Generative model paper discussion (By Alex)

Theory

Deep Image Prior (CVPR18, Dmitry Ulyanov)

- Inductive bias: a great deal of image statistics are captured by the structure of a convolutional image generator independent of learning.
- We now show that, while indeed almost any image can be fitted, the choice of network architecture has a major effect on how the solution space is searched by methods such as gradient descent.
- The parametrization offers high impedance to noise and low impedance to signal.
- Application: 1.Denoising and generic reconstruction 2.Super-resolution 3.Inpainting 4.Natural pre-image 5.Flash-no flash reconstruction

Flow based model

Variational Inference with Normalizing Flows (ICML15, Google, Shakir Mohamed)

- We study deep latent Gaussian models (DLGM), which are a general class of deep directed graphical models that consist of a hierarchy of L layers of Gaussian latent variables z_l for layer l.

$$p(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_L) = p\left(\mathbf{x}|f_0(\mathbf{z}_1)\right) \prod_{l=1}^{L} p\left(\mathbf{z}_l|f_l(\mathbf{z}_{l+1})\right) \quad (4)$$

- Reparameterization $z \sim \mathcal{N}(z|\mu, \sigma^2) \Leftrightarrow z = \mu + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0,1)$ and Montecarlo

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}[f_\theta(z)] \Leftrightarrow \mathbb{E}_{\mathcal{N}(\epsilon|0,1)}[\nabla_\phi f_\theta(\mu + \sigma\epsilon)] .$$

- It is natural to consider the case in which the length of the normalizing flow tends to infinity.

$$\mathbf{z}_K = f_K \circ \ldots \circ f_2 \circ f_1(\mathbf{z}_0) \quad (6)$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \det \left| \frac{\partial f_k}{\partial \mathbf{z}_k} \right|, \quad (7)$$

From into:

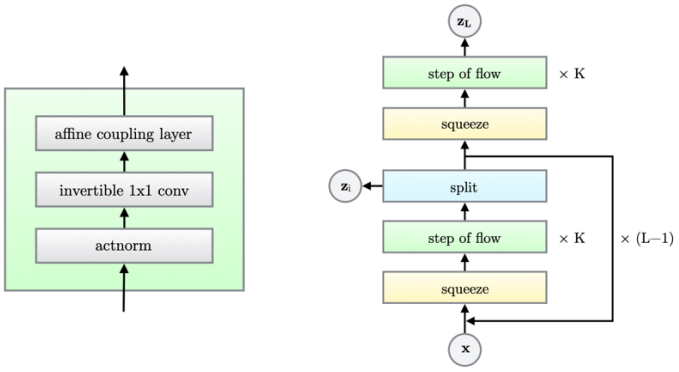
$$d\mathbf{z}(t) = \mathbf{F}(\mathbf{z}(t), t)dt + \mathbf{G}(\mathbf{z}(t), t)d\boldsymbol{\xi}(t), \quad (9)$$

Glow: Generative Flow with Invertible 1×1 Convolutions (NIPS18, OpenAI, Google AI)

- Two major unsolved problems in the field of machine learning are (1) data-efficiency: the ability to learn from few datapoints, like humans; and (2) generalization: robustness to changes of the task or its context.
- A promise of *generative models*, a major branch of machine learning, is to overcome these limitations by: (1) learning realistic world models, potentially allowing agents to plan in a

world model before actual interaction with the world, and (2) learning meaningful features of the input while requiring little or no human supervision or labeling. Since such features can be learned from large unlabeled datasets and are not necessarily task-specific, downstream solutions based on those features could potentially be more robust and more data efficient.

- Merits: (1) Exact latent-variable inference and log-likelihood evaluation(not a lower bound) (2) Efficient (parallelize) (3) Useful latent space for downstream tasks: beat GANs and CNNs (4) Memory savings
- How is 1*1 convolution kernel working: it changes the depth of the data.



(a) One step of our flow. (b) Multi-scale architecture (Dinh et al., 2016).

Description	Function	Reverse Function	Log-determinant
Actnorm. See Section 3.1.	$\forall i,j : \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$	$\forall i,j : \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b})/\mathbf{s}$	$h \cdot w \cdot \texttt{sum}(\log \lvert\mathbf{s}\rvert)$
Invertible 1×1 convolution. $\mathbf{W} : [c \times c]$. See Section 3.2.	$\forall i,j : \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$\forall i,j : \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$	$h \cdot w \cdot \log \lvert\det(\mathbf{W})\rvert$ or $h \cdot w \cdot \texttt{sum}(\log \lvert\mathbf{s}\rvert)$ (see eq. (10))
Affine coupling layer. See Section 3.3 and (Dinh et al., 2014)	$\mathbf{x}_a, \mathbf{x}_b = \texttt{split}(\mathbf{x})$ $(\log \mathbf{s}, \mathbf{t}) = \texttt{NN}(\mathbf{x}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$ $\mathbf{y}_b = \mathbf{x}_b$ $\mathbf{y} = \texttt{concat}(\mathbf{y}_a, \mathbf{y}_b)$	$\mathbf{y}_a, \mathbf{y}_b = \texttt{split}(\mathbf{y})$ $(\log \mathbf{s}, \mathbf{t}) = \texttt{NN}(\mathbf{y}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t})/\mathbf{s}$ $\mathbf{x}_b = \mathbf{y}_b$ $\mathbf{x} = \texttt{concat}(\mathbf{x}_a, \mathbf{x}_b)$	$\texttt{sum}(\log(\lvert\mathbf{s}\rvert))$

Invertible Residual Networks (ICML19)

- One of the main appeals of neural network-based models is that a single model architecture can often be used to solve a variety of related tasks. Generative tasks – flow, discriminative learning – deep residual.

Autoregressive model

Pixel Recurrent Neural Networks (ICML16, Google)

- Recurrent Neural Networks (RNN) are powerful models that offer a compact, shared parametrization of a series of conditional distributions.
- Autoregressive model.

Improved Variational Inference with Inverse Autoregressive Flow (NIPS16, OpenAI)

- Scales well to high-dimensional latent spaces.

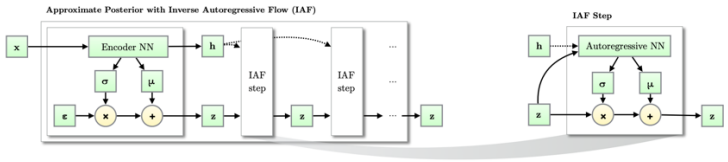


Figure 2: Like other normalizing flows, drawing samples from an approximate posterior with Inverse Autoregressive Flow (IAF) consists of an initial sample $\mathbf{z}$ drawn from a simple distribution, such as a Gaussian with diagonal covariance, followed by a chain of nonlinear invertible transformations of $\mathbf{z}$, each with a simple Jacobian determinants.

-

Language Model Beats Diffusion -- Tokenizer is Key to Visual Generation (ICLR24, Google, CMU)

- One crucial component is the visual tokenizer that maps pixel-space inputs to discrete tokens appropriate for LLM learning.
- Why do language models lag behind diffusion models in visual generation? This paper suggests that a primary reason is the lack of a good visual representation, resembling our natural language system, for effectively modeling the visual world.
- Merits: (1) Compatibility with LLMs (2) Compressed representation (3) Visual understanding benefits.
- The first evidence suggesting that a language model can outperform diffusion models on ImageNet when provided with the same training data, an equivalent model size, and a similar training budget.

