JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Shape, Pose, and Appearance from a Single Image via Bootstrapped Radiance Field Inversion

Dario Pavllo
David Joseph Tan
Marie-Julie Rakotosaona
Federico Tombari

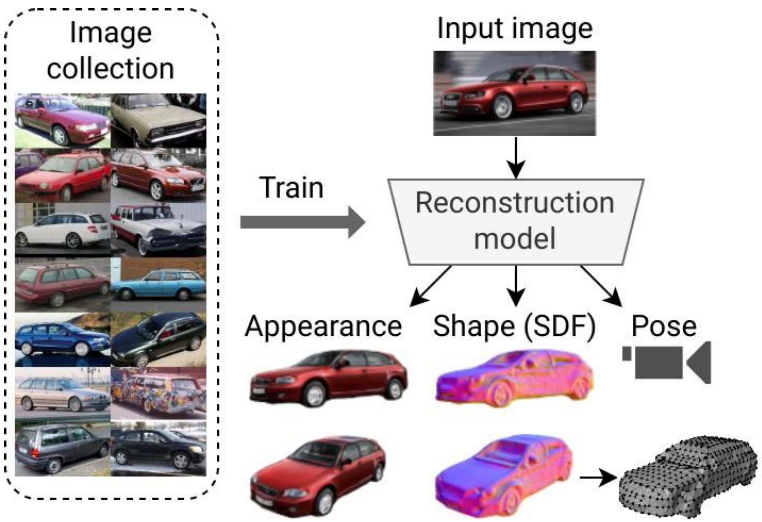
TUE-PM-025

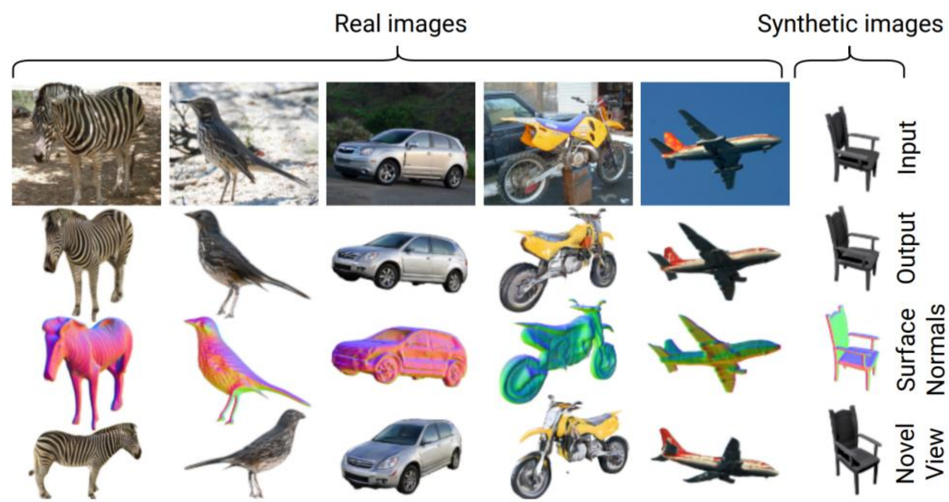
CVPR 2023

Google ETH zürich TUM

