CS380

Introduction to Diffusion Models

Jumin Lee

Advisor : Sung-Eui Yoon

SGVR Lab
KAIST

Diffusion Model

2020.06
DDPM

2022.04
DALLE2

2022.05
Imagen

2022.06
Stable Diffusion

Diffusion Model for Conditional Generation

2020.06
DDPM

2022.04
DALLE2

2022.05
Imagen

2022.06
Stable Diffusion

- Conditional Generation
  - **Inpainting**
  - Outpainting
  - Image to Image Generation
  - Text to Image Generation

Diffusion Model for Conditional Generation

2020.06
DDPM

2022.04
DALLE2

2022.05
Imagen

2022.06
Stable Diffusion

- Conditional Generation
  - Inpainting
  - **Outpainting**
  - Image to Image Generation
  - Text to Image Generation

Diffusion Model for Conditional Generation

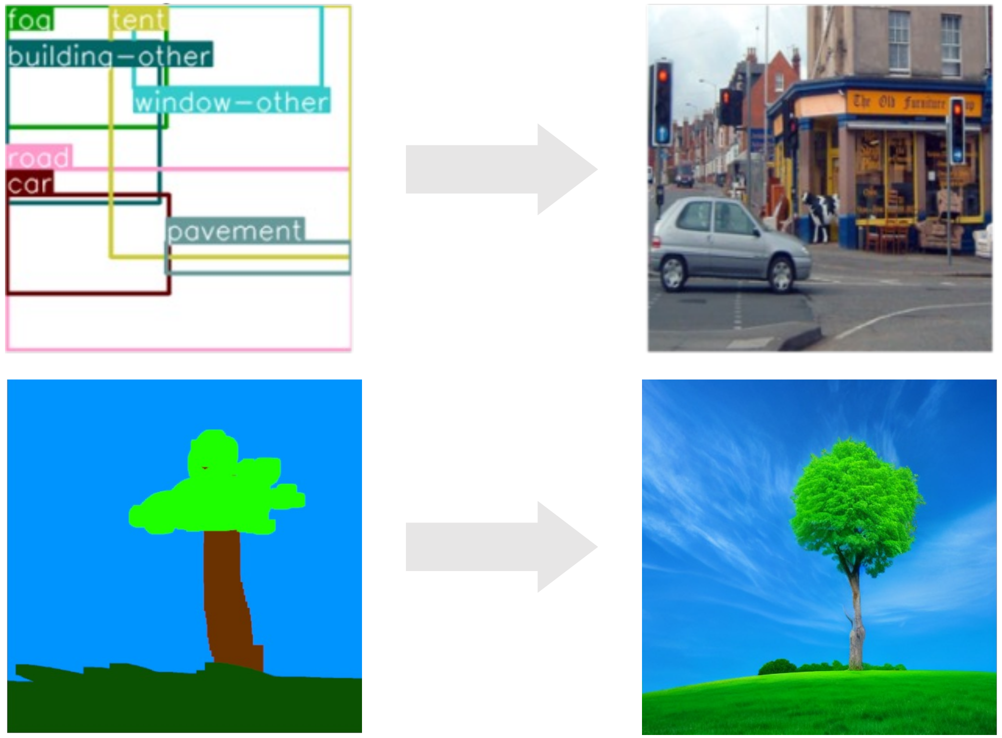
2020.06 DDPM

2022.04 DALLE2

2022.05 Imagen

2022.06 Stable Diffusion

- Conditional Generation
  - Inpainting
  - Outpainting
  - **Image to Image Generation**
  - Text to Image Generation



Diffusion Model for Conditional Generation

2020.06	2022.04	2022.05	2022.06
DDPM	DALLE2	Imagen	Stable Diffusion

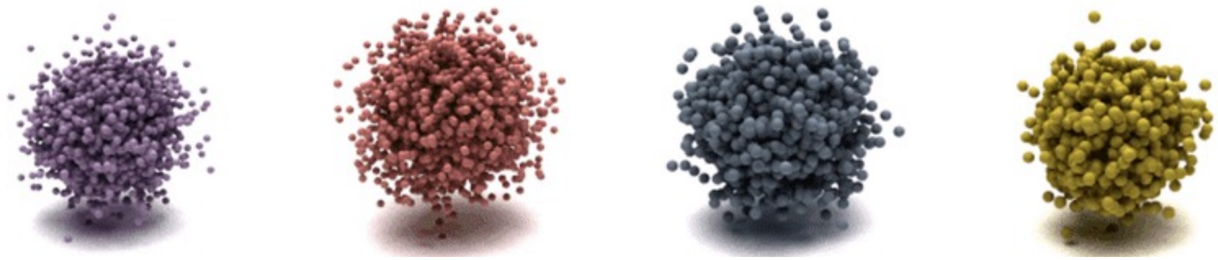
- Conditional Generation
  - Inpainting
  - Outpainting
  - Image to Image Generation
  - **Text to Image Generation**

A street sign that reads “Latent Diffusion” | A zoombie in the style of Picasso | An image of an animal half mouse half octopus

Diffusion Model

2021~
3D Diffusion

- A 3D diffusion process can be used to generate an object from point clouds, meshes, or latent spaces.



2021 PVD

2021
Text2Mesh

2023
Dreamfusion

2023
Magic3D

2023
ProlificDreamer

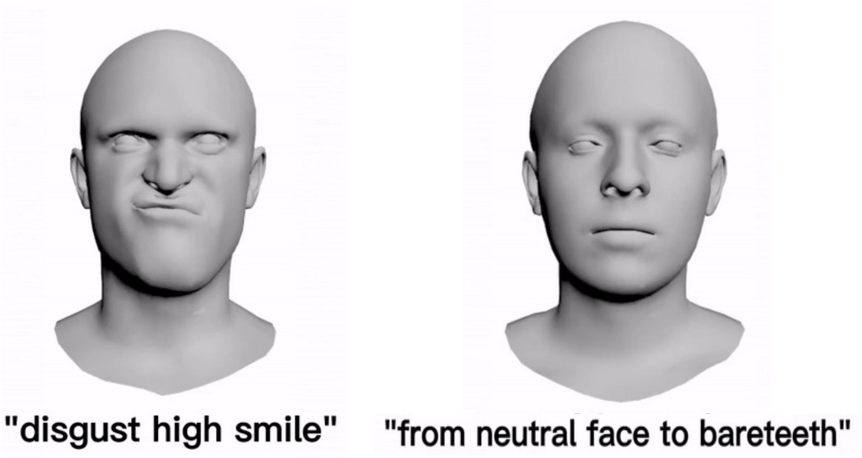
2023
MVdream

Diffusion Model

2021~
3D Diffusion

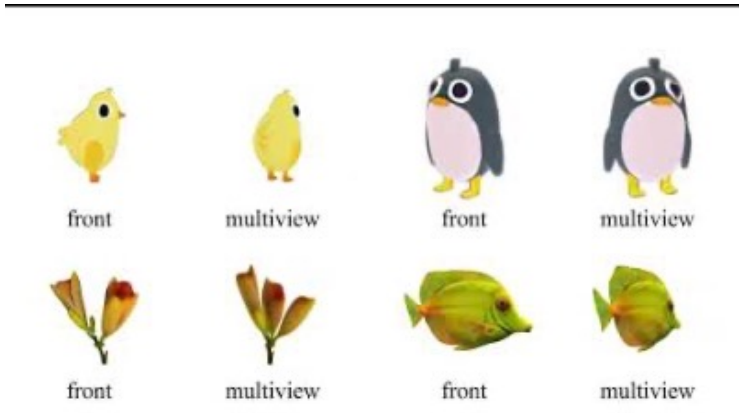
2023~
4D Diffusion

- Extend the diffusion process domain to 4D, including space and time.



2023
4D Facial Expression

2023
Align Your Gaussian

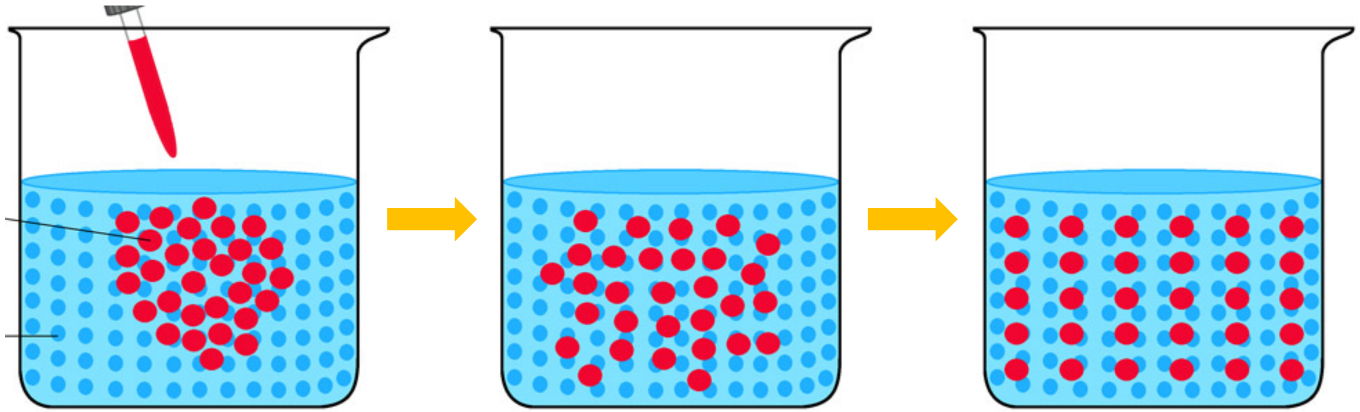


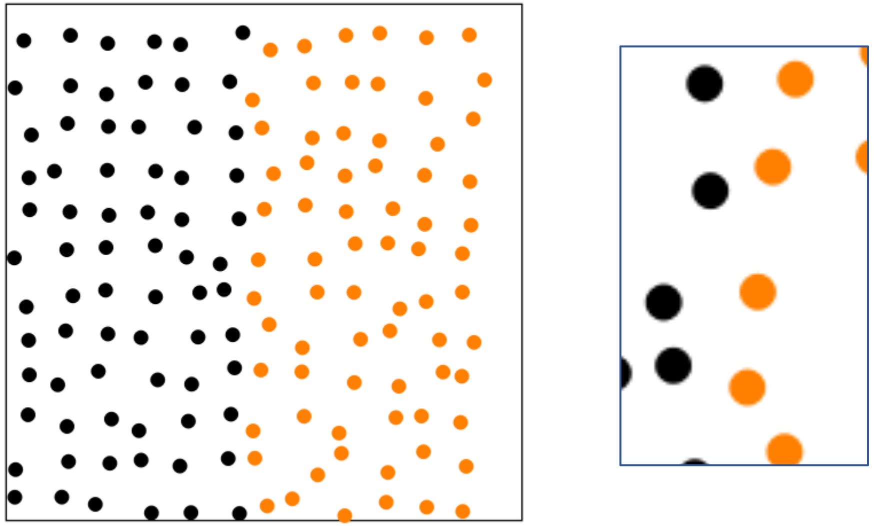
2023
4DGen

Background

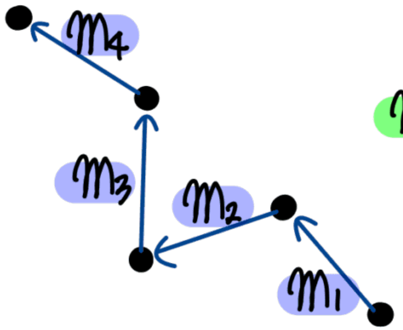
Diffusion Process

- Diffusion models are inspired by non-equilibrium thermodynamics.
- For a small fraction of the time, it is difficult to determine whether particles are moving in the direction of mixing or in the opposite direction.



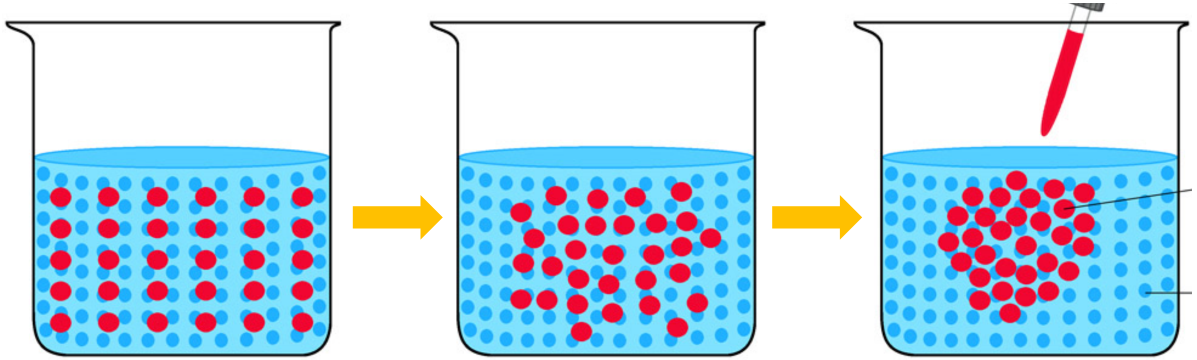


Diffusion Process



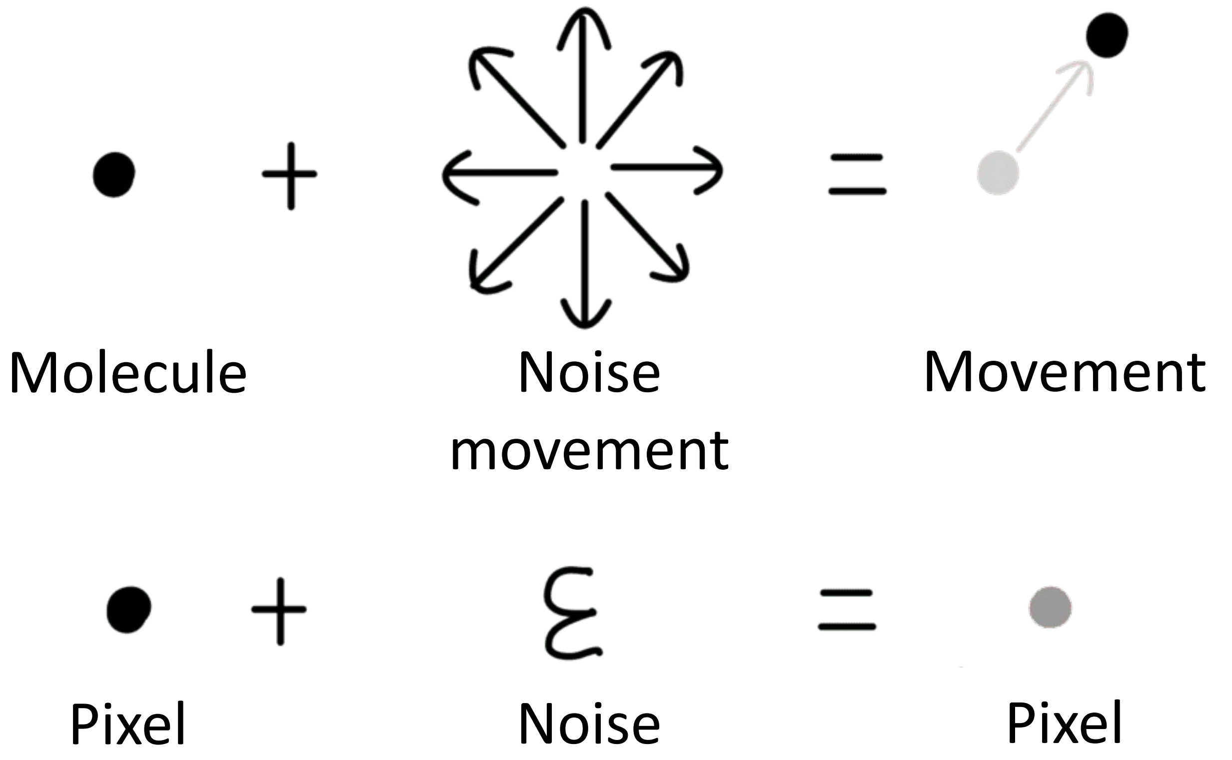
$m_t \sim N(\mu, \sigma)$

- If we look at the movement of a single molecule on a very short time scale, it follows a Gaussian distribution.
- Since the direction of mixing and the opposite direction are the same in a very short time, the opposite direction also follows a Gaussian distribution.



