Controllable Generation

Lecture 06

Qiang Sun

Unconditional Generation

Recall: So far we have focused on **unconditional** generation.

Problem: Sample from p_{data}

Train: Use e.g., the conditional flow matching objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\square} \| u_t^\theta(x) - u_t^{\text{target}}(x|z) \|^2$$
$$\square = z \sim p_{\text{data}},\, t \sim \text{Unif}[0,1),\, x \sim p_t(x|z)$$

Sample: Simulate the corresponding ODE (or SDE):

$$\mathrm{d}X_t = u_t^\theta(X_t)\mathrm{d}t, \qquad X_0 \sim p_{\text{init}}$$

But what about **conditional generation?**

Guidance

Unconditional Generation

A swamp ogre with a pearl earring by Johannes Vermeer

A car made out of vegetables.

heat death of the universe, line art

Image source: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis [1]

Unconditional: “Generate an image.”

Conditional: “Generate an image of a cat baking a cake.”

Unconditional Generation

A swamp ogre with a pearl earring by Johannes Vermeer

A car made out of vegetables.

heat death of the universe, line art

Image source: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis [1]

~~**Unconditional**~~ **Unguided**: “Generate an image.”

~~**Conditional**~~ **Guided:** “Generate an image of a cat baking a cake.”

Guided (conditional) generation

Unguided models: sample from $p_{\text{data}}(x)$.

Guided generation: sample from

$$x \sim p_{\text{data}}(\cdot \mid y),$$

where $y \in \mathcal{Y}$ is additional information:

- class label (discrete), e.g. MNIST: $\mathcal{Y} = \{0, 1, \ldots, 9\}$
- text prompt embedding (continuous), e.g. $\mathcal{Y} = \mathbb{R}^{d_y}$

Terminology: we use *guided* for conditioning on y (to avoid confusion with conditioning on $z \sim p_{\text{data}}$).

Key idea: Guided diffusion model

Key Idea (Guided Generative Model)

A guided diffusion model consists of:

$$
\begin{aligned}
u^\theta : &\mathbb{R}^d \times \mathcal{Y} \times [0,1] \to \mathbb{R}^d, \\
&(x, y, t) \mapsto u_t^\theta(x \mid y), \\
&\sigma : [0,1] \to [0,1), \quad t \mapsto \sigma_t.
\end{aligned}
$$

Sampling for fixed y:

- **Init:** $X_0 \sim p_{\text{init}}$
- **Simulate:** $\mathrm{d}X_t = u_t^\theta(X_t \mid y)\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t$
- **Goal:** $X_1 \sim p_{\text{data}}(\cdot \mid y)$

If $\sigma_t \equiv 0$, this is a **guided flow model**.

Guided Generation: What Changes?

Unguided		Guided	
Marginal probability path	$p_t(x)$	**Guided** marginal probability path	$p_t(x\|y)$
Marginal vector field	$u_t^{\text{target}}(x)$	**Guided** marginal vector field	$u_t^{\text{target}}(x\|y)$
Marginal score	$\nabla \log p_t(x)$	**Guided** marginal score	$\nabla \log p_t(x\|y)$
Model	$u_t^\theta(x)$	**Guided** model	$u_t^\theta(x\|y)$
CFM Objective	$\mathcal{L}_{\text{CFM}}(\theta)$	**Guided** CFM Objective	**???**

Guidance for flow models: Guided CFM objective

Observation: **Fix** y and treat the target distribution as $p_{\text{data}}(x \mid y)$. Then we can use **conditional flow matching** loss:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\square} \left\| u_t^{\theta}(x \mid y) - u_t^{\text{target}}(x \mid z) \right\|^2$$
$$\square = z \sim p_{\text{data}}(z|y),\ t \sim \text{Unif}[0,1),\ x \sim p_t(\cdot \mid z).$$

