The Landscape of Deep Generative Learning

Diffusion Models for Image Generation in Remote Sensing

GISLab Short-Term Course 2025 Summer

Zhenyuan Chen
School of Earth Science, Zhejiang University

June 25, 2025

bili_sakura@zju.edu.cn

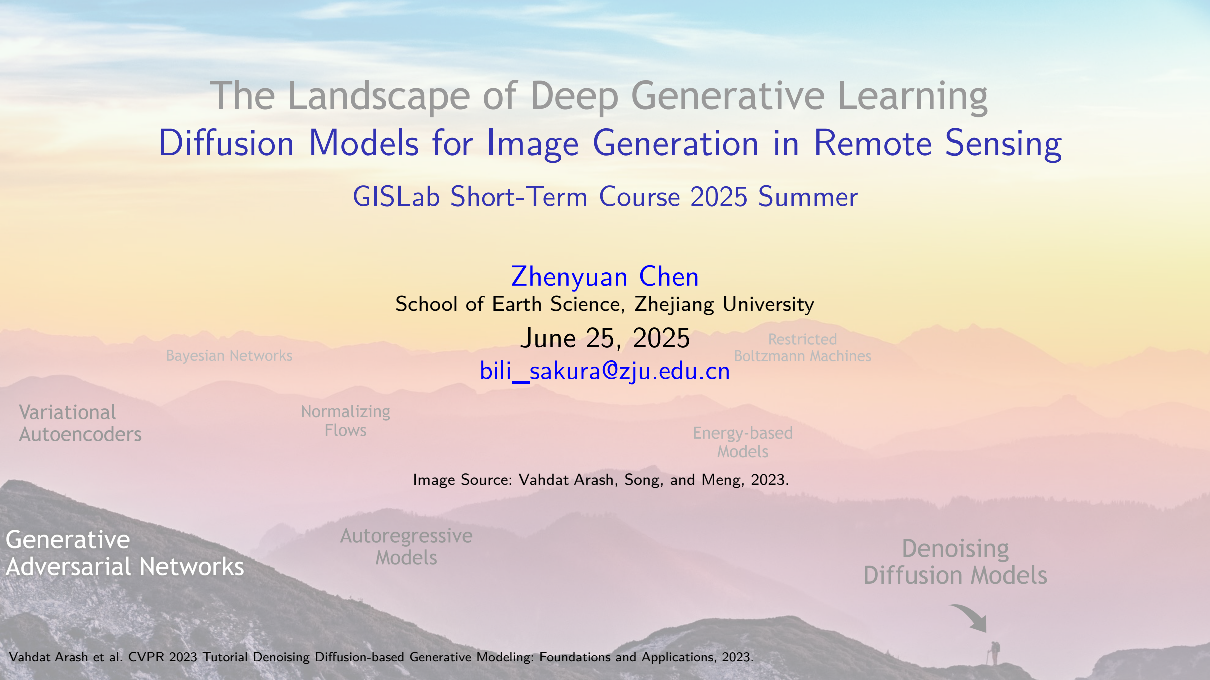


Image Source: Vahdat Arash, Song, and Meng, 2023.

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

Outline

- Background: Generative Models for Image Synthesis
- Diffusion Models: Theory
- Applications in Remote Sensing
- Summary & Q/A

© 2025 Sakura

Background: Generative Models for Image Synthesis

© 2025 Sakura

Background: Generative Models for Image Synthesis

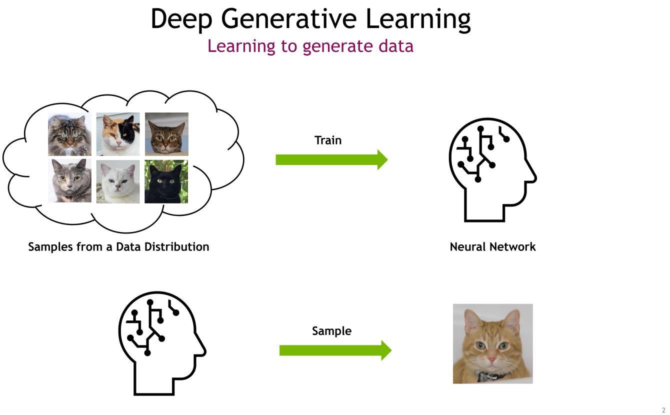


Figure: Illustration of generative modeling (Vahdat Arash, Song, and Meng, 2023).