Diffusion Process

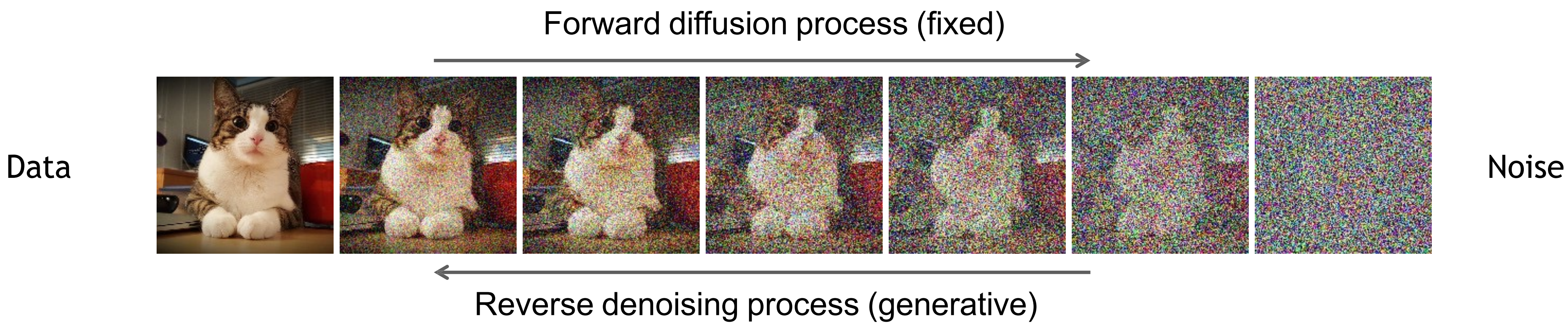
- Just as we viewed the molecule's motion as a Gaussian-distributed noise, we add a Gaussian-distributed noise to the pixel.



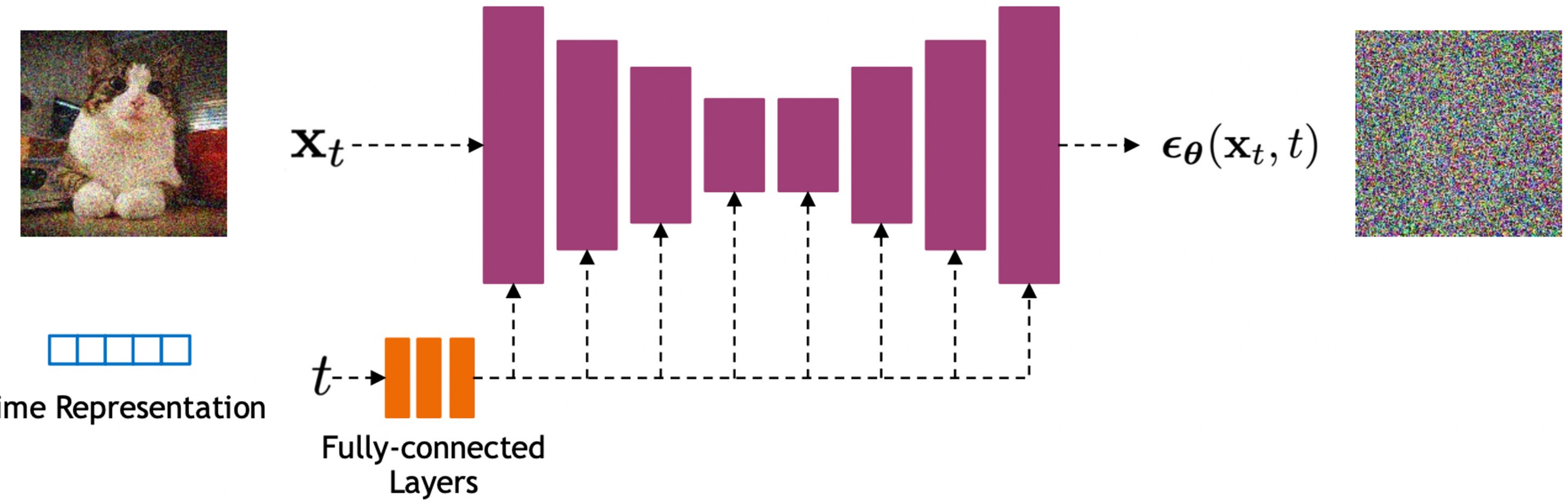
Denoising Diffusion Models

Denoising diffusion models consist of two processes:

- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



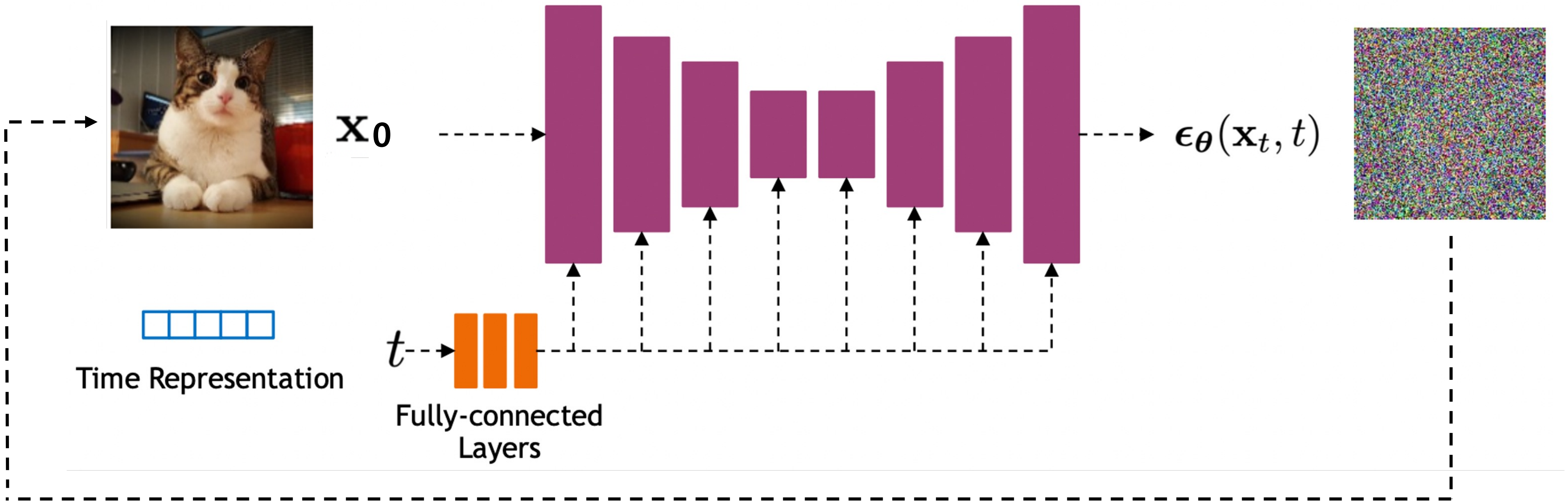
Denoising Diffusion Models : Training



Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
  $$\nabla_\theta \left\| \boldsymbol{\epsilon} - \mathbf{z}_\theta(\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}, t) \right\|^2$$
until converged

Denoising Diffusion Models : Sampling

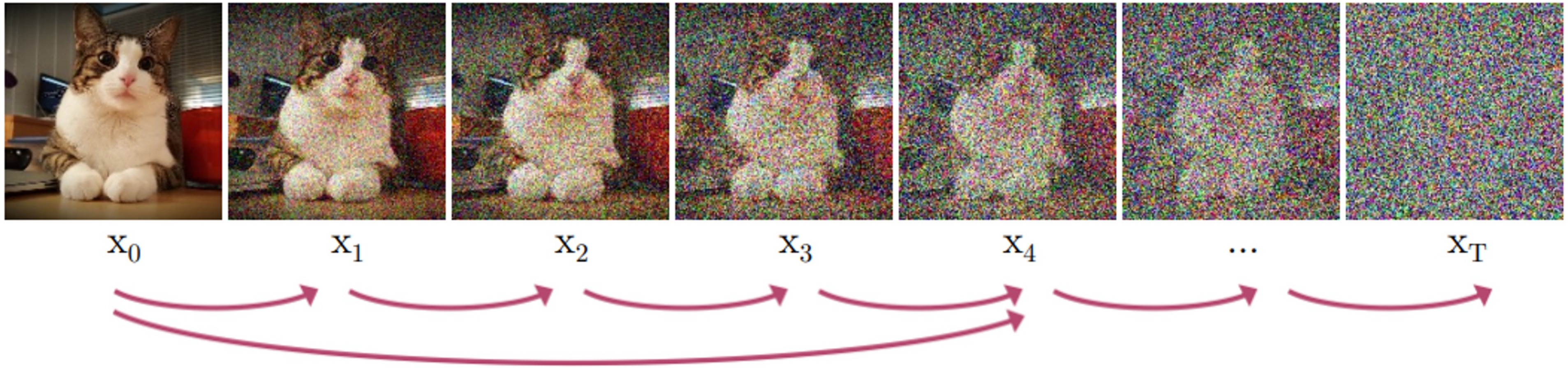


Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \mathbf{z}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Forward Diffusion Process

The formal definition of the forward process in T steps:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Markov Property

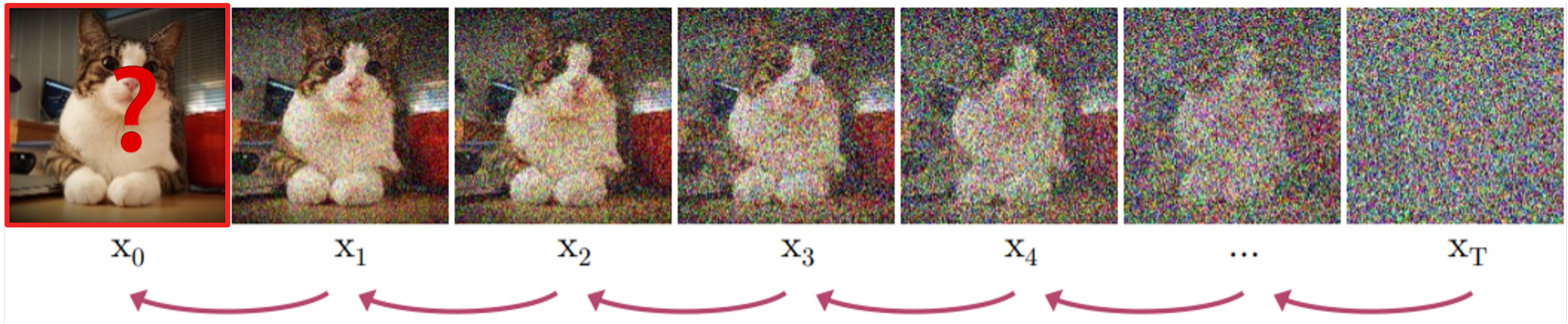
$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$ ← **Diffusion Kernel**

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\alpha_t := 1 - \beta_t \text{ and } \bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$$

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:

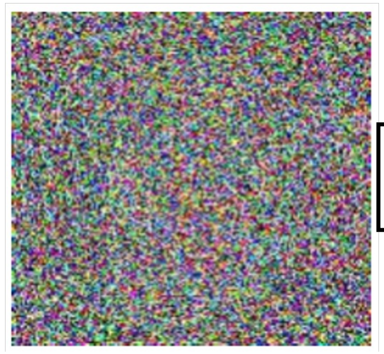


$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

Model

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \boxed{\mu_\theta(x_t, t)}, \Sigma_\theta(x_t, t))$$

Results



Diffusion

Diffusion Model

- Pros
  - Intuitive Understanding: Diffusion in pixel space directly affects image pixels, making the changes visually easy to understand.

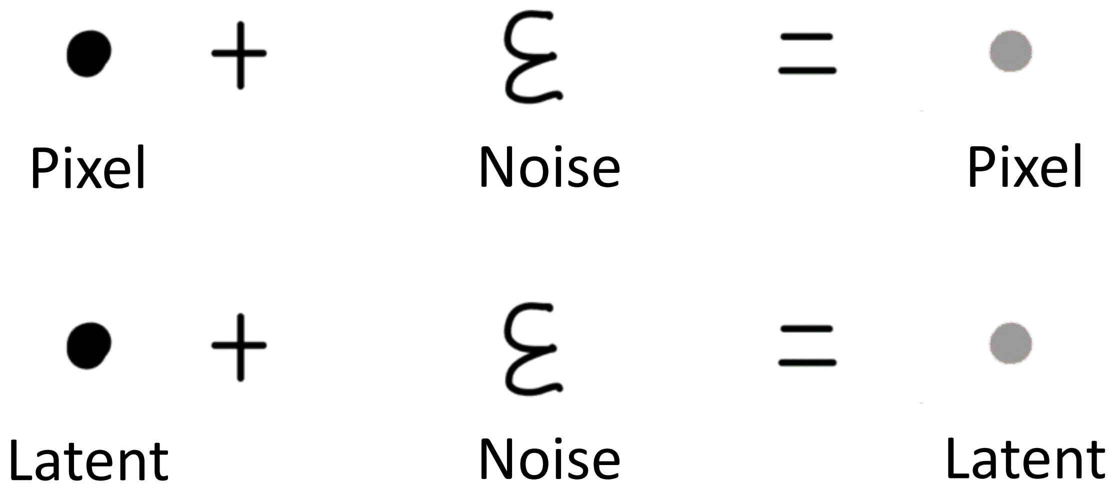
- Cons
  - Computational Cost

  : The larger the number of pixels, the greater the computation.
  - Memory Usage

  : Handling high-resolution images requires substantial memory.

