GANCRAFT

AN UNSUPERVISED 3D NEURAL METHOD FOR WORLD-TO-WORLD TRANSLATION

ARUN MALLYA, SENIOR RESEARCH SCIENTIST

2021 ICCV OCTOBER 11-17 VIRTUAL

AIM 2021

GANCRAFT

Unsupervised 3D Neural Rendering of Minecraft Worlds

(oral presentation)

Zekun Hao[1,2], Arun Mallya[1], Serge Belongie[2], Ming-Yu Liu[1]

[1]NVIDIA [2]Cornell Tech

NVIDIA

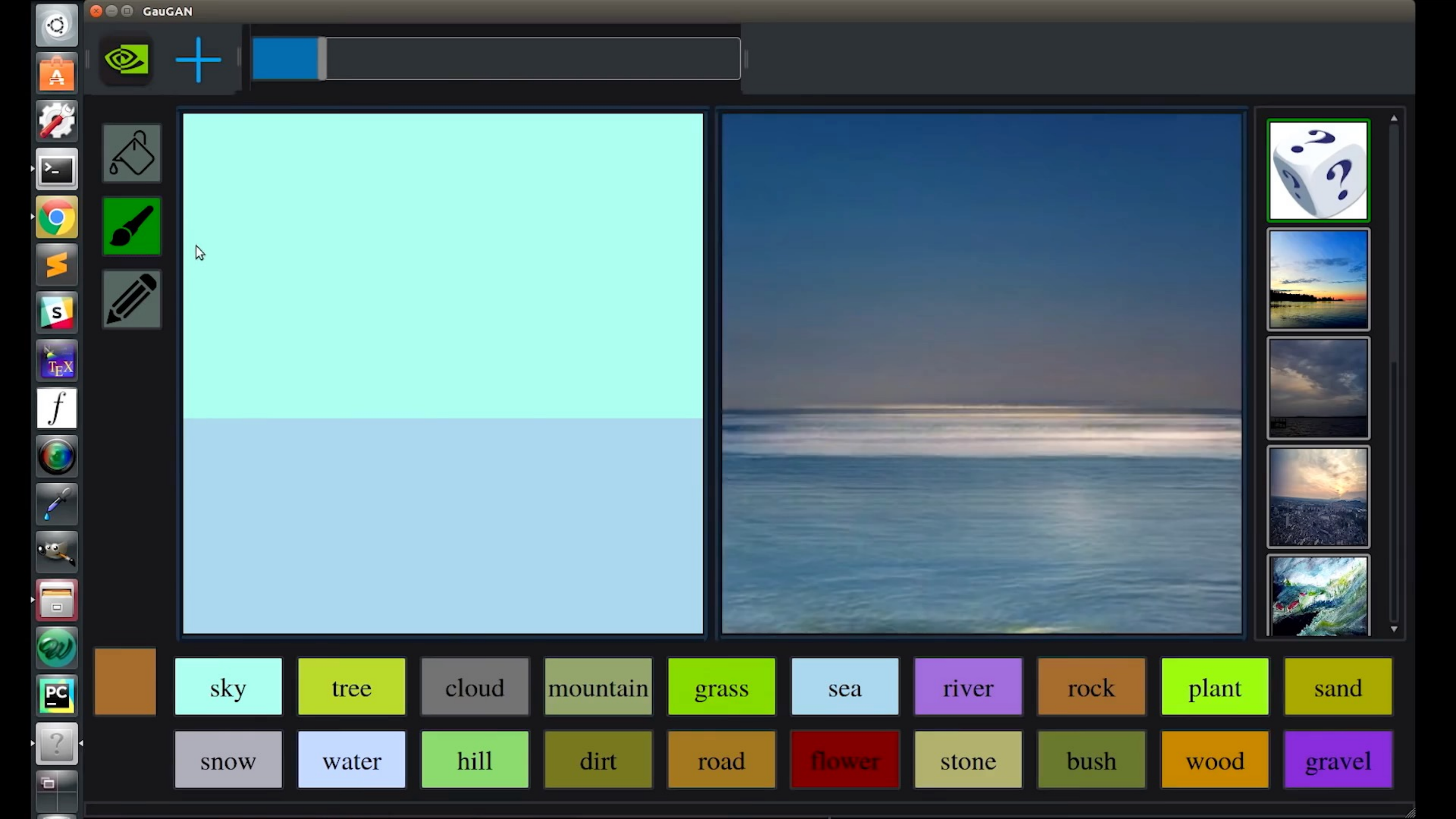
Cornell Bowers CIS Computer Science

CORNELL TECH

NVIDIA

2D image-to-image translation

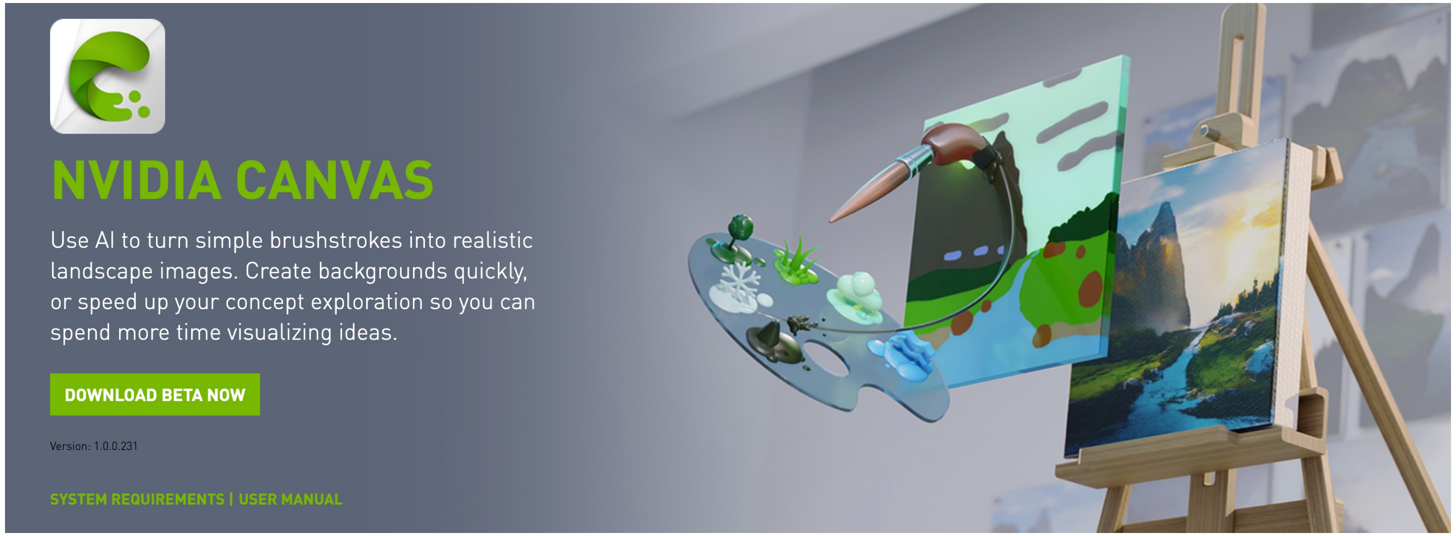
- We have amazing technologies that produce photorealistic outputs given Microsoft Paint-like inputs!


GauGAN
sky
tree
cloud
mountain
grass
sea
river
rock
plant
sand
snow
water
hill
dirt
road
flower
stone
bush
wood
gravel

2D image-to-image translation

- We have amazing technologies that produce photorealistic outputs given Microsoft Paint-like inputs!
- Trained in a supervised fashion using millions of (segmentation map, real image) pairs

- Plug: https://www.nvidia.com/en-us/studio/canvas/ is free and publicly available!



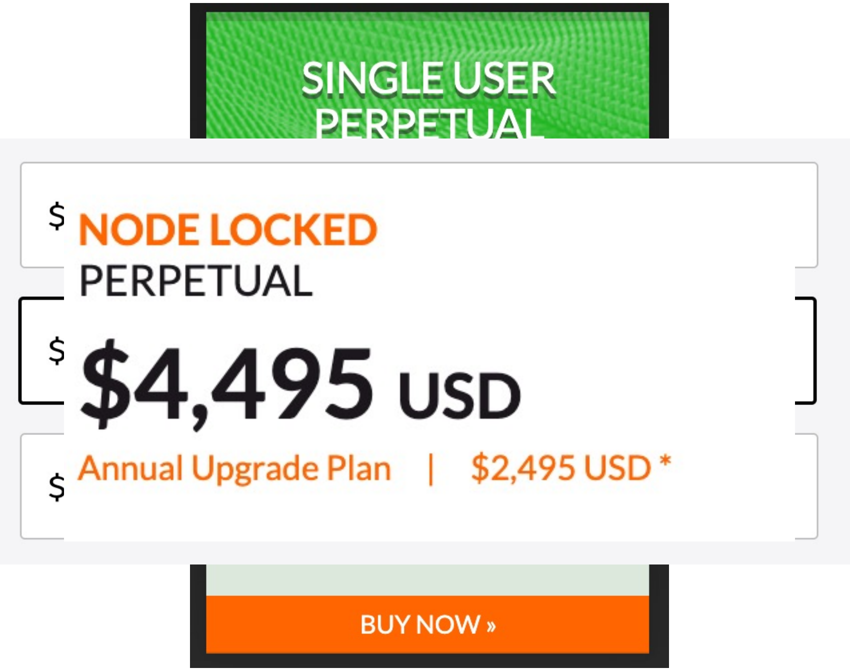
What about 3D content creation?

AND expensive
Fancy and complicated tools

AUTODESK® MAYA®

ZBrush 2021.7.1

Houdini™ 19



- Requires years of experience
- Lots of time and money
- Great for professionals!
- But big barrier to entry for most people

Is there an easier way?