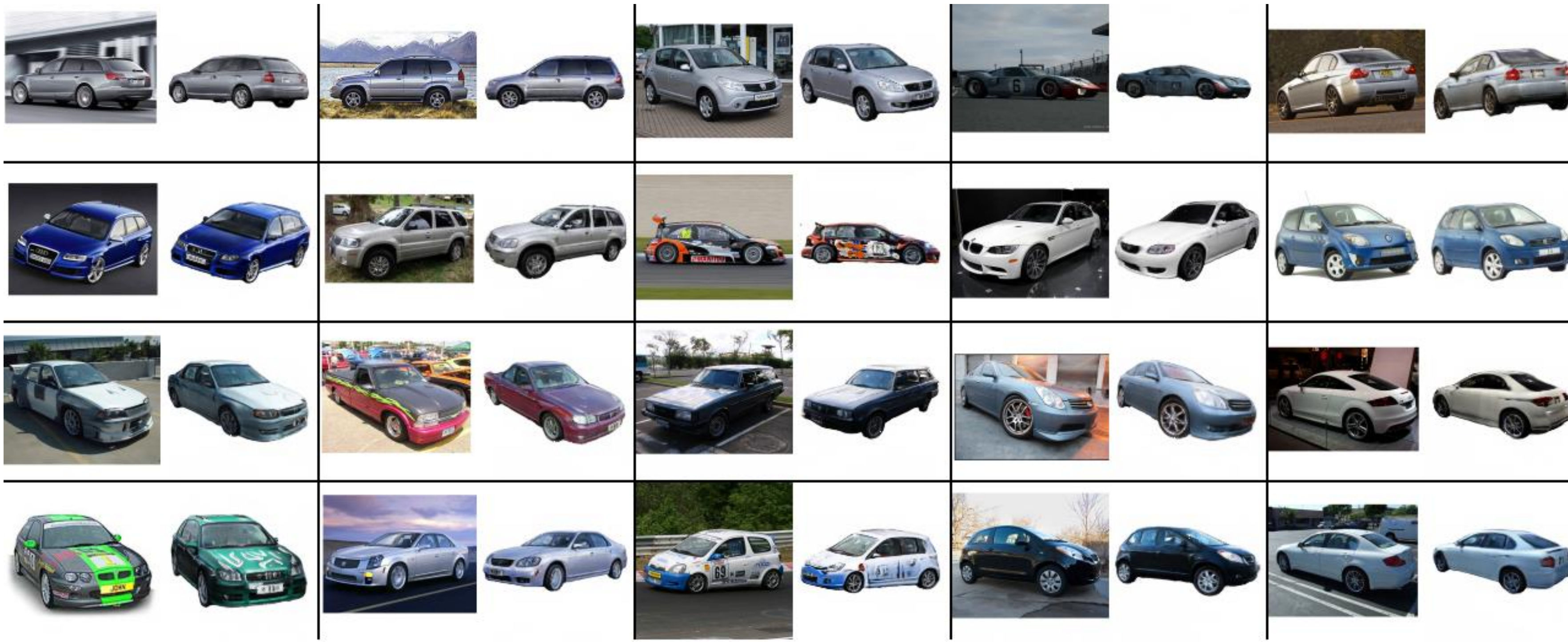
NeRF from a single image

- **Goal:** learn a model to reconstruct 3D shape, pose, and appearance from a single view of an object
  - NeRF with SDF shape parameterization
- Training **without** multiple views
- Focus on real datasets as opposed to synthetic datasets
  - Poses may be inaccurate





