Beyond the Process: Guiding Diffusion Models Under Constraints

Pesaresi Seminar Series

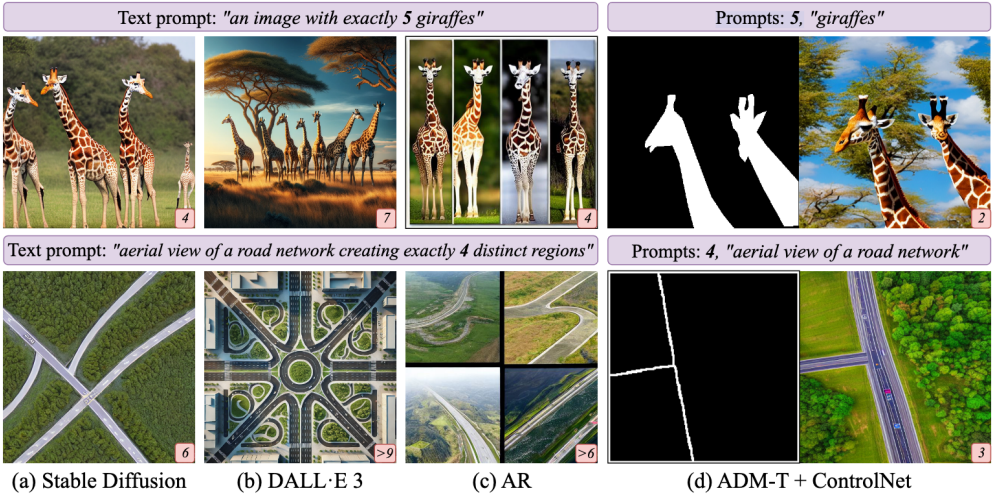
Leonardo Stoppani

University of Pisa

20 March 2026

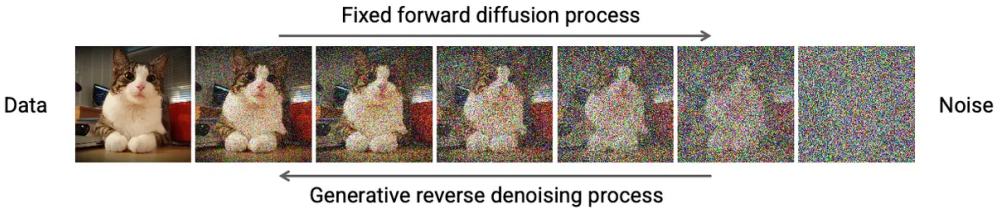
Introduction

Generative models and constraints



Refs: Gupta, Samaras, and Chen (2025)

A class of Generative models: Diffusion Models



The Intuition

Information in images is organized from coarse structure to fine details.

- **Forward process:** gradually corrupt data until it becomes pure noise.
- **Reverse process:** learn to denoise step by step.

Refs: Ho, Jain, and Abbeel (2020), Sohl-Dickstein et al. (2015), Dhariwal, P., & Nichol, A. (2021)

What happens under the hood?

Forward process

Markov chain that adds Gaussian noise at each step.

$$\beta_t \in (0,1), \qquad \alpha_t = 1 - \beta_t \in (0,1)$$

$$q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\, x_{t-1},\, \beta_t \mathbf{I})$$

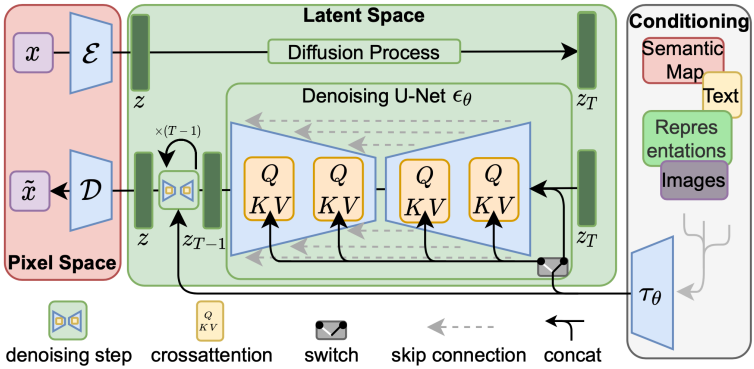
$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{\beta_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

Learned denoiser

The network takes as input x_t and t and predicts the added noise: $\epsilon_\theta(x_t, t) \approx \epsilon$.

