CSCI 699 - ProbGen

Probabilistic and Generative Models

Willie Neiswanger

Lecture 10 - (Final) Generative Models → Connections → Predictive UQ

Today

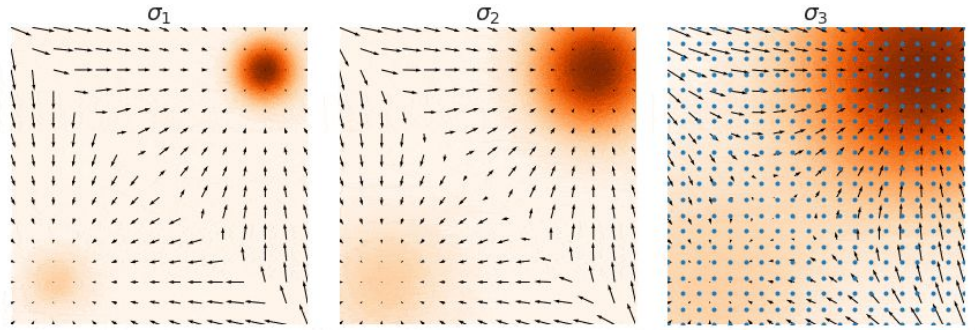
Today

Lecture: Final class on (core) generative modeling methodology → then transition to predictive uncertainty quantification by the end of today/next week.

Today

Lecture: Final class on (core) generative modeling methodology → then transition to predictive uncertainty quantification by the end of today/next week.

- Finish score-based generative models.
  - Connections with diffusion models (LMC in NCSM vs VI in DDPM)
- Also: connections with VAEs and flow-matching from last lecture.
  - *Implicit* vs *Explicit* probabilistic models

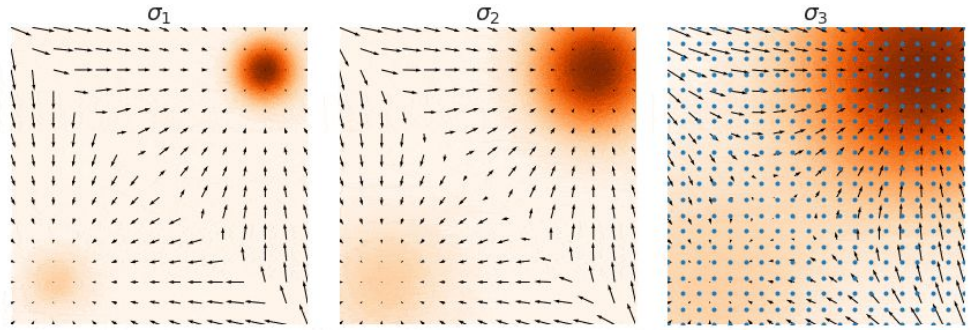


Source: Yang Song, "Generative Modeling by Estimating Gradients of the Data Distribution"

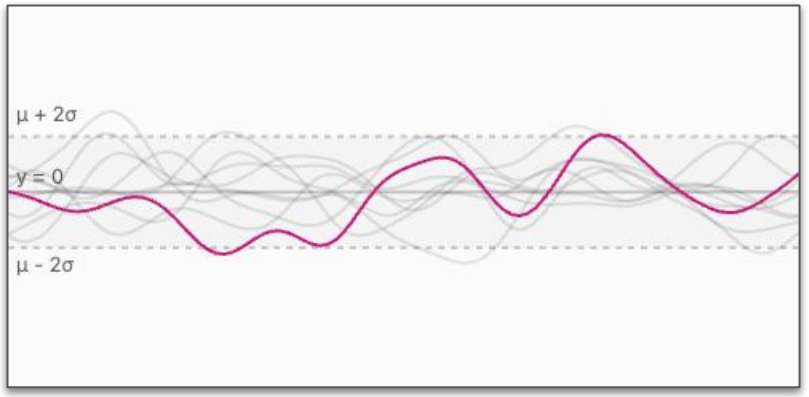
Today

Lecture: Final class on (core) generative modeling methodology → then transition to predictive uncertainty quantification by the end of today/next week.