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

© 2025 Sakura

Timeline of Generative Models

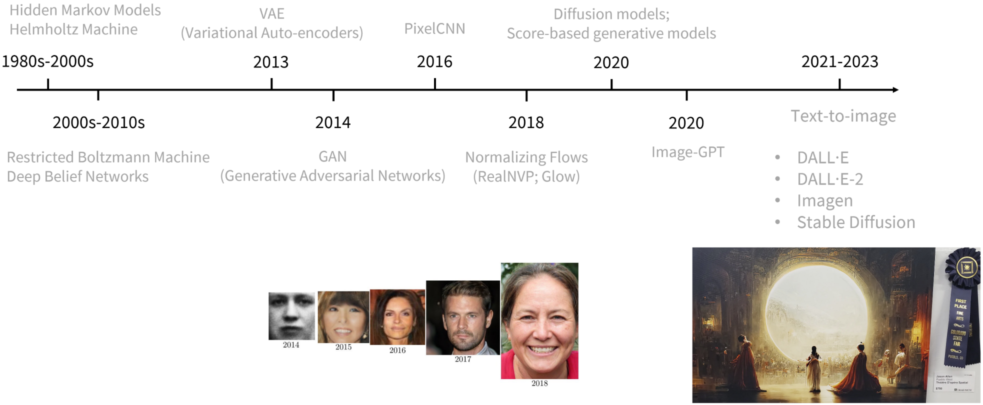


Figure: Timeline of key developments in generative models (Deng, 2024).

Deng, [CS7352] Advanced Neural Network Theory and Application, SJTU Spring, 2024.

© 2025 Sakura

Diffusion Models: Theory

© 2025 Sakura

Denoising Diffusion Models

Denoising diffusion models consist of two processes:

- A forward diffusion process that gradually adds noise to the input.
- A reverse denoising process that learns to generate data by denoising.

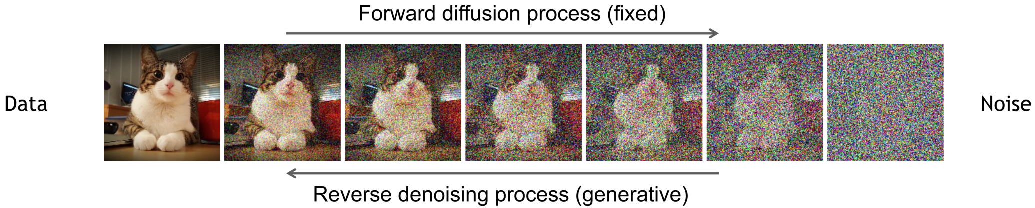


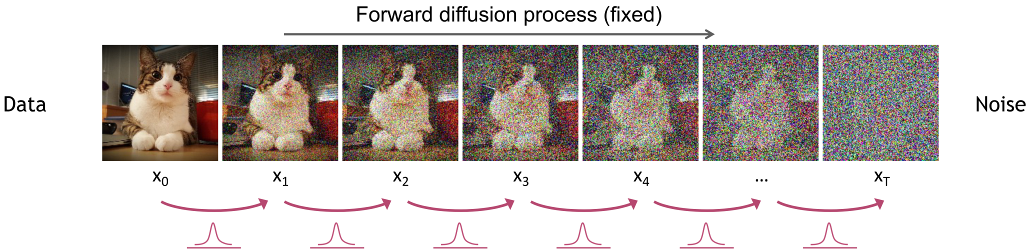
Figure: Diffusion models generate data through iterative denoising (Sohl-Dickstein et al., 2015; Ho, Jain, and Abbeel, 2020). Image credit: Vahdat Arash, Song, and Meng, 2023.

Sohl-Dickstein, et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML, 2015.
Ho, et al. Denoising Diffusion Probabilistic Models, NeurIPS, 2020.
Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

© 2025 Sakura

Forward Diffusion Process

The formal definition of the forward process in T steps:



$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \quad \Longrightarrow \quad q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \quad \text{(joint)}$$

Image credit: Vahdat Arash, Song, and Meng, 2023.