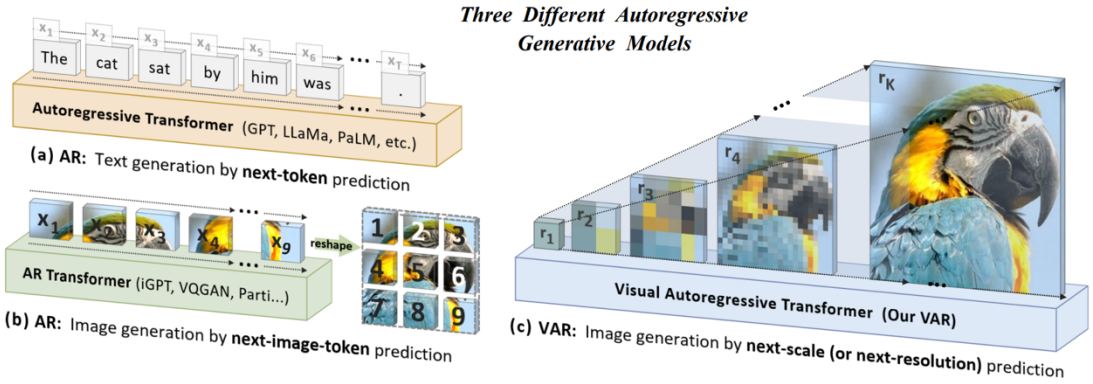
Type	Method	w/o guidance FID↓	w/o guidance IS↑	w/ guidance FID↓	w/ guidance IS↑	#Params	#Steps
GAN	StyleGAN-XL (Sauer et al., 2022)			2.41	267.8	168M	1
Diff. + VAE*	DiT-XL/2 (Peebles & Xie, 2022)	12.03	105.3	3.04	240.8	675M	250
Diffusion	ADM+Upsample (Dhariwal & Nichol, 2021)	9.96	121.8	3.85	221.7	731M	2000
Diffusion	RIN (Jabri et al., 2023)	3.95	216.0			320M	1000
Diffusion	simple diffusion (Hoogeboom et al., 2023)	3.54	205.3	3.02	248.7	2B	512
Diffusion	VDM++ (Kingma & Gao, 2023)	2.99	232.2	2.65	278.1	2B	512
MLM + VQ	MaskGIT (Chang et al., 2022)	7.32	156.0			227M	12
MLM + VQ	DPC+Upsample (Lezama et al., 2023)	3.62	249.4			619M	72
MLM + LFQ	*MAGVIT-v2 (this paper)*	4.61	192.4			307M	12
		3.07	213.1	**1.91**	**324.3**		64

Type	Method	K600 FVD↓	UCF FVD↓	#Params	#Steps
GAN	TrIVD-GAN-FP (Luc et al., 2020)	25.7±0.7			1
Diffusion	Video Diffusion (Ho et al., 2022c)	16.2±0.3		1.1B	256
Diffusion	RIN (Jabri et al., 2023)	10.8		411M	1000
AR-LM + VQ	TATS (Ge et al., 2022)		332±18	321M	1024
MLM + VQ	Phenaki (Villegas et al., 2022)	36.4±0.2		227M	48
MLM + VQ	MAGVIT (Yu et al., 2023a)	9.9±0.3	76±2	306M	12
MLM + LFQ	non-causal baseline	11.6±0.6		307M	12
MLM + LFQ	*MAGVIT-v2 (this paper)*	5.2±0.2		307M	12
		4.3±**0.1**	**58**±3		24

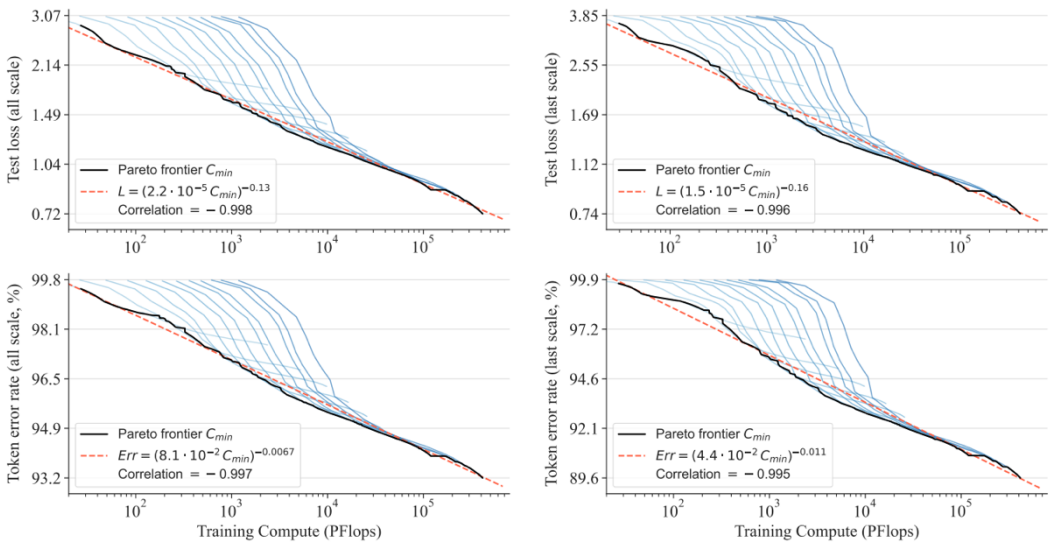
Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction (Keyu

Tian)

- Studies into the success of these large AR models have highlighted their scalability and generalizabilty: the former, as exemplified by scaling laws, allows us to predict large model's performance from smaller ones and thus guides better resource allocation, while the latter, as evidenced by zero-shot and few-shot learning, underscores the unsupervised-trained models' adaptability to diverse, unseen tasks. These properties reveal AR models' potential in learning from vast unlabeled data, encapsulating the essence of "AGI".



- Scaling laws.



An Image is Worth 32 Tokens for Reconstruction and Generation (ByteDance)

- Translating raw pixels into a latent space.
- TiTok provides a more compact latent representation, yielding substantially more efficient and effective representations than conventional techniques. For example, a 256 × 256 × 3 image can be reduced to just 32 discrete tokens, a significant reduction from the 256 or 1024 tokens obtained by prior methods.

Autoregressive Image Generation without Vector Quantization (Google, Kaiming He)

- Is it necessary for autoregressive models to be coupled with vector-quantized representations?
- Vector-quantized tokenizers are difficult to train and are sensitive to gradient approximation strategies. Their reconstruction quality often falls short compared to continuous-valued

counterparts. Our approach allows autoregressive models to enjoy the benefits of higher-quality, non-quantized tokenizers.

- Diffusion per token + autoregressive generation.
- NOVA achieves state-of-the-art text-to-image and text-to-video generation performance with significantly lower training costs and higher inference speed.

Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model (Meta)

- We introduce Transfusion, a recipe for training a model that can seamlessly generate discrete and continuous modalities.

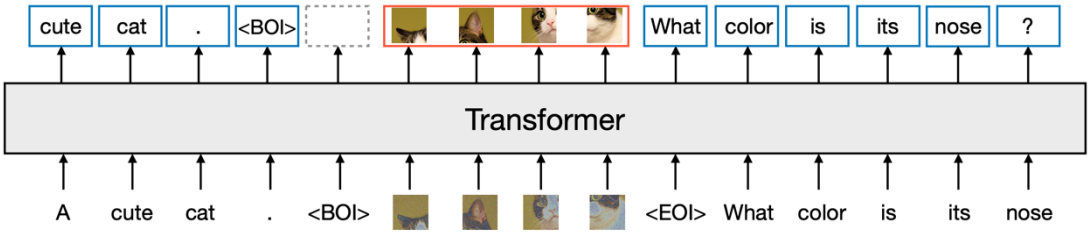


Figure 1: A high-level illustration of Transfusion. A single transformer perceives, processes, and produces data of every modality. Discrete (text) tokens are processed autoregressively and trained on the next token prediction objective. Continuous (image) vectors are processed together in parallel and trained on the diffusion objective. Marker BOI and EOI tokens separate the modalities.