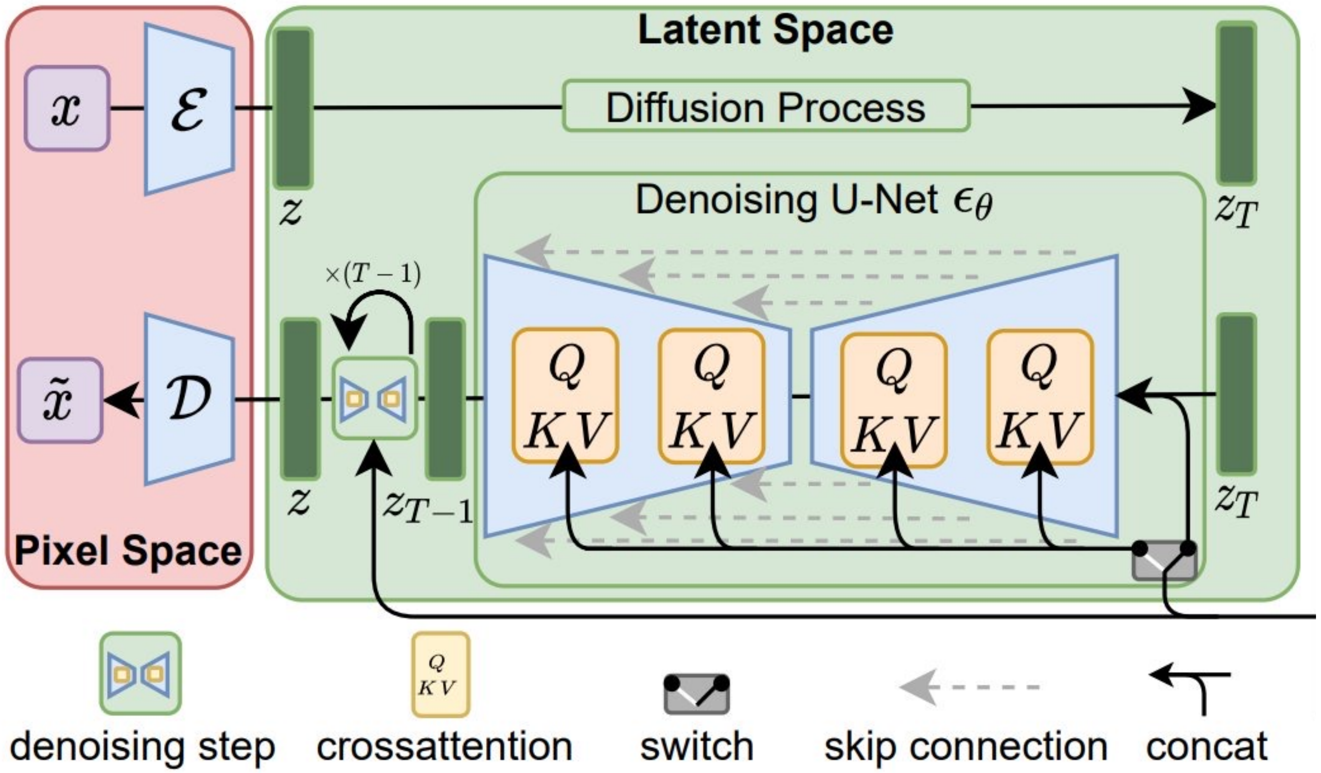
Latent Diffusion Model

- Latent spaces typically have lower dimensions than pixel spaces, resulting in lower computational costs.
  - Pixel Space >> Latent Space



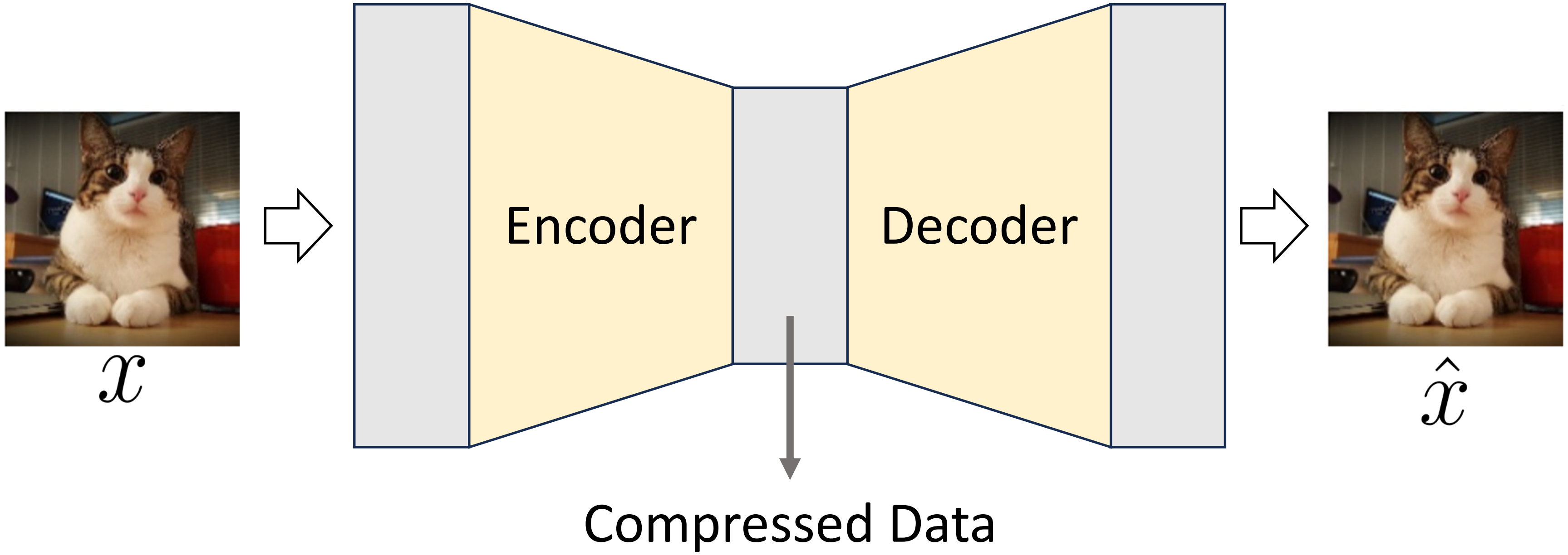
Latent Diffusion Model

- Runs the diffusion process in the latent space instead of pixel space
- 2 Stage Training : Auto-Encoder + Latent Diffusion



Latent Diffusion Model

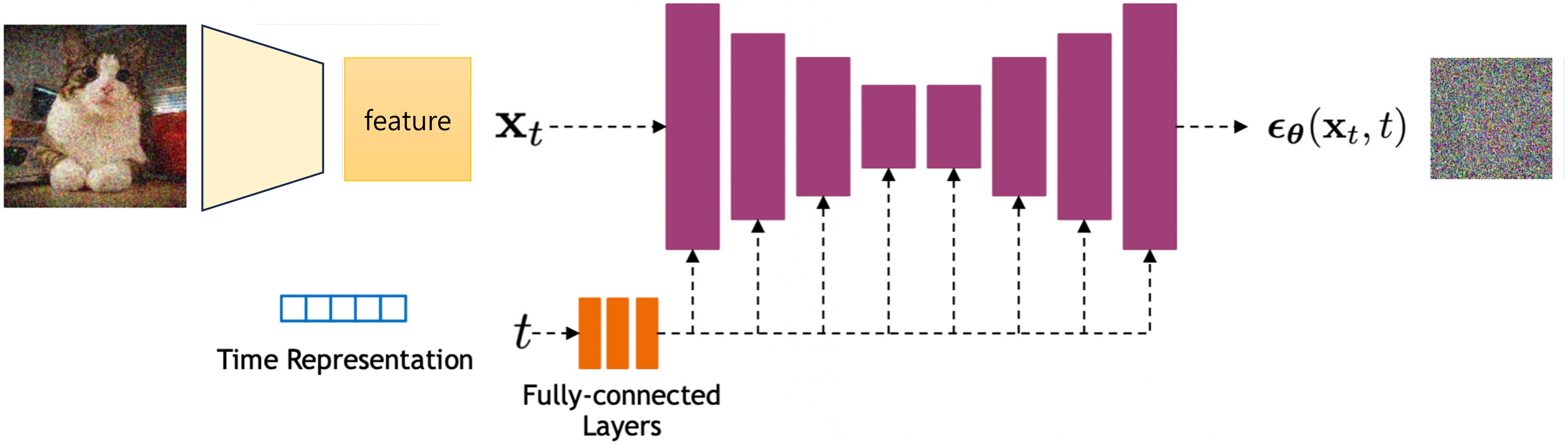
- Autoencoders can be particularly valuable as they enable a compressed yet remaining semantic and conceptual meaning of an image.



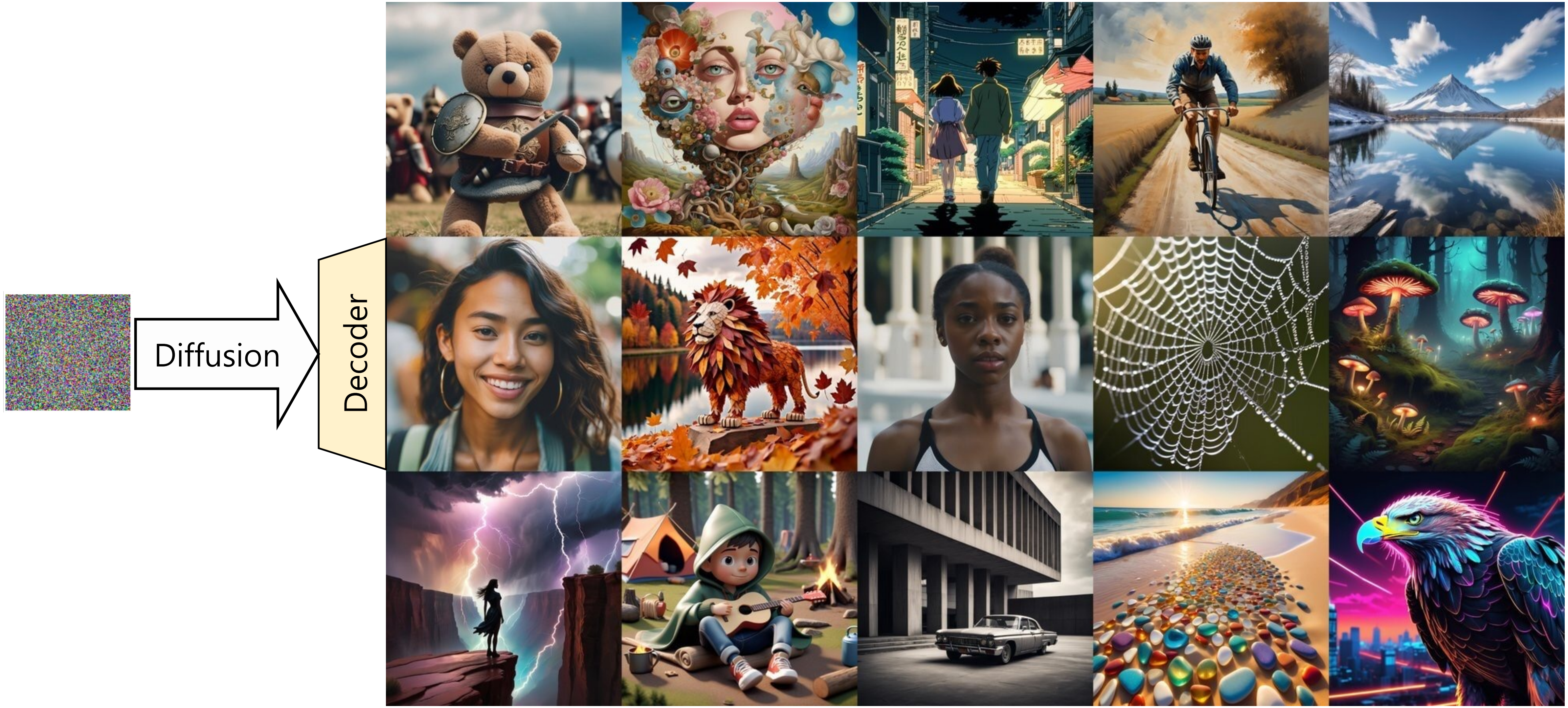
$$loss = \frac{1}{n}\sum_{i=0}^{n}(x_i - \hat{x}_i)^2$$

Latent Diffusion Model

- Runs the diffusion process in the latent space instead of pixel space
- 2 Stage Training : Auto-Encoder + Latent Diffusion