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

© 2025 Sakura

Forward Diffusion Process and Step-by-Step Expansion

Summary of the Forward Diffusion Process:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t, \quad q(x_t \mid x_{t-1}) \sim \mathcal{N}\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right)$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

Detailed Step-by-Step Expansion:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$$

$$x_t = \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-1}\right) + \sqrt{1-\alpha_t}\epsilon_t$$

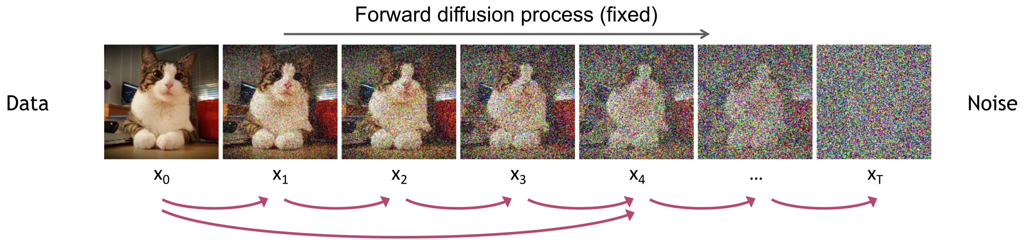
$$x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$$

$$x_t = \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2}} x_{t-3} + \cdots + \sqrt{1-\alpha_t \alpha_{t-1}}\epsilon$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

© 2025 Sakura

Diffusion Kernel



Define: $$\overline{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s) \implies q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\overline{\alpha}_t}\,\mathbf{x}_0, (1-\overline{\alpha}_t)\mathbf{I}\right) \quad \text{(Diffusion Kernel)}$$

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

The noise schedule $\{\beta_t\}$ is chosen so that $\overline{\alpha}_T \to 0$ and $q(\mathbf{x}_T \mid \mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$.

Image credit: Vahdat Arash, Song, and Meng, 2023.

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

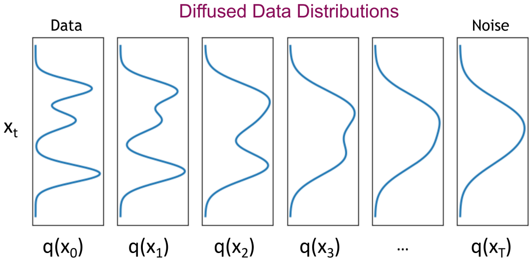
© 2025 Sakura

What happens to a distribution in the forward diffusion?

So far, we discussed the diffusion kernel $q(\mathbf{x}_t|\mathbf{x}_0)$ but what about $q(\mathbf{x}_t)$?

$$\underbrace{q(\mathbf{x}_t)}_{\text{Diffused data dist.}} = \int \underbrace{q(\mathbf{x}_0, \mathbf{x}_t)}_{\text{Joint dist.}}\, d\mathbf{x}_0 = \int \underbrace{q(\mathbf{x}_0)}_{\text{Input data dist.}}\, \underbrace{q(\mathbf{x}_t|\mathbf{x}_0)}_{\text{Diffusion kernel}}\, d\mathbf{x}_0$$

The diffusion kernel is Gaussian convolution.



We can sample $\mathbf{x}_t \sim q(\mathbf{x}_t)$ by first sampling $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ and then sampling $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ (i.e., ancestral sampling).

Image credit: Vahdat Arash, Song, and Meng, 2023.

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

© 2025 Sakura

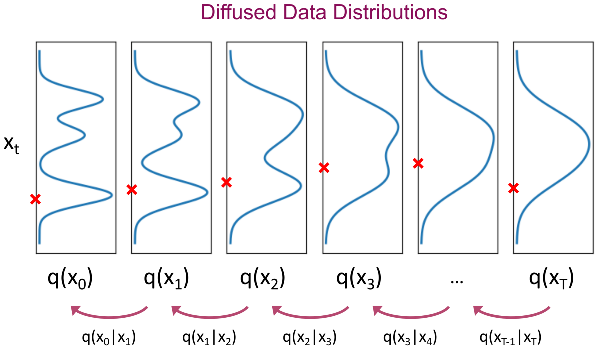
Generative Learning by Denoising

Recall that the diffusion parameters are designed such that $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$.

Generation:

- Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$
- Iteratively sample

$$\mathbf{x}_{t-1} \sim \underbrace{q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}_{\text{True Denoising Dist.}}$$



Can we approximate $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$? Yes, we can use a normal distribution if β_t is small in each forward diffusion step.