Refs: Ho, Jain, and Abbeel (2020), Rombach et al. (2022)

The big picture



Backbone

U-Nets are common in pixel or latent space; DiTs use transformers and scale more cleanly.

Refs: Ho, Jain, and Abbeel (2020), Rombach et al. (2022)

Noise schedules and signal-to-noise ratio

Noise schedule

$\{\beta_t\}_{t=1}^{T}$ controls how fast the forward process destroys structure. With $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$:

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

Signal-to-noise ratio

$$\mathrm{SNR}(t) = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}.$$

Linear schedule

Simple and standard, but it often destroys signal too quickly in the last steps.

Cosine schedule

It keeps useful signal longer and makes the SNR decay smoother.

Refs: Ho, Jain, and Abbeel (2020), Nichol and Dhariwal (2021)

Training the denoiser

Algorithm 1 Training

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

Objective

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2 \right].$$

Why random t?

To ensure the model learns a wide range of SNR values, with one shared denoiser.

Refs: Ho, Jain, and Abbeel (2020), Nichol and Dhariwal (2021)

What about sampling?

Algorithm 2 Sampling

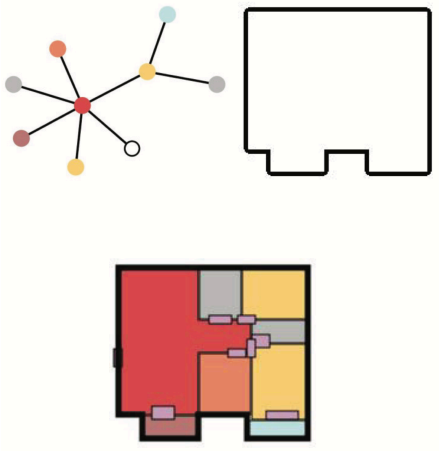
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}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

Faster sampling
DDIM reduces the number of steps for sampling from 1K to **50-100**.

Refs: Ho, Jain, and Abbeel (2020), Song, Meng, and Ermon (2021)

A Concrete Example: Floorplan generation under user-defined constraints

- User-defined constraints: **room connectivity graph** and **boundary**.
- Geometrical constraints: non-overlap, wall alignment, ...



How can we enforce constraints in diffusion models?

1. Conditional guidance
2. Two-stage generation
3. Projected diffusion
4. Neurosymbolic diffusion

1. Conditional guidance

Main idea

Modify the denoising process to steer generation towards satisfying the constraints.

- **Classifier guidance**: train a separate classifier to predict the condition from noisy inputs and use its gradients to guide the denoiser.
- **Classifier-free guidance (CFG)**: mix conditional and unconditional denoising predictions:

$$\hat{\epsilon} = (1 + w)\epsilon_\theta(x_t, c) - w\epsilon_\theta(x_t, \emptyset).$$

Inject conditions via **timestep embedding**, **cross-attention** or simple **concatenation** with the noisy input.

Refs: Dhariwal and Nichol (2021), Rombach et al. (2022), Li et al. (2023), Zhang et al. (2023)

2. Two-stage generation

Stage 1

Train a model to generate a **coarse structure** that satisfies the constraints (e.g. a segmentation map, a layout, or a low-res image).

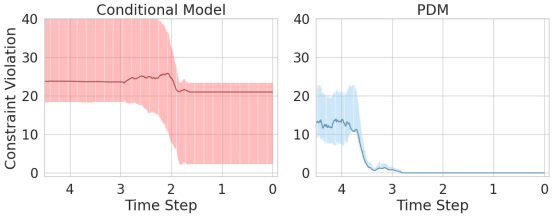
Stage 2

Tipically a foundation model (e.g. ControlNet) that is pretrained on various **conditioning signals**.