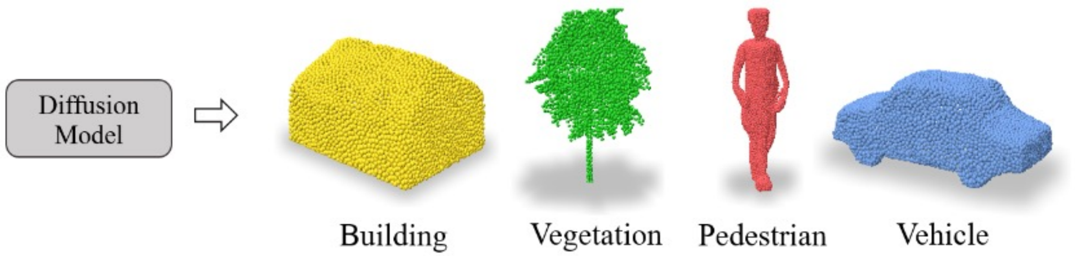


Results

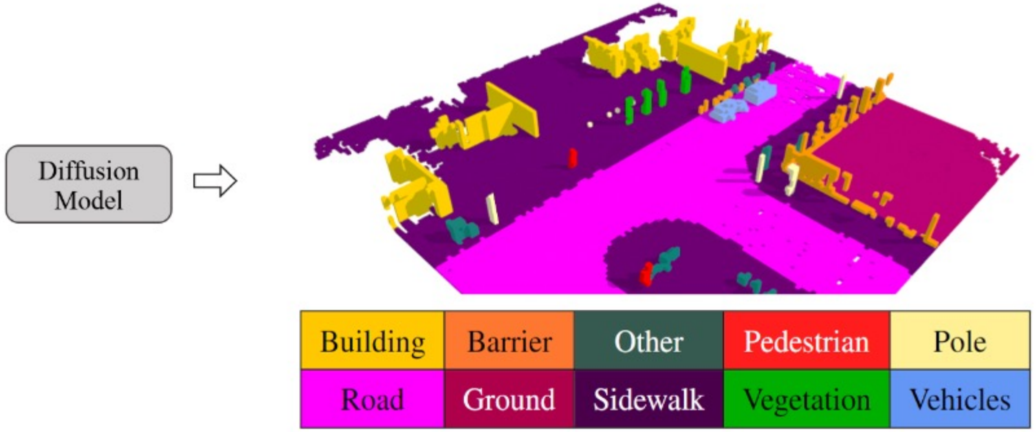


Our Goal

Our Goal

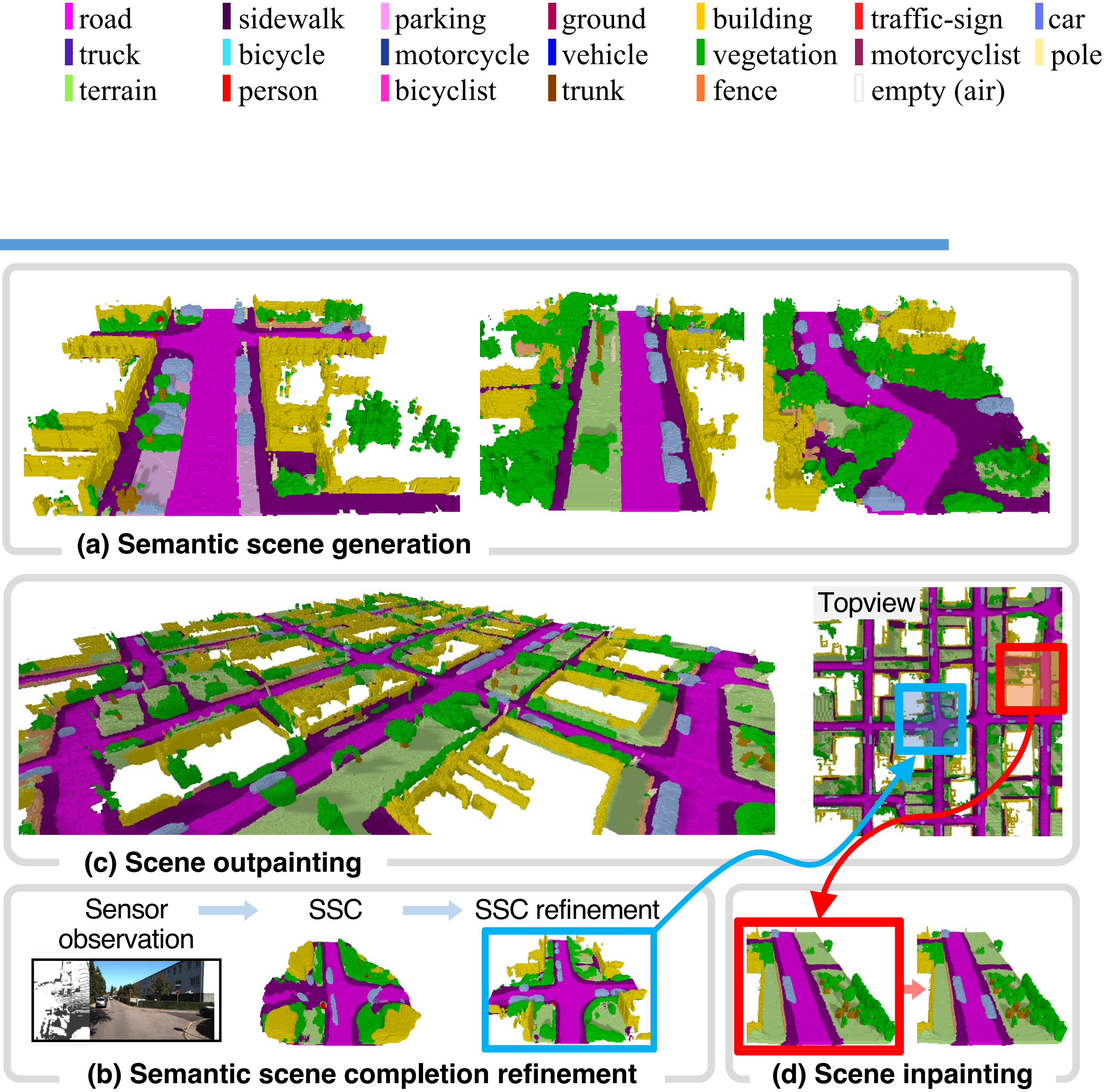


(a) Object-scale generation



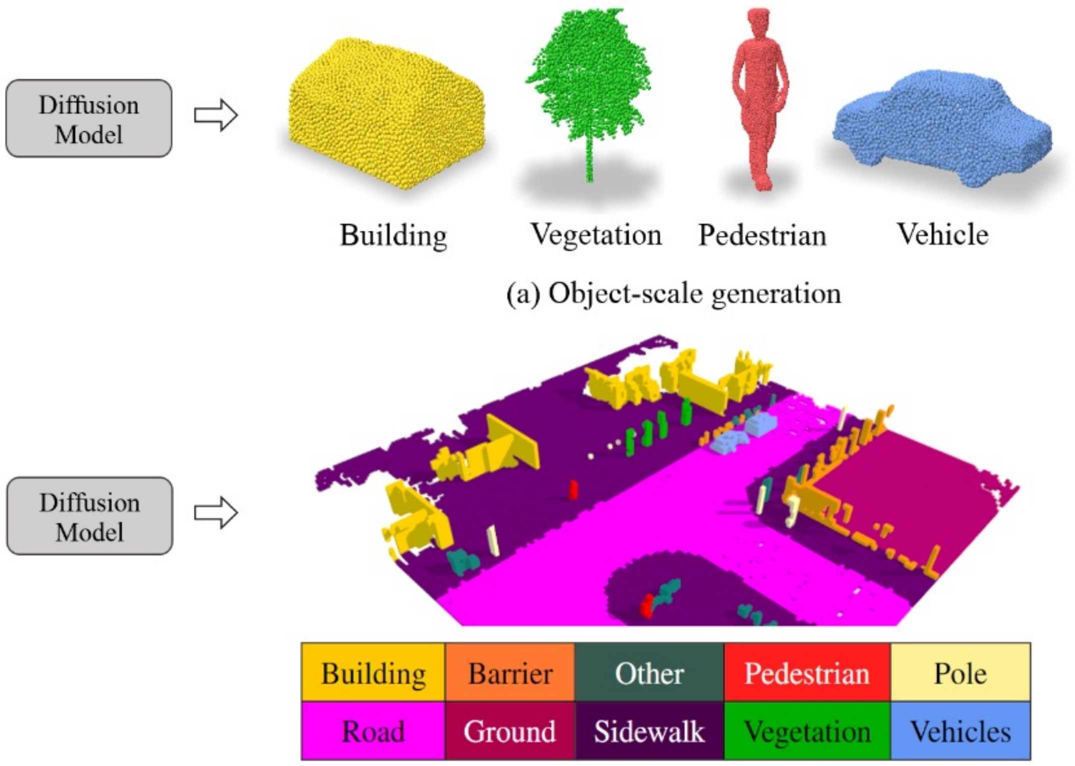
(b) Scene-scale generation (Ours)

Jumin Lee, Woobin Im, Sebin Lee, Sung-Eui Yoon, *Diffusion Probabilistic Models for Scene-Scale 3D Categorical Data,* IPIU 2023 (grand prize)



Jumin Lee*, Sebin Lee*, Changho Jo, Woobin Im, Ju-Hyeong Seon, Sung-Eui Yoon, *SemCity: Semantic Scene Generation with Triplane Diffusion*, CVPR 2024

3D Scene-level Generation

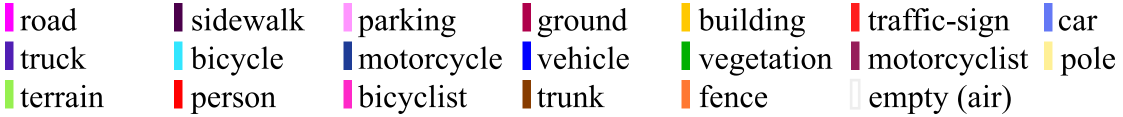


(a) Object-scale generation

(b) Scene-scale generation (Ours)

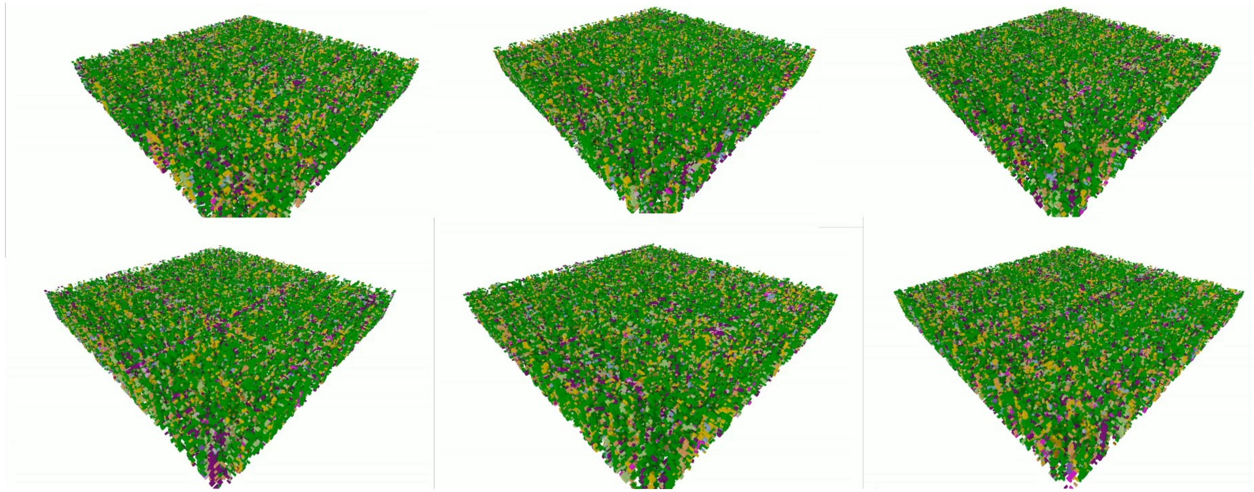
Jumin Lee, Woobin Im, Sebin Lee, Sung-Eui Yoon, *Diffusion Probabilistic Models for Scene-Scale 3D Categorical Data,* IPIU 2023 (grand prize)

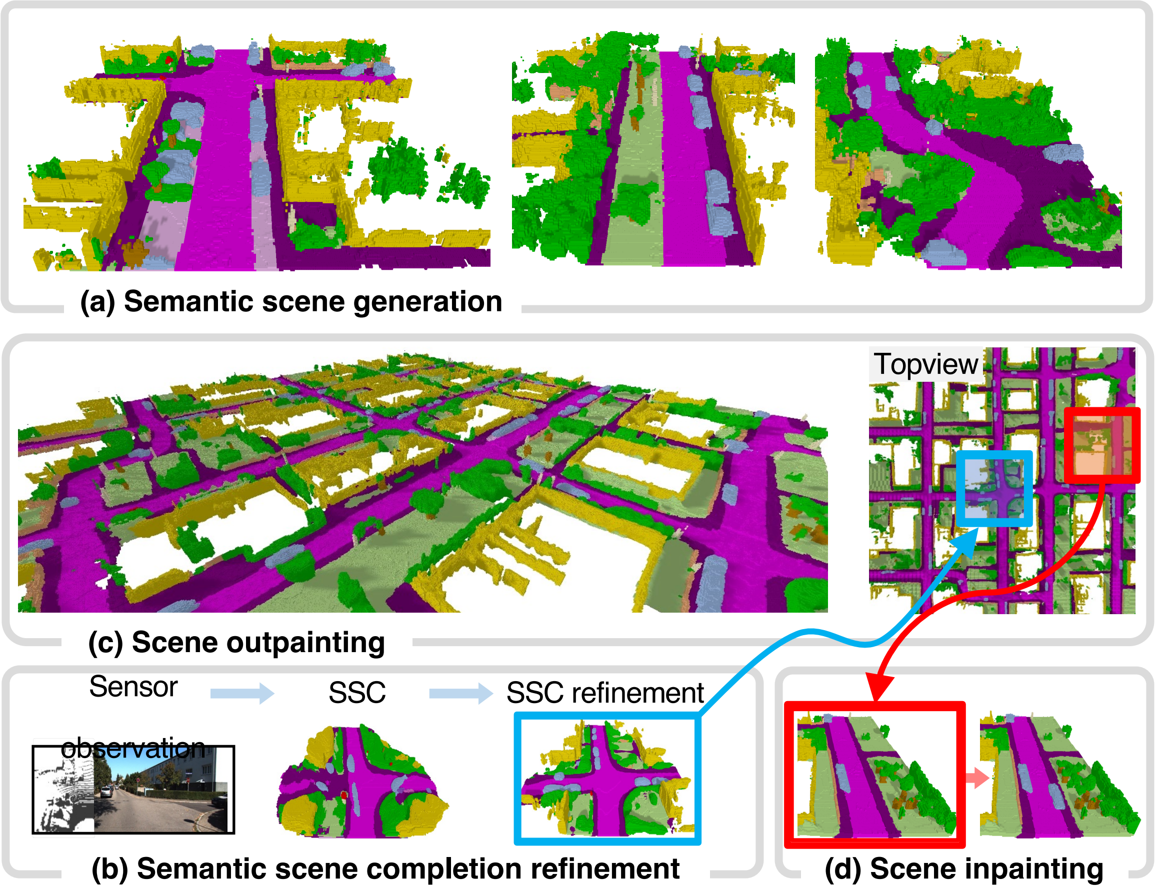
- Firstly apply the diffusion model at the 3D scene level not at the 3D object level.
- Show meaningful results.



3D Scene-level Generation

- Enhance generation power.
- Extend our model with several applications (inpainting, outpainting, semantic scene completion refinement), as in the image domain.