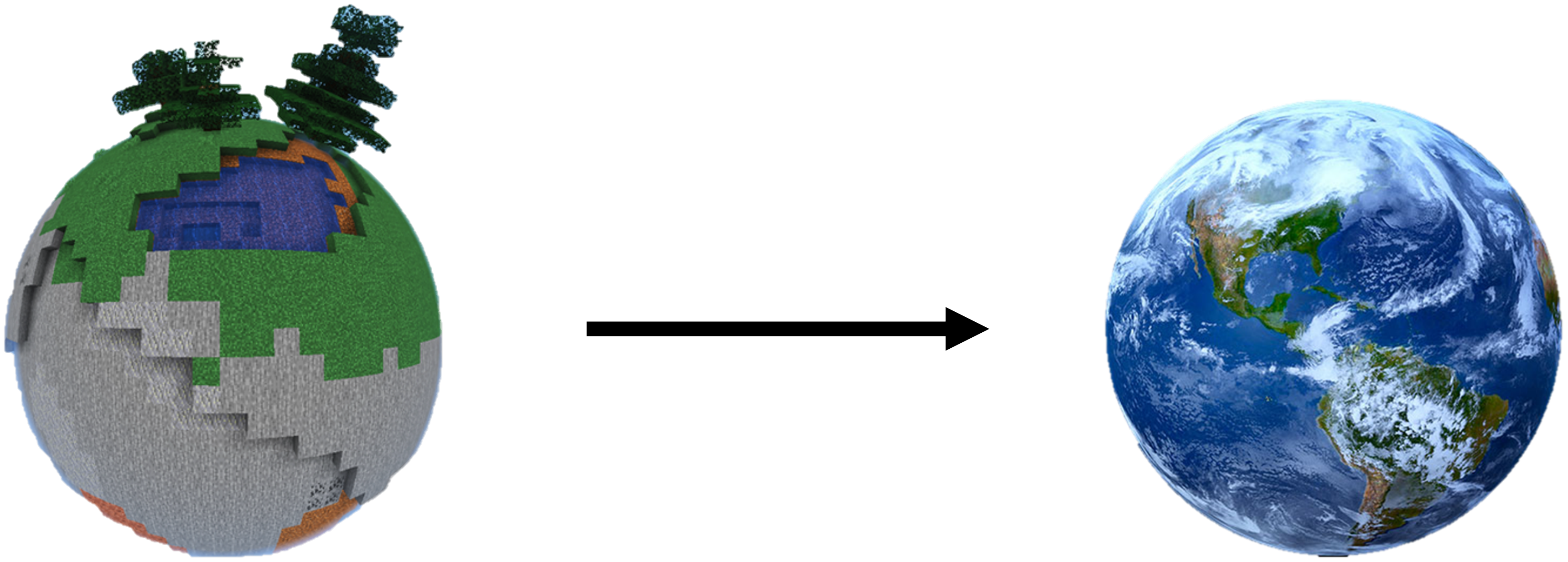
- YES!
- Even kids can make 3D models!

Its digital counterpart

NVIDIA.

World-to-world translation

- GANcraft extends the task of 2D image-to-image translation to 3D
- It translates an input 3D world to another view-consistent 3D world
- Our work focuses on converting Minecraft-style semantically-labeled block worlds to realistic-looking worlds, without paired supervision



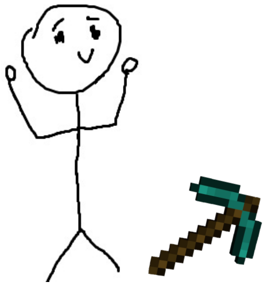
Creating photorealistic 3D worlds is challenging

- 2D landscape images are widely available on the internet

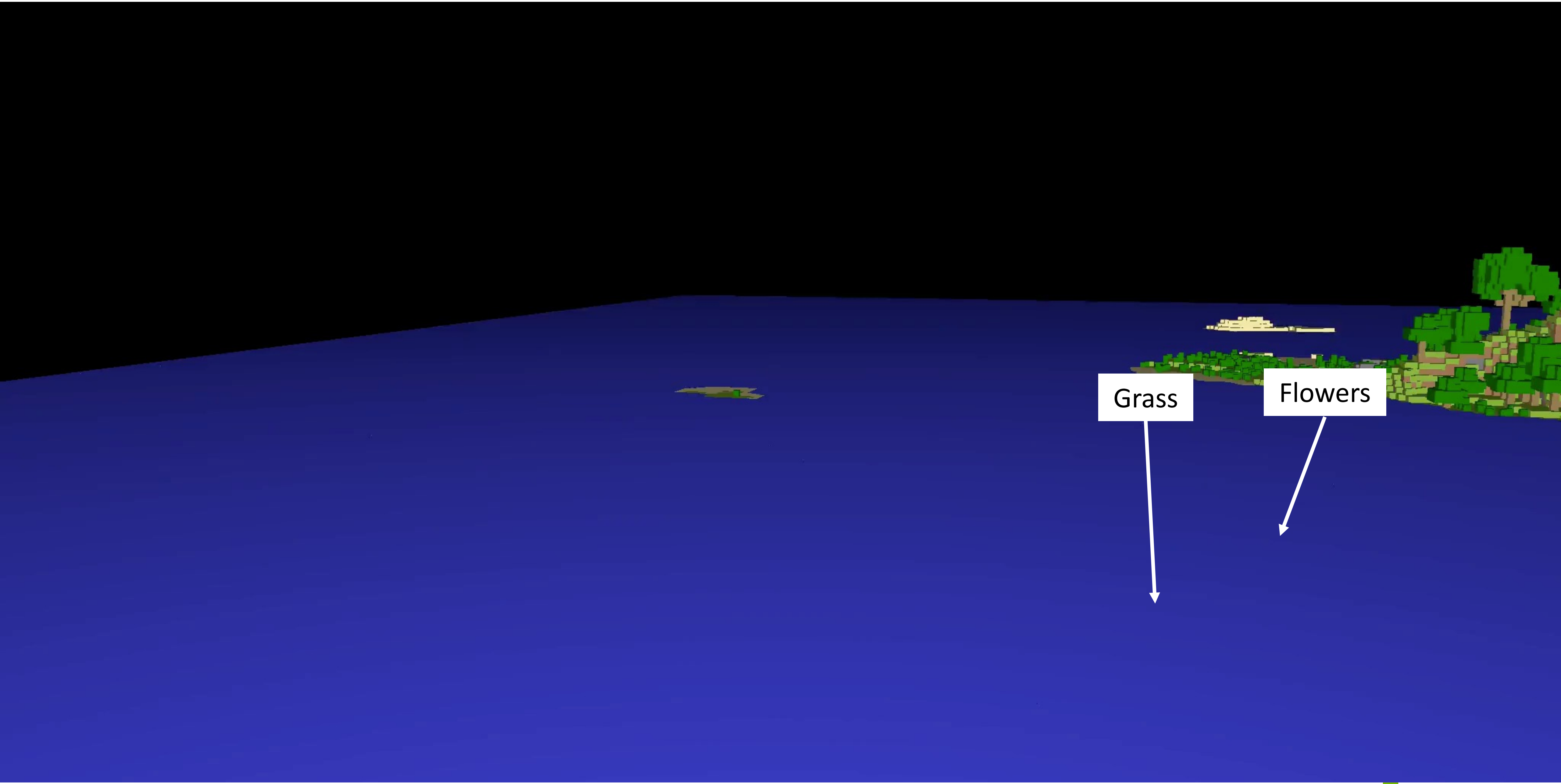
- But what about paired 3D and 2D image data?

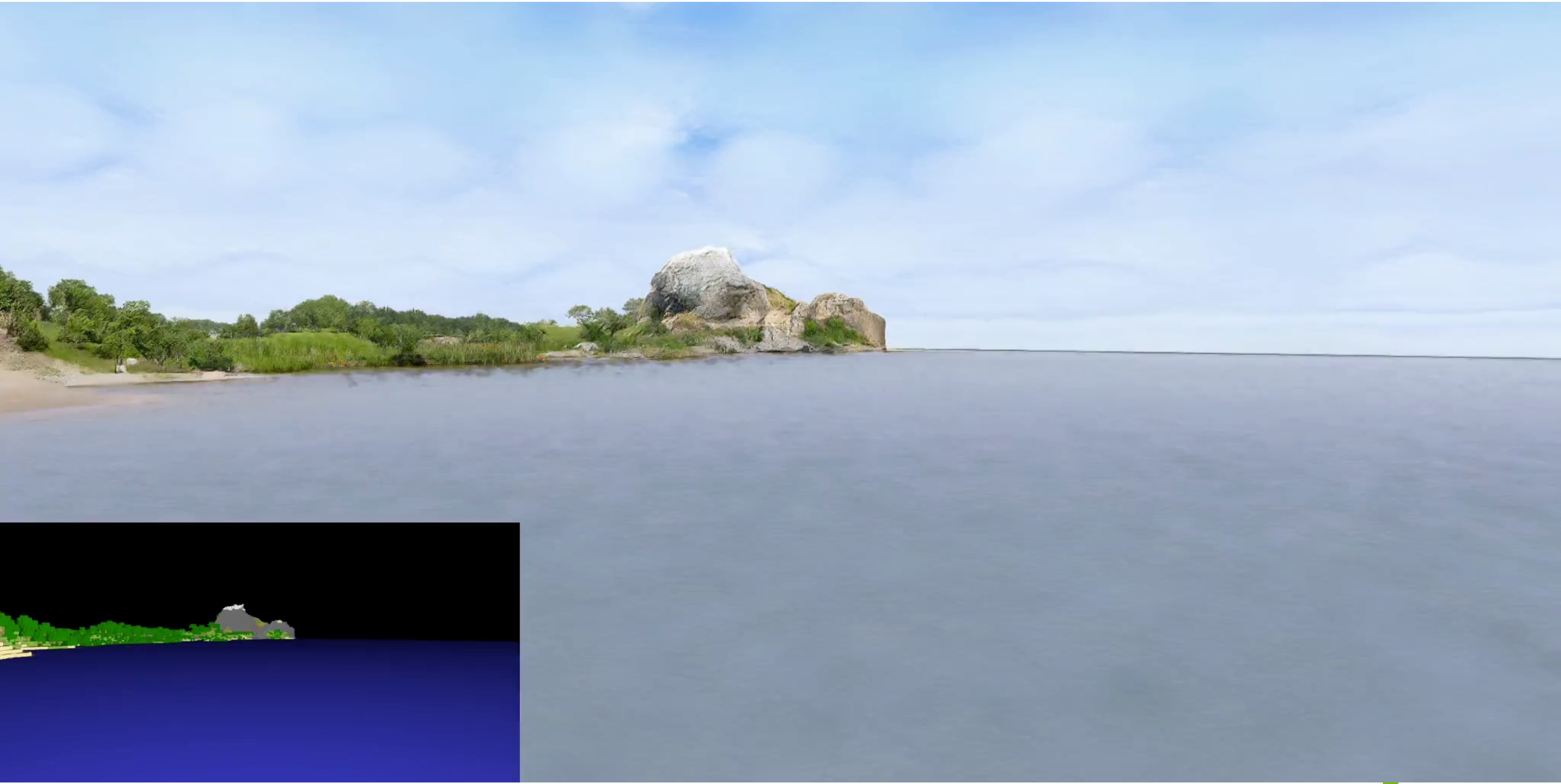
GANcraft converts voxel worlds to reality!