Refs: Gupta, Samaras, and Chen (2025), Zhang, L., Rao, A. (2023)

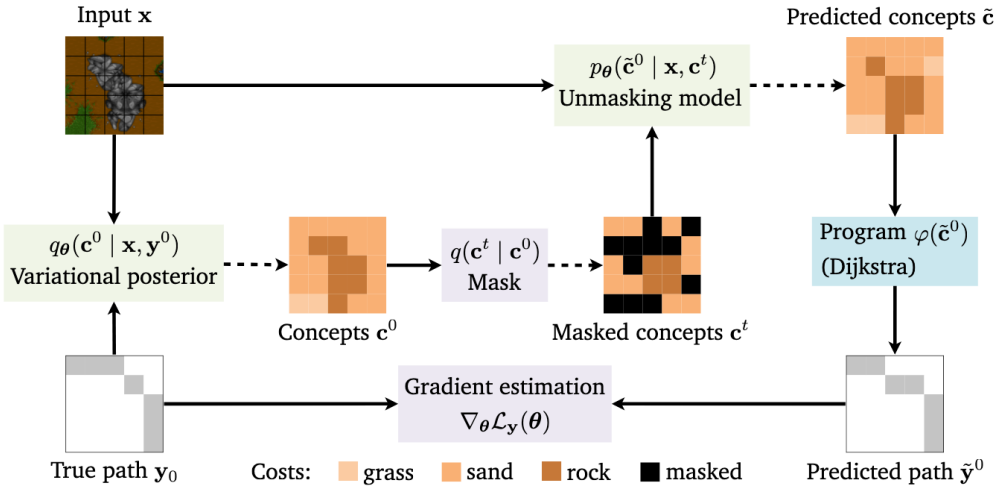
3. Projected diffusion



- Formulate the reverse diffusion process as **constrained optimization** over a feasible region C.
- At each step apply a projection operator $P_C(x) = \arg\min_{y \in C} \|y - x\|_2^2$.

Refs: Fioretto (2024), Christopher, Baek, and Fioretto (2024), Christopher et al. (2025)

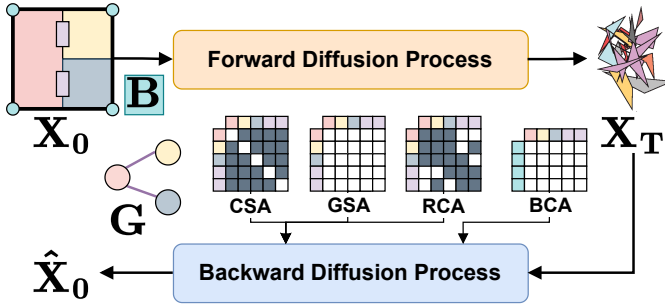
4. Neurosymbolic diffusion



Refs: van Krieken, Tomczak, and Ten Teije (2025)

Remember our example?

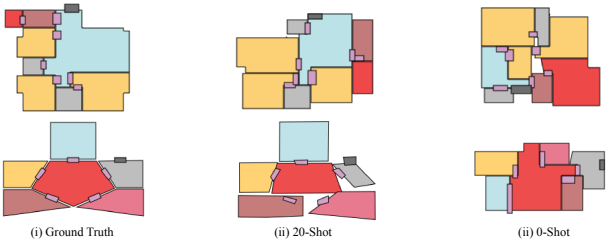
- Diffusion is over **2D corner coordinates**, not pixels.
- **Graph** and **boundary constraints** are injected into latent corner embeddings via specialized cross-attention modules.
- The model is then trained with **classifier-free guidance**.



Refs: Stoppani, Bacciu, and Mokarizadeh (2026)

Low-data regimes

- Real applications often provide only a **small dataset**.
- Ideally, a pretrained model should adapt with **few-shot** learning while retaining validity.
- In practice, it handles small shifts but fails on **out-of-distribution** cases.