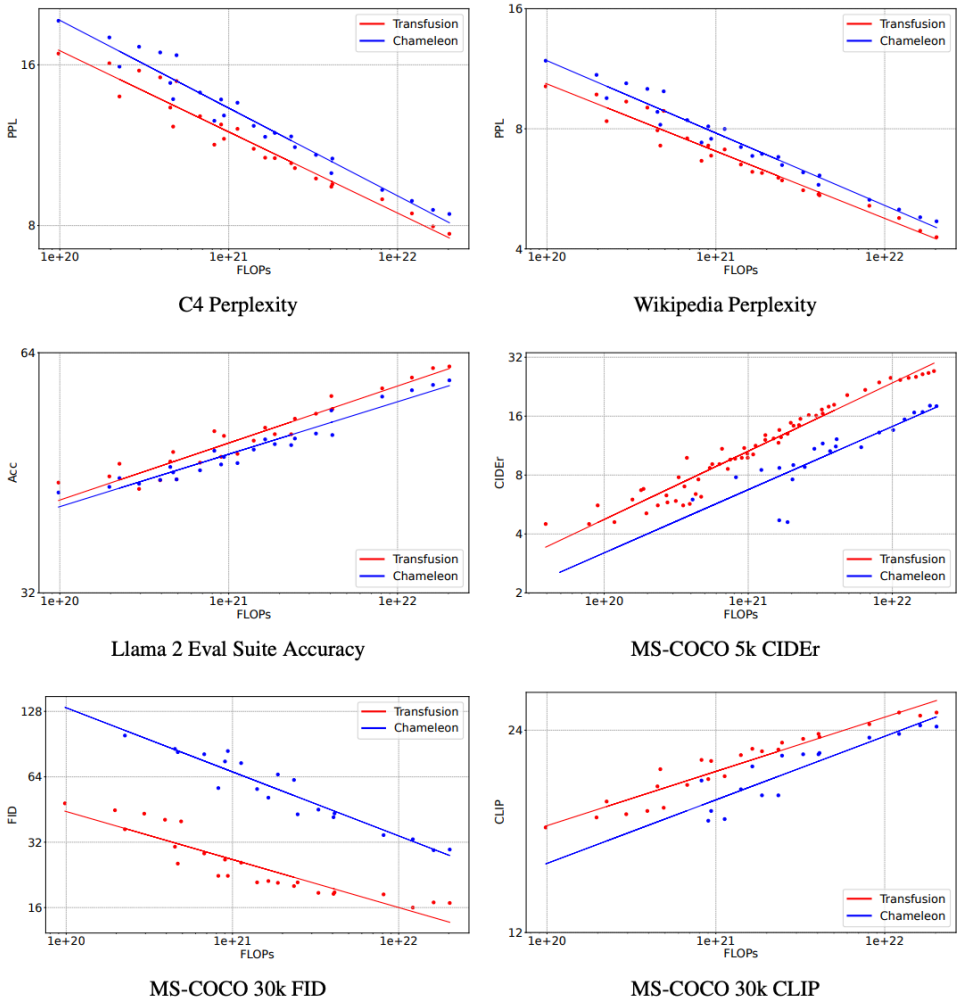
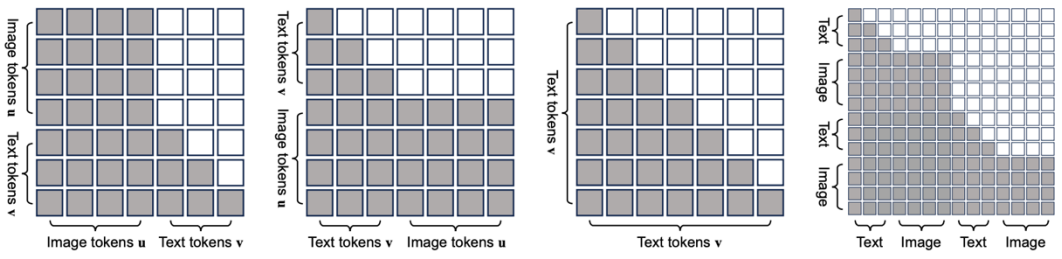


Figure 5: Performance of Transfusion and Chameleon models at different scales, controlled for parameters, data, and compute. All axes are logarithmic.



GAN in the era of diffusion

StyleGAN-T- Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis (ICML23, NVIDIA)

- Generative adversarial networks (GANs) only need a single forward pass -> fast.
- GAN in smaller and less diverse datasets.
- The key benefits of StyleGAN-T include its fast inference speed and smooth latent space interpolation in the context of text-to-image synthesis.

- Modify the architecture!

Scaling up GANs for Text-to-Image Synthesis (CVPR23, POSTECH)

- The now-dominant paradigms, diffusion models and autoregressive models, both rely on iterative inference.
- Can GANs continue to be scaled up and potentially benefit from such resources, or have they plateaued? What prevents them from further scaling, and can we overcome these barriers?

The GAN is dead; long live the GAN! A Modern Baseline GAN (ICML24, Nick Huang)

- We show that by introducing a new regularized training loss, GANs gain improved training stability.
- Our new loss allows us to discard all ad-hoc tricks and replace outdated backbones used in common GANs with modern architectures.

Diffusion Models

Classifier-Free Diffusion Guidance (NIPS21, Google)

- Classifier guidance combines the score estimate of a diffusion model with the gradient of an image classifier and thereby requires training an image classifier separate from the diffusion model. (Bayes)

  can train a classifier $p_\phi(y|x_t, t)$ on noisy images x_t, and then use gradients $\nabla_{x_t} \log p_\phi(y|x_t, t)$ to guide the diffusion sampling process towards an arbitrary class label y.
- Guidance can be indeed performed by a pure generative model without such a classifier. (Implicit bias) Classifier-free guidance instead mixes the score estimates of a conditional diffusion model and a jointly trained unconditional diffusion model.
- When training:

$\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond} ▷ Randomly discard conditioning to train unconditionally

When sampling:

$\tilde{\boldsymbol{\epsilon}}_t = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)$

▷ Sampling step (could be replaced by another sampler, e.g. DDIM)

Progressive Distillation for Fast Sampling of Diffusion Models (ICLR22, Google)

- Diffusion suffers from slow sampling time: generating high quality samples takes many hundreds or thousands of model evaluations.
- First, we present new parameterizations of diffusion models that provide increased stability when using few sampling steps. Second, we present a method to distill a trained deterministic diffusion sampler, using many steps, into a new diffusion model that takes half as many sampling steps.

Alternatively, Song et al. (2021c) show that our denoising model $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ can be used to deterministically map noise $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to samples $\mathbf{x}$ by numerically solving the *probability flow ODE*:

$$d\mathbf{z}_t = [f(\mathbf{z}_t, t) - \frac{1}{2} g^2(t) \nabla_z \log \hat{p}_\theta(\mathbf{z}_t)] dt, \tag{6}$$

where $\nabla_z \log \hat{p}_\theta(\mathbf{z}_t) = \frac{\alpha_t \hat{\mathbf{x}}_\theta(\mathbf{z}_t) - \mathbf{z}_t}{\sigma_t^2}$. Following Kingma et al. (2021), we have $f(\mathbf{z}_t, t) = \frac{d \log \alpha_t}{dt} \mathbf{z}_t$ and $g^2(t) = \frac{d\sigma_t^2}{dt} - 2\frac{d \log \alpha_t}{dt}\sigma_t^2$. Since $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ is parameterized by a neural network, this equation is a special case of a *neural ODE* (Chen et al., 2018), also called a *continuous normalizing flow* (Grathwohl et al., 2018).

-
- The error introduced by numerical integration of the probability flow ODE is guaranteed to vanish as the number of integration steps grows infinitely large. Here, we therefore propose a method to distill these accurate, but slow, ODE integrators into much faster models that are still very accurate.
- Key difference: put one step into two steps and double the stepsize.

$$L_\theta = \|\epsilon - \hat{\epsilon}_\theta(\mathbf{z}_t)\|_2^2 = \left\| \frac{1}{\sigma_t}(\mathbf{z}_t - \alpha_t \mathbf{x}) - \frac{1}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\theta(\mathbf{z}_t)) \right\|_2^2 = \frac{\alpha_t^2}{\sigma_t^2} \|\mathbf{x} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2,$$

Simple diffusion: End-to-end diffusion for high resolution images (ICML23, Google)

- To improve denoising diffusion for high resolution images while keeping the model as simple as possible.
- The noise schedule should be adjusted for high resolution images: we argue that for higher resolutions, this schedule can be changed in a predictable way to retain good visual sample quality.
- In detail, we introduce a method to improve the resolution. We split a color block into 2*2, and adjust the noise (The lower resolution pixel z_t(64*64) only has half the amount of noise).

$$z_t^{64\times64} = (z_t^{(1)} + z_t^{(2)} + z_t^{(3)} + z_t^{(4)})/4.$$

On the Importance of Noise Scheduling for Diffusion Models (Google)

- Empirically.
- The noise scheduling is crucial for the performance, and the optimal one depends on the task (e.g., image sizes).
- When increasing the image size, the optimal noise scheduling shifts towards a noisier one (due to increased redundancy in pixels)
- Simply scaling the input data by a factor of b while keeping the noise schedule function fixed (equivalent to shifting the logSNR by log b) is a good strategy across image sizes.

Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (NIPS23 Arpit Bansal)
(Nice Introduction! Read it carefully!)

- Even when using completely deterministic degradations (e.g., blur, masking, and more), the training and test-time update rules that underlie diffusion models can be easily generalized to create generative models.
- When we apply a sequence of updates at test time that alternate between the image restoration model and the image degradation operation, generative behavior emerges, and we obtain

photo-realistic images.

- Noise in the training process is critically thought to expand the support of the low-dimensional training distribution to a set of full measure in ambient space.
- Also act as data augmentation to improve score predictions in low density regions.
- Degradation: x_t = D(x_0, t).