- Finish score-based generative models.
  - Connections with diffusion models (LMC in NCSM vs VI in DDPM)
- Also: connections with VAEs and flow-matching from last lecture.
  - *Implicit* vs *Explicit* probabilistic models
- Transition to *predictive UQ*.



Source: Yang Song, "Generative Modeling by Estimating Gradients of the Data Distribution"

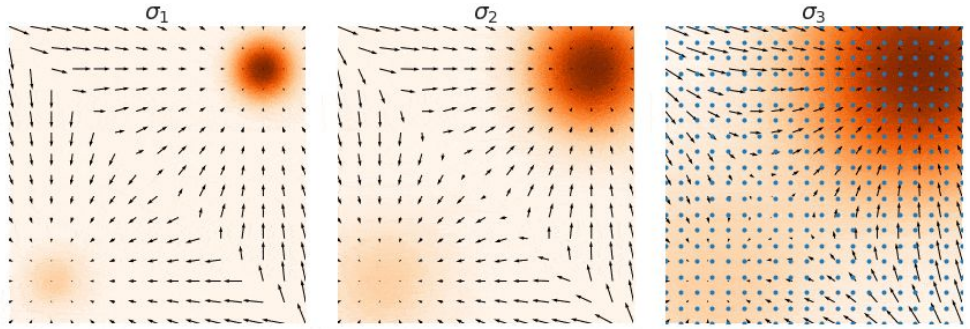


A Visual Exploration of Gaussian Processes, distill.pub

Today

Lecture: Final class on (core) generative modeling methodology → then transition to predictive uncertainty quantification by the end of today/next week.

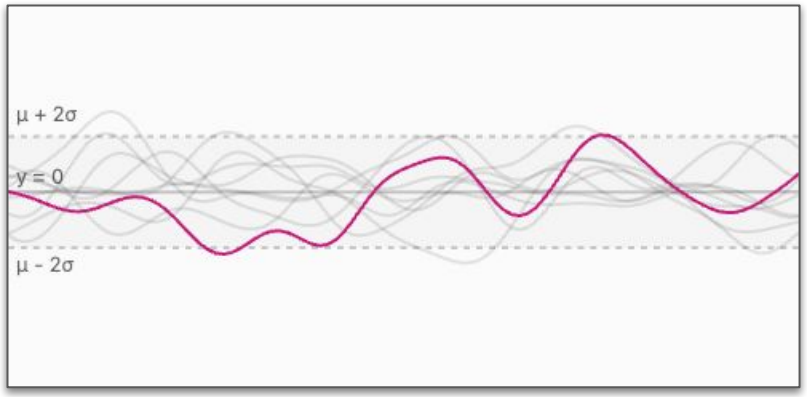
- Finish score-based generative models.
  - Connections with diffusion models (LMC in NCSM vs VI in DDPM)
- Also: connections with VAEs and flow-matching from last lecture.
  - *Implicit* vs *Explicit* probabilistic models
- Transition to *predictive UQ*.



Source: Yang Song, "Generative Modeling by Estimating Gradients of the Data Distribution"

After:

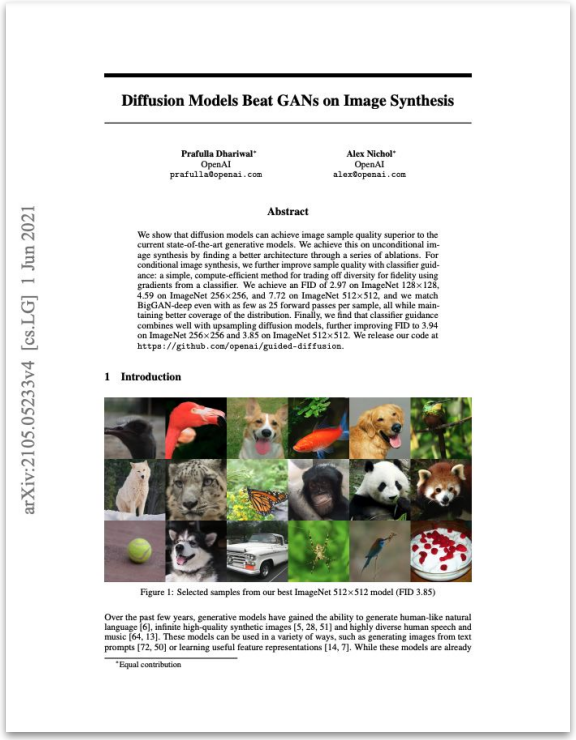
- Fourth batch of four **paper presentations**.



