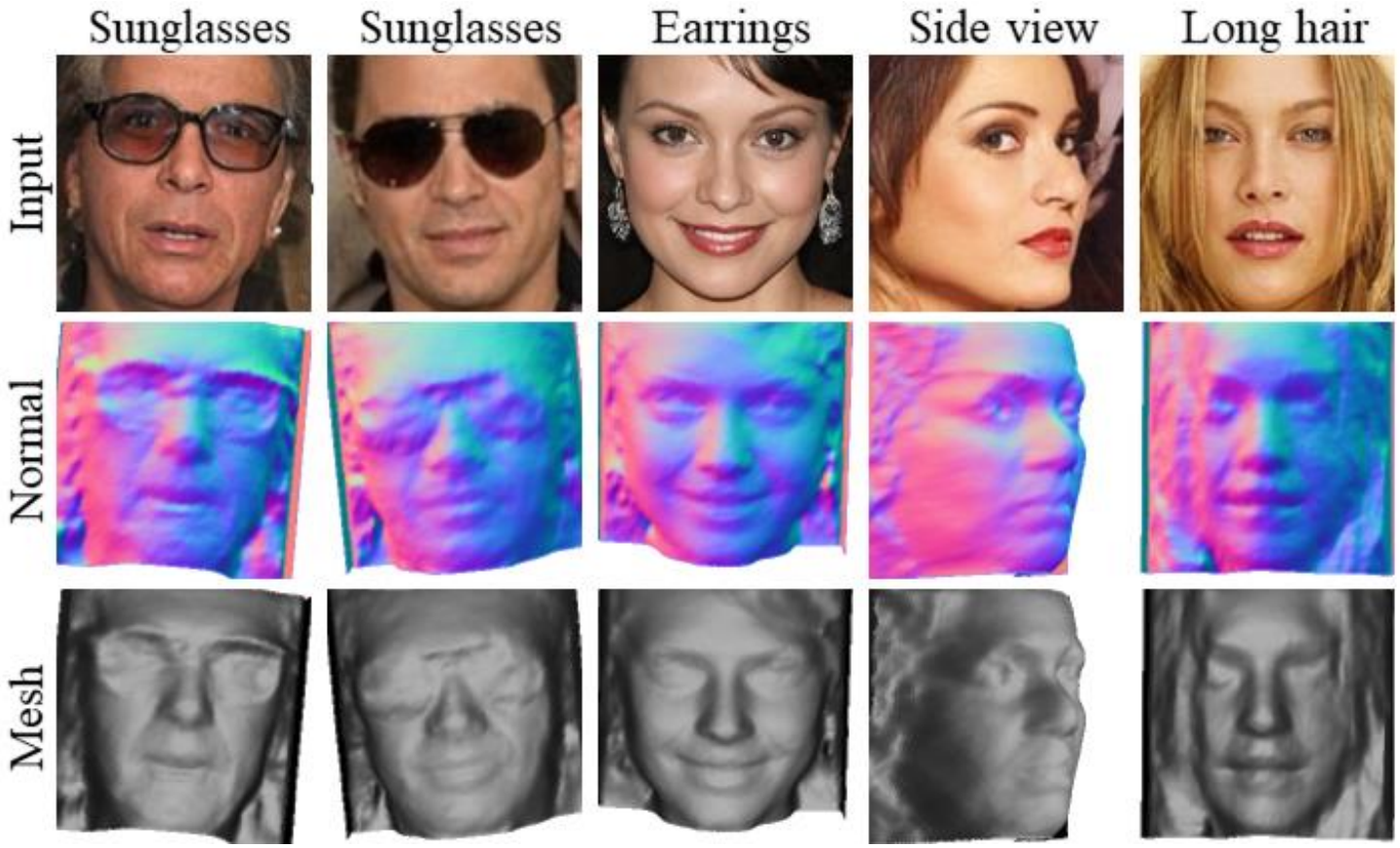
More Results

More Results

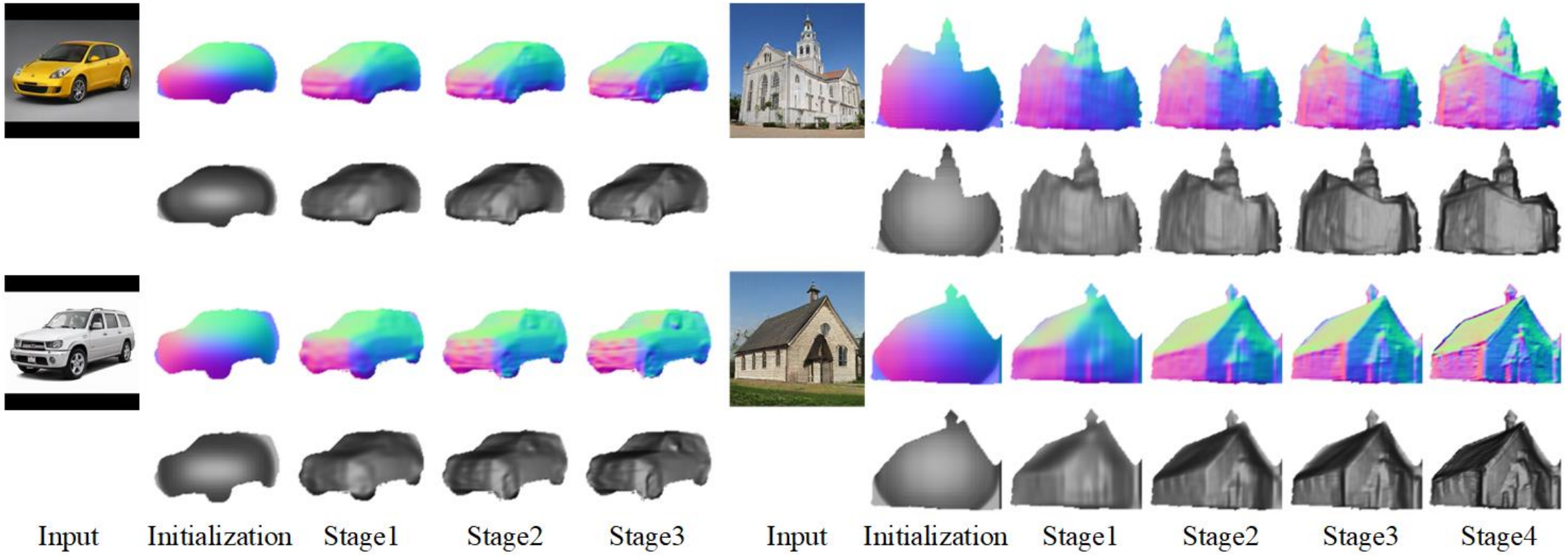
Demo


Rotation
Relighting
Input
Shape
Normal
Textured
GAN
Textured
GAN

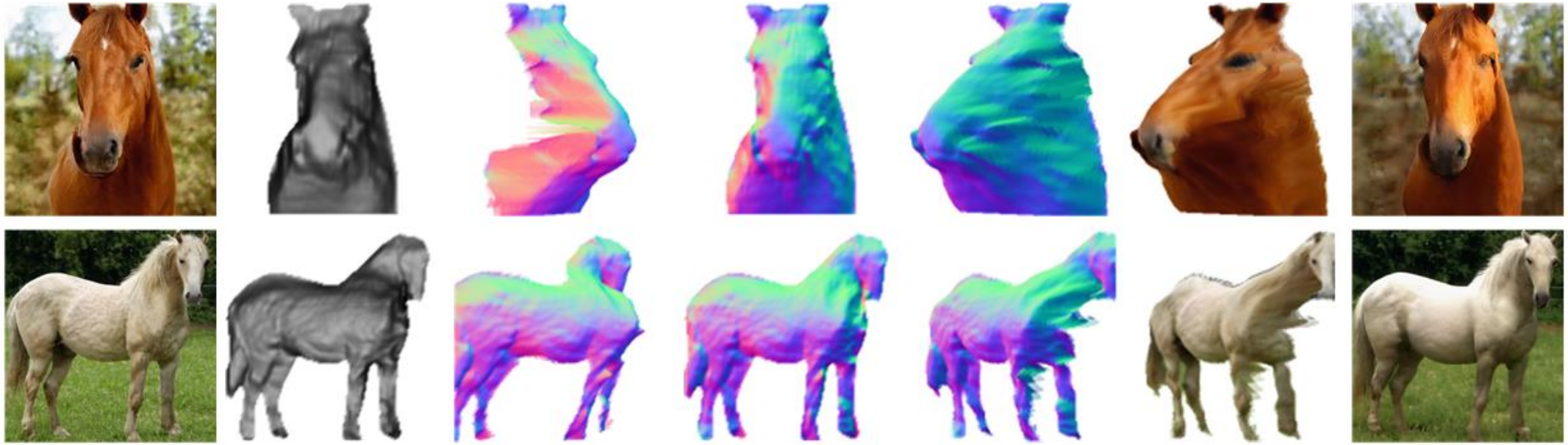
Challenging Cases



Effects of Iterative Training



Results on Horses



Summary

- We demonstrate that ***2D GANs inherently capture the underlying 3D geometry of objects by learning from RGB images***.
- Our method is a powerful approach for unsupervised 3D shape learning from unconstrained 2D images, and ***does not rely on the symmetry assumption***.

  We are doing ***Shape-from-X***, where ***X=GAN***.
- We achieve accurate 3D-aware image manipulation via GANs ***without borrowing external 3D models***.
- Our method provides a new perspective for 3D shape generation.

ICLR

Thanks!

Code on github