GANcraft

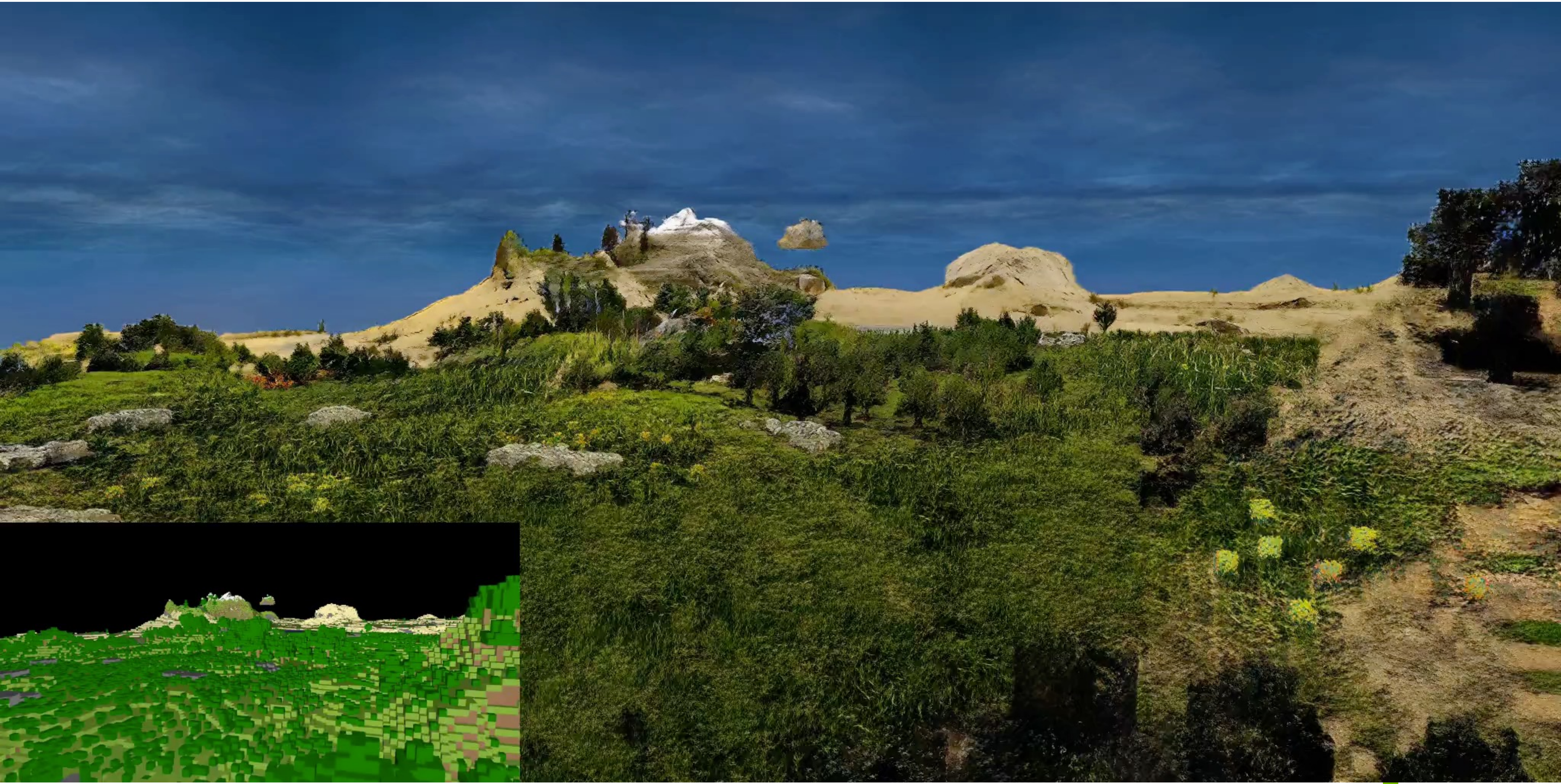


High-resolution results
1024 x 2048 pixels, 30 fps

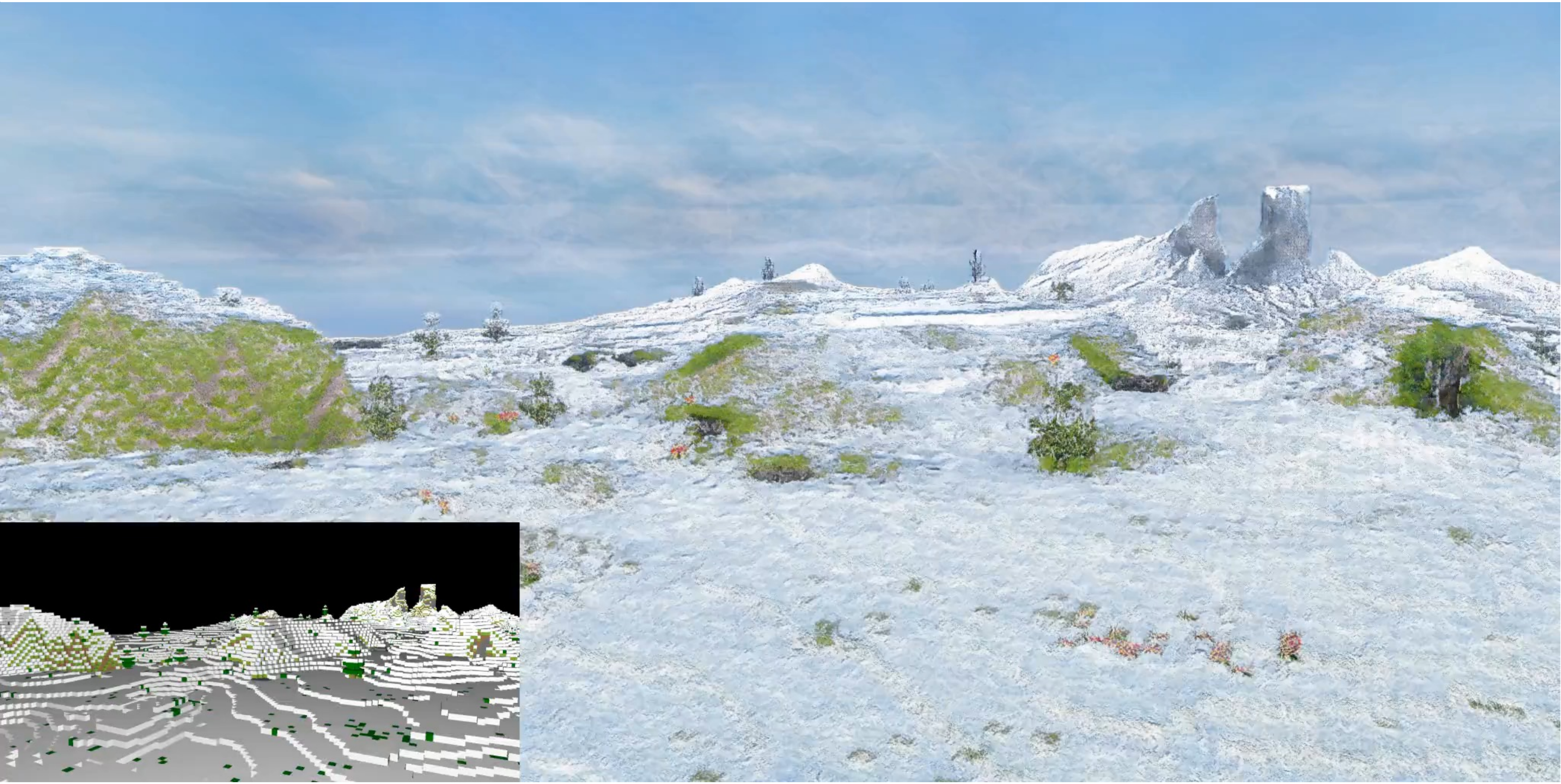

Grass
Flowers



GANcraft generalizes to new worlds with significant label distribution shifts - **desert**



GANcraft generalizes to new worlds with significant label distribution shifts - **snow**



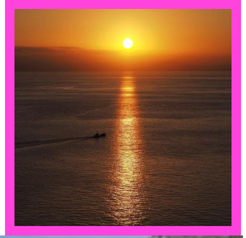
GANcraft generalizes to new worlds with unique input geometry – **valleys and arches**

Style interpolation

GANcraft can render worlds with different style-conditioning images style-conditioning image



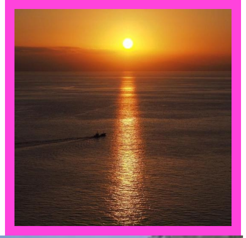


style-conditioning image





The
"Why don't you just use im2im translation? "
Question
aka
Comparison with baselines

MUNIT (ECCV’18)

Flickering - generates one image at a time, with no memory of past
Mismatch between segmentation label and texture due to unsupervised training

SPADE (CVPR’19)

Flickering - generates one image at a time, with no memory of past wc-vid2vid (ECCV'20)

View consistent, but fails for large motions due to incremental inpainting

Does not refine blocky geometry

NSVF-W (NeurIPS’20, CVPR’21)