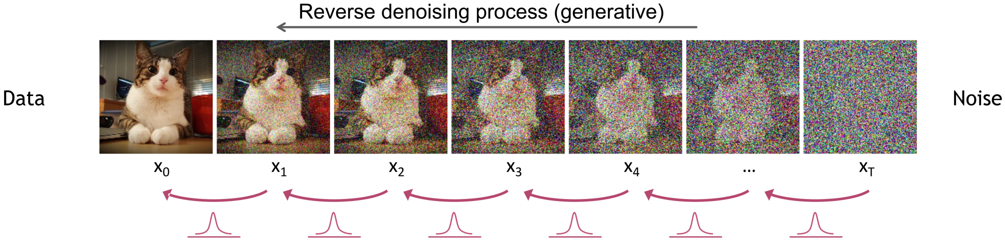
Image credit: Vahdat Arash, Song, and Meng, 2023.

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

© 2025 Sakura

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



$$p(x_T) = \mathcal{N}(x_T; 0, I)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I\right) \quad \Rightarrow \quad p_\theta(x_{0:T}) \; = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t).$$

where $\mu_\theta(x_t, t)$ is a trainable network (e.g., U-Net, Denoising Autoencoder)
Image credit: Vahdat Arash, Song, and Meng, 2023.

Vahdat Arash et al. CVPR 2023 Tutorial Denoising Diffusion-based Generative Modeling: Foundations and Applications, 2023.

© 2025 Sakura

Learning Denoising Model

Variational upper bound

- For training, use a variational upper bound (as in VAEs):

$$\mathbb{E}_{q_\lambda}[\log p_\theta(\mathbf{x})] \leq \mathbb{E}_{q_\lambda}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\lambda(\mathbf{z}|\mathbf{x})}\right] = L$$

- $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, mean parameterized as (Ho, Jain, and Abbeel, 2020):

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right)$$

- Variational objective:

$$L = \mathbb{E}_{q(\mathbf{x}_0,\epsilon)}\left[\sum_{t=1}^{T}\lambda_t \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right]\right]$$

- Set $\lambda_t = 1$ for all t works best (Ho, Jain, and Abbeel, 2020).

Ho, et al. Denoising Diffusion Probabilistic Models, NeurIPS, 2020.

© 2025 Sakura

Summary

Training and Sample Generation

Algorithm1-Training

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

Algorithm2-Sampling

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

Algorithms are derived from (Ho, Jain, and Abbeel, 2020)

Ho, et al. Denoising Diffusion Probabilistic Models, NeurIPS, 2020.

© 2025 Sakura

Applications in Remote Sensing

© 2025 Sakura

Introduction to DiffusionSat

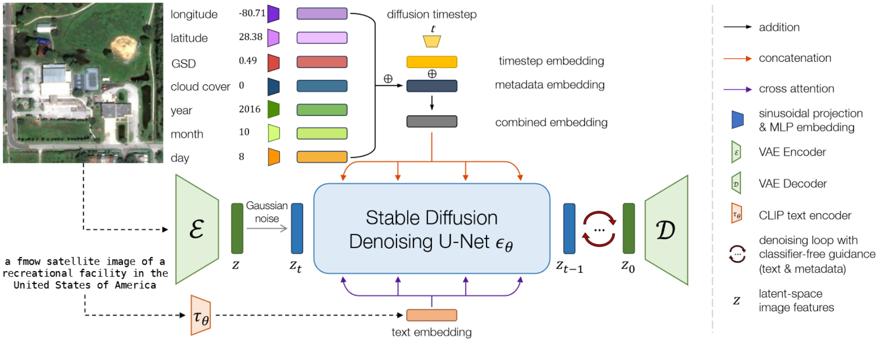


Figure: Overall architecture of the DiffusionSat *base* model, showing how freely-available metadata (sensor type, date, location) is fused with a diffusion backbone to generate high-fidelity satellite imagery (Khanna et al., 2024).