Restoration: R(x_t, t) \approx x_0

Loss: $$\min_{\theta} \mathbb{E}_{x\sim\mathcal{X}} \|R_\theta(D(x,t),t) - x\|,$$

Algorithm 1 Naive Sampling

Input: A degraded sample x_t
for $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = D(\hat{x}_0, s-1)$
end for
Return: x_0

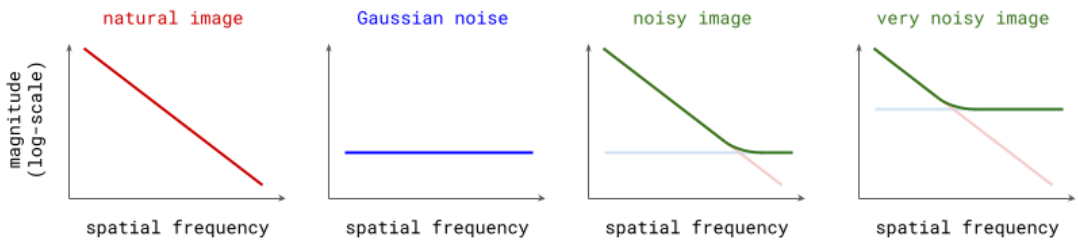
Algorithm 2 Improved Sampling for Cold Diffusion

Input: A degraded sample x_t
for $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = x_s - D(\hat{x}_0, s) + D(\hat{x}_0, s-1)$
end for

-

Diffusion is spectral autoregression (Sander Dieleman's blog)

- Autoregression does this by casting the data to be modelled into the shape of a sequence, and recursively predicting one sequence element at a time. Diffusion instead works by defining a corruption process that gradually destroys all structure in the data, and training a model to learn to invert this process step by step.
- Underlying iterative approach.
- Spectral analysis:



-

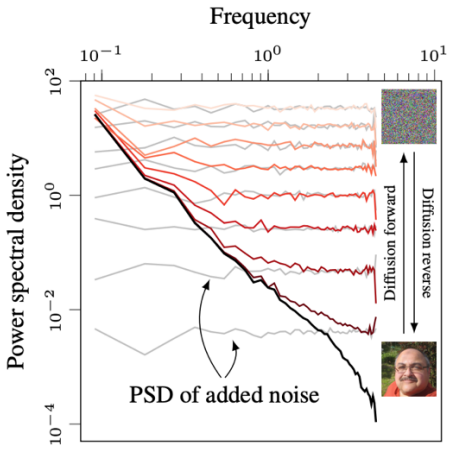


Figure 4: The $1/f^{\alpha}$ power spectral density in natural images induces an implicit coarse-to-fine inductive bias in diffusion models.

- Gradually filtering out more and more high-frequency information from the input image. This is a soft version of autoregression in frequency space, or if you want to make it sound fancier, approximate spectral autoregression.
- For individual images, the spectrum will not be a perfectly straight line, and it will not typically be monotonically decreasing.
- $$\mathcal{R}[\alpha(t)\mathbf{x}_0](f) > \tau \cdot \mathcal{R}[\sigma(t)\varepsilon](f).$$ noise of the f.

Going beyond images, one could use the same line of reasoning to try and understand why diffusion models haven't really caught on in the domain of language modelling so far. The interpretation in terms of a frequency decomposition is not really applicable there, and hence being able to change the relative weighting of noise levels in the loss doesn't quite have the same impact on the quality of generated outputs.

- Unstable equilibrium, because the future is multimodal.
- This flexibility also enables various distillation methods to reduce the number of steps required, and classifier-free guidance to improve sample quality.

Generative Modelling With Inverse Heat Dissipation (ICLR23, Severi Rissanen)

- We propose a new diffusion-like model that generates images through stochastically reversing the heat equation.

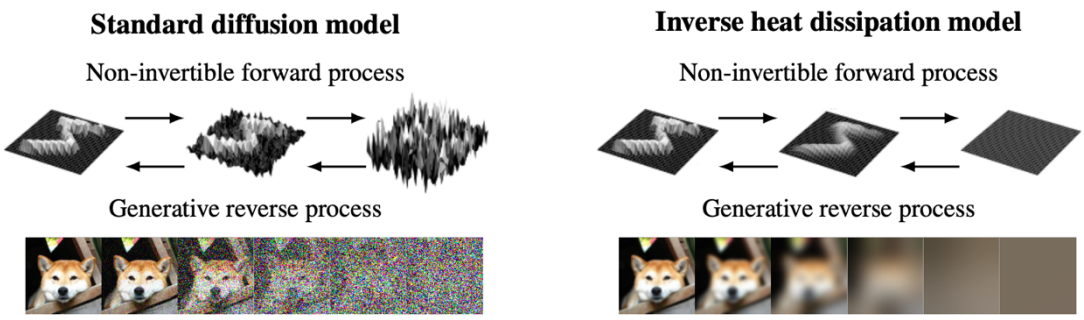


Figure 2: Comparison of generation by generative denoising and inverse heat diffusion, where the focus of the forward process is in the *pixel space* in the left and the *2D image plane* on the right.

- We write the forward equation!

Forward PDE model: $$\frac{\partial}{\partial t}u(x,y,t) = \Delta u(x,y,t),$$

And the solution is in the form:

The PDE model in Eq. (1) can be formally written in evolution equation form as $u(x,y,t) = \mathcal{F}(t)\, u(x,y,t)|_{t=t_0}$, where $\mathcal{F}(t) = \exp[(t-t_0)\,\Delta]$ is an evolution operator given in terms of the operator exponential function (see, *e.g.*, Da Prato & Zabczyk, 1992). We can use this general

- Parameterize the reverse process!
- Implicit bias: circularly symmetric and localized.

$$u(x,y,t-\mathrm{d}t) \approx u(x,y,t) - \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} * u(x,y,t)\,\mathrm{d}t.$$

Inversion by Direct Iteration- An Alternative to Denoising Diffusion for Image Restoration (TMLR23, Google)

- Recovering a high-quality image from a low-quality observation is a fundamental problem in computer vision and computational imaging.
- This evidently results in an image that is the (weighted) average of all plausible reconstructions.

$$\min_\theta \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|F_\theta(\boldsymbol{y}) - \boldsymbol{x}\|_p \approx \min_\theta \sum \|F_\theta(\boldsymbol{y}^i) - \boldsymbol{x}^i\|_p.$$

The solution is $$\boldsymbol{x}_{\mathrm{MMSE}}(\boldsymbol{y}) = \mathbb{E}[\,\boldsymbol{x}\,|\,\boldsymbol{y}\,] = \int \boldsymbol{x} p(\boldsymbol{x}\mid \boldsymbol{y}) d\boldsymbol{x}.$$

- In this work, we explicitly address this problem by avoiding single-step prediction of the clean image, and instead iterating a series of inferences, where at each step we solve an 'easier' (i.e., less ill-posed) inverse problem than the original.
- INDI: $$\hat{\boldsymbol{x}}_{t-\delta} = \frac{\delta}{t} F_\theta(\hat{\boldsymbol{x}}_t, t) + \left(1 - \frac{\delta}{t}\right)\hat{\boldsymbol{x}}_t,$$ predict the slightly less corrupted signal at time t - \delta.

Structured Denoising Diffusion Models in Discrete State-Spaces (NIPS21, Google)

- Diffusion models with discrete state spaces have been explored for text and image segmentation domains, but they have not yet been demonstrated as a competitive model class

for large scale text or image generation.

- The state space is a discrete space and we describe the distribution transition by a Markov chain.

Let us Build Bridges: Understanding and Extending Diffusion Generative Models (Xingchao Liu)

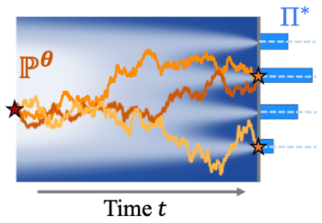


Figure 1: Ω-Bridges for discrete $\Omega = \{1, 2, 3, 4\}$.