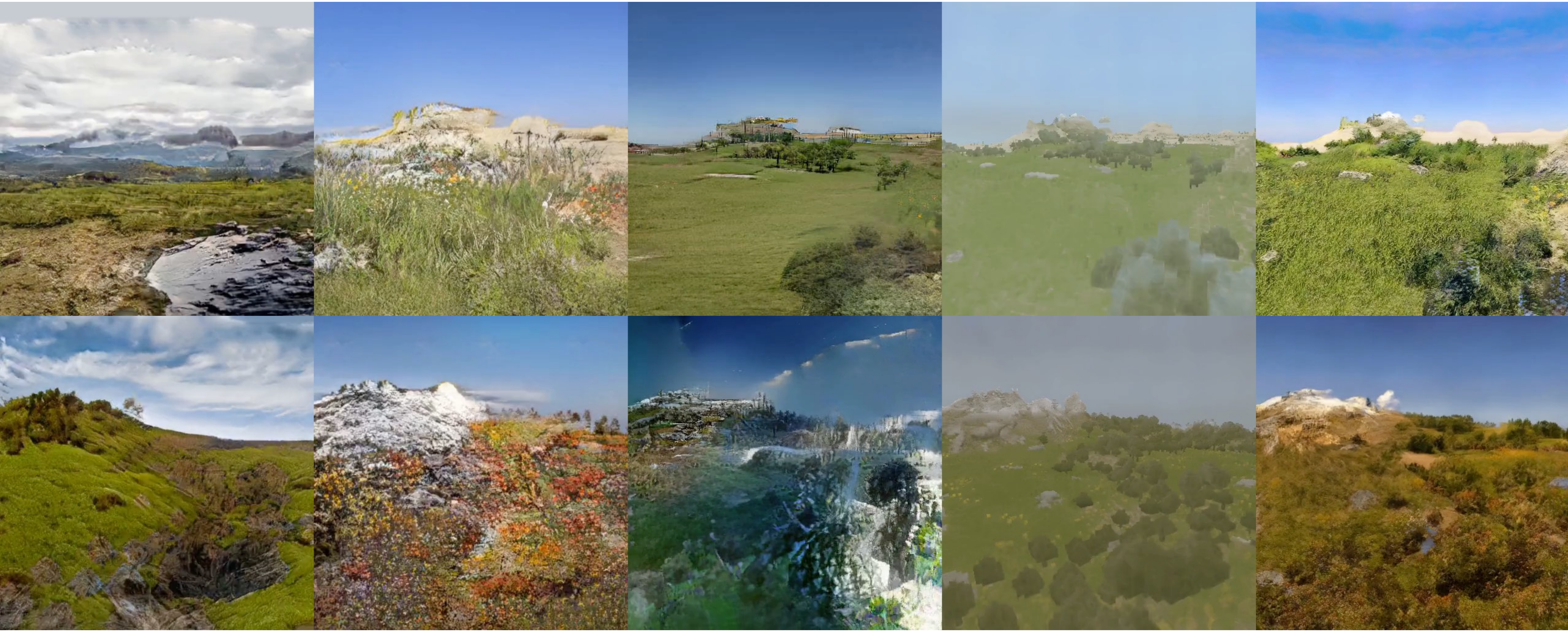
View consistent, but dull unrealistic outputs due to lack of GAN loss

Single-stage rendering, difficult to scale up

GANcraft (ours)

Our full model: view consistent, vivid colors, more realistic
Implicitly refines blocky geometry to learn fine details

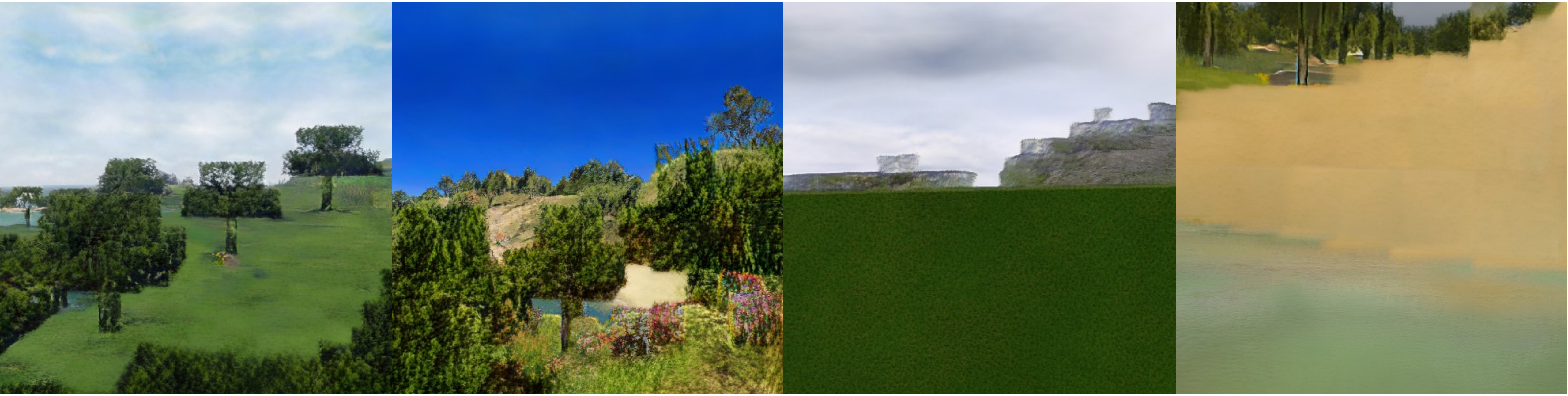
MUNIT (ECCV’18) | SPADE (CVPR’19) | wc-vid2vid (ECCV’20) | NSVF-W (NeurIPS’20, CVPR’21) | GANcraft (ours)



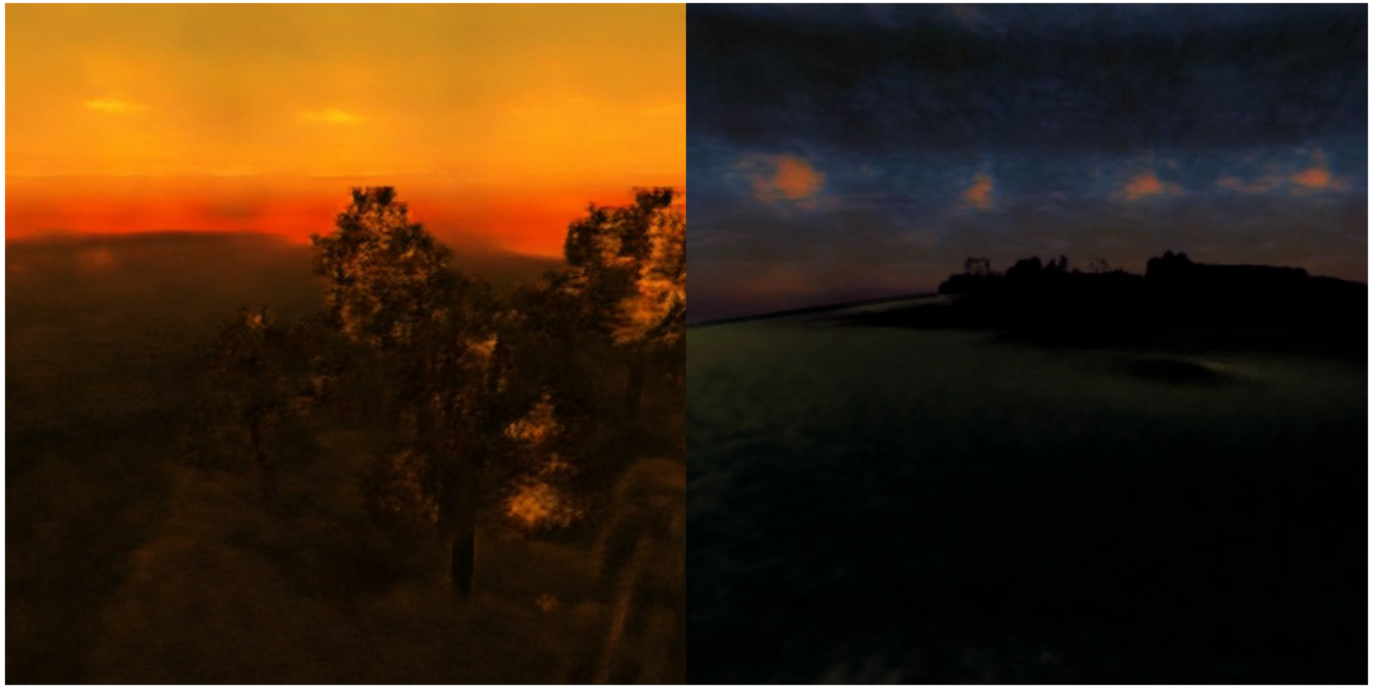
MUNIT (ECCV’18) | SPADE (CVPR’19) | wc-vid2vid (ECCV’20) | NSVF-W (NeurIPS’20, CVPR’21) | GANcraft (ours)

Failure cases

Some regions look blocky due to underlying input geometry



Certain scene-style combinations don't work well



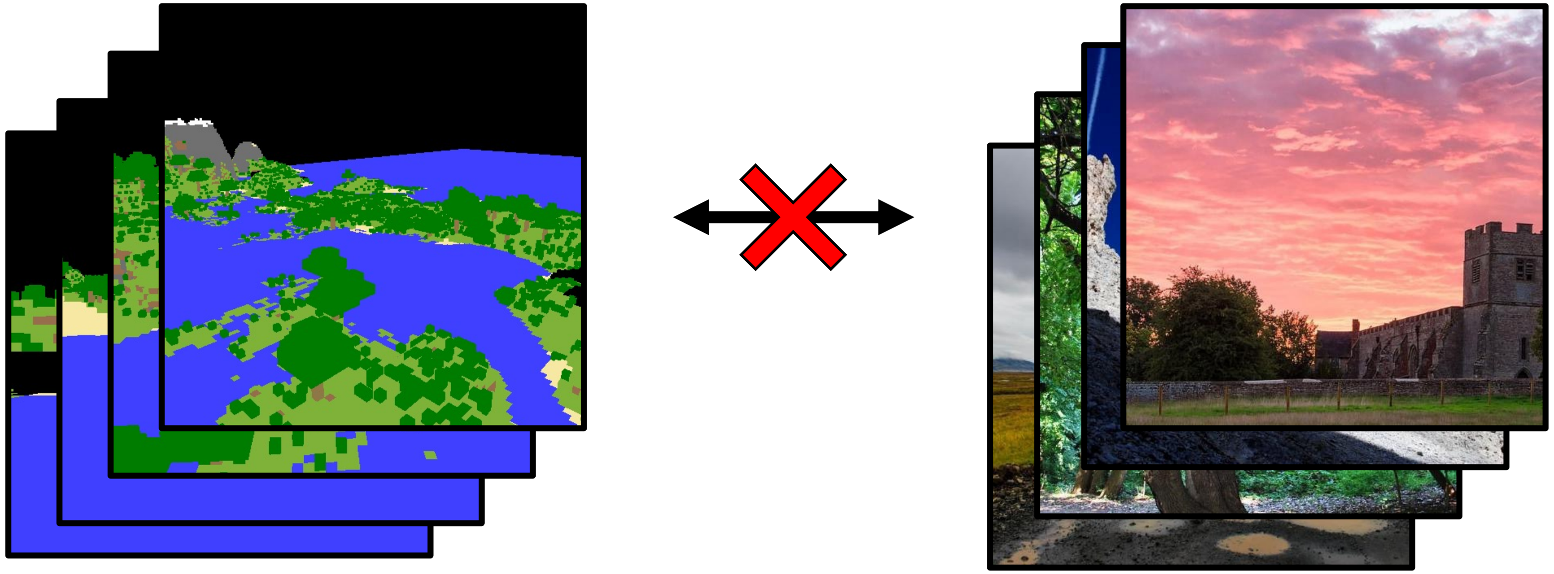
GANcraft details

Problem setting

- We want to render the semantically-labeled voxel world (as in Minecraft) as a realistic-looking world