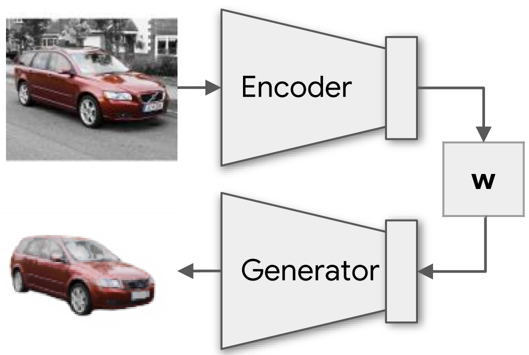
Reconstruction demo



Reconstruction demo

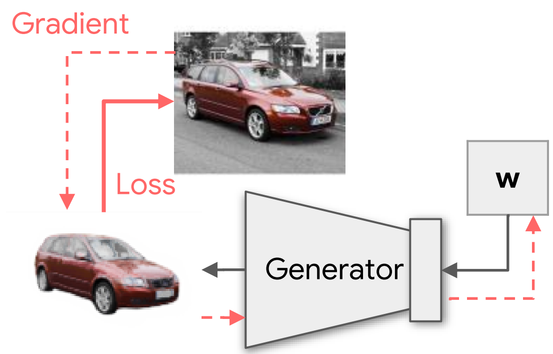
Reconstruction demo

Reconstruction frameworks & motivation



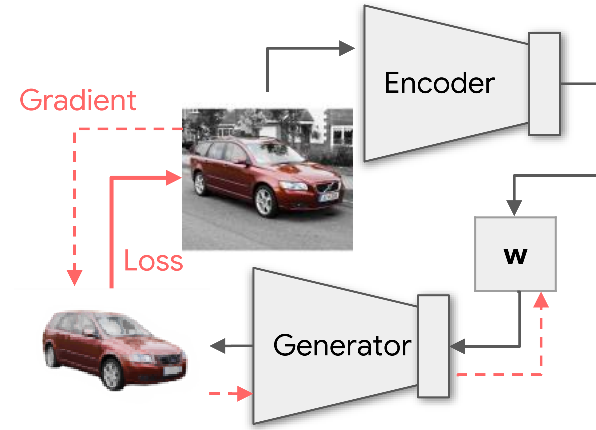
Encoder-based

- E.g. CMR, PixelNeRF, Pix2NeRF
- Autoencoder setup: a ConvNet encoder predicts a latent code **w** which is decoded into a 3D scene
- **Fast but relies on accurate poses**, which are typically available only on synthetic datasets



GAN Inversion via Optimization

- Leverages a pretrained unconditional GAN (e.g. pi-GAN, EG3D)
- Gradient-based optimization w.r.t. pose and latent code **w**
- **Better results, robust to inaccurate poses, but very slow**

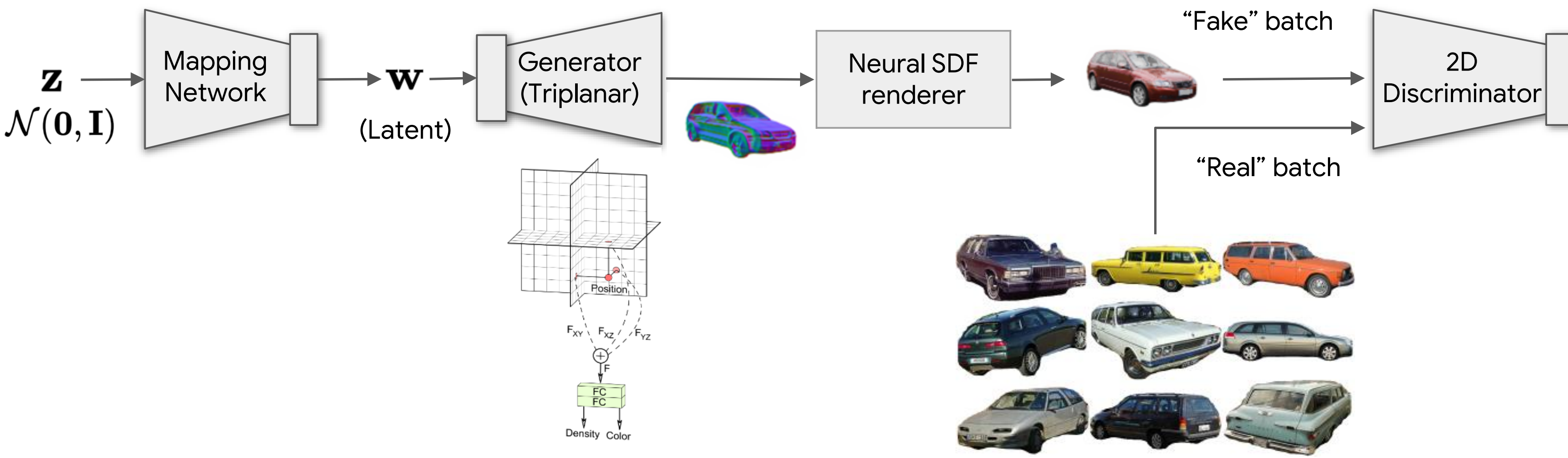


Bootstrapped Inversion (Ours)

- **Not explored for NeRFs**
- A ConvNet encoder produces a first guess of the pose and latent code
- These are then refined via optimization for a small number of steps