- Constructing imputation mechanisms to generate latent trajectories that would have generated a given data point x(x-bridge) 2. specifying and training the diffusion generative model to generate data on the domain Ω of interest by maximizing likelihood using the imputed trajectories(Ω-bridge)
- Construction: Time-reversal and h-transform:

$$\mathrm{d}\tilde{Z}_t^x = \tilde{\eta}(\tilde{Z}_t^x, T-t)\mathrm{d}t + \sigma(\tilde{Z}_t^x, T-t)\mathrm{d}\tilde{W}_t, \qquad \tilde{Z}_0^x = x,$$

Using time reversion formula:

$$\mathrm{d}Z_t^x = \left(-\tilde{\eta}(Z_t^x, t) + \frac{\nabla_z(\sigma^2(Z_t^x, t)q_t^x(Z_t^x))}{q_t^x(Z_t^x)}\right)\mathrm{d}t + \sigma(Z_t^x, t)\mathrm{d}W_t, \qquad Z_0^x \sim \mathbb{Q}_0^x,$$

Consider the conditioned process $\mathbb{Q}^x(\cdot) \coloneqq \mathbb{Q}(\cdot \mid Z_T = x)$ exists:

$$\mathrm{d}Z_t^x = \left(b(Z_t^x, t) + \sigma^2(Z_t^x, t)\nabla_z \log q_{T|t}(x \mid Z_t^x)\right)\mathrm{d}t + \sigma(Z_t^x, t)\mathrm{d}W_t, \quad Z_0 \sim \mathbb{Q}_{0|T}(\cdot \mid x), \tag{10}$$

- Mixtures: mixtures of bridges are bridges, which allows us to decouple the choice of initialization and dynamics in bridges.
- Markov: Let $\mathbb{Q}^{\Pi^*}(\cdot) = \int \mathbb{Q}^x(\cdot)\Pi^*(\mathrm{d}x)$, and

Proposition 3.4. *Take $\mathbb{Q}^x$ to be the dynamics in (11) initialized from $Z_0 \sim \mathcal{N}(0, v_0)$. Assume $\varsigma_t > 0$, $\forall t \in [0, T]$. Then $\mathbb{Q}^{\Pi^*}$ is Markov only when $v_0 = 0$, or $v_0 = +\infty$.*

- General construction:

In the first step, for any $\mathbb{Q}$ following $\mathrm{d}Z_t = b(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$, the h-transform method shows that the conditioned process $\mathbb{Q}^\Omega \coloneqq \mathbb{Q}(\cdot \mid Z_T \in \Omega)$ follows $\mathrm{d}Z_t = \eta^\Omega(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$ with

$$\eta^\Omega(z, t) = b(z, t) + \sigma^2(z, t)\mathbb{E}_{x\sim\mathbb{Q}_{T|t,z,\Omega}}[\nabla_z \log q_{T|t}(x \mid z)], \qquad Z_0 \sim \mathbb{Q}_{0|T}(\cdot \mid X_T \in \Omega).$$

In the second step, given an Ω-bridge $\mathbb{Q}^\Omega$, we construct a parametric model $\mathbb{P}^\theta$ by adding a learnable neural network f^θ in the drift and (optionally) starting from a learnable initial distribution μ^θ:

$$\mathbb{P}^\theta: \qquad \mathrm{d}Z_t = (\sigma(Z_t, t)f^\theta(Z_t, t) + \eta^\Omega(Z_t, t))\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \quad Z_0 \sim \mathbb{P}_0^\theta. \tag{12}$$

Safety (guest Junyan Zhu):

1. Ground truth influence is unknown.
2. Learn Attribution from Customized Models
3.Contrastive Learning of exemplar and synthesized image (CLIP)

Flow Matching

Resource: **T. Fjelde et al., Post on "An Introduction to Flow Matching"**

Flow Matching for Generative Modeling (ICLR23, Meta, Lipman)

- FM: a simulation-free approach. Using OT.

Given a target probability density path $p_t(x)$ and a corresponding vector field $u_t(x)$, which generates $p_t(x)$, we define the Flow Matching (FM) objective as

- $$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)}\|v_t(x) - u_t(x)\|^2, \tag{5}$$

Parameterize the vector field v_t.

- However, we have no prior knowledge for what an appropriate p_t and u_t are.
- $$p_t(x) = \int p_t(x|x_1)q(x_1)dx_1,$$

$$p_1(x) = \int p_1(x|x_1)q(x_1)dx_1 \approx q(x).$$ (q(x) is data distribution)

$$u_t(x) = \int u_t(x|x_1)\frac{p_t(x|x_1)q(x_1)}{p_t(x)}dx_1,$$ is a marginal vector field.

Combine them together, we get an alternative loss function which is tractable:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)}\left\|v_t(x) - u_t(x|x_1)\right\|^2,$$

At last we give a parameterization of p_t and u_t:

$$p_t(x|x_1) = \mathcal{N}(x \mid \mu_t(x_1), \sigma_t(x_1)^2 I)$$

We have $\psi_t(x) = \sigma_t(x_1)x + \mu_t(x_1).$ and our loss function has the form:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)}\left\|v_t(\psi_t(x_0)) - \frac{d}{dt}\psi_t(x_0)\right\|^2.$$

Additionally, the vector field can be constructed as:

Theorem 3. *Let $p_t(x|x_1)$ be a Gaussian probability path as in equation 10, and ψ_t its corresponding flow map as in equation 11. Then, the unique vector field that defines ψ_t has the form:*

$$u_t(x|x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}\left(x - \mu_t(x_1)\right) + \mu_t'(x_1). \tag{15}$$

Consequently, $u_t(x|x_1)$ generates the Gaussian path $p_t(x|x_1)$.

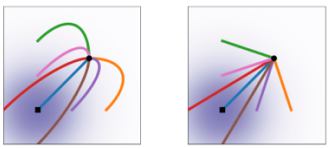


Figure 3: Diffusion and OT trajectories.

Intuitively, particles under the OT displacement map always move in straight line trajectories and with constant speed.

- Two weakness of CFM (conditional flow matching) integrate == expectation

1. Non-straight marginal paths ⇒ ODE hard to integrate ⇒ slow sampling at inference.
2. Many possible x_1 for a noised x_t ⇒ high CFM loss variance ⇒ slow training convergence.

- Faster training (free of simulation) + sampling efficiency (ode>sde and OT performs better) + SOTA.

Catalog of one-step generative models

- **VAEs**
  - Stable training (maximum likelihood)
  - Tractable likelihood estimation
  - Low sample quality
- **GANs**
  - Unstable training (adversarial games)
  - High sample quality
  - No likelihoods
- **Normalizing flows**
  - Stable training (maximum likelihood)
  - Exact likelihood computation
  - Restricted model architecture
  - Low sample quality
- **Consistency models**
  - Stable training (pseudo-objective)
  - High sample quality
  - No likelihoods
  - Moderate architecture constraints.

Building Normalizing Flows with Stochastic Interpolants (ICLR23, Michael S. Albergo)

- Interpolation perspective: $\partial_t \rho_t + \nabla \cdot (v_t \rho_t) = 0$ with $\rho_{t=0} = \rho_0$ and $\rho_{t=1} = \rho_1$,

$$x_t = I_t(x_0, x_1), \qquad x_0 \sim \rho_0, \quad x_1 \sim \rho_1 \quad \text{independent.}$$

Main results: (view the flow matching as an optimizing problem)

Proposition 1. *The stochastic interpolant x_t defined in* (6) *with $I_t(x_0, x_1)$ satisfying* (4) *has a probability density $\rho_t(x)$ that satisfies the continuity equation* (3) *with a velocity $v_t(x)$ which is the unique minimizer over $\hat{v}_t(x)$ of the objective*

$$G(\hat{v}) = \mathbb{E}\left[|\hat{v}_t(I_t(x_0, x_1))|^2 - 2\partial_t I_t(x_0, x_1) \cdot \hat{v}_t(I_t(x_0, x_1))\right] \tag{9}$$

In addition the minimum value of this objective is given by

$$G(v) = -\mathbb{E}\left[|v_t(I_t(x_0, x_1))|^2\right] = -\int_0^1 \int_{\mathbb{R}^d} |v_t(x)|^2 \rho_t(x) dx dt > -\infty \tag{10}$$

Continuity equation:

$$\rho_t(x) = \int_{\mathbb{R}^d \times \mathbb{R}^d} \delta\left(x - I_t(x_0, x_1)\right) \rho_0(x_0) \rho_1(x_1) dx_0 dx_1.$$

Therefore,

$$\partial_t \rho_t(x) = -\int_{\mathbb{R}^d \times \mathbb{R}^d} \partial_t I_t(x_0, x_1) \cdot \nabla \delta\left(x - I_t(x_0, x_1)\right) \rho_0(x_0) \rho_1(x_1) dx_0 dx_1 \equiv -\nabla \cdot j_t(x) \quad (13)$$

where we defined the probability current

$$j_t(x) = \int_{\mathbb{R}^d \times \mathbb{R}^d} \partial_t I_t(x_0, x_1) \delta\left(x - I_t(x_0, x_1)\right) \rho_0(x_0) \rho_1(x_1) dx_0 dx_1. \quad (14)$$

We introduce v_t:

$$v_t(x) = \begin{cases} j_t(x)/\rho_t(x) & \text{if } \rho_t(x) > 0, \\ 0 & \text{else} \end{cases}$$

Observe that maximize G(v) equals to

$$\min_{(\hat{v}, \hat{\rho})} \int_0^1 \int_{\mathbb{R}^d} |\hat{v}_t(x)|^2 \hat{\rho}_t(x) dx dt$$

$$\text{subject to:} \quad \partial_t \hat{\rho}_t + \nabla \cdot \left(\hat{v}_t \hat{\rho}_t\right) = 0, \quad \hat{\rho}_{t=0} = \rho_0, \quad \hat{\rho}_{t=1} = \rho_1.$$

So we are going to find $\max_{\hat{I}} \min_{\hat{v}} G(\hat{v})$, which can give the solution $x_t^* = I_t^*(x_0, x_1)$.