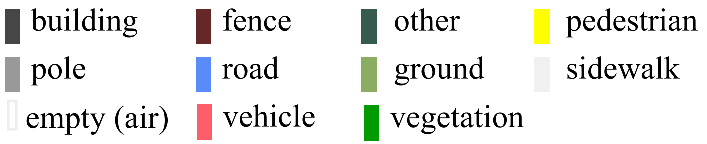
Jumin Lee*, Sebin Lee*, Changho Jo, Woobin Im, Ju-Hyeong Seon, Sung-Eui Yoon, *SemCity: Semantic Scene Generation with Triplane Diffusion*, CVPR 2024

CS380

SSD: Diffusion Probabilistic Models for Scene-Scale 3D Categorical Data

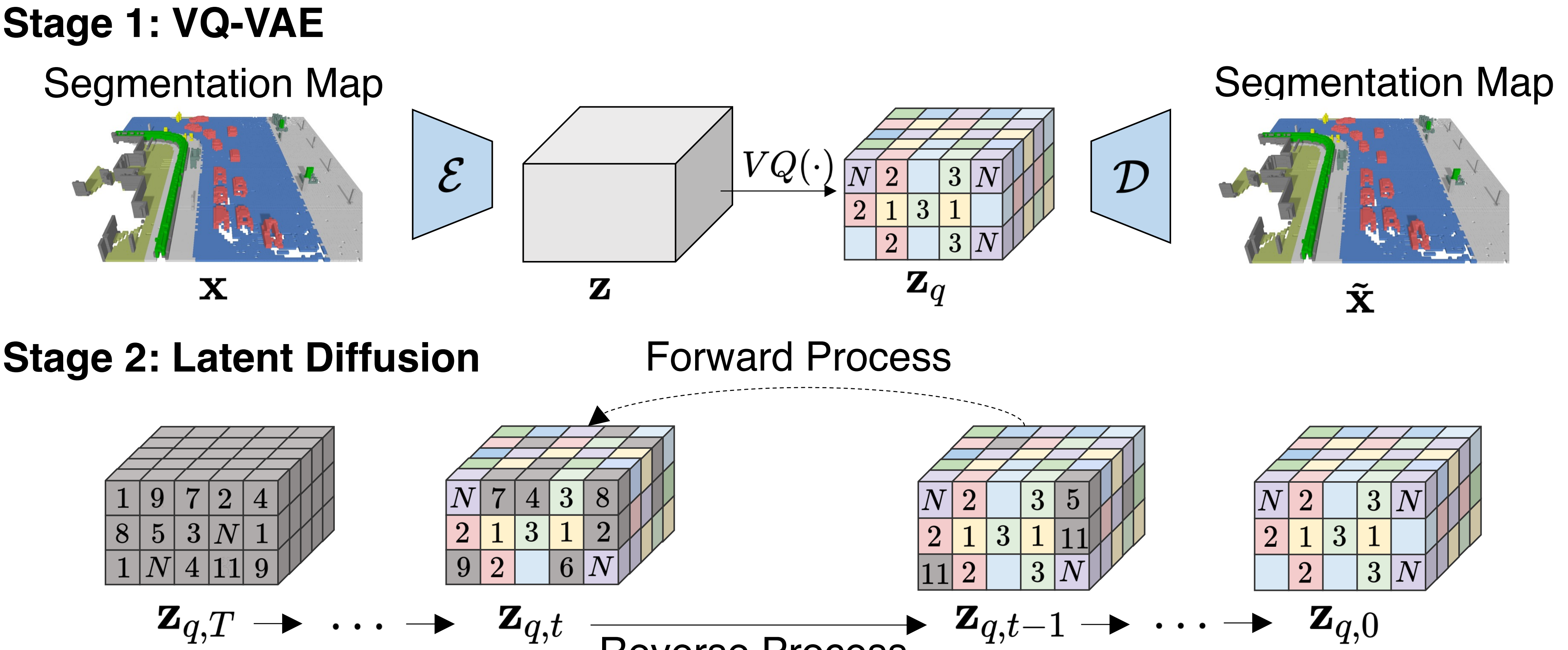
Jumin Lee, Woobin Im, Sebin Lee, Sung-Eui Yoon, *Diffusion Probabilistic Models for Scene-Scale 3D Categorical Data, IPIU 2023*

SGVR Lab KAIST



Method

- Diffusion process on 3D latent space.

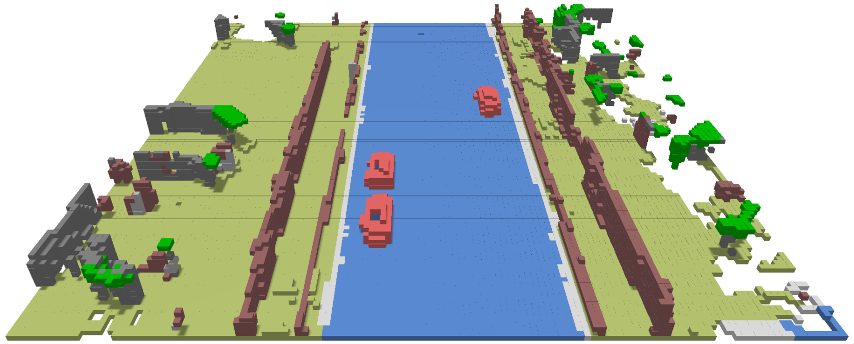


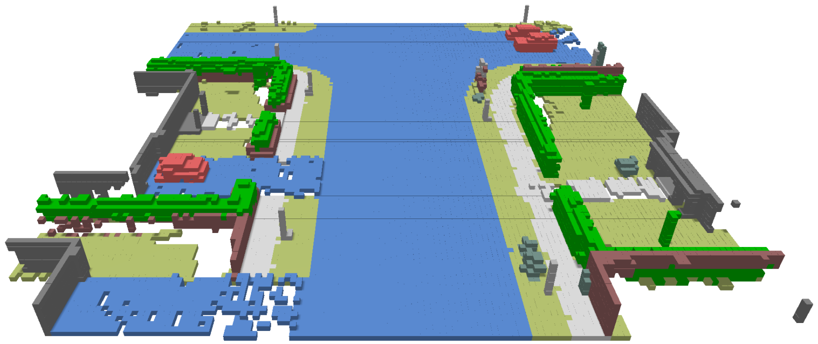
< Scene-scale Diffusion(SSD) >

building fence other pedestrian
pole road ground sidewalk
empty (air) vehicle vegetation

Results

- Show quite good results on synthetic datasets.





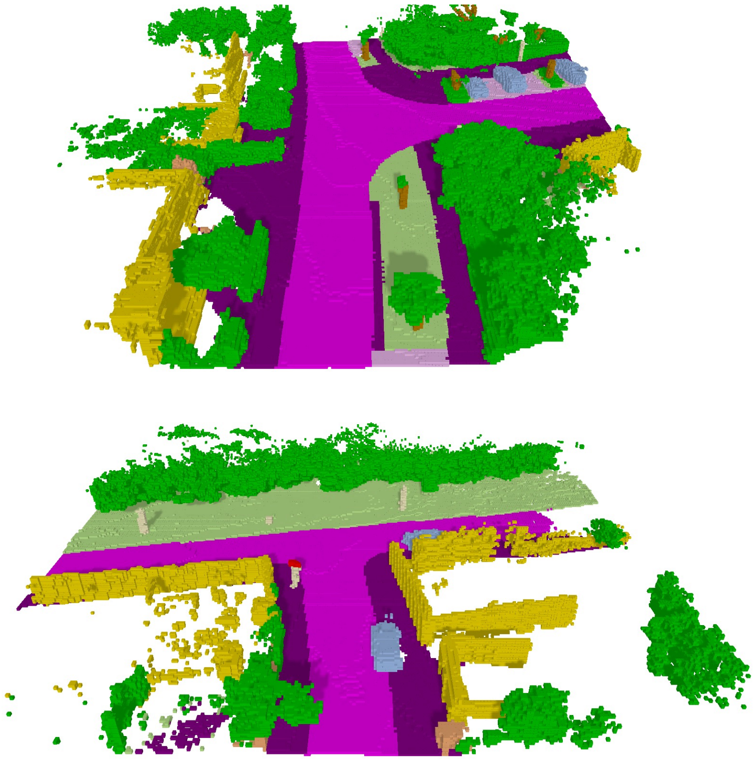
- Limitation
  - Suffers heavy computation burden.
  - Have to represent redundant empty region like sky.

road sidewalk parking ground building traffic-sign car
truck bicycle motorcycle vehicle vegetation motorcyclist pole
terrain person bicyclist trunk fence empty (air)

Challenges

- Scene-level dataset
  - High resolution.
  - A lots of empty region (e.g., sky).
    - Sensor limitations.

      e.g., occlusions, range constraints.
  - Different size of objects.



Voxels
H x W x Z x #Classes

CS380

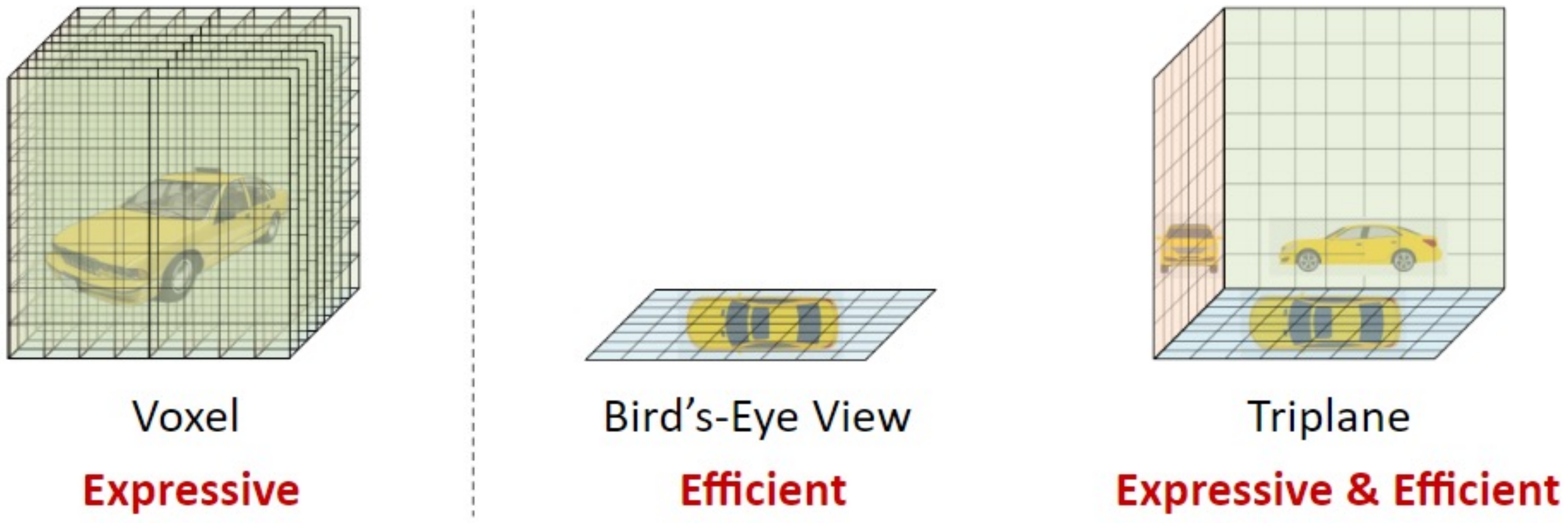
SemCity: Semantic Scene Generation with Triplane Diffusion

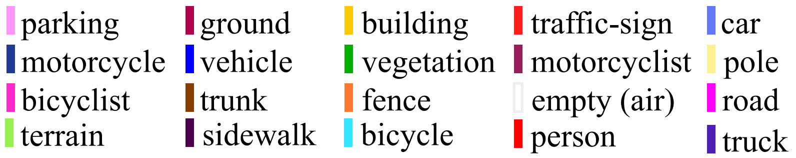
Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Ju-Hyeong Seon and Sung-Eui Yoon, *SemCity: Semantic Scene Generation with Triplane Diffusion, CVPR 2024*

SGVR Lab
KAIST

Ideas

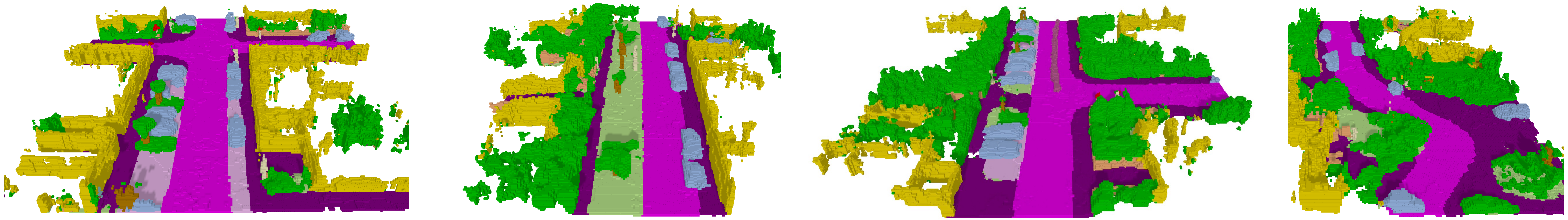
- Decompose a scene into 3 orthogonal 2D planes.
- Utilized in 3D object reconstruction.





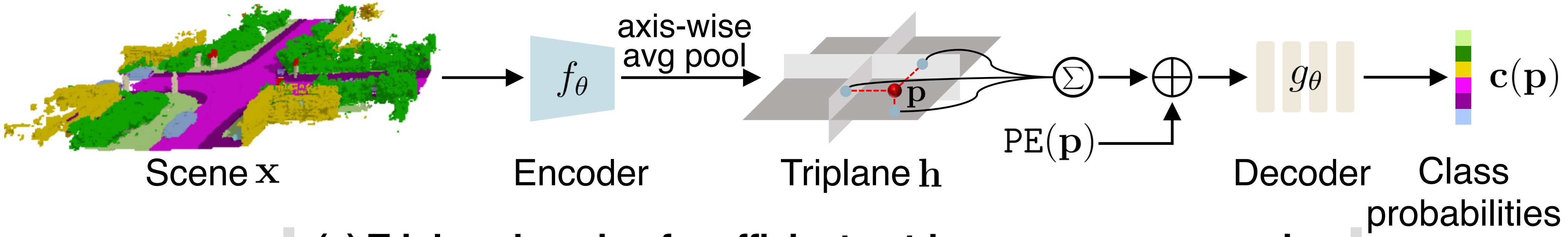
Ideas

- Leverage the triplane representation for the generation of real outdoor scenes.
  - Efficient and expressive.
  - Better focus on objects rather than empty region.
  - Spatial awareness representation helps capture semantic and geometric complexity within a scene.