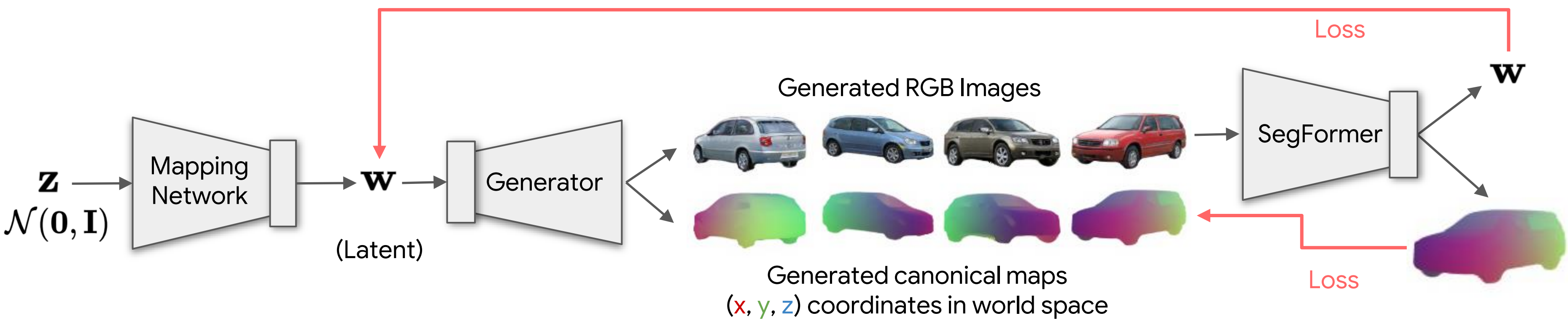
Method (1/3): unconditional GAN training

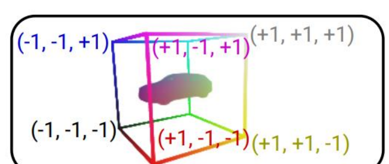
- We train an unconditional 3D GAN on the collection of images (foundation model)
- Backbone inspired by EG3D (triplanar NeRF representation), with some improvements
  - **SDF representation (VolSDF)** → better surface reconstruction
  - **Color mapping:** disentanglement between color and semantics → facilitates inversion & manipulation
  - **Path length regularization** → facilitates inversion



Method (2/3): bootstrapping & pose estimation

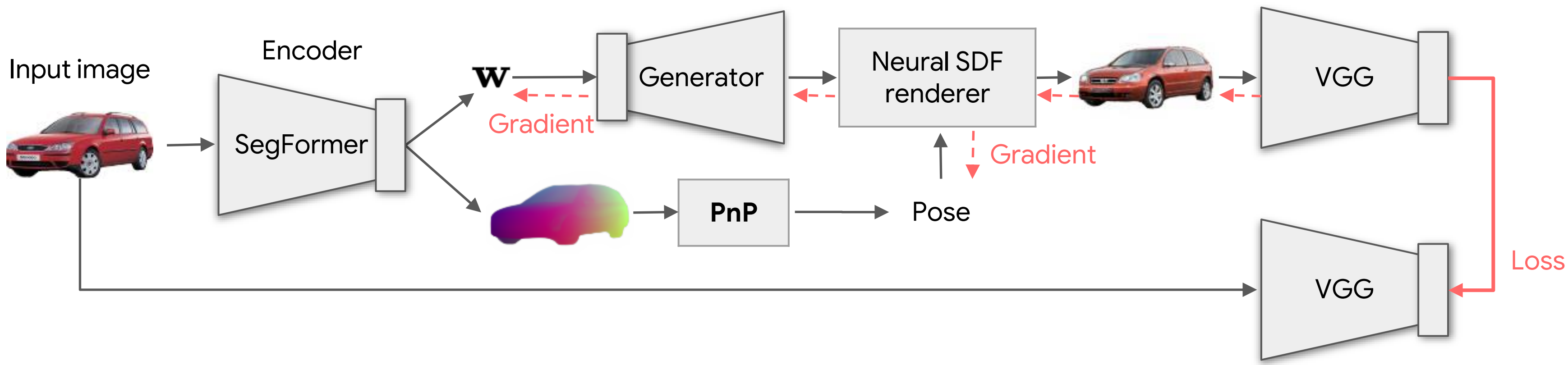
- Using generated data, we learn a model that jointly predicts the pose and the latent code **w**
- **NOCS approach**: predict a canonical map, convert to point cloud, and recover the pose using a PnP solver
  - More robust than directly regressing the pose parameters
  - Pseudo-ground-truth canonical maps generated using the GAN itself (rasterize xyz instead of rgb)