Guidance for flow models: Guided CFM objective

Observation: **Fix** y and treat the target distribution as $p_{\text{data}}(x \mid y)$. Then we can use **conditional flow matching** loss:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\square} \left\| u_t^{\theta}(x \mid y) - u_t^{\text{target}}(x \mid z) \right\|^2$$
$$\square = z \sim p_{\text{data}}(z|y),\ t \sim \text{Unif}[0,1),\ x \sim p_t(\cdot \mid z).$$

Observation: Averaging over $(z, y) \sim p_{\text{data}}(z, y)$ and $t \sim \text{Unif}[0,1)$ gives the **guided conditional flow matching** loss:

$$\mathcal{L}_{\text{CFM}}^{\text{guided}}(\theta) = \mathbb{E}_{\square} \left\| u_t^{\theta}(x \mid y) - u_t^{\text{target}}(x \mid z) \right\|^2,$$
$$\square = (z, y) \sim p_{\text{data}}(z, y),\ t \sim \text{Unif}[0,1),\ x \sim p_t(\cdot \mid z).$$

Guided Sampling

Algorithm 7 Guided Sampling Procedure

Require: A trained guided vector field $u_t^\theta(x|y)$.

1: Select a prompt $y \in \mathcal{Y}$, such as "a cat baking a cake".
2: Initialize $X_0 \sim p_{\text{init}}$.
3: Simulate $\mathrm{d}X_t = u_t^\theta(X_t|y)\mathrm{d}t$ from $t = 0$ to $t = 1$.

Can we do better? At least empirically, the answer is yes…

Classifier-Free Guidance (CFG)

Motivation: why CFG?

Empirically, purely guided training can yield samples that do not adhere strongly enough to y.

- Improve perceptual alignment by **amplifying** guidance at inference
- CFG is widely used in modern diffusion/flow-based image models

We illustrate intuition using Gaussian probability paths.

Gaussian probability paths (recall)

Gaussian conditional probability path:

$$p_t(\cdot \mid z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d),$$

where α_t, β_t are smooth monotone schedulers with $\alpha_0 = \beta_1 = 0$ and $\alpha_1 = \beta_0 = 1$.

We will connect guided vector fields to guided scores $\nabla \log p_t(x \mid y)$.

Recall the conversion formula from Lecture 5

Proposition (Conversion formula for Gaussian probability paths)

For $p_t(\cdot \mid z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$,

$$u_t^{\text{target}}(x \mid z) = \left(\frac{\beta_t^2 \dot{\alpha}_t}{\alpha_t} - \beta_t \dot{\beta}_t \right) \nabla \log p_t(x \mid z) + \frac{\dot{\alpha}_t}{\alpha_t} x,$$

$$u_t^{\text{target}}(x) = \left(\frac{\beta_t^2 \dot{\alpha}_t}{\alpha_t} - \beta_t \dot{\beta}_t \right) \nabla \log p_t(x) + \frac{\dot{\alpha}_t}{\alpha_t} x.$$

The marginal formula corresponds to the **probability flow ODE** *viewpoint.*

Guided vector field via guided score

Using the Gaussian conversion form, write the guided vector field as

$$u_t^{\text{target}}(x \mid y) = a_t x + b_t \nabla \log p_t(x \mid y),$$

where

$$(a_t, b_t) = \left(\frac{\dot{\alpha}_t}{\alpha_t}, \frac{\dot{\alpha}_t \beta_t^2 - \dot{\beta}_t \beta_t \alpha_t}{\alpha_t} \right).$$

Bayes rule (gradient w.r.t. x):

$$\nabla \log p_t(x \mid y) = \nabla \log p_t(x) + \nabla \log p_t(y \mid x).$$

So

$$u_t^{\text{target}}(x \mid y) = u_t^{\text{target}}(x) + b_t \nabla \log p_t(y \mid x).$$

Classifier(-free) guidance scale

Heuristic: amplify the "classifier" term using guidance scale $w > 1$:

$$\tilde{u}_t(x \mid y) = u_t^{\text{target}}(x) + w\, b_t\, \nabla \log p_t(y \mid x).$$

Equivalently,

$$\tilde{u}_t(x \mid y) = (1 - w)\, u_t^{\text{target}}(x) + w\, u_t^{\text{target}}(x \mid y).$$

Key trick: learn both $u_t(x \mid y)$ and $u_t(x)$ in *one model* by introducing a "null" label $\varnothing$ and treating $u_t(x) = u_t(x \mid \varnothing)$.