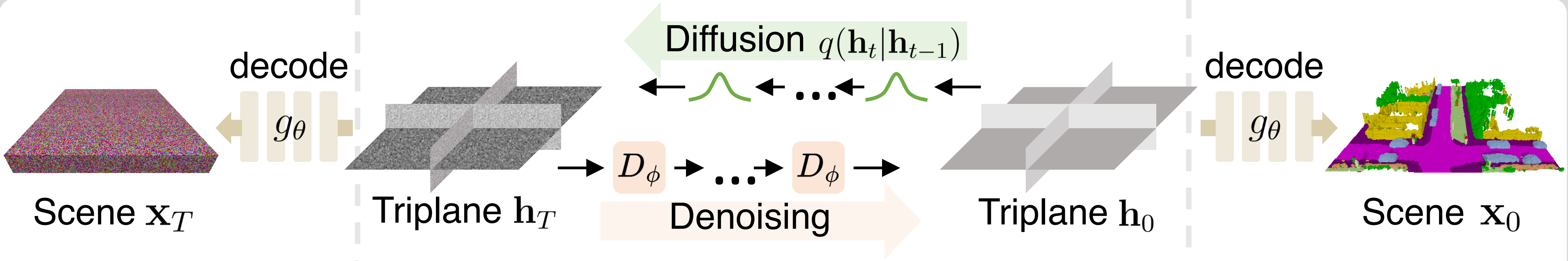


Scene generation

Method : Training

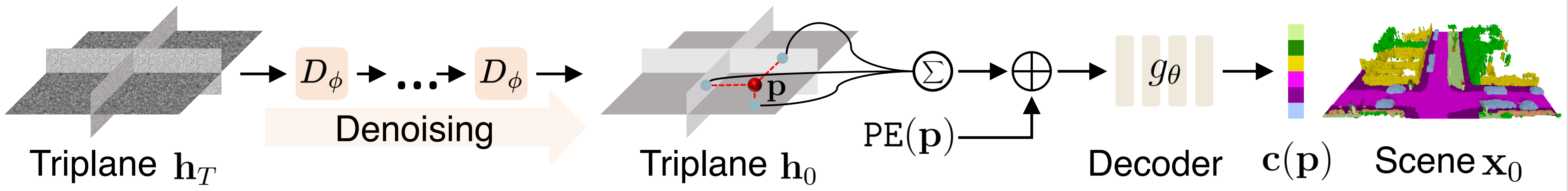


(a) Triplane learning for efficient outdoor scene compression

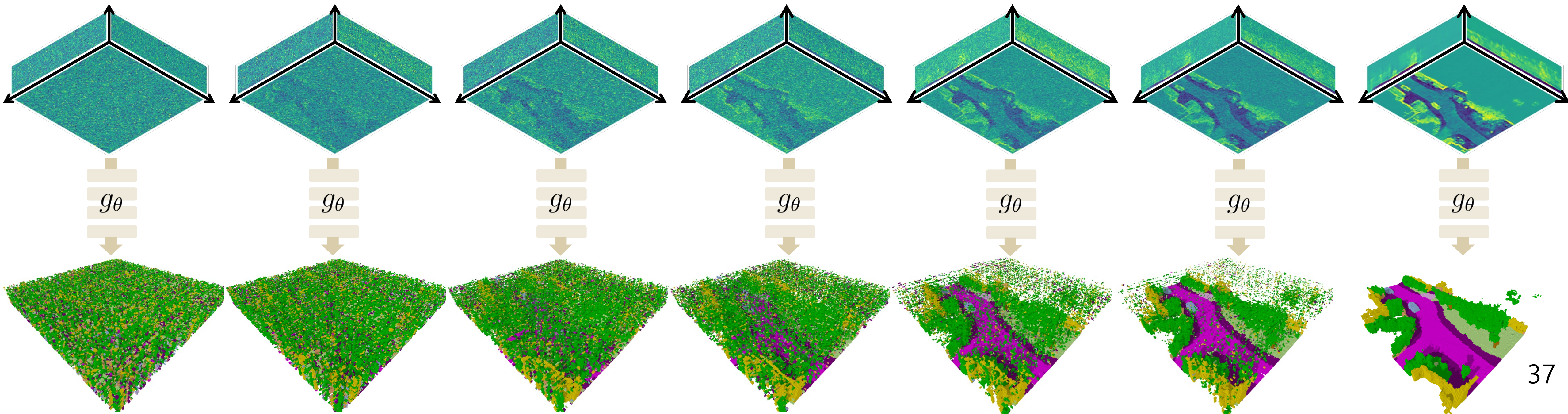


(b) Triplane diffusion for outdoor scene generation

Method : Sampling

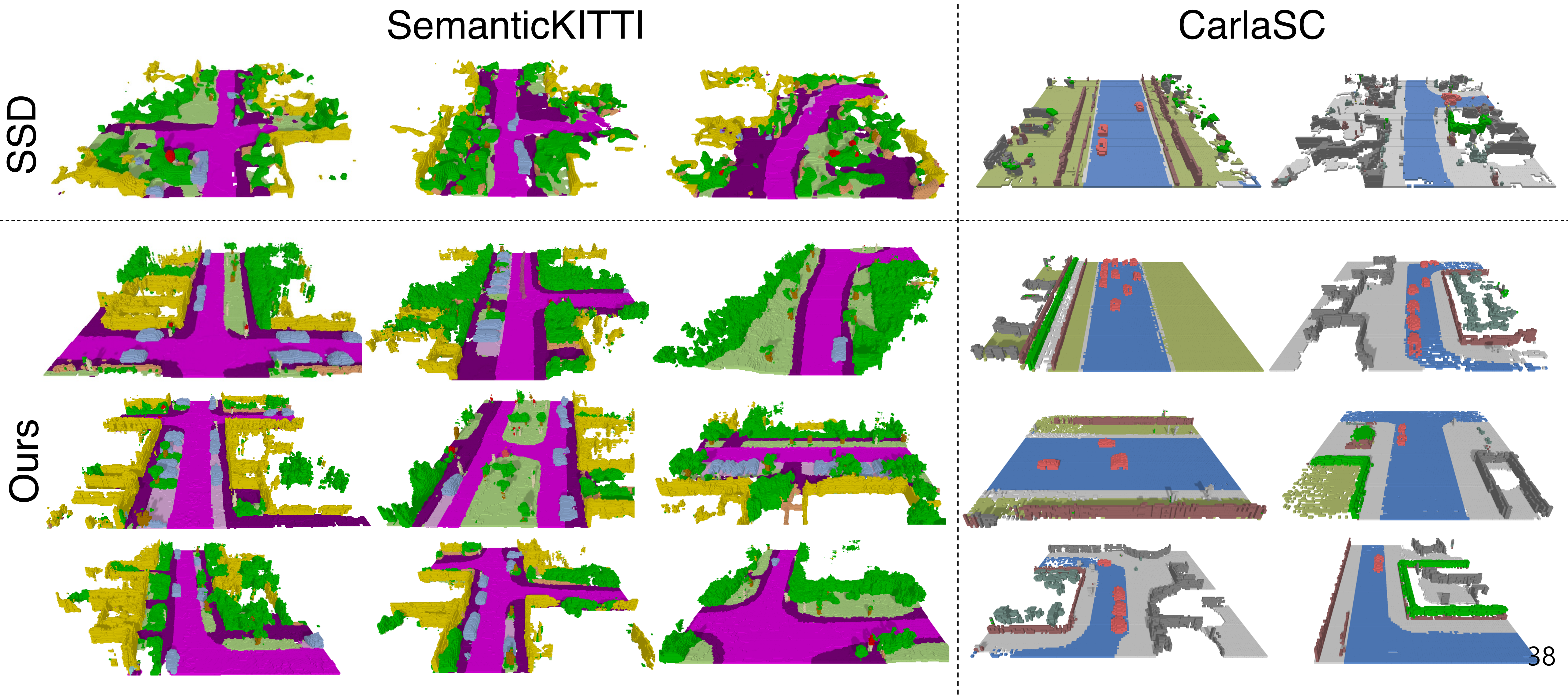


Triplane diffusion for outdoor scene generation



road truck terrain sidewalk bicycle person parking motorcycle bicyclist ground vehicle trunk building vegetation fence traffic-sign motorcyclist empty (air) car pole

Generation Results



Generation Results

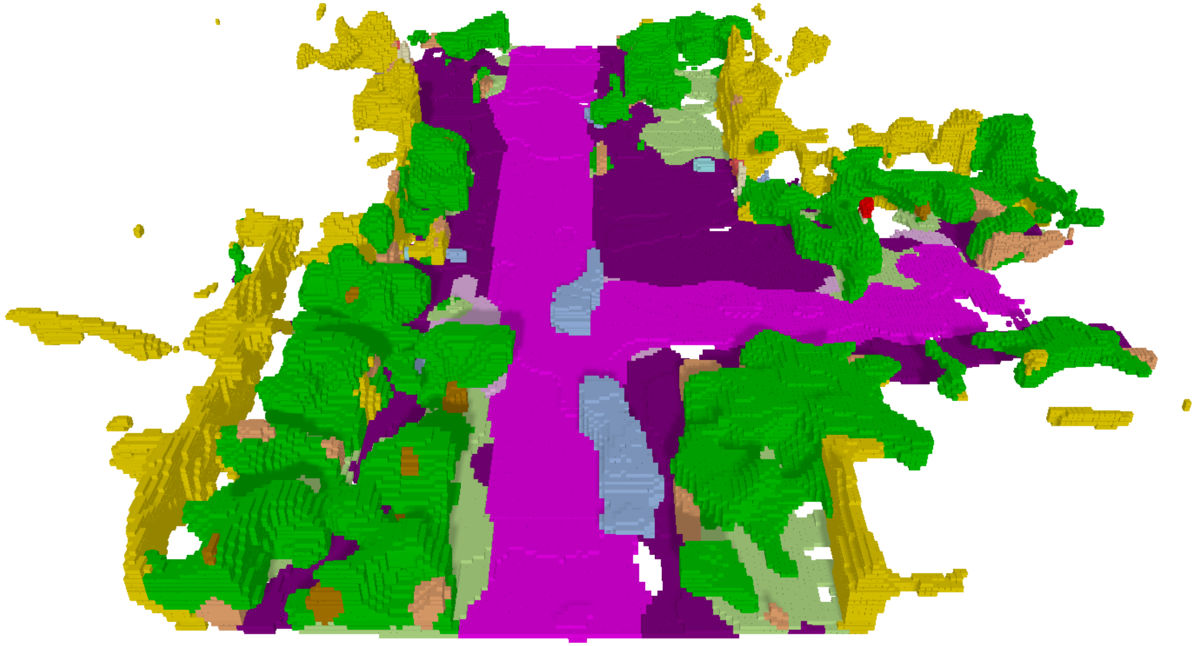
Model	FID ↓	KID ↓	IS ↑	Prec ↑	Rec ↑
	Diversity & Fidelity		Fidelity		Diversity
SemanticKITTI [6]					
SSD [24]	112.82	0.12	2.23	0.01	0.08
SemCity (Ours)	56.55	0.04	3.25	0.39	0.32
CarlaSC [50]					
SSD [24]	87.39	0.09	2.44	0.14	0.07
SemCity (Ours)	40.63	0.02	3.51	0.31	0.09

Quantitative results of semantic scene generation

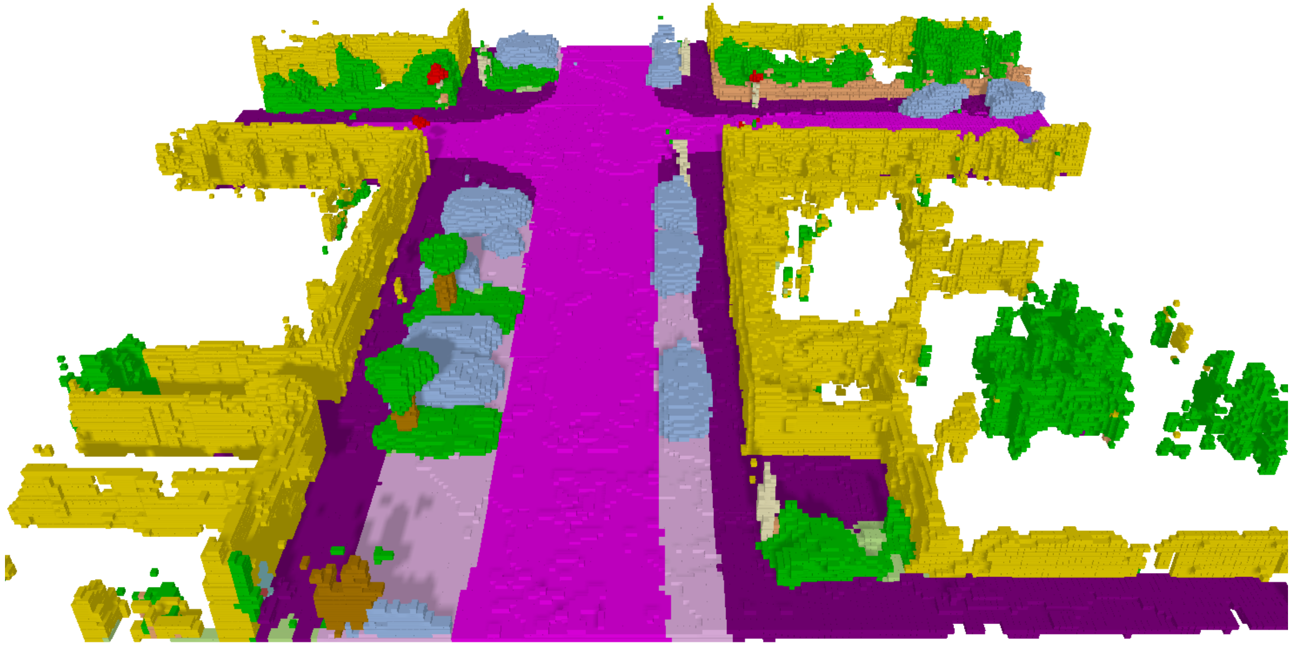
road truck terrain sidewalk bicycle person parking motorcycle bicyclist ground vehicle trunk building vegetation fence traffic-sign motorcyclist empty (air) car pole