Method (3/3): hybrid inversion

- At inference, we initially estimate the latent code **w** and pose using the previous model
- These are then refined via optimization using a VGG loss
  - In practice, we use multiple crops to reduce the variance of the gradient
- We further investigated strategies to achieve maximum speed
  - We can invert an image in as few as **10 steps** (vs 100s of related work)



Datasets & Evaluation

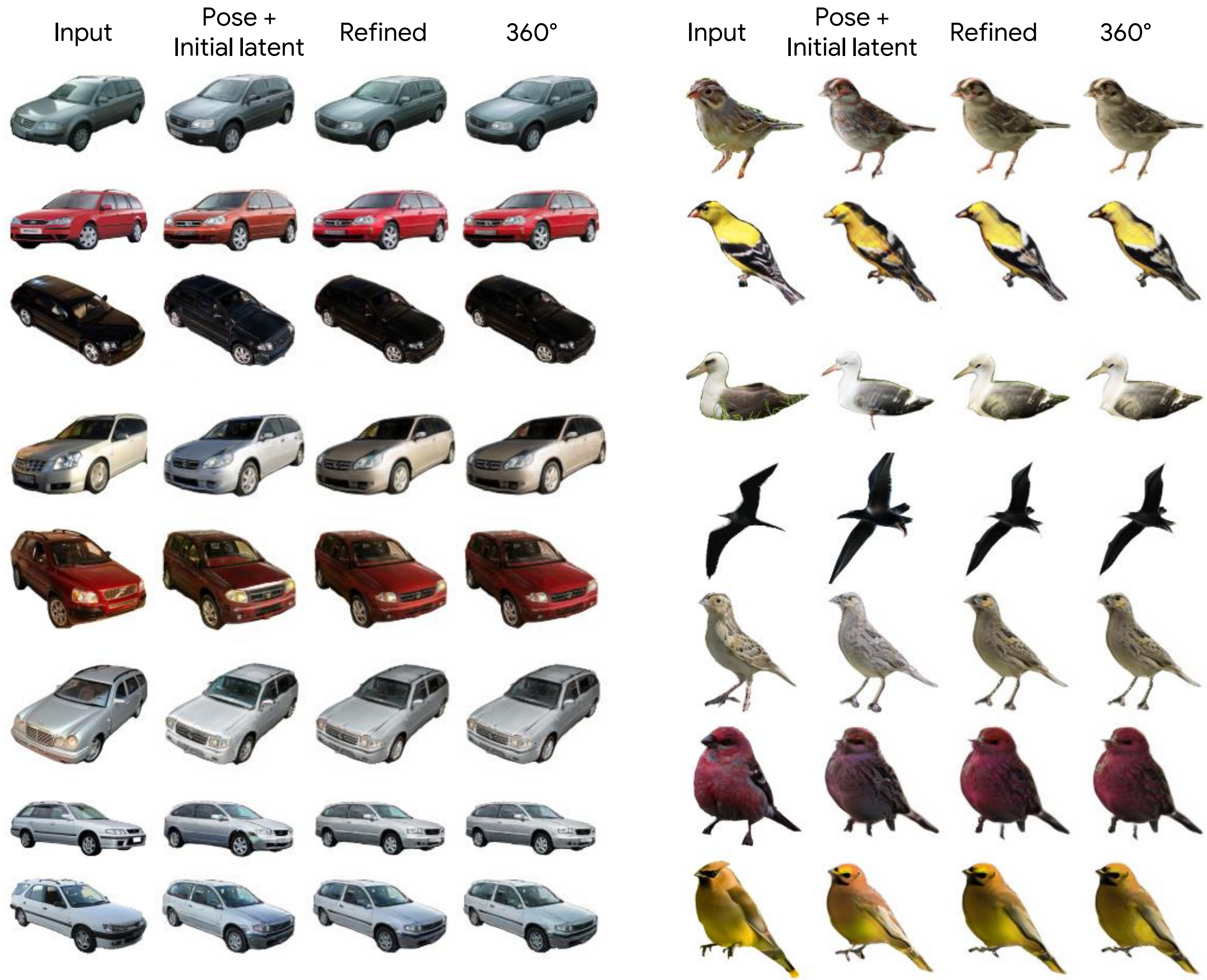
- Approach evaluated on a mix of synthetic and real datasets
- For real datasets, we compare to CMR (Kanazawa et al. 2018) & follow-up papers (U-CMR, UMR, ...)
    - Reconstruction evaluated using IoU against input view ⟶ easy to overfit
    - No ground-truth novel views ⟶ quality evaluated using FID on renderings from random views
- For ShapeNet, in addition to the FID, we evaluate the PSNR on novel views from the test set
    - Comparison against Pix2NeRF (Cai et al. 2022)