Classifier guidance vs. Classifier-free guidance

$\log p_t(y \mid x)$ can be viewed as a classifier on noised data. Early diffusion works used an explicit trained classifier: **classifier guidance**.

Classifier-free guidance (CFG) avoids training a separate classifier by using a single conditional model with label dropping.

CFG training for flow models

Observation: We may treat the unguided vector field as conditioned on nothing. But, nothing is something:

$$u_t^{\text{target}}(x) = u_t^{\text{target}}(x \mid \varnothing).$$

CFG-CFM objective: We can train a single model $u_t^\theta(\cdot \mid y),\ y \in \{\mathcal{Y}, \varnothing\}$ by re-using $\mathcal{L}_{\text{CFM}}^{\text{guided}}(\theta)$ and occasionally setting $y = \varnothing$:

$$\mathcal{L}_{\text{CFM}}^{\text{CFG}}(\theta) = \mathbb{E}_{\square}\left\| u_t^\theta(x \mid y) - u_t^{\text{target}}(x \mid z)\right\|^2,$$

where sampling $\square$ is:

$$(z, y) \sim p_{\text{data}}(z, y), \quad \text{replace } y = \varnothing \text{ with prob. } \eta.$$
$$t \sim \text{Unif}[0, 1), \quad x \sim p_t(\cdot \mid z).$$

CFG for flow models

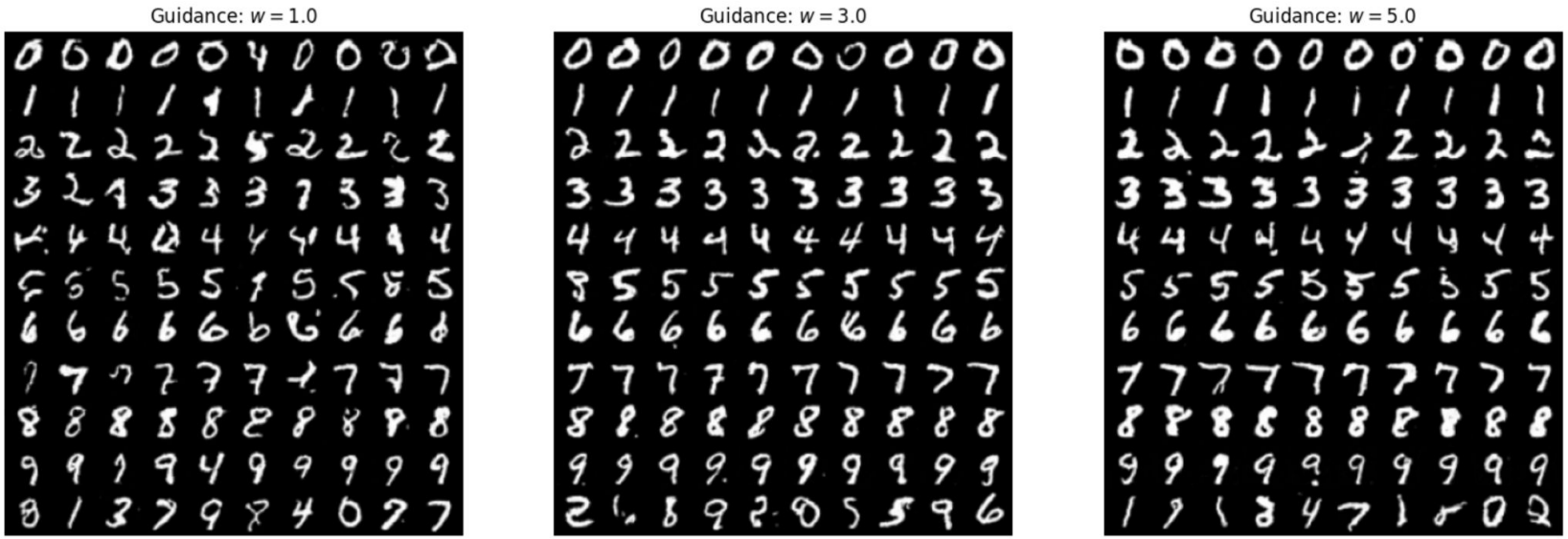
Summary (Classifier-Free Guidance for Flow Models)