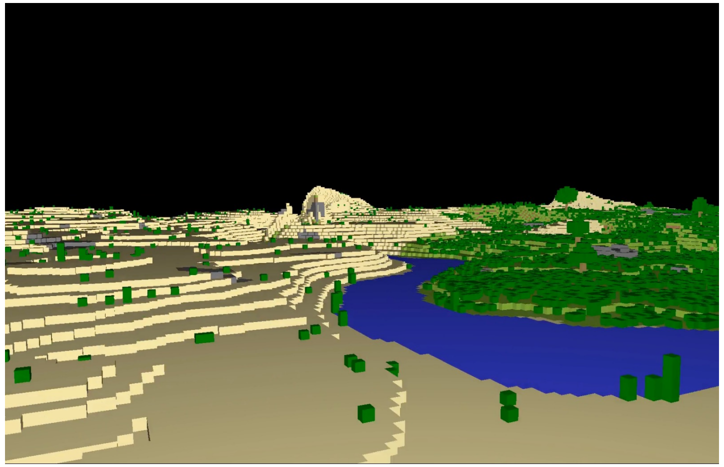
Problem setting

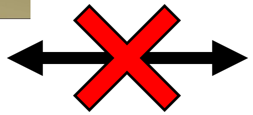
- We want to render the semantically-labeled voxel world (as in Minecraft) as a realistic-looking world
- There is no paired data mapping Minecraft segmentations to real images



Problem setting

- We want to render the semantically-labeled voxel world (as in Minecraft) as a realistic-looking world
- There is no paired data mapping Minecraft segmentations to real images
- Label, geometry and camera pose distribution between Minecraft scenes and real images is very different



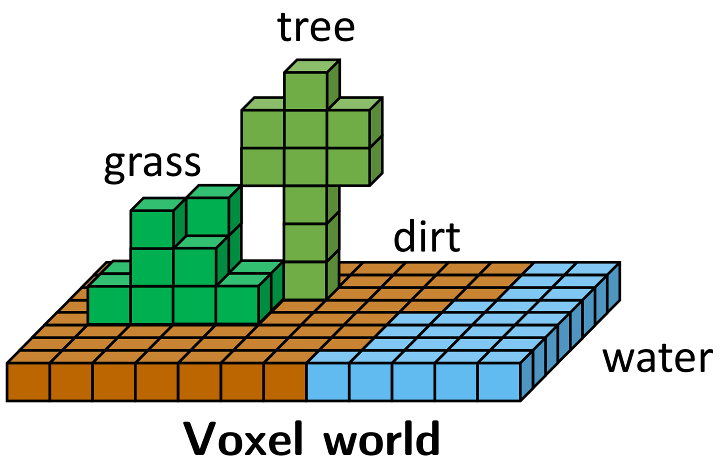


Problem setting

- We want to render the semantically-labeled voxel world (as in Minecraft) as a realistic-looking world
- There is no paired data mapping Minecraft segmentations to real images
- Label, geometry and camera pose distribution between Minecraft scenes and real images is very different

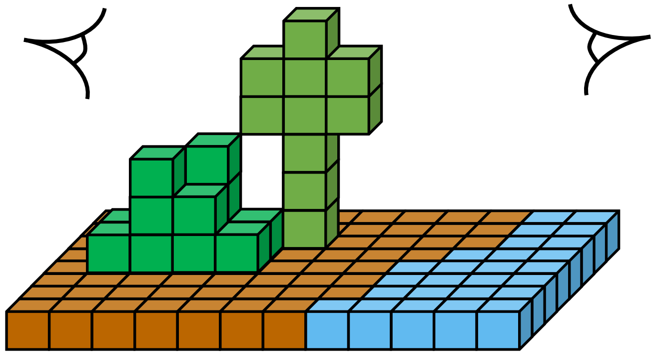
- Solution: **pseudo-ground truths**, and **adversarial training**

Pseudo-ground truth generation



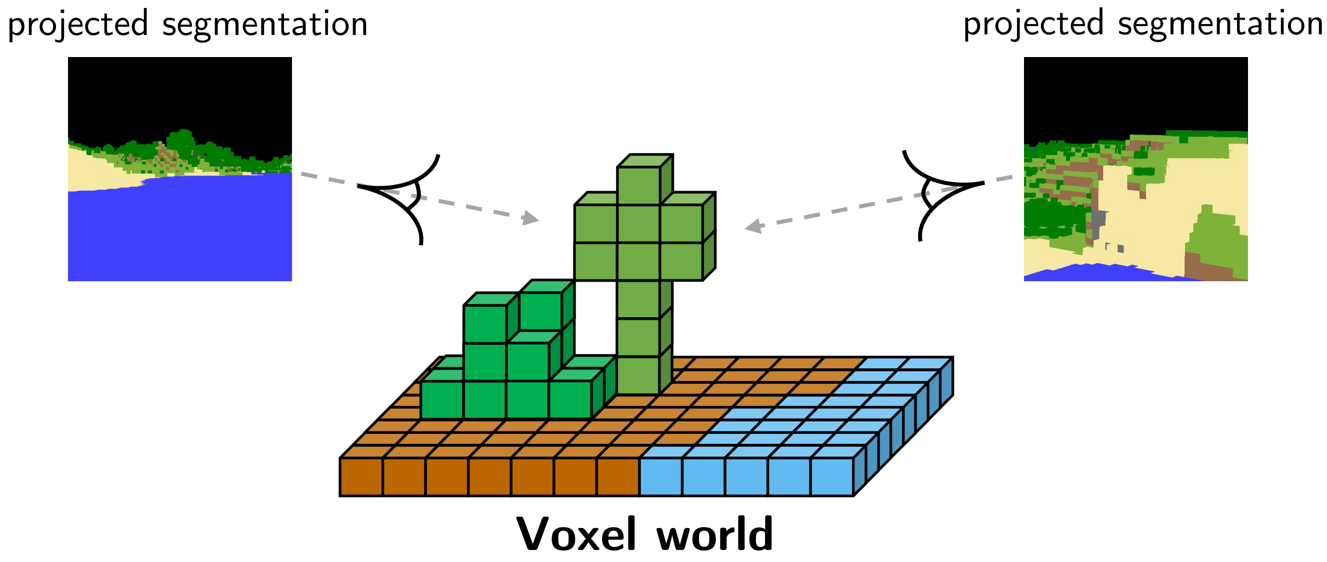
We are given a semantically-labeled voxel world as input