A Visual Exploration of Gaussian Processes, distill.pub

Today

After: Fourth batch of four **paper presentations.**



Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com

Alex Nichol* OpenAI alex@openai.com

Abstract

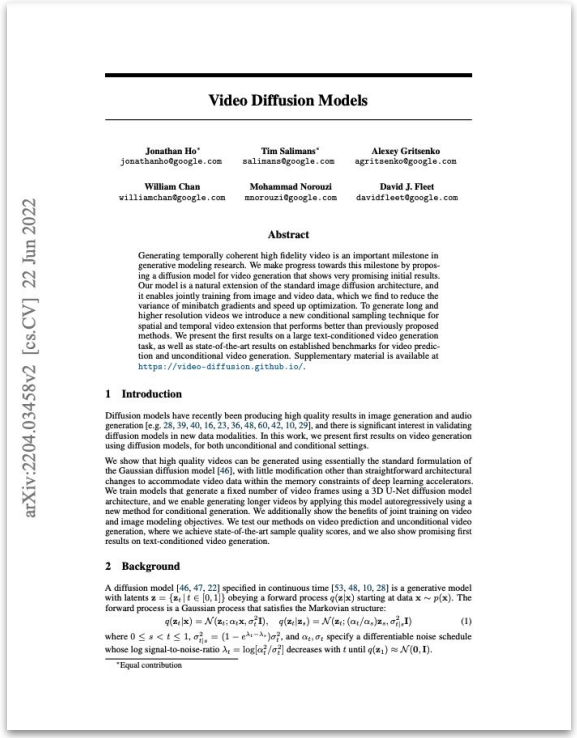
We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 28, 51] and highly diverse human speech and music [64, 13]. These models can be used in a variety of ways, such as generating images from text prompts [72, 50] or learning useful feature representations [14, 7]. While these models are already

*Equal contribution



Video Diffusion Models

Jonathan Ho* jonathanho@google.com

Tim Salimans* salimans@google.com

Alexey Gritsenko agritsenko@google.com

William Chan williamchan@google.com

Mohammad Norouzi mnorouzi@google.com

David J. Fleet davidfleet@google.com

Abstract

Generating temporally coherent high fidelity video is an important milestone in generative modeling research. We make progress towards this milestone by proposing a diffusion model for video generation that shows very promising initial results. Our model is a natural extension of the standard image diffusion architecture, and it enables jointly training from image and video data, which we find to reduce the variance of minibatch gradients and speed up optimization. To generate long and higher resolution videos we introduce a new conditional sampling technique for spatial and temporal video extension that performs better than previously proposed methods. We present the first results on a large text-conditioned video generation task, as well as state-of-the-art results on established benchmarks for video prediction and unconditional video generation. Supplementary material is available at https://video-diffusion.github.io/.

1 Introduction

Diffusion models have recently been producing high quality results in image generation and audio generation [e.g. 28, 39, 40, 16, 23, 36, 48, 60, 42, 10, 29], and there is significant interest in validating diffusion models in new data modalities. In this work, we present first results on video generation using diffusion models, for both unconditional and conditional settings.

We show that high quality videos can be generated using essentially the standard formulation of the Gaussian diffusion model [46], with little modification other than straightforward architectural changes to accommodate video data within the memory constraints of deep learning accelerators. We train models that generate a fixed number of video frames using a 3D U-Net diffusion model architecture, and we enable generating longer videos by applying this model autoregressively using a new method for conditional generation. We additionally show the benefits of joint training on video and image modeling objectives. We test our methods on video prediction and unconditional video generation, where we achieve state-of-the-art sample quality scores, and we also show promising first results on text-conditioned video generation.