Guidance scale $w > 1$:

$$\tilde{u}_t(x \mid y) = (1 - w)\, u_t(x \mid \varnothing) + w\, u_t(x \mid y).$$

Train a single network $u_t^\theta(\cdot \mid y)$ with label dropping (η) using CFG-CFM. At inference: simulate ODE with $\tilde{u}_t^\theta(\cdot \mid y)$ to increase adherence to y.

Figure: CFG on MNIST



Algorithm 8 Classifier-Free Guidance Sampling Procedure

Require: A trained guided vector field $u_t^\theta(x|y)$.

1: Select a prompt $y \in \mathcal{Y}$, or take $y = \varnothing$ for unguided sampling.
2: Select a **guidance scale** $w > 1$.
3: Initialize $X_0 \sim p_{\text{init}}$.
4: Simulate $\mathrm{d}X_t = \left[(1-w)u_t^\theta(X_t|\varnothing) + w u_t^\theta(X_t|y)\right] \mathrm{d}t$ from $t = 0$ to $t = 1$.

Figure: CFG effect on Corgi generation

Figure 1: Effect of classifier-free guidance: $w = 1$ vs $w > 1$ (from (Ho and Salimans, 2022)).

Guidance for Diffusion Models

CFG for diffusions

Summary (Classifier-Free Guidance for Diffusions)

Define

$$\tilde{s}_t^\theta(x \mid y) = (1-w)s_t^\theta(x \mid \varnothing) + w\, s_t^\theta(x \mid y),$$
$$\tilde{u}_t^\theta(x \mid y) = (1-w)u_t^\theta(x \mid \varnothing) + w\, u_t^\theta(x \mid y).$$

Sample by simulating the SDE:

$$\mathrm{d}X_t = \left[\tilde{u}_t^\theta(X_t \mid y) + \tfrac{\sigma_t^2}{2}\tilde{s}_t^\theta(X_t \mid y)\right]\mathrm{d}t + \sigma_t\, \mathrm{d}W_t.$$

Neural Network Architectures

Architecture requirements

We need a neural network for a guided vector field:

$$u^\theta : \mathbb{R}^d \times \mathcal{Y} \times [0,1] \to \mathbb{R}^d, \quad (x, y, t) \mapsto u_t^\theta(x \mid y).$$

- Low-dimensional toy data: an MLP on concatenated (x, y, t) can work.
- Images/videos: use specialized architectures in high dimensions (as MLP is in sufficient in this case).

Two common choices:

- **U-Net** (Ronneberger et al., 2015)
- **Diffusion Transformer (DiT)** (Peebles and Xie, 2023)

Images as tensors

An image is a tensor

$$x \in \mathbb{R}^{C_{\text{image}} \times H \times W}.$$

Example: RGB images have $C_{\text{image}} = 3$.

Videos add time:

$$x \in \mathbb{R}^{T \times C \times H \times W}.$$

U-Net intuition

U-Net: encoder $\rightarrow$ bottleneck $\rightarrow$ decoder, with skip/residual connections. For example, processing $x_t \in \mathbb{R}^{3\times256\times256}$:

- $x_t^{\text{input}} \in \mathbb{R}^{3\times256\times256}$
- $x_t^{\text{latent}} = E(x_t^{\text{input}}) \in \mathbb{R}^{512\times32\times32}$
- midcoder $M(\cdot)$: keeps latent size
- $x_t^{\text{output}} = D(x_t^{\text{latent}}) \in \mathbb{R}^{3\times256\times256}$

Common enhancements: attention layers, residual blocks.