Generation Results : Comparison

SSD



SemCity

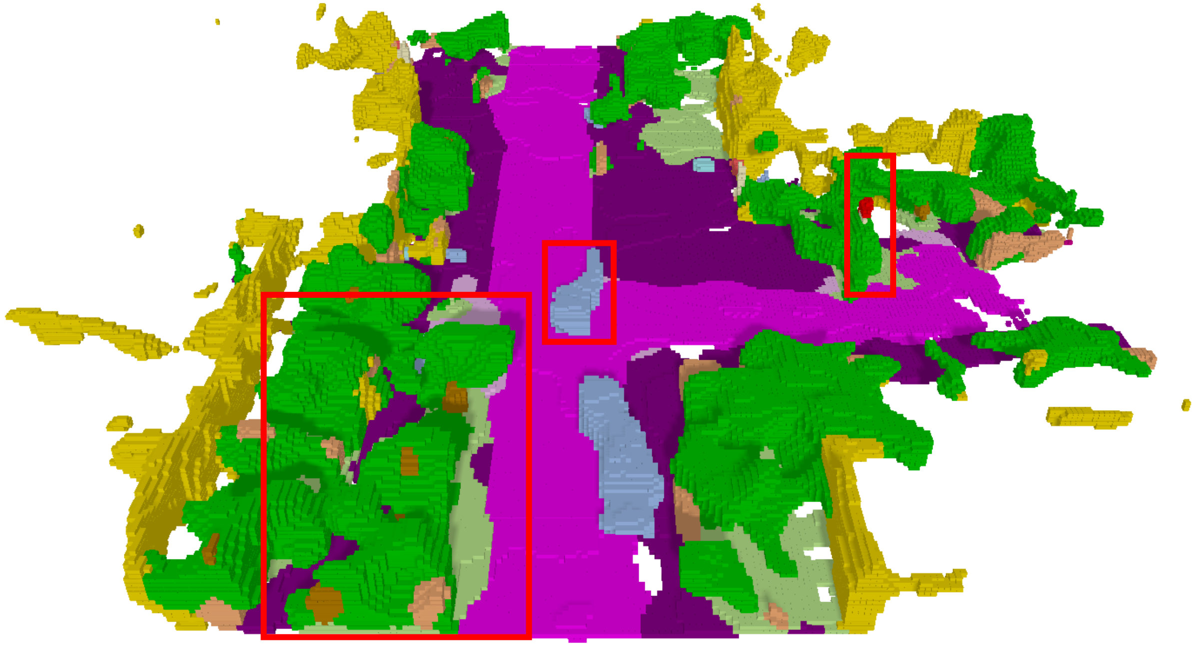


- Overall contours : road, building

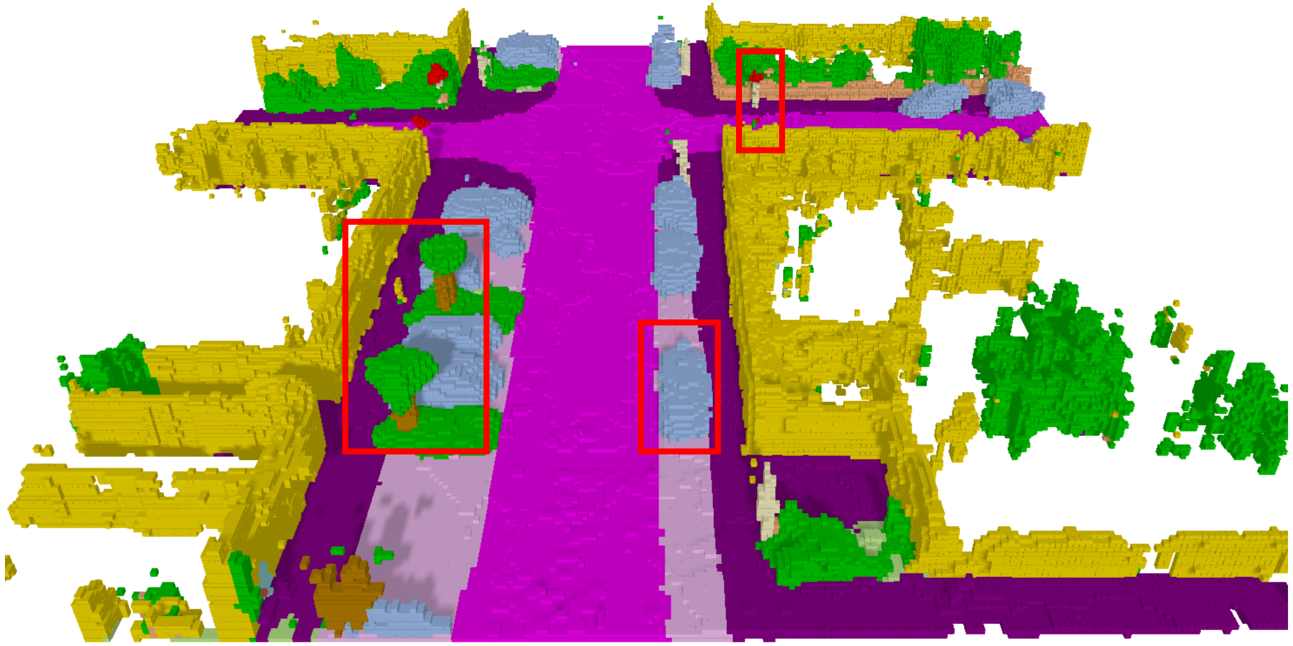
road truck terrain sidewalk bicycle person parking motorcycle bicyclist ground vehicle trunk building vegetation fence traffic-sign motorcyclist empty (air) car pole

Generation Results : Comparison

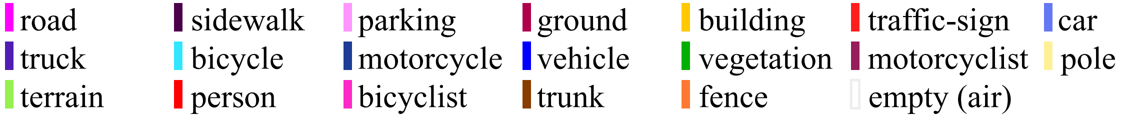
SSD



SemCity

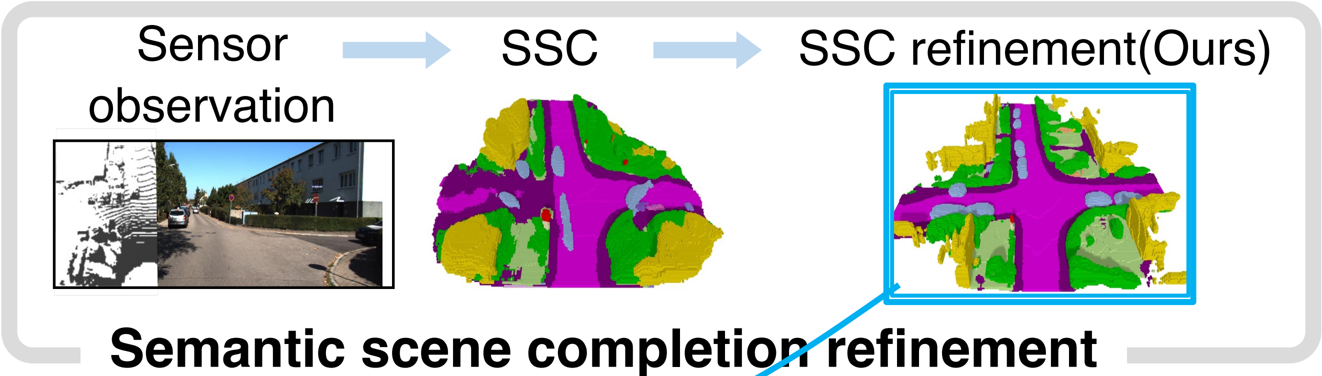


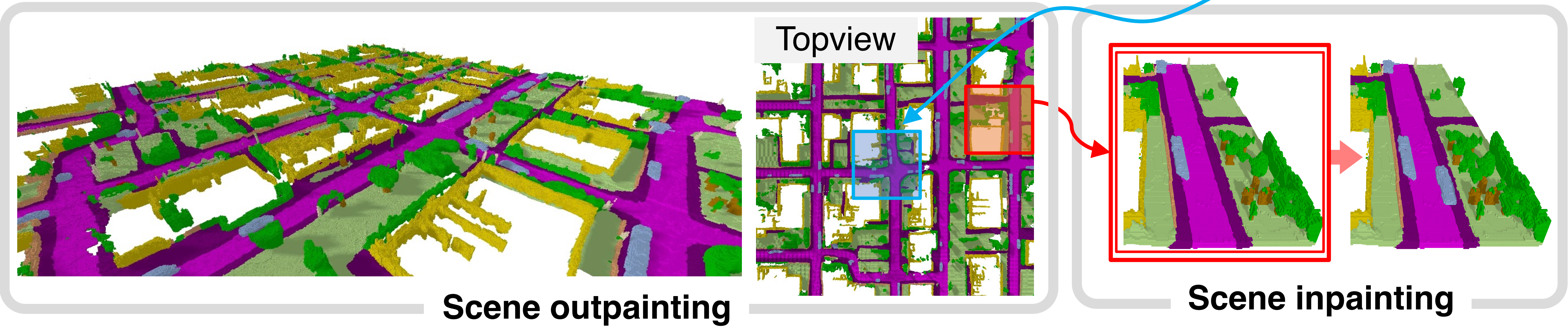
- Overall contours : road, building
- Finer structures : trunk and leave, traffic light and pole, car



Conditional Generation

- We extend our model to refine the predictions of SSC models.



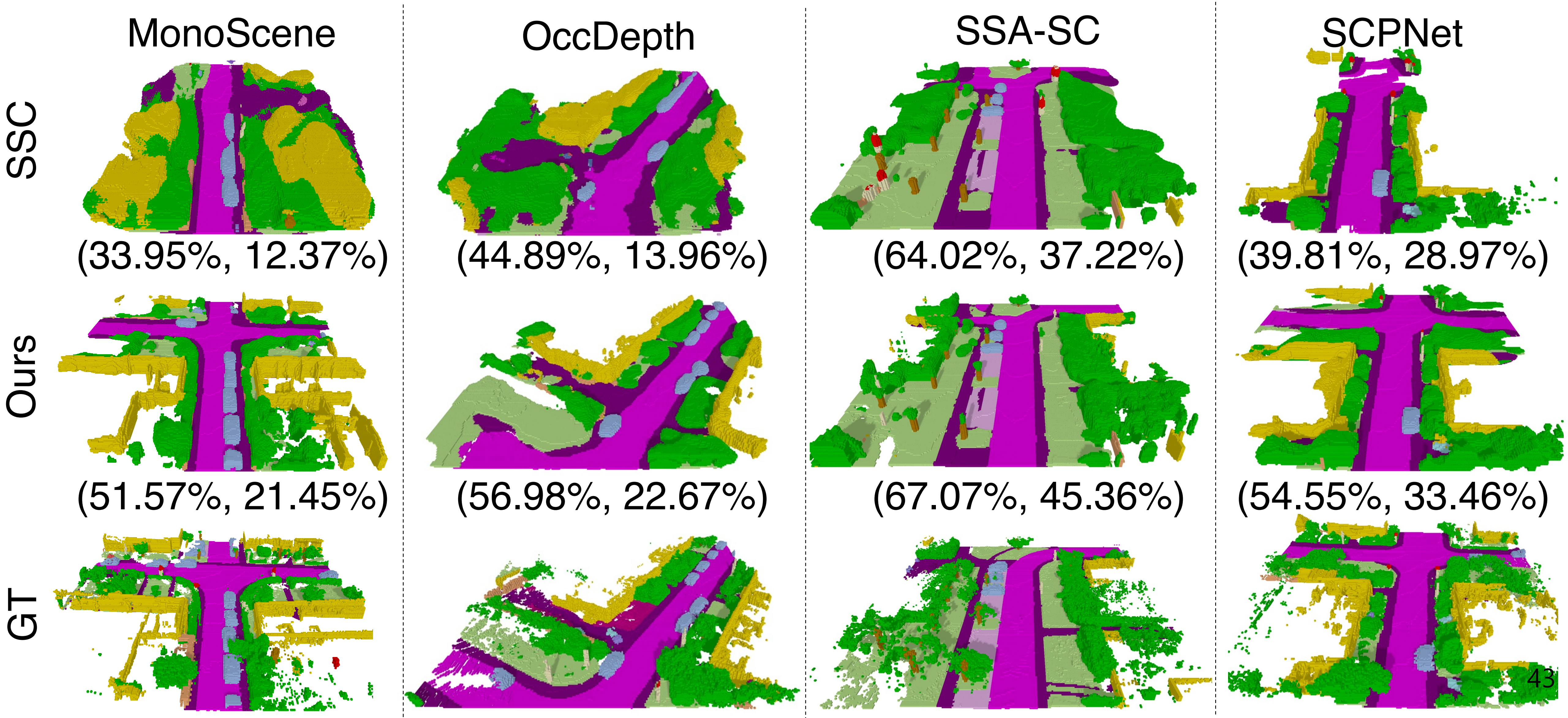


- We propose to manipulate triplane features during our diffusion process for scene outpainting and inpainting.

road sidewalk parking ground building traffic-sign car
truck bicycle motorcycle vehicle vegetation motorcyclist pole
terrain person bicyclist trunk fence empty (air)

Semantic Scene Completion Refinement

(·, ·) : IoU, mIoU

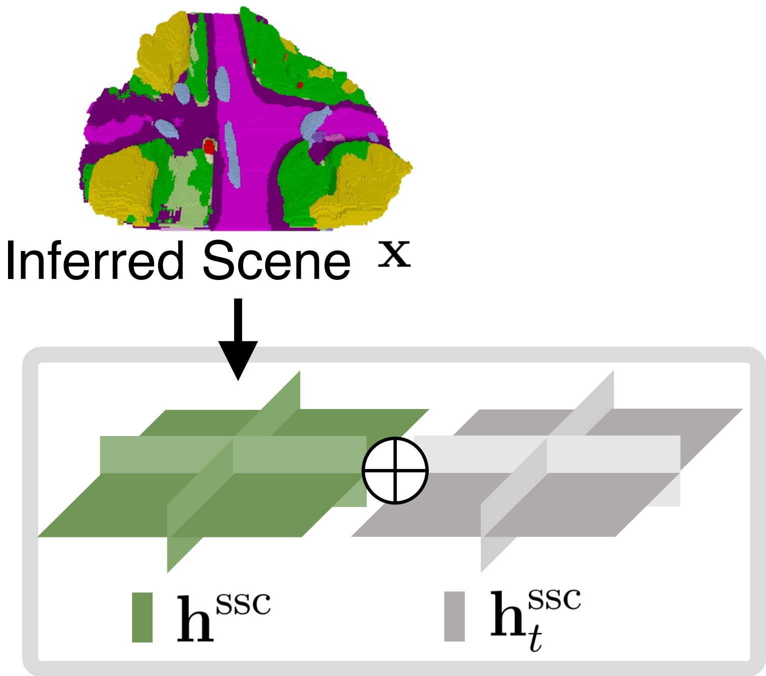


Semantic Scene Completion Refinement

Completeness

Semantic segmentation of completed scene

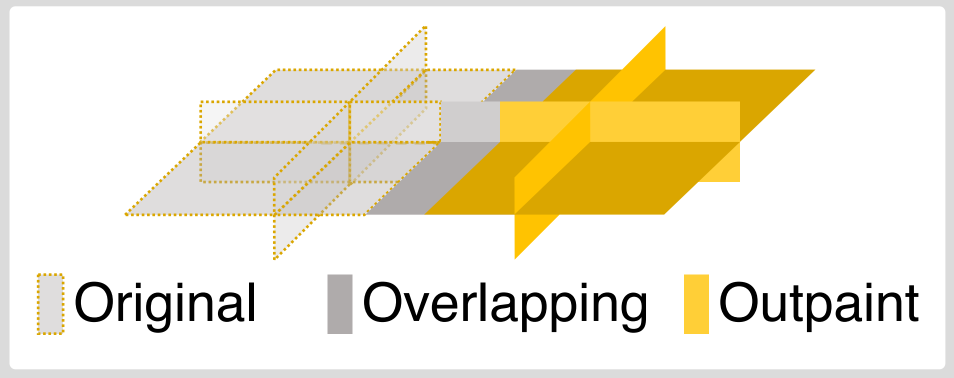
SSC Input	Method	IoU ↑	mIoU ↑
RGB	MonoScene [9]	37.12	11.50
	MonoScene + Ours	50.44	17.08
	OccDepth [32]	41.60	12.84
	OccDepth + Ours	50.20	16.79
Point Cloud	SSA-SC [54]	58.25	24.54
	SSA-SC + Ours	60.71	25.58
	SCPNet [52]	50.24	37.55
	SCPNet + Ours	59.25	38.19