Qualitative results

Real images

Left:
Pascal3D+ Cars
(test set)

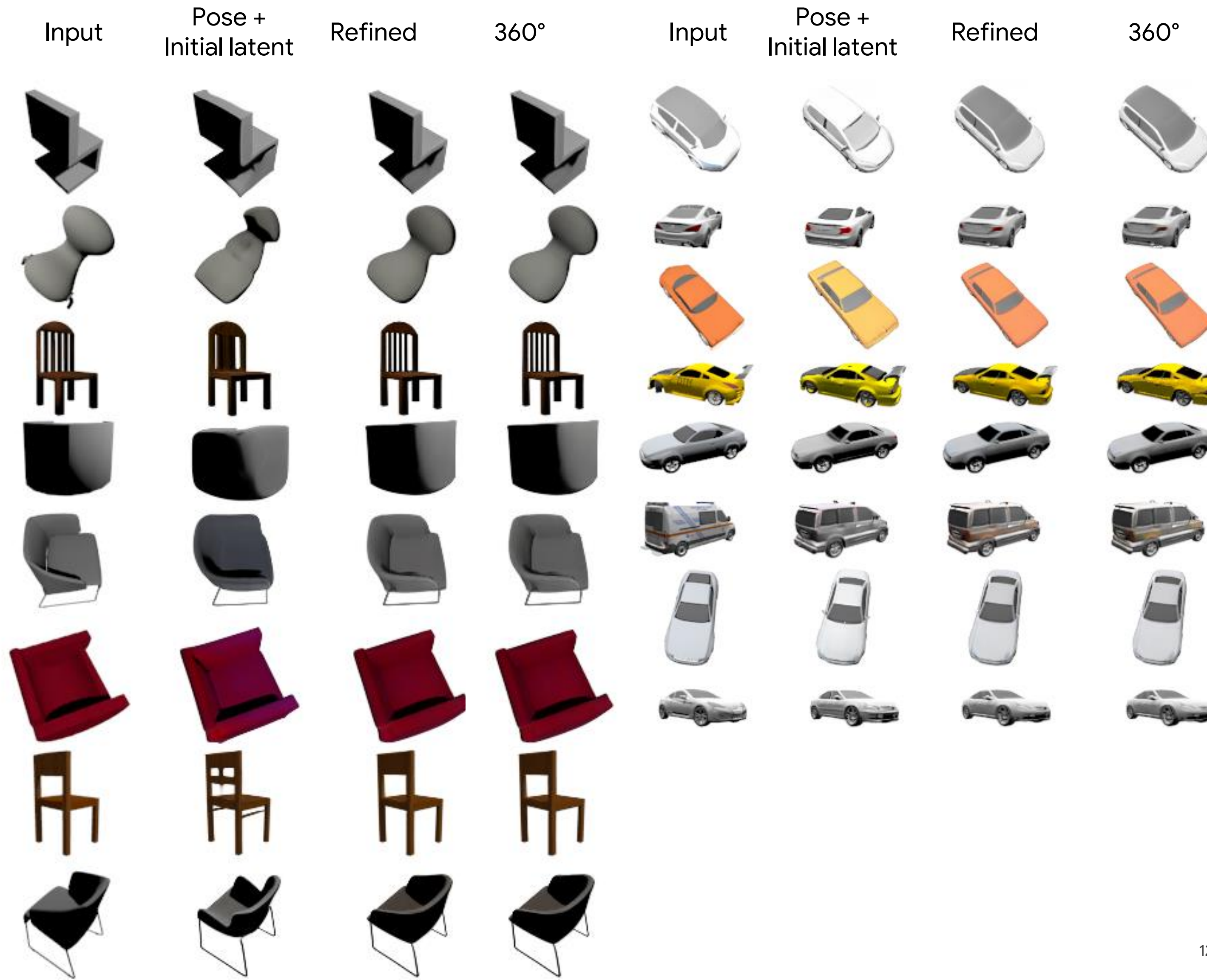
Right:
CUB Birds
(test set)