Simplified U-Net architecture

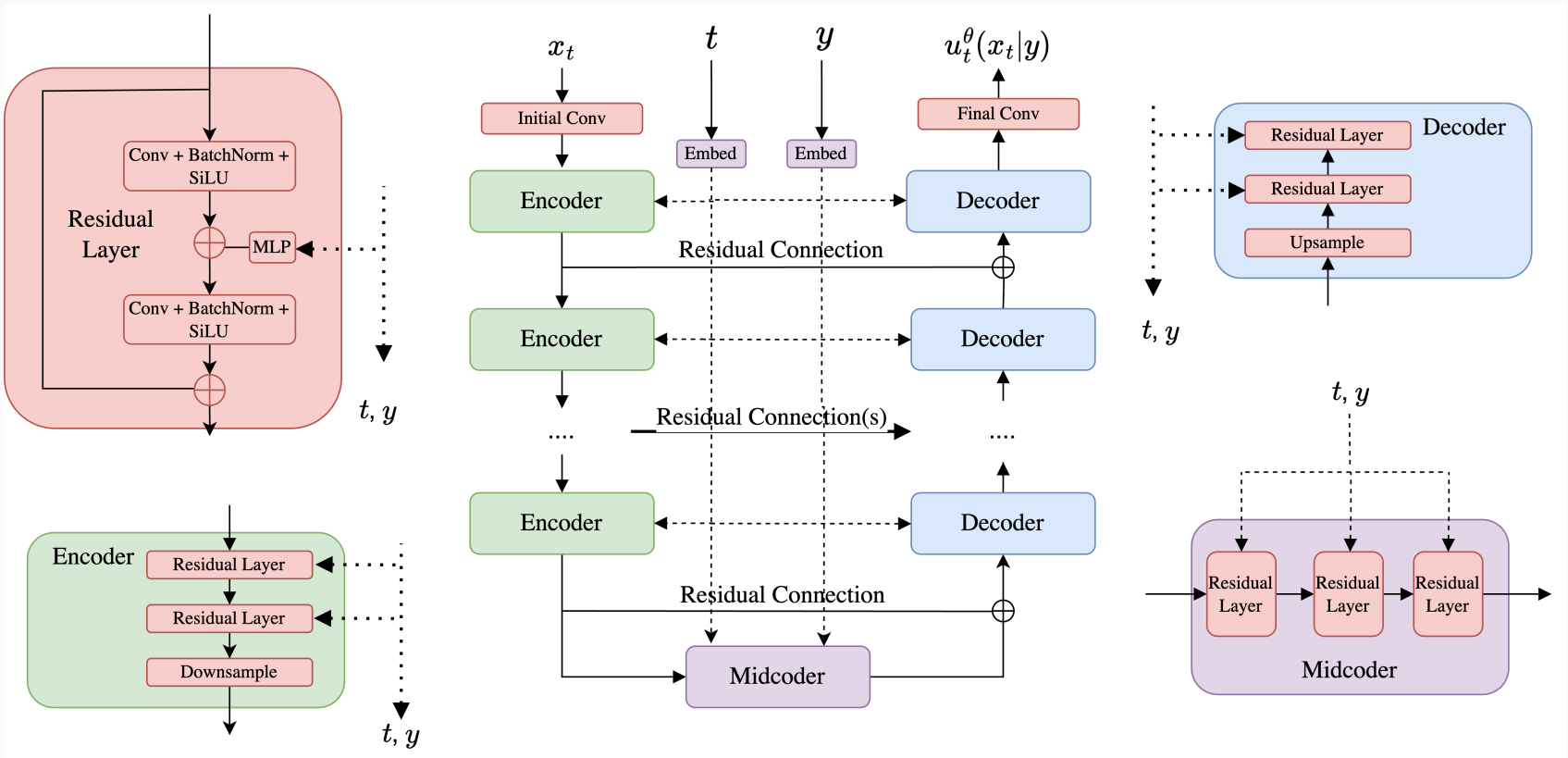


Figure 2: Simplified U-Net architecture (see also the lecture notes).