- (Related works): Schroedinger bridges, which are an entropic regularized version of the optimal transportation plan connecting two densities in finite time, using the framework of score-based diffusion.

Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow (ICLR23, Xingchao Liu)

- Theoretical work!
- Provide a unified solution to generative modeling and domain transfer, among various other tasks involving distribution transport.
- The idea of rectified flow is to learn the ODE to follow the straight paths connecting the points drawn from π0 and π1 as much as possible.

Algorithm 1 Rectified Flow: Main Algorithm

Procedure: $\boldsymbol{Z} = \texttt{RectFlow}((X_0, X_1))$:

Inputs: Draws from a coupling (X_0, X_1) of π_0 and π_1; velocity model $v_\theta \colon \mathbb{R}^d \to \mathbb{R}^d$ with parameter θ.

Training: $\hat{\theta} = \arg\min_\theta \mathbb{E}\left[\|X_1 - X_0 - v(tX_1 + (1-t)X_0,\ t)\|^2\right]$, with $t \sim \text{Uniform}([0,1])$.

Sampling: Draw (Z_0, Z_1) following $\mathrm{d}Z_t = v_{\hat{\theta}}(Z_t, t)\mathrm{d}t$ starting from $Z_0 \sim \pi_0$ (or backwardly $Z_1 \sim \pi_1$).

Return: $\boldsymbol{Z} = \{Z_t \colon t \in [0,1]\}$.

Reflow (optional): $\boldsymbol{Z}^{k+1} = \texttt{RectFlow}((Z_0^k, Z_1^k))$, starting from $(Z_0^0, Z_1^0) = (X_0, X_1)$.

Distill (optional): Learn a neural network $\hat{T}$ to distill the k-rectified flow, such that $Z_1^k \approx \hat{T}(Z_0^k)$.

-

It prefers a straight line flows yield fast simulation (can be exactly simulated without time discretization).

non-crossing property – causal.

Marginal preserving property [Theorem 3.3] *The pair (Z_0, Z_1) is a coupling of π_0 and π_1. In fact, the marginal law of Z_t equals that of X_t at every time t, that is,* $\text{Law}(Z_t) = \text{Law}(X_t), \forall t \in [0,1]$.

Reducing transport costs [Theorem 3.5] *The coupling (Z_0, Z_1) yields lower or equal convex transport costs than the input (X_0, X_1) in that $\mathbb{E}[c(Z_1 - Z_0)] \le \mathbb{E}[c(X_1 - X_0)]$ for any convex cost c: $\mathbb{R}^d \to \mathbb{R}$.*

- Read the relation work carefully!!! P18-P22

Discrete Flow Matching (Meta AI, FAIR)

- For language. Experiments on language modeling:

Small model beat baseline, large model beat AR.

Task: Conditional text generation & Code generation.

METHOD	DATA	HUMANEVAL↑			MBPP (1-SHOT)↑		
		Pass@1	Pass@10	Pass@25	Pass@1	Pass@10	Pass@25
Autoregressive	Text	1.2	3.1	4.8	0.2	1.7	3.3
	Code	14.3	21.3	27.8	17.0	34.3	44.1
FM	Text	1.2	2.6	4.0	0.4	1.1	3.6
	Code	6.7	13.4	18.0	6.7	20.6	26.5
FM (Oracle length)	Code	11.6	18.3	20.6	13.1	28.4	34.2

Table 4 Execution based code generation evaluation.

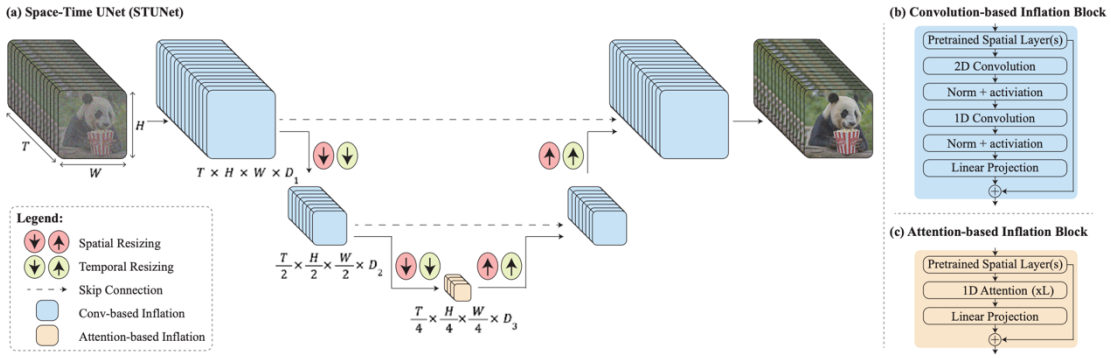
Flow Matching Guide and Code (Meta) (Need to read!)

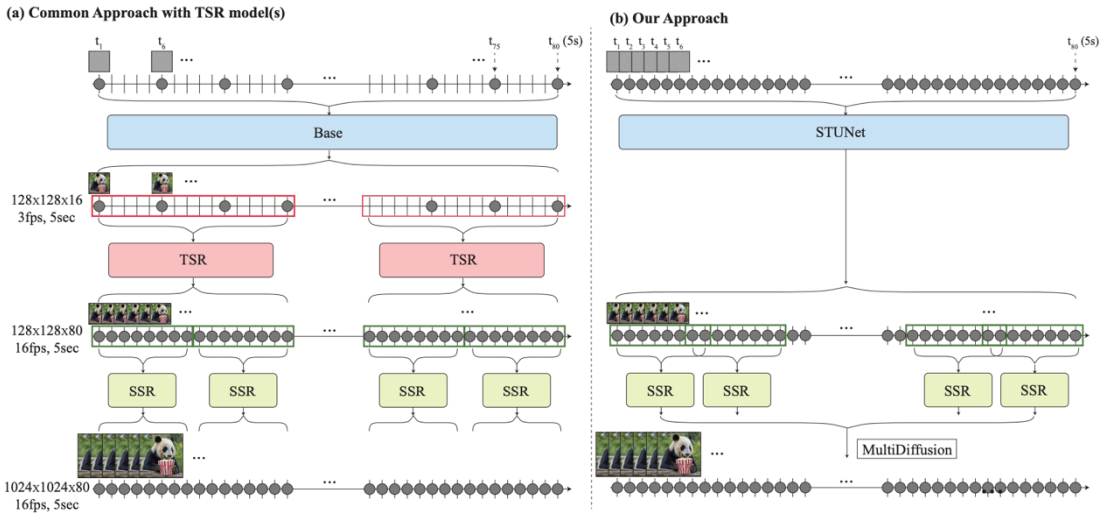
Applications

Videos

Lumiere: A Space-Time Diffusion Model for Video Generation (Google, Weizmann Institute)

- Space-Time U-Net: process all the frames at once to learn the uniform motion.
- Diffusion models can learn a conditional distribution by incorporating additional guiding signals, such as text embedding, or spatial conditioning (e.g., depth map).