2 Background

A diffusion model [46, 47, 22] specified in continuous time [53, 48, 10, 28] is a generative model with latents $\mathbf{z} = \{\mathbf{z}_t | t \in [0, 1]\}$ obeying a forward process $q(\mathbf{z}|\mathbf{x})$ starting at data $\mathbf{x} \sim p(\mathbf{x})$. The forward process is a Gaussian process that satisfies the Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}) \tag{1}$$

where $0 \leq s < t \leq 1$, $\sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$, and α_t, σ_t specify a differentiable noise schedule whose log signal-to-noise-ratio $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ decreases with t until $q(\mathbf{z}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

*Equal contribution

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]

[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML

https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

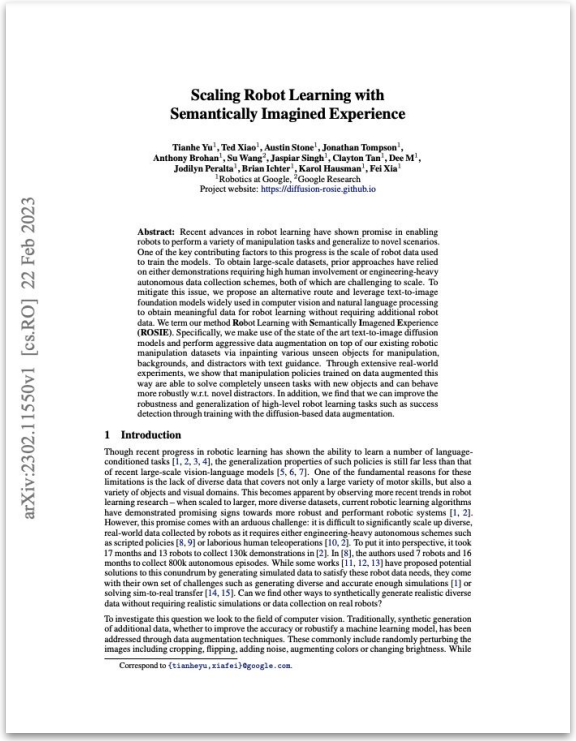
Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [66, 67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [30, 85] and beyond [7, 45, 48, 57], and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,



Scaling Robot Learning with Semantically Imagined Experience

Tianhe Yu[1], Ted Xiao[1], Austin Stone[1], Jonathan Tompson[1], Anthony Brohan[1], Su Wang[2], Jaspiar Singh[1], Clayton Tan[1], Dee M[1], Jodilyn Peralta[1], Brian Ichter[1], Karol Hausman[1], Fei Xia[1]

[1]Robotics at Google, [2]Google Research

Project website: https://diffusion-rosie.github.io

Abstract: Recent advances in robot learning have shown promise in enabling robots to perform a variety of manipulation tasks and generalize to novel scenarios. One of the key contributing factors to this progress is the scale of robot data used to train the models. To obtain large-scale datasets, prior approaches have relied on either demonstrations requiring high human involvement or engineering-heavy autonomous data collection schemes, both of which are challenging to scale. To mitigate this issue, we propose an alternative route and leverage text-to-image foundation models widely used in computer vision and natural language processing to obtain meaningful data for robot learning without requiring additional robot data. We term our method **R**obot Learning with **S**emantically Imagened **E**xperience (**ROSIE**). Specifically, we make use of the state of the art text-to-image diffusion models and perform aggressive data augmentation on top of our existing robotic manipulation datasets via inpainting various unseen objects for manipulation, backgrounds, and distractors with text guidance. Through extensive real-world experiments, we show that manipulation policies trained on data augmented this way are able to solve completely unseen tasks with new objects and can behave more robustly w.r.t. novel distractors. In addition, we find that we can improve the robustness and generalization of high-level robot learning tasks such as success detection through training with the diffusion-based data augmentation.