Diffusion Transformers (DiT)

DiT replaces convolutions with attention:

- Patchify image into tokens
- Embed tokens and apply transformer blocks (self-attention)
- Condition on y (and time t) via modulation / cross-attention

DiT builds on ViT (Dosovitskiy et al., 2020) and attention (Vaswani et al., 2017). Stable Diffusion 3 uses a modified, multi-modal DiT backbone (Esser et al., 2024).

DiT + CLIP

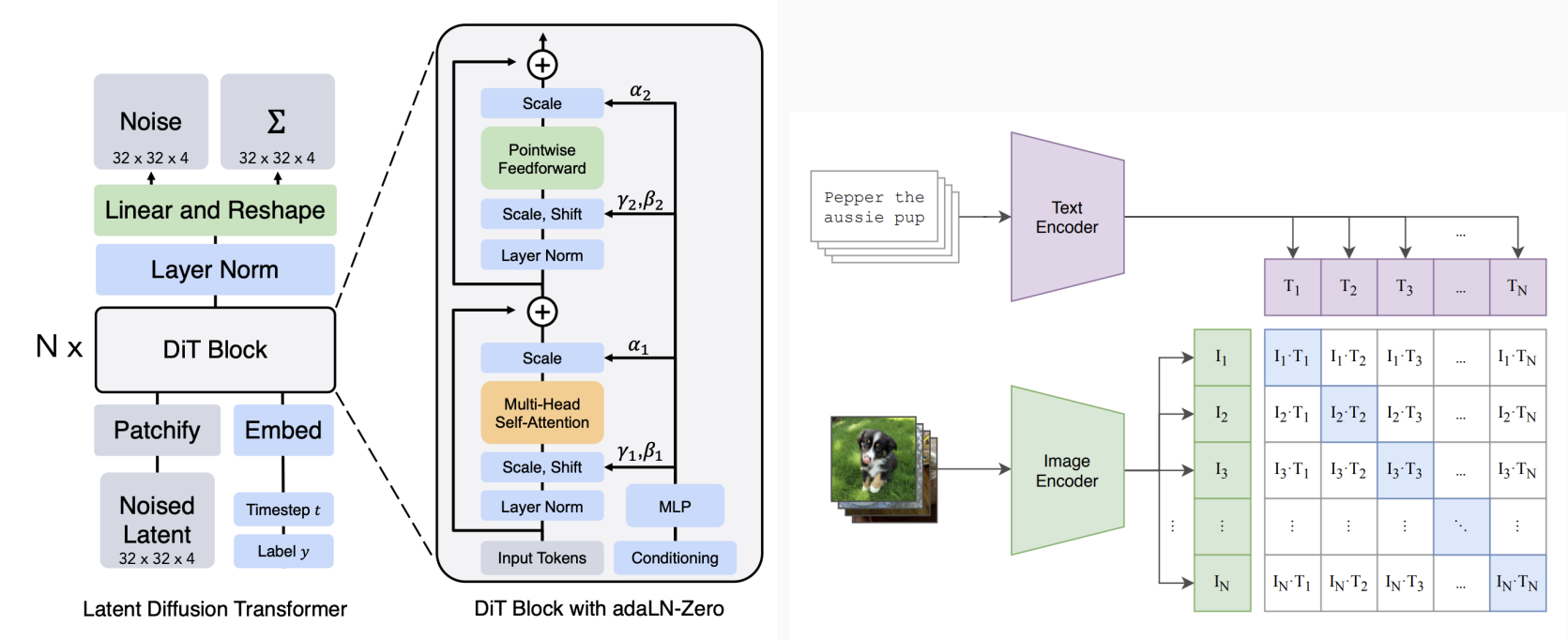


Figure 3: Left: diffusion transformer overview (Esser et al., 2024). Right: CLIP contrastive loss (Radford et al., 2021).