© 2025 Sakura

Text Encoder in DiffusionSat

- **Input Prompt:** `"A satellite image of a farmland"`
- **Tokenization:**
  - Split into subwords → tokens
  - Example mapping: {"A" : 101, " satellite" : 564, ..., <EOS> : 102}
- **CLIP Text Encoder:**
  - Token IDs → 512-dim embedding
  - Captures semantic features

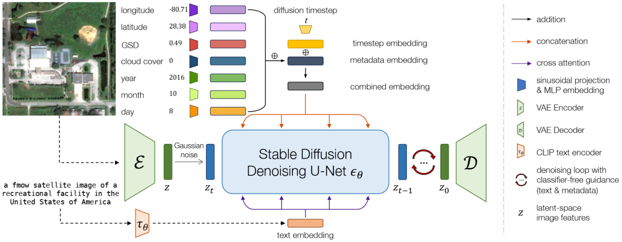


Figure: The text encoder module (OpenAI CLIP) tokenizes the input prompt and produces a 512-dimensional embedding to condition the diffusion backbone (Khanna et al., 2024).

Khanna, et al. DiffusionSat: A Generative Foundation Model for Satellite Imagery, ICLR, 2024.

© 2025 Sakura

Metadata Encoder in DiffusionSat

- **Input Metadata Example:**
  - Sensor: Sentinel-2
  - Location: (lat: 37.7749, lon: -122.4194)
  - Date: 2022-06-01
  - GSD: 10 m
  - Cloud cover: 5%
- **Processing Module:** Metadata Encoder
- **Output:** 512-dim conditioning embedding

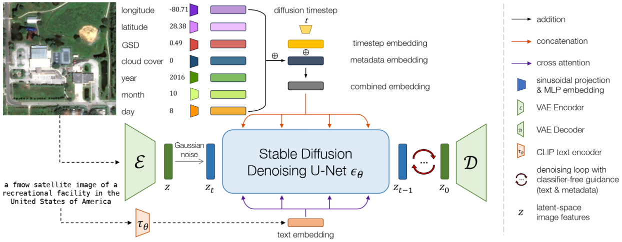


Figure: The Metadata Encoder transforms raw satellite metadata into a fixed-length embedding to condition the diffusion backbone (Khanna et al., 2024).

Khanna, et al. DiffusionSat: A Generative Foundation Model for Satellite Imagery, ICLR, 2024.

© 2025 Sakura

Image Processing & Diffusion Steps

- **Training Process:**
  - Input: Clean satellite image
  - Encode: Image Encoder → latent
  - Forward Diffusion: Add Gaussian noise over T steps
  - Conditioning: Inject Text & Metadata embeddings
  - Learn: UNet predicts and removes noise
- **Inference Process:**
  - Input: Random noise latent $x_T \sim \mathcal{N}(0, I)$
  - Reverse Diffusion: Iteratively denoise via UNet (conditioned)
  - Decode: Latent Decoder → final high-fidelity image

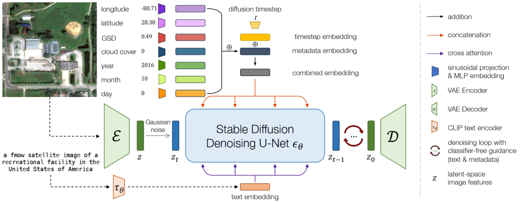


Figure: Comparison of the training (forward diffusion) and inference (reverse denoising) pipelines in the DiffusionSat base model, showing how images traverse the encoders, diffusion backbone, and decoder (Khanna et al., 2024).

Khanna, et al. DiffusionSat: A Generative Foundation Model for Satellite Imagery, ICLR, 2024.

© 2025 Sakura

Reverse Diffusion: Sampling

- **High-Level Sampling:**
  $x \sim p(\text{noise}, c_{\text{text}}, c_{\text{meta}})$
- **Where:**
  - noise: $x_T \sim \mathcal{N}(0, I)$
  - c_{text}: text embedding
  - c_{meta}: metadata embedding

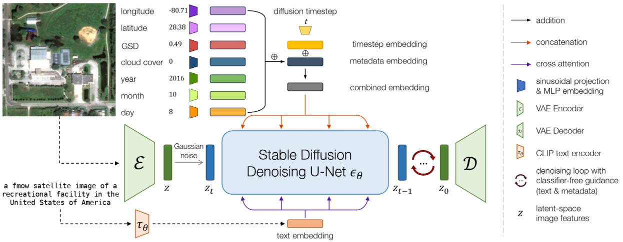


Figure: High-level sampling distribution for the DiffusionSat base model.