1 Introduction

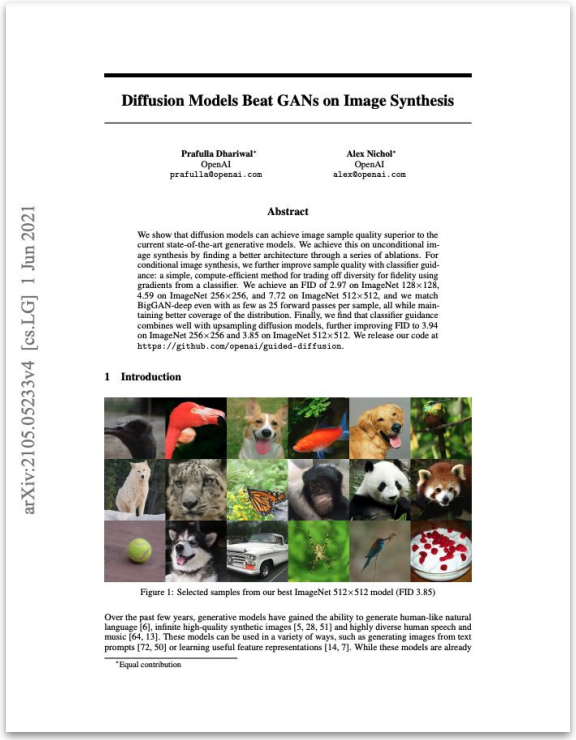
Though recent progress in robotic learning has shown the ability to learn a number of language-conditioned tasks [1, 2, 3, 4], the generalization properties of such policies is still far less than that of recent large-scale vision-language models [5, 6, 7]. One of the fundamental reasons for these limitations is the lack of diverse data that covers not only a large variety of motor skills, but also a variety of objects and visual domains. This becomes apparent by observing more recent trends in robot learning research – when scaled to larger, more diverse datasets, current robotic learning algorithms have demonstrated promising signs towards more robust and performant robotic systems [1, 2]. However, this promise comes with an arduous challenge: it is difficult to significantly scale up diverse, real-world data collected by robots as it requires either engineering-heavy autonomous schemes such as scripted policies [8, 9] or laborious human teleoperations [10, 2]. To put it into perspective, it took 17 months and 13 robots to collect 130k demonstrations in [2]. In [8], the authors used 7 robots and 16 months to collect 800k autonomous episodes. While some works [11, 12, 13] have proposed potential solutions to this conundrum by generating simulated data to satisfy these robot data needs, they come with their own set of challenges such as generating diverse and accurate enough simulations [1] or solving sim-to-real transfer [14, 15]. Can we find other ways to synthetically generate realistic diverse data without requiring realistic simulations or data collection on real robots?

To investigate this question we look to the field of computer vision. Traditionally, synthetic generation of additional data, whether to improve the accuracy or robustify a machine learning model, has been addressed through data augmentation techniques. These commonly include randomly perturbing the images including cropping, flipping, adding noise, augmenting colors or changing brightness. While

Correspond to {tianheyu,xiaf ei}@google.com.

Today

After: Fourth batch of four **paper presentations.**



arXiv:2105.05233v4 [cs.LG] 1 Jun 2021

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com

Alex Nichol* OpenAI alex@openai.com

Abstract

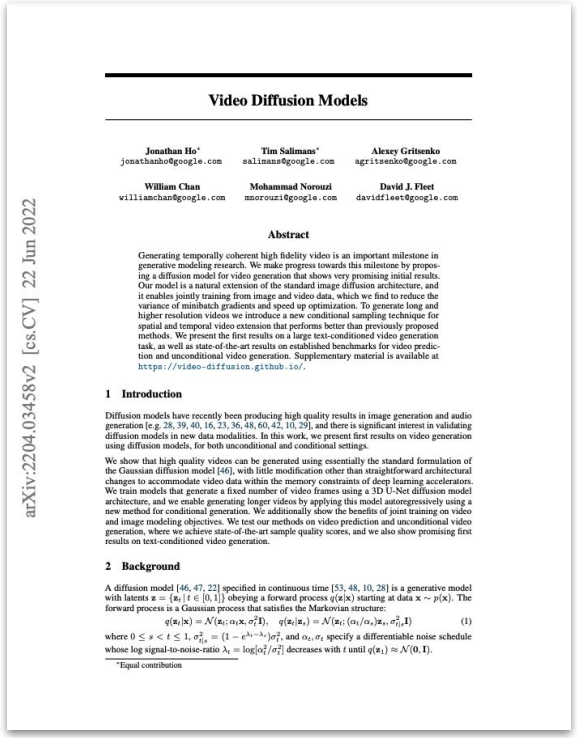
We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 28, 51] and highly diverse human speech and music [64, 13]. These models can be used in a variety of ways, such as generating images from text prompts [72, 50] or learning useful feature representations [14, 7]. While these models are already