Pseudo-ground truth generation



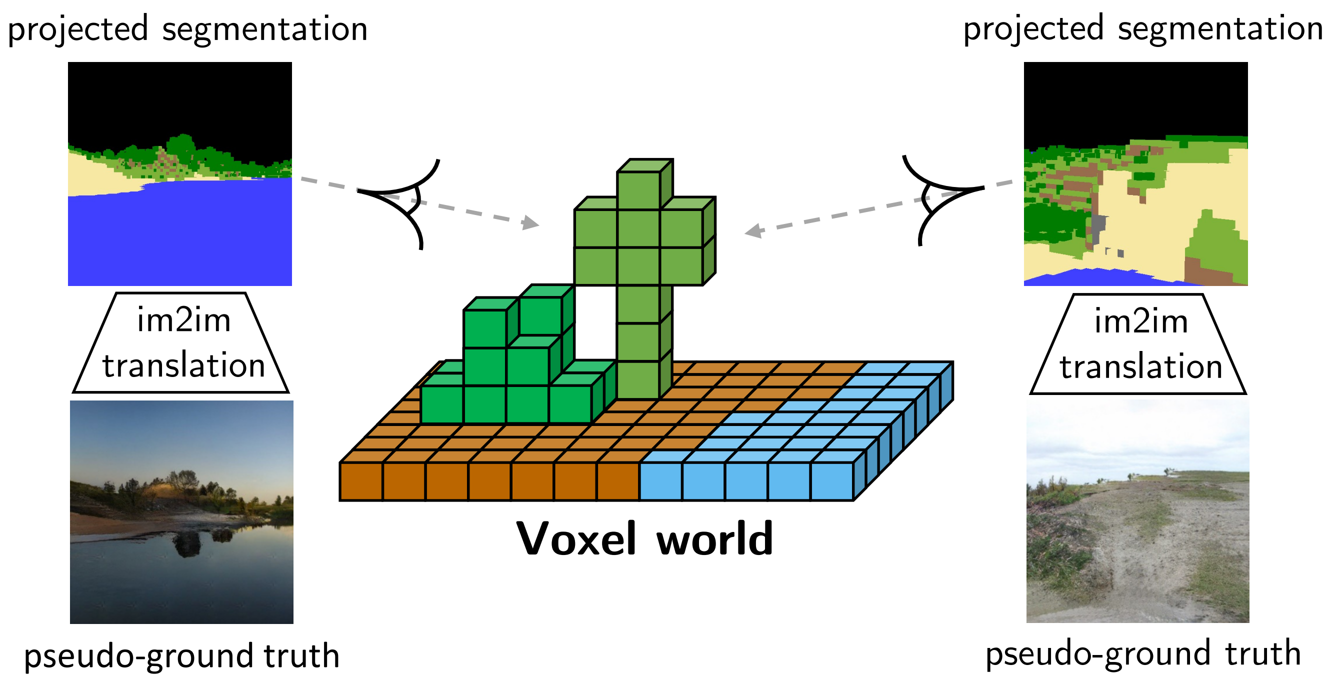
We sample random camera locations

Pseudo-ground truth generation

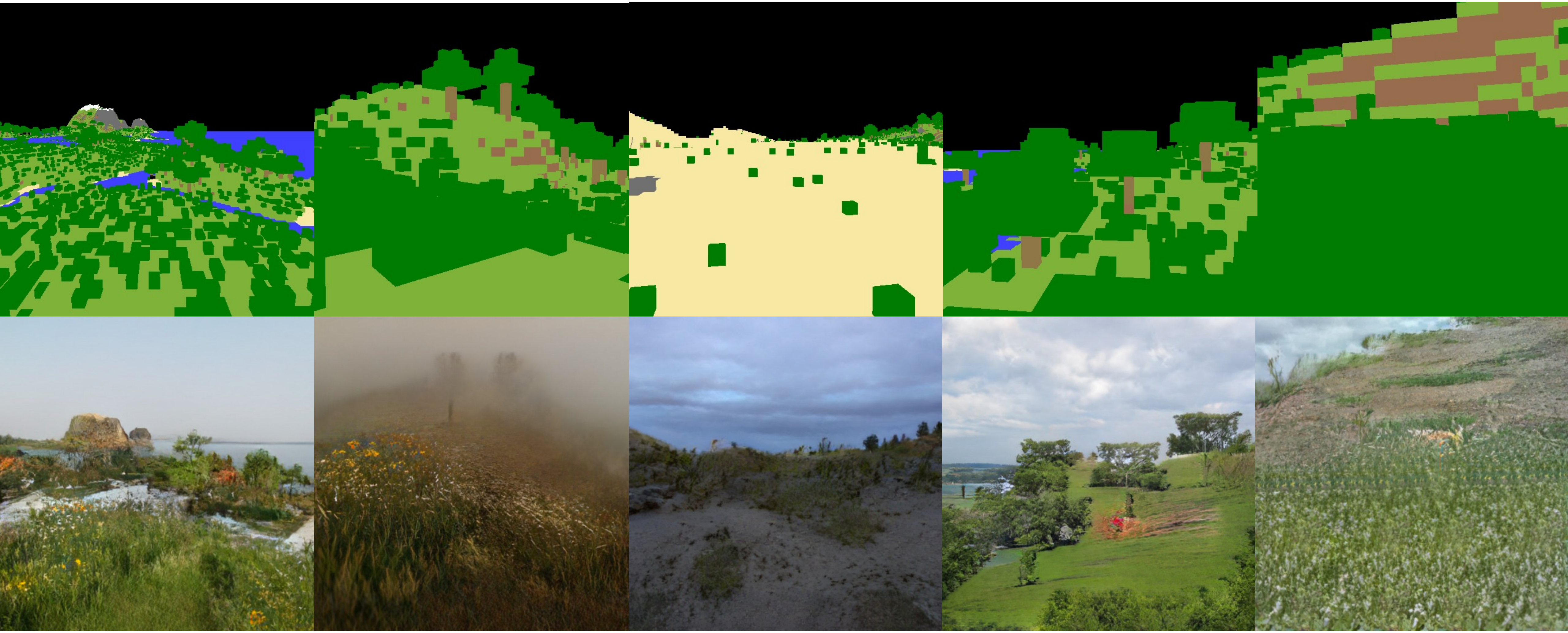


and project the voxel world to obtain segmentation maps

Pseudo-ground truth generation



These segmentation maps are fed to a pretrained image-to-image translation network to obtain pseudo-ground truths



Such pseudo-ground truths are not guaranteed to be 3D consistent, but our method is designed to be robust to such noisy training data

Why bother with Pseudo-ground truth?

- Enables us to use pixel-wise losses such as the L_1, L_2, and the VGG Perceptual loss!

- Why not simply use a GAN loss?

Pseudo-ground truth significantly improves the quality

GANcraft without using pseudo-ground truth

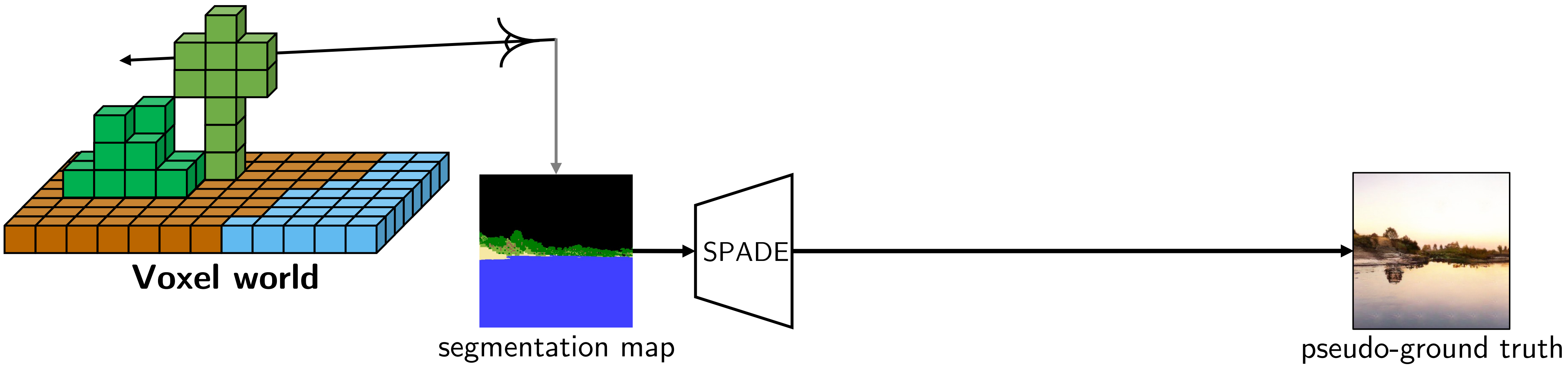
GANcraft with pseudo-ground truth

Why bother with Pseudo-ground truth?

- Enables us to use pixel-wise losses such as the L_1, L_2, and the VGG Perceptual loss!

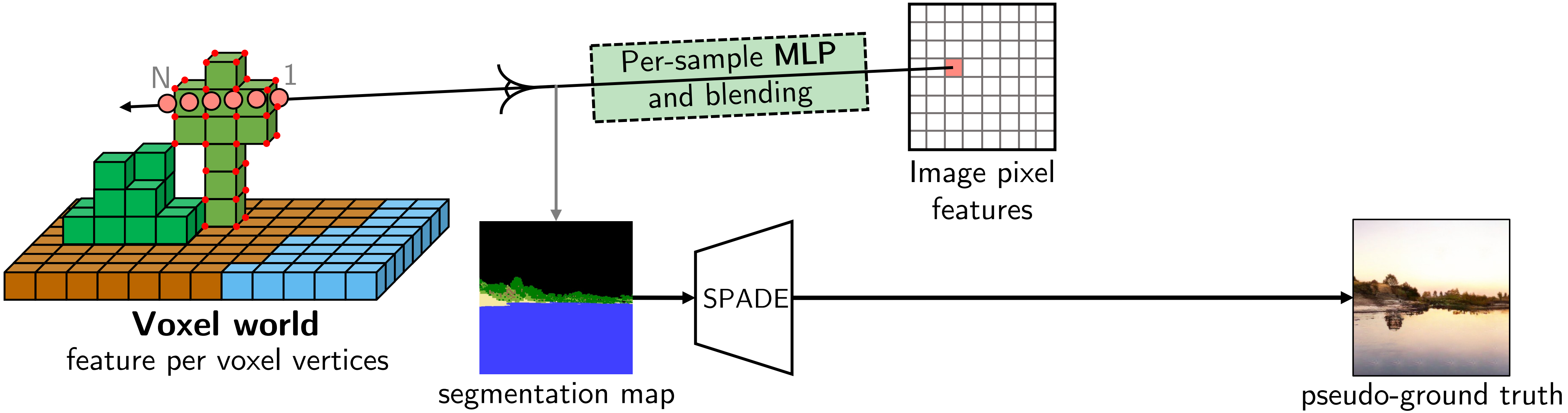
- Why not simply use a GAN loss?
- Truth hurts, but just GAN losses may not be enough for complicated tasks

Training



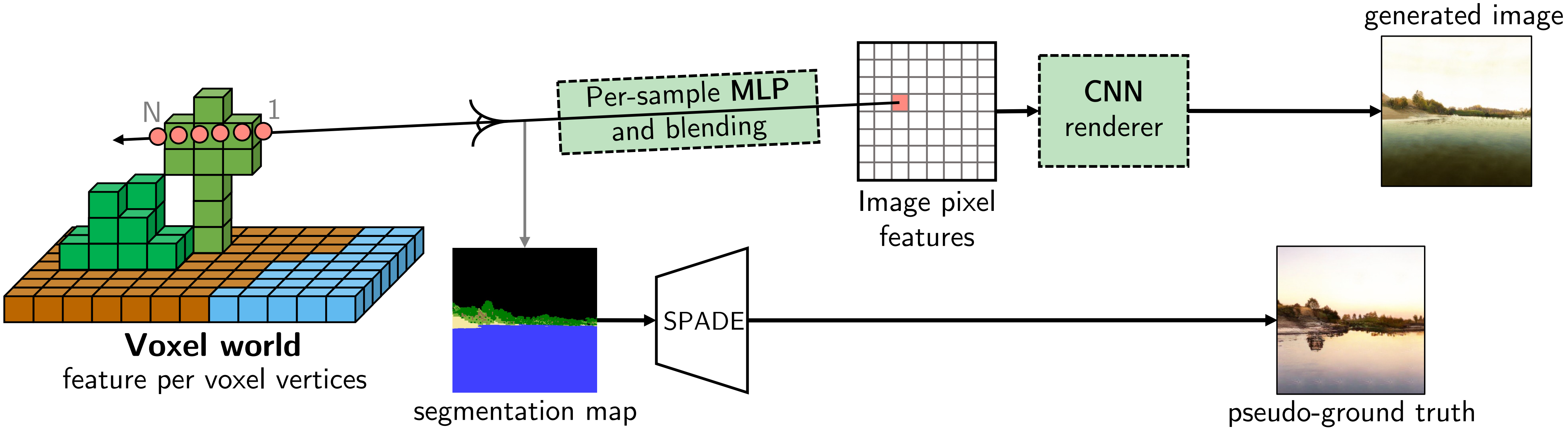
We sample a camera location,
obtain the segmentation map,
and generate the pseudo-ground truth