© 2025 Sakura

DiffusionSat+3DControlNet: Framework Overview

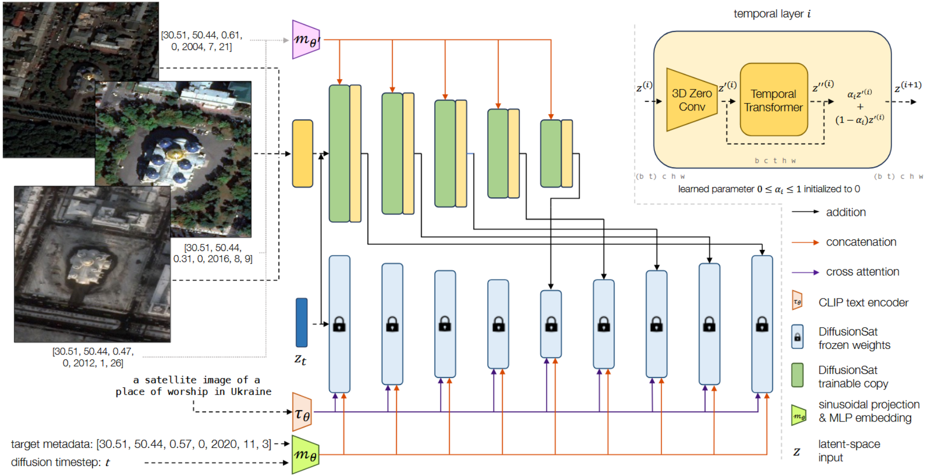


Figure: 3DControlNet in DiffusionSat (Khanna et al., 2024).

Khanna, et al. DiffusionSat: A Generative Foundation Model for Satellite Imagery, ICLR, 2024.

© 2025 Sakura

DiffusionSat+3DControlNet: Temporal Prediction Results

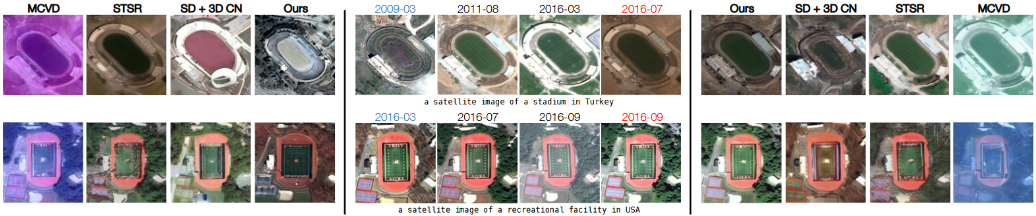


Figure: Generated samples from the fMoW-temporal, for temporal prediction (Khanna et al., 2024).

Model	$t' > t$			$t' < t$		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
SD + 3D CN	0.2027	11.0536	0.5523	0.2181	11.3004	0.5342
DiffusionSat + CN	0.3297	13.6938	0.5062	0.2862	12.4990	0.5307
DiffusionSat + 3D CN	**0.3983**	**13.7886**	**0.4304**	**0.4293**	**14.8699**	**0.3937**

Table: Table 4: Sample quality quantitative results on fMoW-temporal validation data. $t' > t$ represents generating an image in the past given a future image, and $t' < t$ is the task for generating a future image given a past image.

Khanna, et al. DiffusionSat: A Generative Foundation Model for Satellite Imagery, ICLR, 2024.

© 2025 Sakura

DiffusionSat+3DControlNet:Super-Resolution Results

Figure: Example results: DiffusionSat for multi-spectral super-resolution (Khanna et al., 2024).

Khanna, et al. DiffusionSat: A Generative Foundation Model for Satellite Imagery, ICLR, 2024.

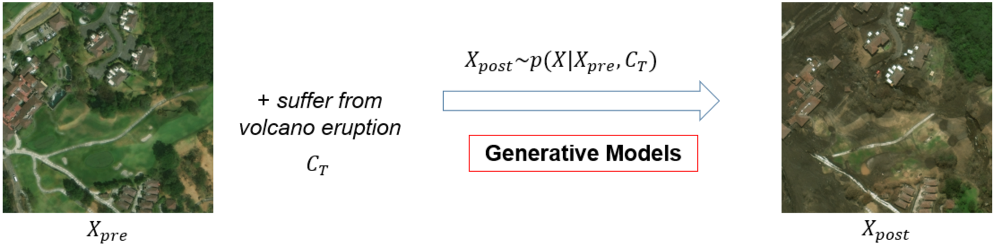
© 2025 Sakura

Further Discussion

© 2025 Sakura

Using Gen AI to Make New Images

- Generative models can create new, realistic images.
- We can use them to make more training data.
- Example: Give a "before" image and a description, get a new "after" image.