Quantitative results of semantic scene completion refinement

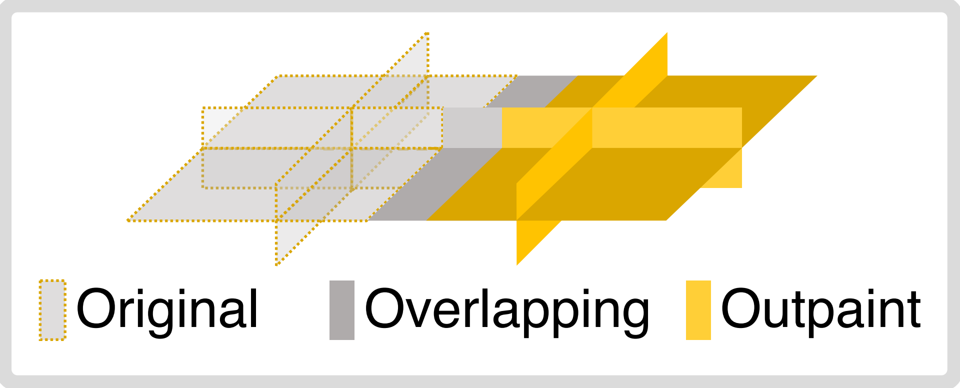
Scene Outpainting

256 x 256 x 32 → 1792 x 2816 x 32



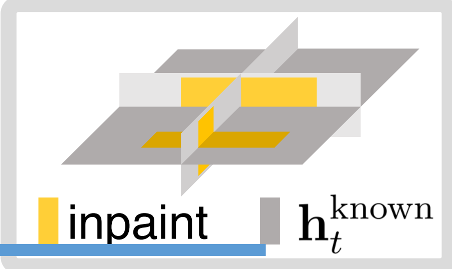
Scene Outpainting

256 x 256 x 32 → 1792 x 2816 x 32



Scene Outpainting

Scene Inpainting



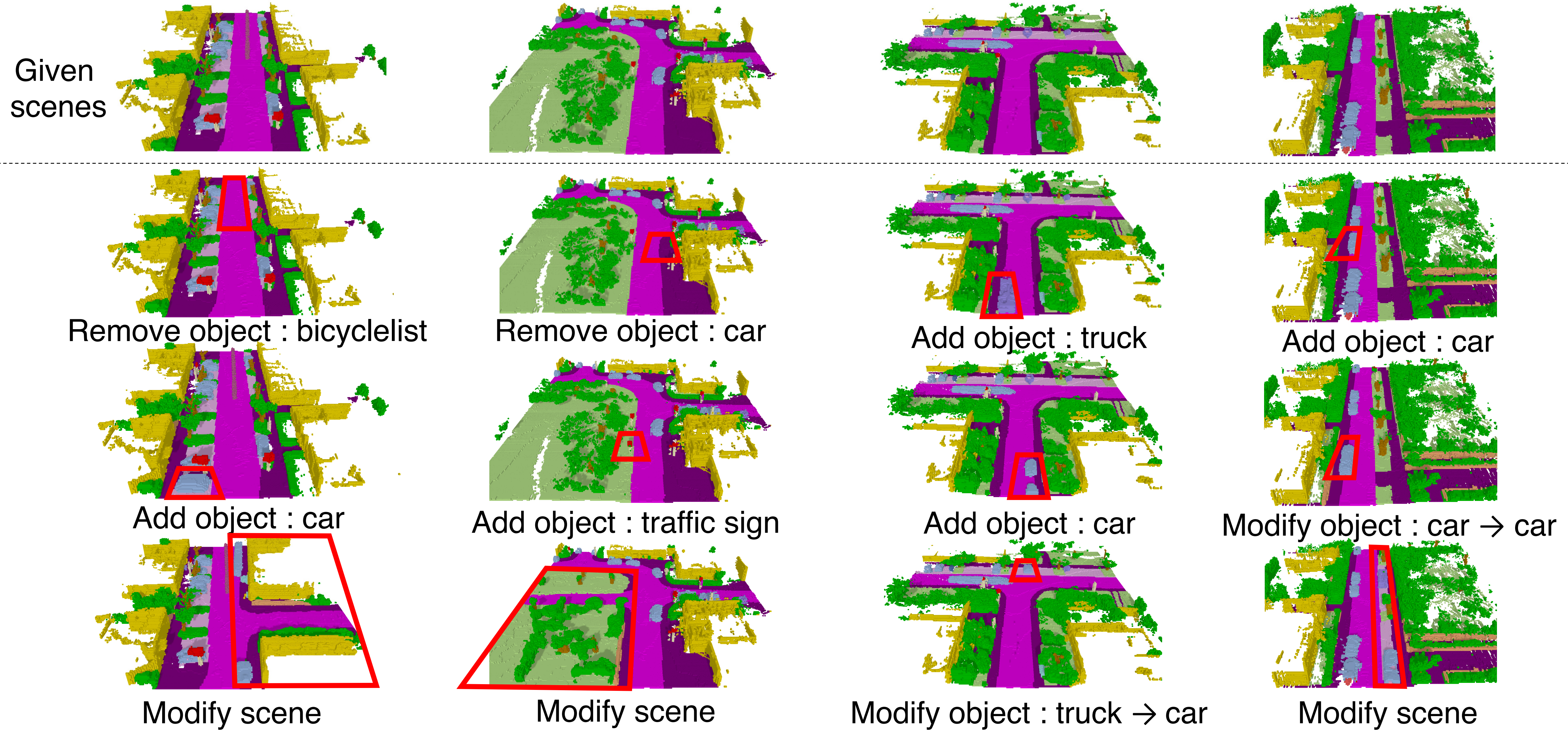
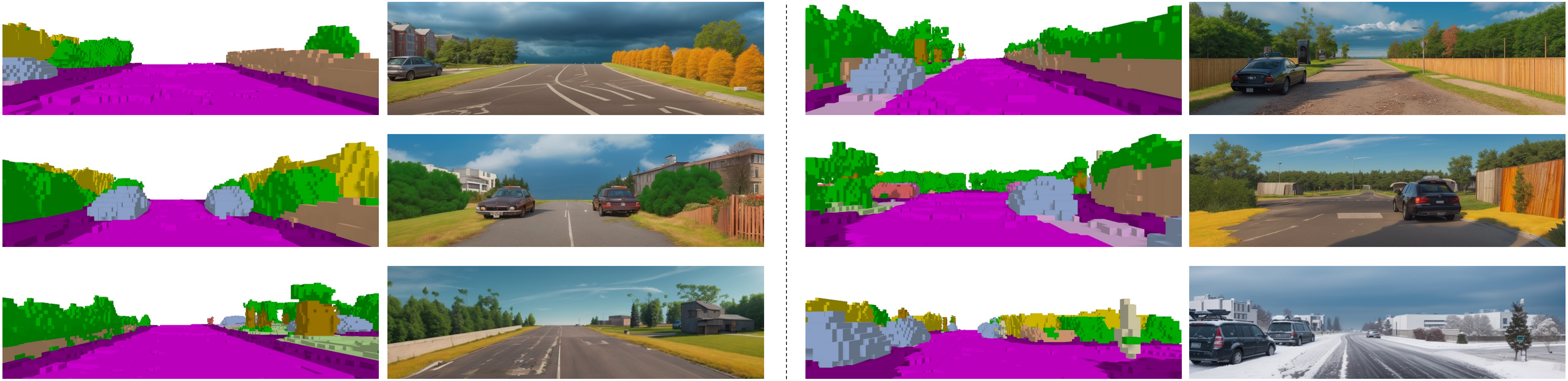


Image to Image Generation

- Exploit ControlNet to generate RGB images by conditioning semantic and depth maps rendered from our generated scene.

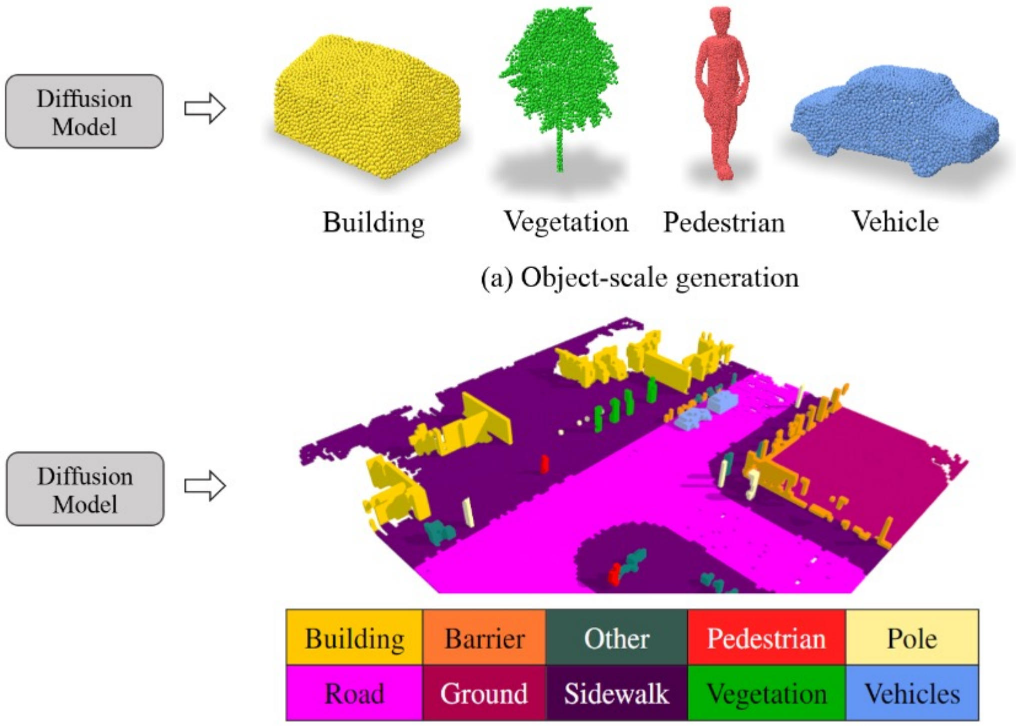


Generated scene   Generated image   Generated scene   Generated image

Conclusion

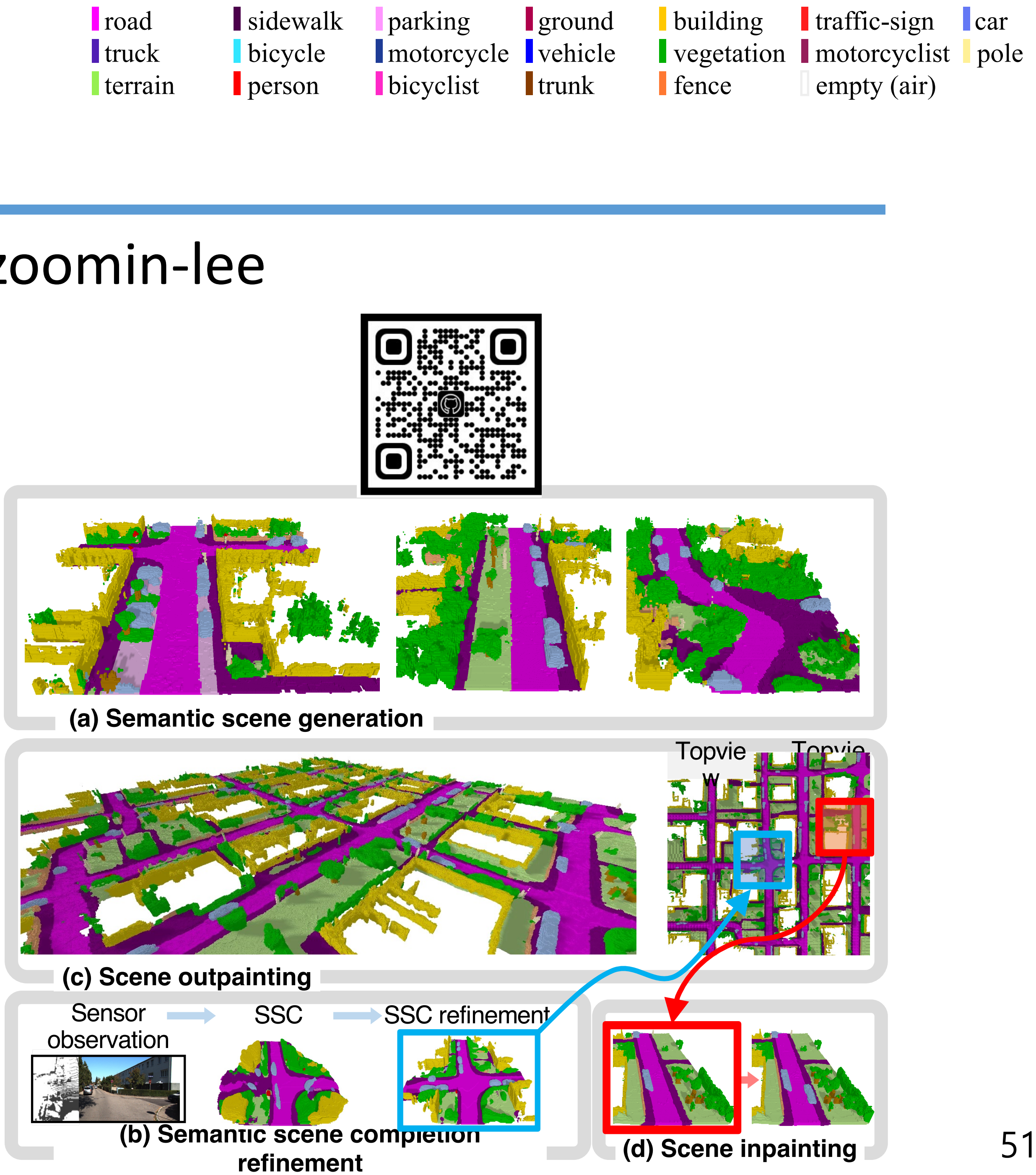
Conclusion

- Open Source : https://github.com/zoomin-lee



(a) Object-scale generation

(b) Scene-scale generation (Ours)



(a) Semantic scene generation

(c) Scene outpainting

(b) Semantic scene completion refinement

(d) Scene inpainting

Diffusion Model for Scene-level Generation

- Firstly utilized the diffusion model on a 3D outdoor dataset.
- Enhancing outdoor scenes generation through a triplane representation.
- By manipulating triplane, our model can both inpaint and outpaint scenes.
- Our model can refine the outcomes of existing semantic scene completion model by utilizing learned 3D scene prior.

Thank you.