Validity vs. Diversity

Steps (10^3)	FID (↓)	GC (↓)	BC (↓)	DS (↑)
400	10.74	**1.73±1.42**	**0.04±0.028**	33.59±10.10
300	**10.71**	1.88±1.44	0.04±0.028	35.49±8.94
200	10.77	1.89±1.42	0.05±0.028	34.28±8.84
100	11.38	2.10±1.48	0.06±0.032	37.84±5.75
50	14.14	2.57±1.73	0.08±0.033	40.27±5.38
30	16.33	3.17±1.71	0.10±0.035	42.95±4.56
20	18.26	3.41±1.83	0.11±0.034	48.40±6.76
10	25.98	3.95±1.77	1.00±0.000	**52.69±5.91**

Table 2: Comparison of FID, GC, BC and DS across different training steps. The best result for each metric is highlighted in bold.

- Generative models are often evaluated on realism, e.g. **FID**.
- Better constraint satisfaction often comes **at the cost of diversity**.
- In low-data settings, soft guidance may **overfit a few valid combinations**.

Conclusions

Wrapping up

Diffusion Models give a flexible generative process
We have seen how we can train and sample from a diffusion model to generate images.

Classifier-free guidance is a powerful tool
It allowed us to inject boundary constraint into the floorplan generation process.

Projection and neurosymbolic approaches can further improve validity
They can be used to enforce hard constraints, but they may be expensive or require domain-specific design.

What are we still missing?

- Guidance and penalty terms remain **soft**: invalid samples can still survive sampling.
- In few-shot and OOD settings, the model may **memorize a few feasible patterns** instead of learning the real feasible set.
- Hard validity guarantees are expensive and can reduce sample quality.

Open question

How can we obtain **hard validity guarantees** while preserving efficiency and sample quality of diffusion models?

References I

Binyamin, L., Tewel, Y., Segev, H., Hirsch, E., Rassin, R., & Chechik, G.. (2025). Make it count: Text-to-image generation with an accurate number of objects. *Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 13242–13251*.

Christopher, J. K., Baek, S., & Fioretto, F.. (2024). Constrained synthesis with projected diffusion models. *NeurIPS*.

Christopher, J. K., Cardei, M., Liang, J., & Fioretto, F.. (2025). Neuro-symbolic generative diffusion models for physically grounded, robust, and safe generation. *NeuS*.

Dhariwal, P., & Nichol, A.. (2021). Diffusion models beat GANs on image synthesis. *NeurIPS*.

Gupta, S., Samaras, D., & Chen, C.. (2025). TopoDiffusionNet: A topology-aware diffusion model. *ICLR*.

Ho, J., Jain, A., & Abbeel, P.. (2020). Denoising diffusion probabilistic models. *NeurIPS*.

Ho, J., & Salimans, T.. (2022). Classifier-free diffusion guidance. *arXiv*.

Nauata, N., Hosseini, S., Chang, K.-H., Chu, H., Cheng, C.-Y., & Furukawa, Y.. (2021). House-GAN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. *CVPR*.

References II

Nichol, A., & Dhariwal, P.. (2021). Improved denoising diffusion probabilistic models. *ICML*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B.. (2022). High-resolution image synthesis with latent diffusion models. *CVPR*.

Shabani, M. A., Hosseini, S. A., & Furukawa, Y.. (2023). HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. *CVPR*.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S.. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*.

Song, J., Meng, C., & Ermon, S.. (2021). Denoising diffusion implicit models. *ICLR*.

Stoppani, L., Bacciu, D., & Mokarizadeh, S.. (2026). Boundary-constrained diffusion models for floorplan generation: Balancing realism and diversity. *ESANN*.

Zhang, L., Rao, A., & Agrawala, M.. (2023). Adding conditional control to text-to-image diffusion models. *Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847*.

Thank you

Questions?