© 2025 Sakura

Supplymentary

© 2025 Sakura

How Do We Augment Data?

Classic Methods:

- Flip, rotate, crop, change colors, etc.

Modern Methods:

- Mix two images together (Mixup) (Zhang et al., 2018).
- Cut and paste parts of images (CutMix) (Yun et al., 2019).

Figure: Illustration of modern augmentation methods. From Left to Right: Mixup (Zhang et al., 2018), Cutout (DeVries and Taylor, 2017), and CutMix (Yun et al., 2019).

Soft Label Example (CutMix):

$$\text{cutmix\_label} = \lambda \cdot \text{label}_A + (1 - \lambda) \cdot \text{label}_B$$

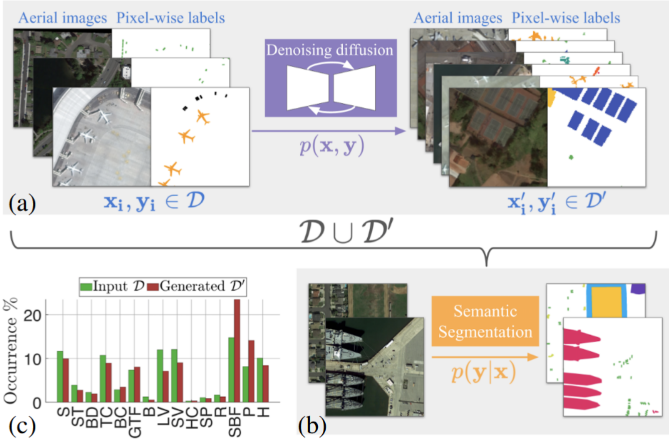
$$\text{Example:} \quad \lambda = 0.5, \quad \text{label}_A = [1, 0], \quad \text{label}_B = [0, 1]$$

$$\text{cutmix\_label} = 0.5 \times [1, 0] + 0.5 \times [0, 1] = [0.5, 0.5]$$

Zhang, et al. Mixup: Beyond Empirical risk minimazation, ICLR, 2018.
Yun, et al. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features, ICCV, 2019.
DeVries, et al. Improved Regularization of Convolutional Neural Networks with Cutout, arXiv, 2017.

© 2025 Sakura

Generative Models for Data Augmentation



SatSyn (Toker et al., 2024) proposes a generative model (diffusion model) to generate both images and corresponding masks for satellite segmentation. The synthetic dataset is used for data augmentation, yielding significant quantitative improvements in satellite semantic segmentation compared to other data augmentation methods.

Toker, et al. SatSynth: Augmenting Image-Mask Pairs through Diffusion Models for Aerial Semantic Segmentation, CVPR, 2024.

© 2025 Sakura

Application in Remote Sensing Image Generation: Text2Earth

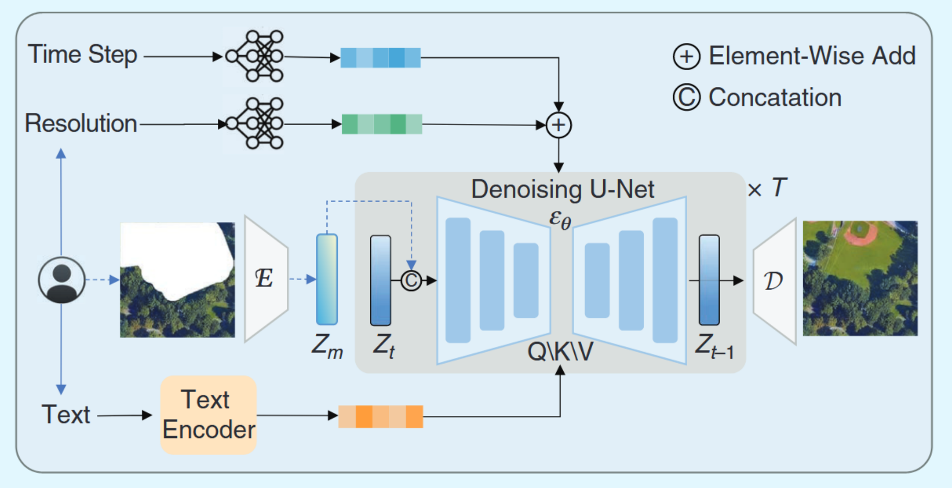


Figure: Text2Earth: Foundation model for text-driven Earth observation (Liu et al., 2025).