*Equal contribution



arXiv:2204.03458v2 [cs.CV] 22 Jun 2022

Video Diffusion Models

Jonathan Ho* jonathanho@google.com

Tim Salimans* salimans@google.com

Alexey Gritsenko agritsenko@google.com

William Chan williamchan@google.com

Mohammad Norouzi mnorouzi@google.com

David J. Fleet davidfleet@google.com

Abstract

Generating temporally coherent high fidelity video is an important milestone in generative modeling research. We make progress towards this milestone by proposing a diffusion model for video generation that shows very promising initial results. Our model is a natural extension of the standard image diffusion architecture, and it enables jointly training from image and video data, which we find to reduce the variance of minibatch gradients and speed up optimization. To generate long and higher resolution videos we introduce a new conditional sampling technique for spatial and temporal video extension that performs better than previously proposed methods. We present the first results on a large text-conditioned video generation task, as well as state-of-the-art results on established benchmarks for video prediction and unconditional video generation. Supplementary material is available at https://video-diffusion.github.io/.

1 Introduction

Diffusion models have recently been producing high quality results in image generation and audio generation [e.g. 28, 39, 40, 16, 23, 36, 48, 60, 42, 10, 29], and there is significant interest in validating diffusion models in new data modalities. In this work, we present first results on video generation using diffusion models, for both unconditional and conditional settings.

We show that high quality videos can be generated using essentially the standard formulation of the Gaussian diffusion model [46], with little modification other than straightforward architectural changes to accommodate video data within the memory constraints of deep learning accelerators. We train models that generate a fixed number of video frames using a 3D U-Net diffusion model architecture, and we enable generating longer videos by applying this model autoregressively using a new method for conditional generation. We additionally show the benefits of joint training on video and image modeling objectives. We test our methods on video prediction and unconditional video generation, where we achieve state-of-the-art sample quality scores, and we also show promising first results on text-conditioned video generation.

2 Background

A diffusion model [46, 47, 22] specified in continuous time [53, 48, 10, 28] is a generative model with latents $\mathbf{z} = \{\mathbf{z}_t | t \in [0, 1]\}$ obeying a forward process $q(\mathbf{z}|\mathbf{x})$ starting at data $\mathbf{x} \sim p(\mathbf{x})$. The forward process is a Gaussian process that satisfies the Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}) \tag{1}$$

where $0 \leq s < t \leq 1$, $\sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$, and α_t, σ_t specify a differentiable noise schedule whose log signal-to-noise-ratio $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ decreases with t until $q(\mathbf{z}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

*Equal contribution

arXiv:2112.10752v2 [cs.CV] 13 Apr 2022

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]

[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML

https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [66, 67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

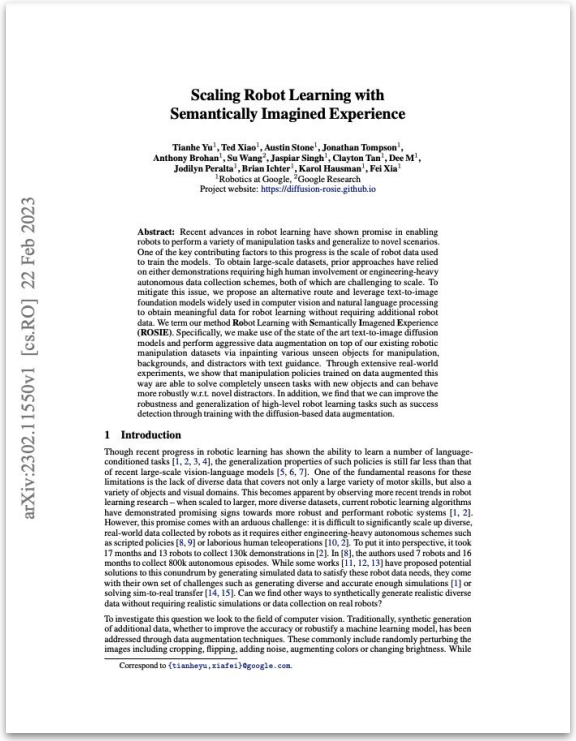
*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [30, 85] and beyond [7, 45, 48, 57], and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