Training



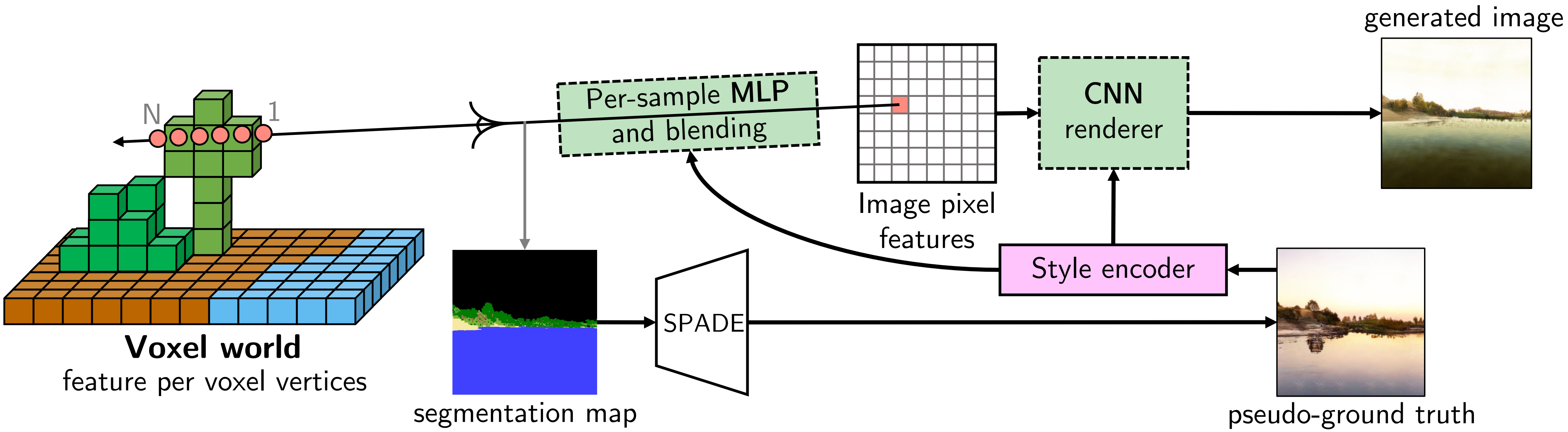
We sample N points from voxels along the ray,
trilinearly interpolate the corner features and pass them through a per-sample MLP,
and blend them to obtain image pixel features

Training



We pass the image pixel features to a CNN
and generate an output image

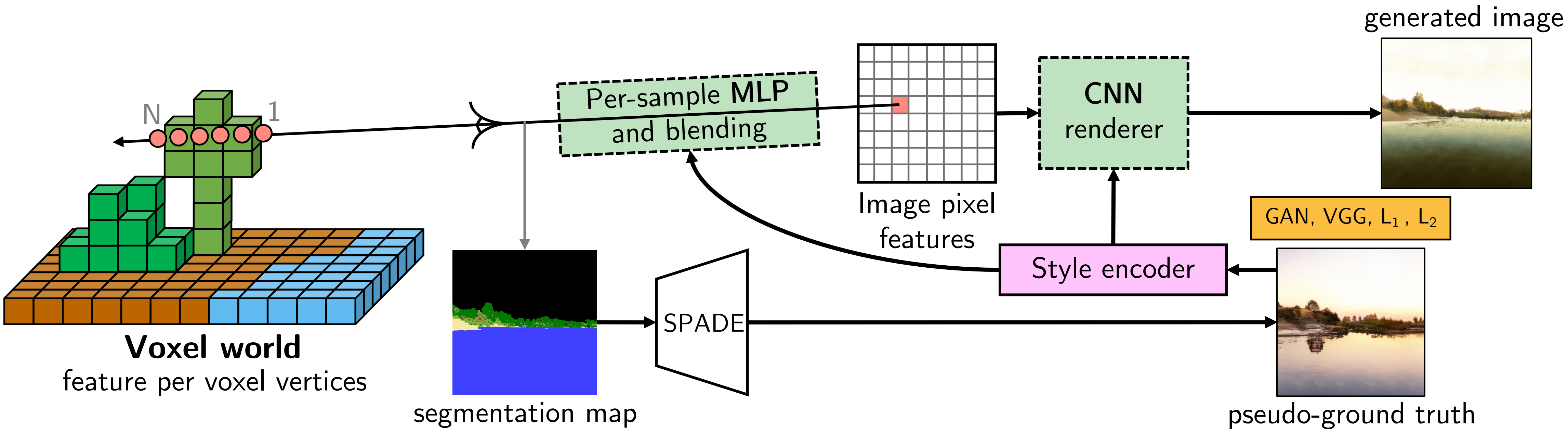
Training



Both the MLP and the CNN are conditioned on the style of the pseudo-ground truth image

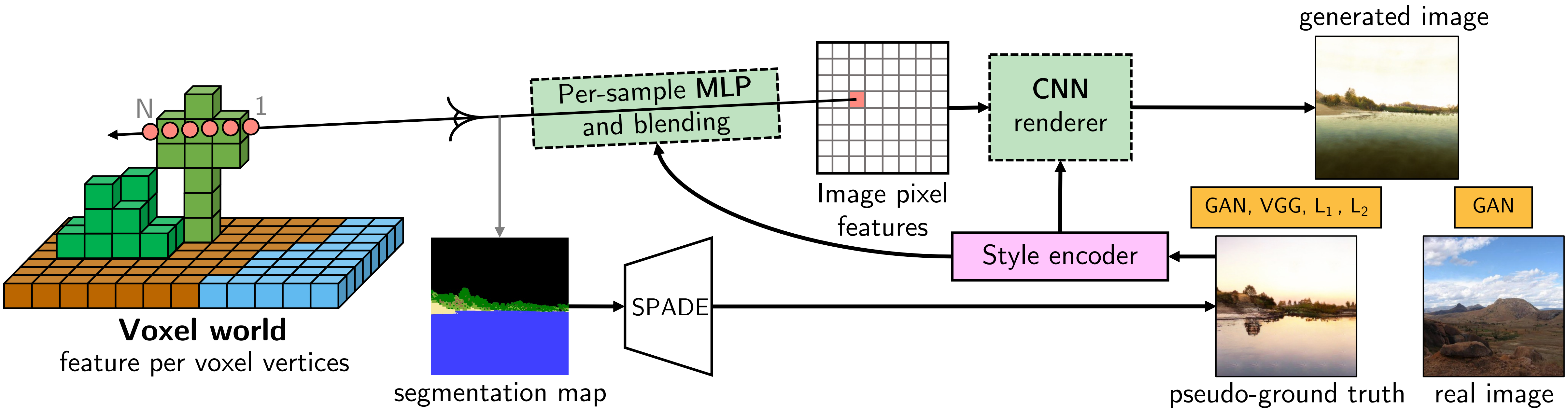
The style encoder explains away the view-inconsistency of pseudo-ground truth image

Losses



We apply a GAN loss, VGG-19 perceptual loss, and pixel-wise losses between the output and pseudo-ground truth