Genie: Generative Interactive Environments (Google)

- The first generative interactive environment trained in an unsupervised manner from unlabelled Internet videos. Data: Video.
- Model: latent action model + video tokenizer + dynamics model

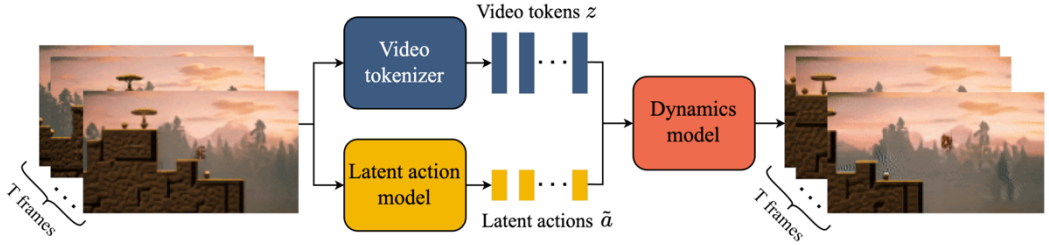


Figure 3 | **Genie model training**: Genie takes in T frames of video as input, tokenizes them into discrete tokens $\boldsymbol{z}$ via the video tokenizer, and infers the latent actions $\tilde{\boldsymbol{a}}$ between each frame with the latent action model. Both are then passed to the dynamics model to generate predictions for the next frames in an iterative manner.

Dynamics Model The dynamics model is a decoder-only MaskGIT (Chang et al., 2022) transformer (Figure 7). At each time step $t \in [1, T]$, it takes in the tokenized video $\boldsymbol{z}_{1:t-1}$ and stopgrad latent actions $\tilde{\boldsymbol{a}}_{1:t-1}$ and predicts the next frame tokens $\hat{\boldsymbol{z}}_t$. We again utilize an ST-transformer,

Movie Gen: A Cast of Media Foundation Models (Meta) **(Need to read: How to convince others this architecture is useful?)**

- Movie Gen: a cast of foundation models that generates high-quality, 1080p HD videos with different aspect ratios and synchronized audio.
- Our models set a new state-of-the-art on multiple tasks: text-to-video synthesis, video personalization, video editing, video-to-audio generation, and text-to-audio generation.

3D and Geometry

DreamFusion: Text-to-3D using 2D Diffusion (ICLR 23, Google)

- Adapting diffusion to 3D synthesis would require large-scale datasets of labeled 3D data and efficient architectures for denoising 3D data, neither of which currently exist.
- In this work, we circumvent these limitations by using a pretrained 2D text-to-image diffusion model to perform text-to-3D synthesis.
- Technique: Score Distillation Sampling approach + NeRF-like rendering engine.

Score Distillation Sampling approach: (we instead want to create 3D models that look like good images when rendered from random angles instead of pixels)

The diffusion gradient:

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x}=g(\theta)) = \mathbb{E}_{t,\epsilon}\left[w(t) \underbrace{\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right)}_{\text{Noise Residual}} \underbrace{\frac{\partial \hat{\epsilon}_\phi(\mathbf{z}_t; y, t)}{\mathbf{z}_t}}_{\text{U-Net Jacobian}} \underbrace{\frac{\partial \mathbf{x}}{\partial \theta}}_{\text{Generator Jacobian}} \right]$$

Simplify:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x}=g(\theta)) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t) \left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right) \frac{\partial \mathbf{x}}{\partial \theta} \right]$$

We can write the loss as:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x}=g(\theta)) = \nabla_\theta \mathbb{E}_t \left[\sigma_t / \alpha_t w(t) \mathbf{KL}(q(\mathbf{z}_t | g(\theta); y, t) \| p_\phi(\mathbf{z}_t; y, t)) \right].$$

NeRF-like rendering engine: a volumetric raytracer is combined with a neural mapping from spatial coordinates to color and volumetric density

LRM: Large Reconstruction Model for Single Image to 3D (Adobe)

- In light of this, we pose the same question for 3D: given sufficient 3D data and a large-scale training framework, is it possible to learn a generic 3D prior for reconstructing an object from a single image?
- Transformer.

Robotics

Planning with Diffusion for Flexible Behavior Synthesis (ICML22)

- However, learned models are often poorly suited to the types of planning algorithms designed with ground-truth models in mind, leading to planners that exploit learned models by finding adversarial examples.
- Target: $\mathbf{a}^*_{0:T} = \arg\max_{\mathbf{a}_{0:T}} \mathcal{J}(\mathbf{s}_0, \mathbf{a}_{0:T}) = \arg\max_{\mathbf{a}_{0:T}} \sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t)$
- Trajectory: $\boldsymbol{\tau} = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)$
- Diffusion model: $p_\theta(\boldsymbol{\tau}^0) = \int p(\boldsymbol{\tau}^N) \prod_{i=1}^{N} p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i) \mathrm{d}\boldsymbol{\tau}^{1:N}$
- $\tilde{p}_\theta(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) h(\boldsymbol{\tau})$. where h(t) contains prior information.

- Merits: 1.Long-horizon 2.Task combination (adapt to new reward function).

Material Science

Junction Tree Variational Autoencoder for Molecular Graph Generation (ICML 2018)

- The key challenge of drug discovery is to find target molecules with desired chemical properties.
- We decompose the challenge into two complementary subtasks: learning to represent molecules in a continuous manner that facilitates the prediction and optimization of their properties (encoding); and learning to map an optimized continuous representation back into a molecular graph with improved properties (decoding). – latent representation
- SMILES strings (1988). Drawbacks: First, the SMILES representation is not designed to capture molecular similarity. Second, essential chemical properties such as molecule validity are easier to express on graphs rather than linear SMILES representations.

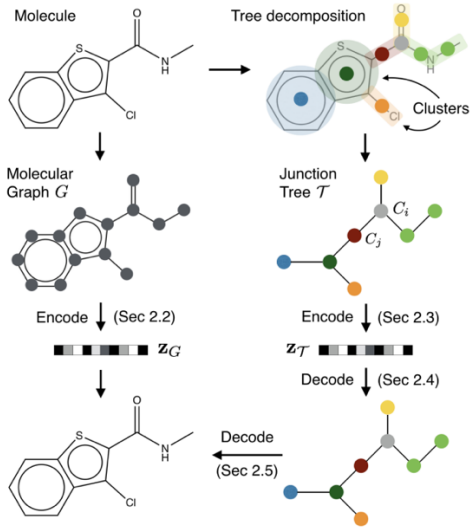


Figure 3. Overview of our method: A molecular graph G is first decomposed into its junction tree $\mathcal{T}_G$, where each colored node in the tree represents a substructure in the molecule. We then encode both the tree and graph into their latent embeddings $\mathbf{z}_\mathcal{T}$ and $\mathbf{z}_G$. To decode the molecule, we first reconstruct junction tree from $\mathbf{z}_\mathcal{T}$, and then assemble nodes in the tree back to the original molecule.

- Node: a chemical structure, tree: potential structure, score: the probability of this structure.

Equivariant Diffusion for Molecule Generation in 3D (ICML22)

- Our E(3) Equivariant Diffusion Model (EDM) learns to denoise a diffusion process with an equivariant network that jointly operates on both continuous (atom coordinates) and categorical features (atom types).

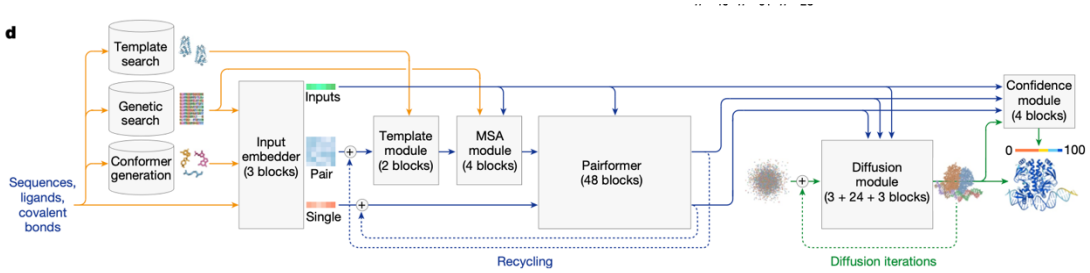
Uni-Mol: A Universal 3D Molecular Representation Learning Framework (ICLR 2023)

- In most MRL methods, molecules are treated as 1D sequential tokens or 2D topology graphs, limiting their ability to incorporate 3D information for downstream tasks and, in particular, making it almost impossible for 3D geometry prediction/generation.

Protein and Biology

Accurate structure prediction of biomolecular interactions with AlphaFold 3 (Nature)

- Inference:



More papers are coming!