1



arXiv:2302.11550v1 [cs.RO] 22 Feb 2023

Scaling Robot Learning with Semantically Imagined Experience

Tianhe Yu[1], Ted Xiao[1], Austin Stone[1], Jonathan Tompson[1], Anthony Brohan[1], Su Wang[2], Jaspiar Singh[1], Clayton Tan[1], Dee M[1], Jodilyn Peralta[1], Brian Ichter[1], Karol Hausman[1], Fei Xia[1]

[1]Robotics at Google, [2]Google Research

Project website: https://diffusion-rosie.github.io

Abstract: Recent advances in robot learning have shown promise in enabling robots to perform a variety of manipulation tasks and generalize to novel scenarios. One of the key contributing factors to this progress is the scale of robot data used to train the models. To obtain large-scale datasets, prior approaches have relied on either demonstrations requiring high human involvement or engineering-heavy autonomous data collection schemes, both of which are challenging to scale. To mitigate this issue, we propose an alternative route and leverage text-to-image foundation models widely used in computer vision and natural language processing to obtain meaningful data for robot learning without requiring additional robot data. We term our method **R**obot Learning with **S**emantically Imagened **E**xperience (**ROSIE**). Specifically, we make use of the state of the art text-to-image diffusion models and perform aggressive data augmentation on top of our existing robotic manipulation datasets via inpainting various unseen objects for manipulation, backgrounds, and distractors with text guidance. Through extensive real-world experiments, we show that manipulation policies trained on data augmented this way are able to solve completely unseen tasks with new objects and can behave more robustly w.r.t. novel distractors. In addition, we find that we can improve the robustness and generalization of high-level robot learning tasks such as success detection through training with the diffusion-based data augmentation.

1 Introduction

Though recent progress in robotic learning has shown the ability to learn a number of language-conditioned tasks [1, 2, 3, 4], the generalization properties of such policies is still far less than that of recent large-scale vision-language models [5, 6, 7]. One of the fundamental reasons for these limitations is the lack of diverse data that covers not only a large variety of motor skills, but also a variety of objects and visual domains. This becomes apparent by observing more recent trends in robot learning research – when scaled to larger, more diverse datasets, current robotic learning algorithms have demonstrated promising signs towards more robust and performant robotic systems [1, 2]. However, this promise comes with an arduous challenge: it is difficult to significantly scale up diverse, real-world data collected by robots as it requires either engineering-heavy autonomous schemes such as scripted policies [8, 9] or laborious human teleoperations [10, 2]. To put it into perspective, it took 17 months and 13 robots to collect 130k demonstrations in [2]. In [8], the authors used 7 robots and 16 months to collect 800k autonomous episodes. While some works [11, 12, 13] have proposed potential solutions to this conundrum by generating simulated data to satisfy these robot data needs, they come with their own set of challenges such as generating diverse and accurate enough simulations [1] or solving sim-to-real transfer [14, 15]. Can we find other ways to synthetically generate realistic diverse data without requiring realistic simulations or data collection on real robots?

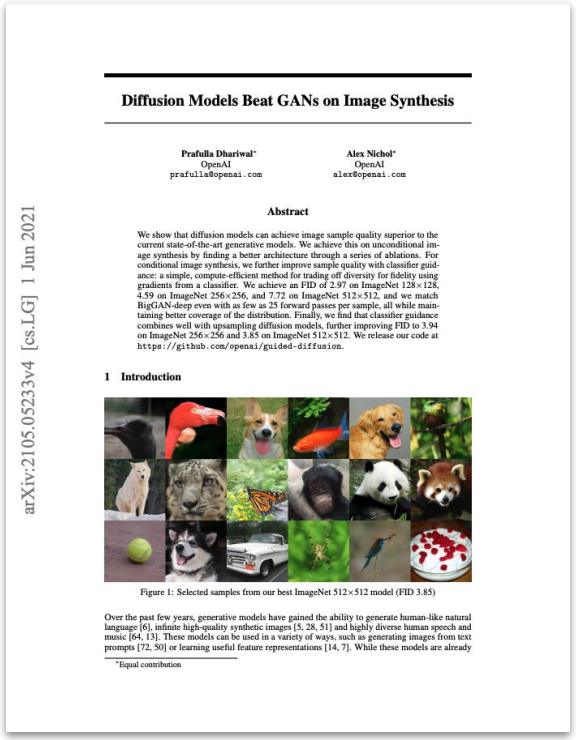
To investigate this question we look to the field of computer vision. Traditionally, synthetic generation of additional data, whether to improve the accuracy or robustify a machine learning model, has been addressed through data augmentation techniques. These commonly include randomly perturbing the images including cropping, flipping, adding noise, augmenting colors or changing brightness. While

Correspond to {tianheyu,xiaf}@google.com.