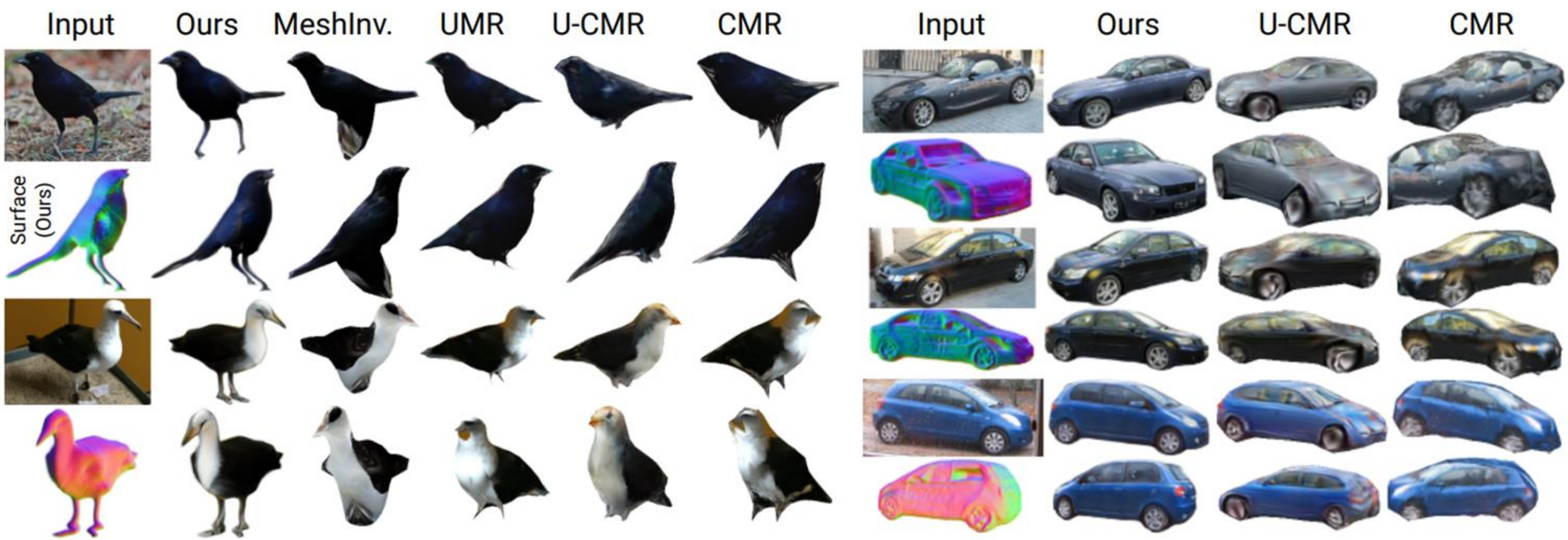
Qualitative results
Synthetic images

Left:
ShapeNet Chairs
(test set)

Right:
ShapeNet Cars
(test set)



Side-by-side comparison



Qualitative results

End-to-end reconstruction pipeline on real images in the wild (ImageNet)

End-to-end reconstruction: quantitative results

- Even without inversion (N = 0 steps), we show an improvement over existing work
  - 36% decrease in FID on CUB over SOTA; 9% increase in IoU on P3D Cars
  - 68% decrease in FID on ShapeNet Chairs; 83% decrease in FID on CARLA
- Applying our hybrid inversion approach further widens the gap