Liu, et al. Text2Earth: Unlocking text-driven remote sensing image generation with a global-scale dataset and a foundation model. GRSM. 2025.

© 2025 Sakura

Text2Earth: Example Results

Figure: Example results generated by Text2Earth (Liu et al., 2025).

Liu, et al. Text2Earth: Unlocking text-driven remote sensing image generation with a global-scale dataset and a foundation model. GRSM. 2025.

© 2025 Sakura

Application in Remote Sensing Image Generation: CRS-Diff

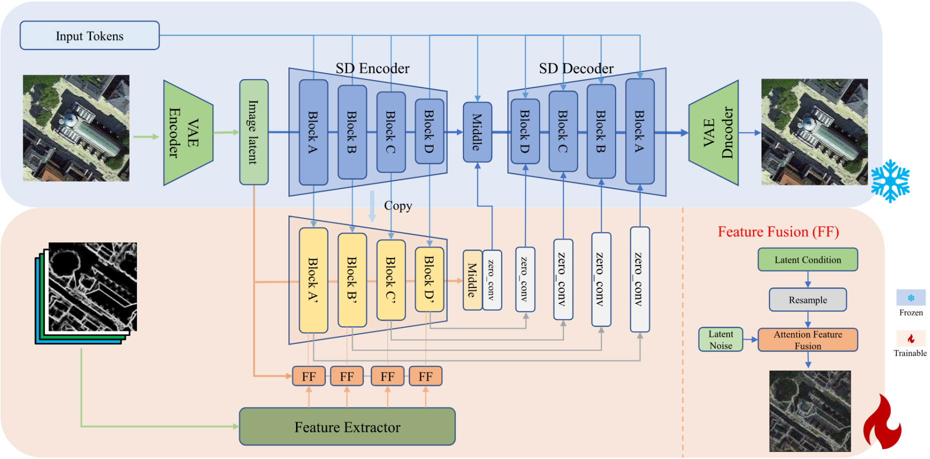


Figure: CRS-Diff: Controllable remote sensing image generation framework (Tang, Li, et al., 2024).

Tang, et al. CRS-Diff: Controllable Remote Sensing Image Generation with Diffusion Model. TGRS. 2024.

© 2025 Sakura

CRS-Diff: Example Results

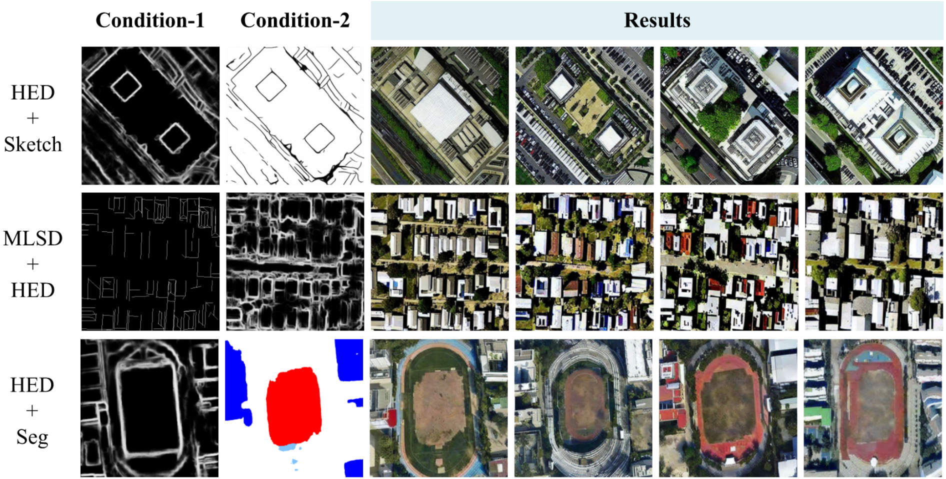


Figure: Example results generated by CRS-Diff (Tang, Li, et al., 2024).

Tang, et al. CRS-Diff: Controllable Remote Sensing Image Generation with Diffusion Model. TGRS. 2024.

© 2025 Sakura