Dimension-free Score Matching and Time Bootstrapping for Diffusion Models

- Establish the first (nearly) dimension-free sample complexity bounds for learning these score functions. (log-log-d bound)
- Key idea: martingale-based error decomposition + variance bound.
- BSM (Bootstrapped Score Matching): a variance reduction technique.
- $$\hat{\mathcal{L}}(f) := \frac{1}{mN}\sum_{i=1}^{m}\sum_{t\in\mathcal{T}}\left\| f(t, x_t^{(i)}) + \frac{z_t^{(i)}}{\sigma_t^2}\right\|_2^2.$$ represents a regression task with noisy labels.

$$\epsilon_{\text{score}}^2(\hat{f}) := \sum_{i=2}^{N}\gamma_i \mathbb{E}_{x\sim p_{t_i}}\|\hat{f}(t_i,x) - s(t_i,x)\|^2, \text{ where } \gamma_i := t_i - t_{i-1}$$ is what we need to bound.

- Define the martingale like

The martingale difference decomposition of $H^{\hat{f}}$, exploiting the Markovian structure of (1), has terms of the form $Q_i := \langle G_i, Y_i - \mathbb{E}[Y_i|\mathcal{F}_{i-1}]\rangle$ adapted to the filtration $\{\mathcal{F}_i\}_{i\in[n]}$, where G_i is a $\mathcal{F}_{i-1}$ measurable random variable. The proof primarily uses the fact that for $t_1 \le t_2 \le t_3$, $\mathbb{E}[x_{t_1}|x_{t_2}, x_{t_3}] = \mathbb{E}[x_{t_1}|x_{t_2}]$ due to the Markov property.

Lemma 3. *Let $\zeta = \frac{s-f}{m}$ for any $f \in \mathcal{H}$. Define*

$$\bar{G}_i := \sum_{j=1}^{N}\frac{\gamma_j e^{-(t_j-t_1)}\zeta\left(t_j, x_{t_j}^{(i)}\right)}{\sigma_{t_j}^2}, \quad G_{i,k} := \sum_{j=N-k+2}^{N}\frac{\gamma_j e^{-t_j}\zeta\left(t_j, x_{t_j}^{(i)}\right)}{\sigma_{t_j}^2}$$

and define $R_{i,k}$ as

$$R_{i,k} := \begin{cases} 0, & \text{for } k = 0, \\ \langle G_{i,k+1}, \mathbb{E}[x_0^{(i)}|x_{t_{N-k+1}}^{(i)}] - \mathbb{E}[x_0^{(i)}|x_{t_{N-k}}^{(i)}]\rangle, & \text{for } k \in [N-1], \\ \langle \bar{G}_i, z_{t_1}^{(i)} - \mathbb{E}\left[z_{t_1}^{(i)}|x_{t_1}^{(i)}\right]\rangle, & \text{for } k = N. \end{cases}$$

Let $t_0 = 0$. Consider the filtration defined by the sequence of σ-algebras,

$$\mathcal{F}_{i,k} := \sigma(\{x_t^{(j)} : 1 \le j < i, t \in \mathcal{T}\} \cup \{x_t^{(i)} : t \ge t_{N-k}\})$$

Calculate the variance and derive concentration inequality.

Remark: I think there's some typo, and no insight for the result. (Maybe variance reduction)

Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion
History-guided Video Diffusion. Thousand frames video era comes!
Noise level – mask mechanism.

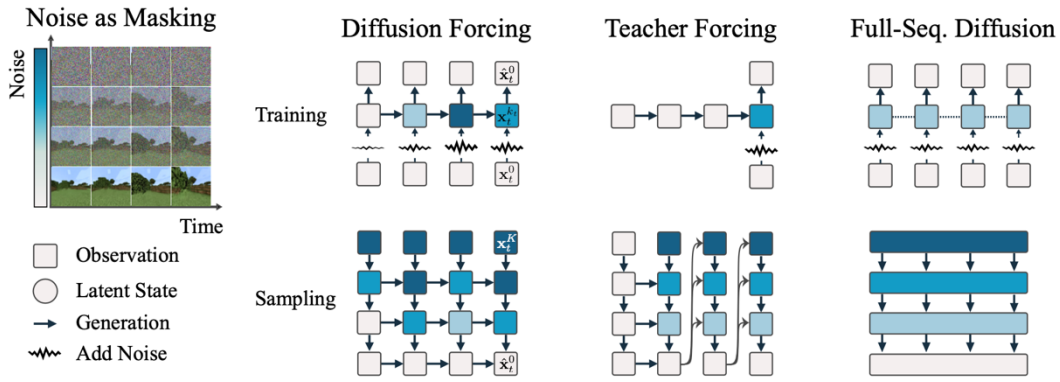


Figure 2: **Method Overview.** Diffusion Forcing trains causal sequence neural networks (such as an RNN or a masked transformer) to denoise flexible-length sequences where each frame of the sequence can have a *different* noise level. In contrast, next-token prediction models, common in language modeling, are trained to predict a single next token from a *ground-truth* sequence (teacher forcing [65]), and full-sequence diffusion, common in video generation, train non-causal architectures to denoise all frames in a sequence at once with the *same* noise level. Diffusion Forcing thus *interleaves* the time axis of the sequence and the noise axis of diffusion, unifying strengths of both alternatives and enabling completely new capabilities (see Secs. 3.2,3.4).

Simple and Effective Masked Diffusion Language Models

- Diffusion + AR
- Mask mechanism

```
          changes in his starting lineup, Brees was      hoping they
had little to prove          Carolina                    I felt like
we didn   have       enough       on     there was so pun
that          more    our guys at     same          so we'd up our
game," he said         said       Carolina was well at
"If that's part    it, if you          to try          with what
you          [    with   What'd         ?       play   Brees said.
               say? That you're always ready to     .       're
          strong and ready to go to      football."
```

The sample generation process begins with a sequence of all masked tokens. MDLM then replaces these masked tokens with actual tokens in a random order.

How Much Is A Noisy Image Worth? Data Scaling Laws For Ambient Diffusion

- Ambient Diffusion and related frameworks train diffusion models with solely corrupted data (which are usually cheaper to acquire) but ambient models significantly underperform models trained on clean data.
- Some ideas: Noise is all you need.
- If the data's noise is larger than the t-level noise, than we don't learn it.

$$J_{\text{Ambient DSM}}(\theta) = \mathbb{E}_{\boldsymbol{x}_{t_n} \sim p_{t_n}} \mathbb{E}_{t \sim U(t_n, T]} \mathbb{E}_{\boldsymbol{x}_t \sim p_t(\boldsymbol{x}_t | \boldsymbol{X}_{t_n} = \boldsymbol{x}_{t_n})} \left\| \frac{\sigma_t^2 - \sigma_{t_n}^2}{\sigma_t^2} \boldsymbol{h}_\theta(\boldsymbol{x}_t, t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \boldsymbol{x}_t - \boldsymbol{x}_{t_n} \right\|^2 . \quad (4)$$

This idea is closely related to Noisier2Noise (Moran et al., 2020). We underline that there are also alternative ways to learn the optimal denoiser in this regime, such as SURE (Stein, 1981). However, SURE-based methods usually bring a computational overhead, as one needs to compute (or approximate) a Jacobian Vector Product.

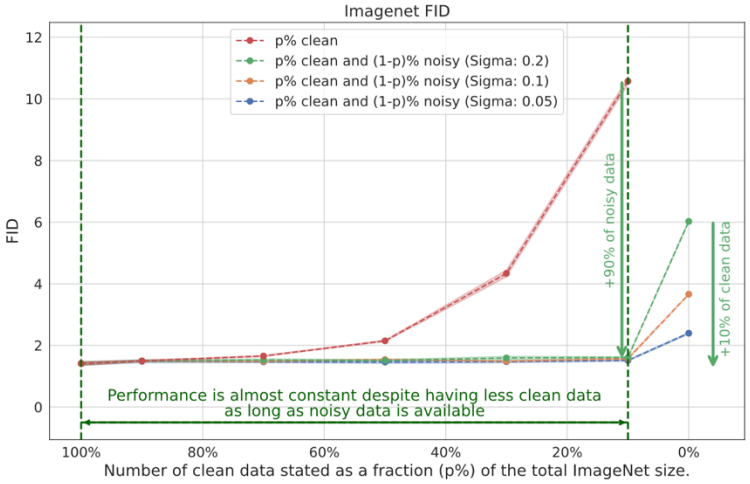


Figure 1: ImageNet FID performance (lower is better) for models trained with different amounts of clean and noisy data. Performance of models trained with only clean data (red curve) reduces as we decrease the amount of data used. Training with purely noisy data (right-most points) also gives poor performance – even if 100% of the dataset is available. Training with a mix of noisy and clean data strikes an interesting balance: *a model trained with* 90% *noisy and* 10% *clean data is almost as good as a model trained with* 100% *clean data.*