Working in latent space

High-dimensional data can be memory-heavy (e.g. megapixel images or long videos). A common pattern is to train in the latent space of a (variational) autoencoder:

- encode data into a compressed latent representation
- train flow/diffusion model on latents
- sample in latent space, then decode to pixel space

This is the basis of **latent diffusion models** (Rombach et al., 2022; Vahdat et al., 2021).

Encoding the guiding variable y

Two steps:

- Embed raw input y_{raw} into a vector (or sequence) y
- Inject y into the model at multiple layers

Cases:

- class labels: learned embedding table
- text prompts: frozen pretrained text encoders (e.g. CLIP (Radford et al., 2021), T5 (Raffel et al., 2020))

Feeding the embedding into a U-Net (example)

Inject $y \in \mathbb{R}^{d_y}$ into an intermediate activation $x_t^{\text{intermediate}} \in \mathbb{R}^{C \times H \times W}$:

$$y \leftarrow \text{MLP}(y) \in \mathbb{R}^{C}, \quad y \leftarrow \text{reshape}(y) \in \mathbb{R}^{C \times 1 \times 1}, \quad x_t^{\text{intermediate}} \leftarrow x_t^{\text{inte}}$$

If y is a *sequence* (token embeddings), use cross-attention between image tokens and text tokens.

Survey of Large-Scale Models

Survey: Stable Diffusion 3 and Movie Gen Video

We briefly examine:

- Stable Diffusion 3 (image generation) (Esser et al., 2024)
- Meta Movie Gen Video (video generation) (Polyak et al., 2024)

Common themes:

- conditional flow matching + CFG training
- latent space generation via pretrained autoencoders
- transformer backbones with rich text conditioning

Stable Diffusion 3: key points

Stable Diffusion 3:

- uses conditional flow matching (and CFG training with label dropping)
- trains in latent space of a pretrained autoencoder (Rombach et al., 2022)
- uses multiple text embeddings (e.g. CLIP + T5 encoder outputs)
- proposes a multi-modal DiT (MM-DiT) with attention to both image patches and text tokens
- largest model: **8B parameters**
- sampling: ~ 50 Euler steps; guidance $w \approx 2$–5