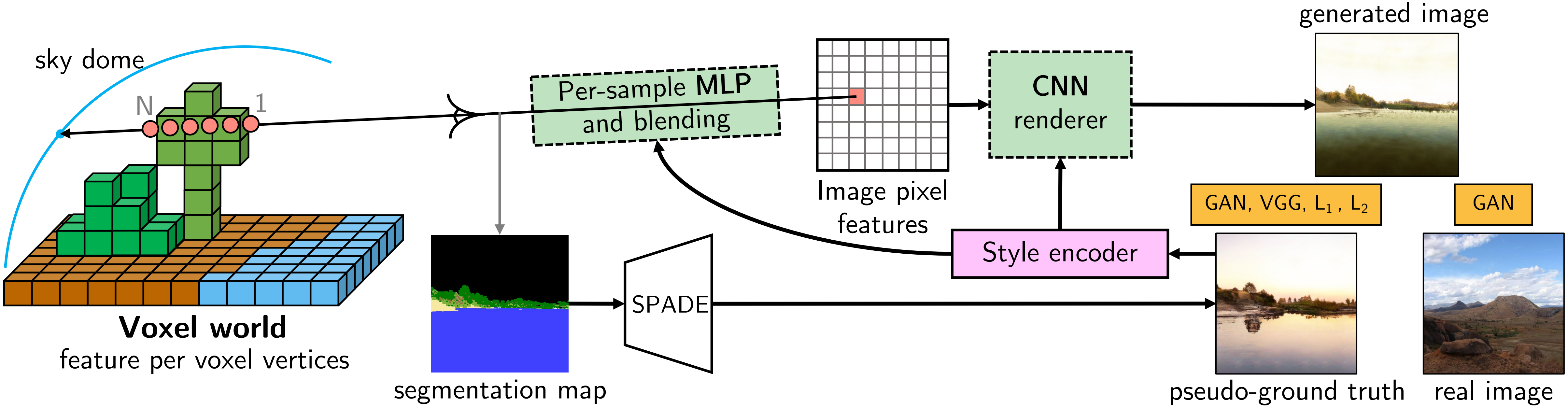
Losses



We also apply a GAN loss between the output and real images to improve realism

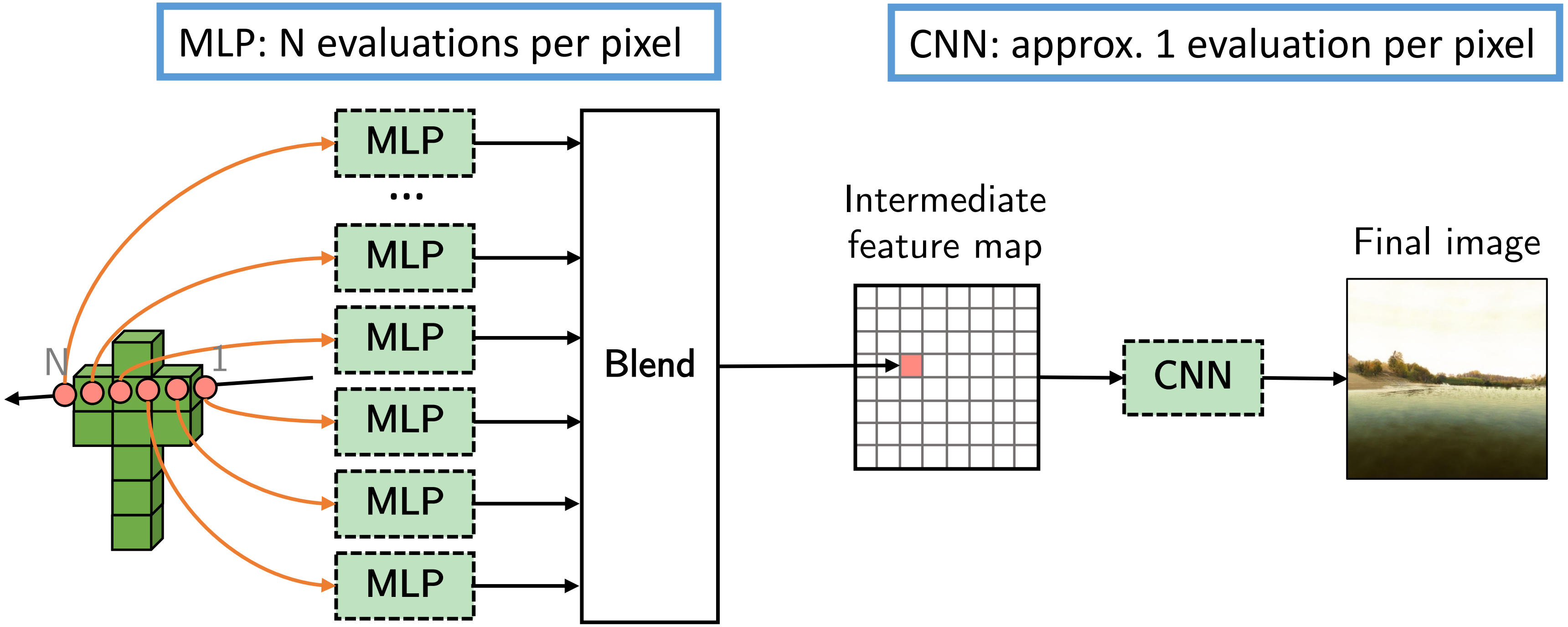
Additional Details

Neural sky dome



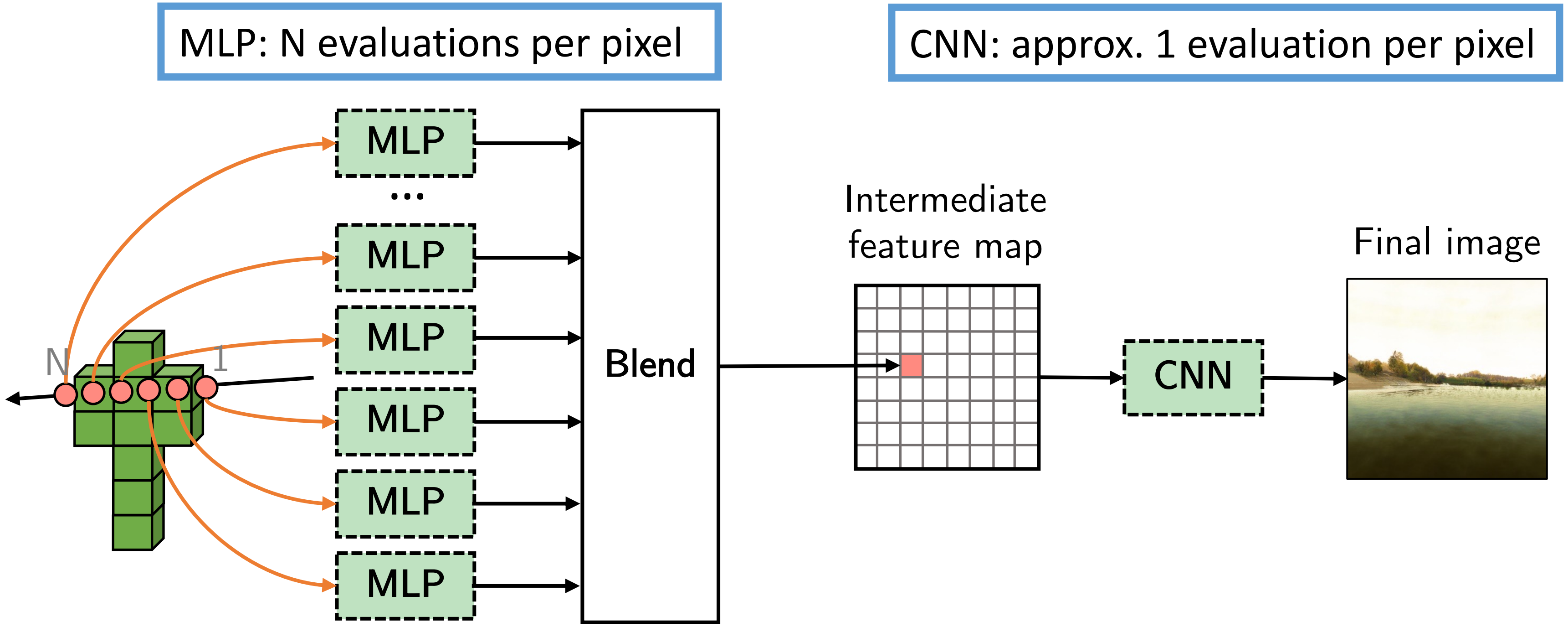
A neural sky dome located at infinitely far away catches the residual transmittance.

Two-stage renderer improves scalability



GANcraft only uses 24 samples per ray in the volumetric rendering stage – noisy feature map

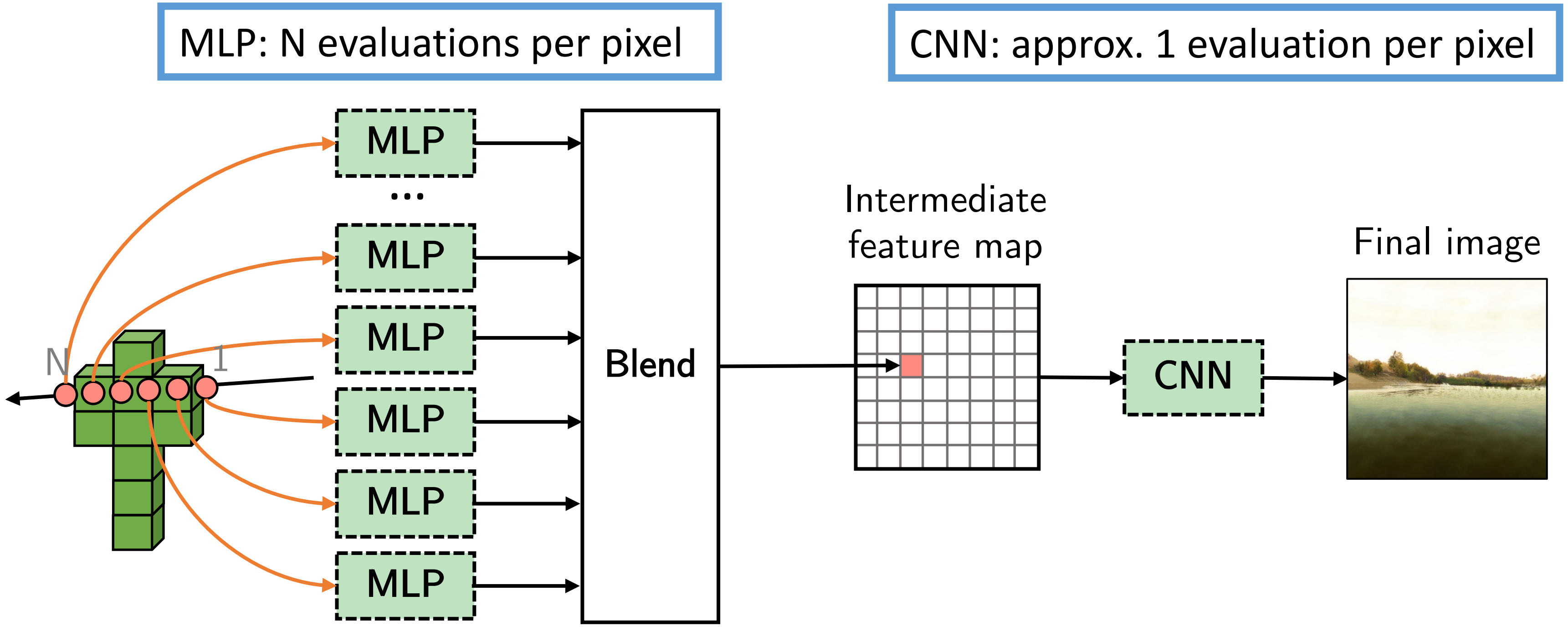
Two-stage renderer improves scalability



GANcraft only uses 24 samples per ray in the volumetric rendering stage – noisy feature map
WHY?? NeRF uses over a 100!

- NeRF applies L_2 loss per pixel only
- We need to produce the whole image (not just a subset) to apply perceptual and GAN losses

Two-stage renderer improves scalability



GANcraft only uses 24 samples per ray in the volumetric rendering stage – noisy feature map
The CNN aggregates information within local patches and removes noise
The CNN is more flops-efficient than the radiance field MLP due to fewer number of evaluations
Need small CNN to preserve view-consistency!

Two-stage renderer improves quality

One-stage (MLP only)

Two-stage (MLP + CNN)

Two-stage rendering pipeline produces images with better detail under the same computation and memory budget

Please refer to the main paper for further details and quantitative results

Website: https://nvlabs.github.io/GANcraft/

Code available at https://github.com/nvlabs/imaginaire/

NVIDIA®

2021 ICCV OCTOBER 11-17 VIRTUAL

GANCRAFT

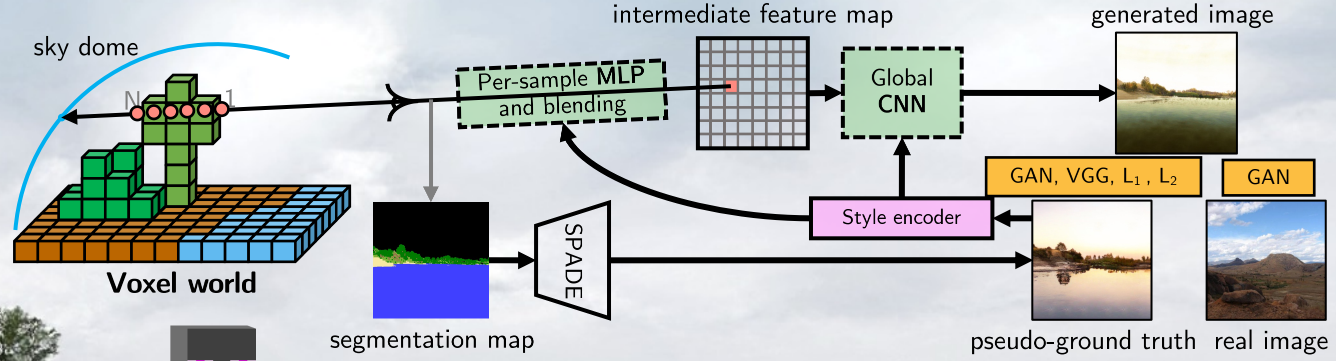
Unsupervised 3D Neural Rendering of Minecraft Worlds

Zekun Hao

Arun Mallya

Serge Belongie

Ming-Yu Liu



Everyone can be a *3D painter*!

nvlabs.github.io/GANcraft

NVIDIA

Cornell Bowers CIS Computer Science

CORNELL TECH