Method	Pascal3D+ Cars IoU ↑	Pascal3D+ Cars FID ↓	CUB Birds IoU ↑	CUB Birds FID ↓
CMR [27]	0.64	273.28	0.706	105.04
U-CMR [17]	0.646	223.12	0.644	69.42
UMR [33]	-	-	0.734	43.83
SDF-SRN [34]	0.81	254.90	-	-
ViewGeneralization [2]	0.78	-	0.629	-
StyleGANRender [71]	0.80	-	-	-
Ours Init. (N=0)	**0.883**	**75.90** (15.08)	**0.739**	**28.15**
MeshInv. (N=200) (∗) (†) [69]	-	-	0.752	31.60
Ours Hybrid Slow (N=30) (†)	**0.920**	73.53 (14.36)	**0.844**	**24.70**
Ours Hybrid Fast (N=10) (†)	0.917	**73.12** (14.36)	0.835	25.65

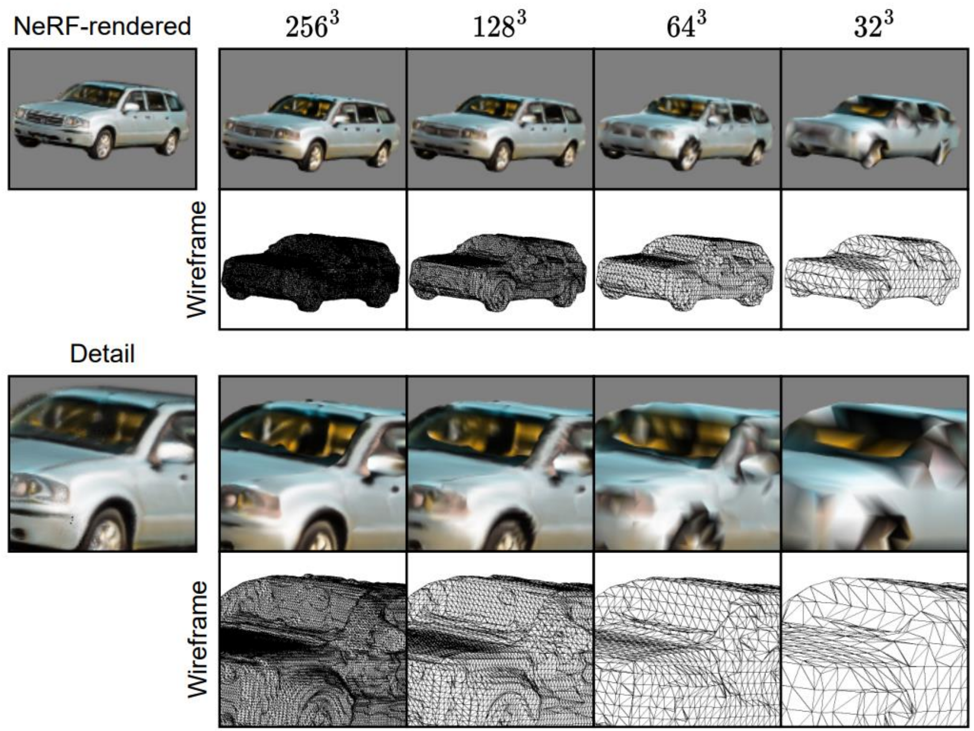
Evaluation on real datasets
(Novel views **not** available)

Method	SRN Cars PSNR ↑	SRN Cars SSIM ↑	SRN Cars FID ↓	SRN Chairs PSNR ↑	SRN Chairs SSIM ↑	SRN Chairs FID ↓	CARLA FID ↓
Pix2NeRF [4]	-	-	-	18.14	0.84	26.81	38.51
Ours Init. (N=0)	18.54	0.848	12.39	18.26	0.857	8.64	6.49
Ours Hybrid Slow (N=30)	**19.55**	**0.864**	**11.37**	**19.36**	**0.875**	**7.44**	**5.97**
Ours Hybrid Fast (N=10)	19.24	0.861	12.26	19.02	0.871	7.62	6.18

Evaluation on synthetic datasets
(Novel views **available** on ShapeNet)

Bonus: extraction of a triangle mesh from the SDF

- The adoption of an SDF representation allows us extract its 0-level set and obtain a colored triangle mesh
- Results at different step sizes for the marching cubes algorithm



JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Thank you for your attention!

Feel free to visit our poster:

TUE-PM-025

(Tuesday afternoon session, stand 25)