MM-DiT

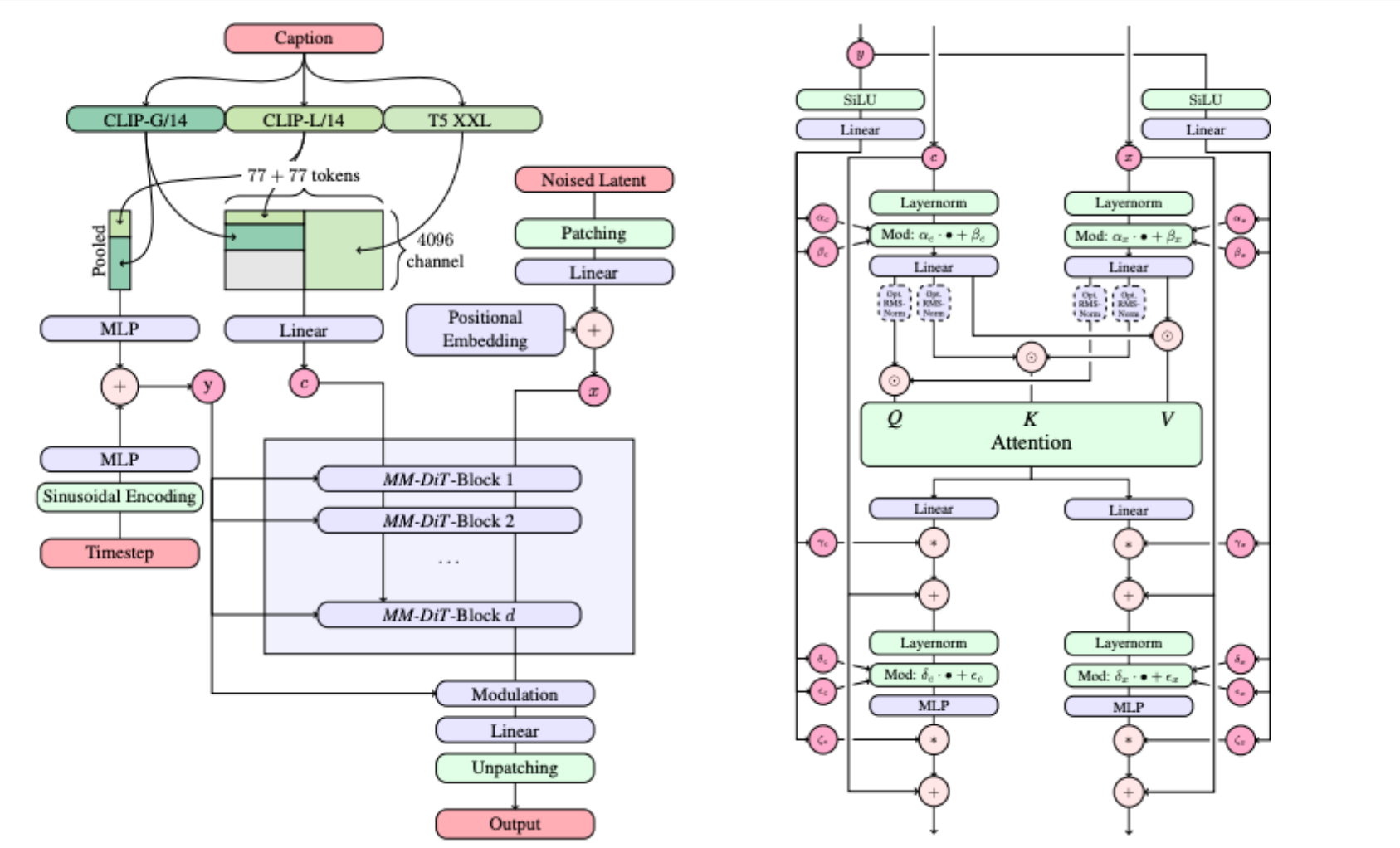


Figure 4: Multi-modal DiT (MM-DiT) architecture from (Esser et al., 2024).

Meta Movie Gen Video: key points

Movie Gen Video:

- data: $x \in \mathbb{R}^{T \times C \times H \times W}$ (video)
- conditional flow matching with CondOT path + CFG
- trains in latent space of a frozen pretrained **temporal autoencoder** (TAE)
- downsampling ratios: $\frac{T'}{T} = \frac{H'}{H} = \frac{W'}{W} = 8$
- transformer backbone with spatiotemporal patchification + self/cross-attention
- text conditioning: UL2 (Tay et al., 2022), ByT5 (Xue et al., 2022), MetaCLIP (Lavoie et al., 2024)
- largest model: **30B parameters** (Polyak et al., 2024)

Takeaways

- Guidance targets $p_{\text{data}}(x \mid y)$ via guided $u_t^\theta(\cdot \mid y)$ / $s_t^\theta(\cdot \mid y)$
- CFG strengthens conditioning at inference:

$$\tilde{u}_t = (1-w)u_t(\cdot \mid \varnothing) + w u_t(\cdot \mid y), \quad \tilde{s}_t = (1-w)s_t(\cdot \mid \varnothing) + w s_t(\cdot \mid y)$$

- Modern image/video models use latent autoencoders + transformer backbones + rich text embeddings

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

References ii

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. (2024). Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Lavoie, S., Kirichenko, P., Ibrahim, M., Assran, M., Wilson, A. G., Courville, A., and Ballas, N. (2024). Modeling caption diversity in contrastive vision-language pretraining. *arXiv preprint arXiv:2405.00740*.

References iii

Peebles, W. and Xie, S. (2023). Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205.

Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.-Y., Chuang, C.-Y., et al. (2024). Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

References iv

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

References

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., et al. (2022). Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Vahdat, A., Kreis, K., and Kautz, J. (2021). Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

References vi

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. (2022). Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.