Early OpenAI diffusion model paper

Today

After: Fourth batch of four **paper presentations.**



Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com
Alex Nichol* OpenAI alex@openai.com

Abstract

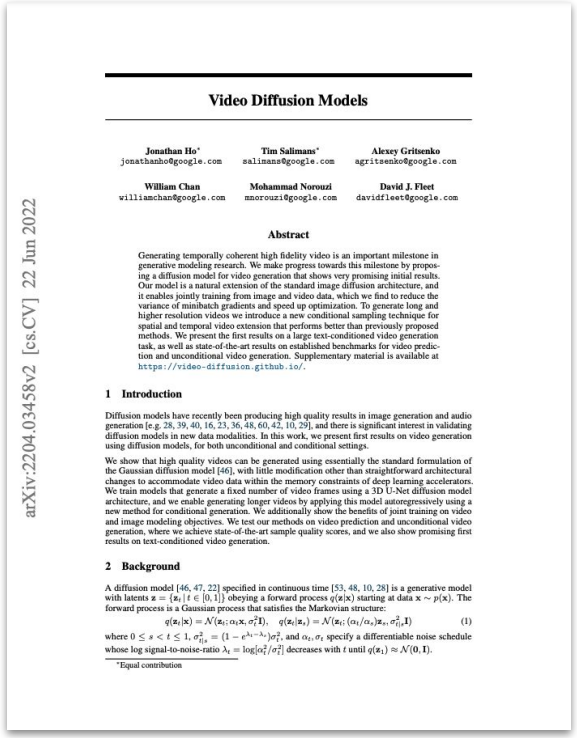
We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 28, 51] and highly diverse human speech and music [64, 13]. These models can be used in a variety of ways, such as generating images from text prompts [72, 50] or learning useful feature representations [14, 7]. While these models are already

*Equal contribution



Video Diffusion Models

Jonathan Ho* jonathanho@google.com
Tim Salimans* salimans@google.com
Alexey Gritsenko agritsenko@google.com
William Chan williamchan@google.com
Mohammad Norouzi mnorouzi@google.com
David J. Fleet davidfleet@google.com

Abstract

Generating temporally coherent high fidelity video is an important milestone in generative modeling research. We make progress towards this milestone by proposing a diffusion model for video generation that shows very promising initial results. Our model is a natural extension of the standard image diffusion architecture, and it enables jointly training from image and video data, which we find to reduce the variance of minibatch gradients and speed up optimization. To generate long and higher resolution videos we introduce a new conditional sampling technique for spatial and temporal video extension that performs better than previously proposed methods. We present the first results on a large text-conditioned video generation task, as well as state-of-the-art results on established benchmarks for video prediction and unconditional video generation. Supplementary material is available at https://video-diffusion.github.io/.

1 Introduction

Diffusion models have recently been producing high quality results in image generation and audio generation [e.g. 28, 39, 40, 16, 23, 36, 48, 60, 42, 10, 29], and there is significant interest in validating diffusion models in new data modalities. In this work, we present first results on video generation using diffusion models, for both unconditional and conditional settings.

We show that high quality videos can be generated using essentially the standard formulation of the Gaussian diffusion model [46], with little modification other than straightforward architectural changes to accommodate video data within the memory constraints of deep learning accelerators. We train models that generate a fixed number of video frames using a 3D U-Net diffusion model architecture, and we enable generating longer videos by applying this model autoregressively using a new method for conditional generation. We additionally show the benefits of joint training on video and image modeling objectives. We test our methods on video prediction and unconditional video generation, where we achieve state-of-the-art sample quality scores, and we also show promising first results on text-conditioned video generation.

2 Background

A diffusion model [46, 47, 22] specified in continuous time [53, 48, 10, 28] is a generative model with latents $\mathbf{z} = \{\mathbf{z}_t | t \in [0, 1]\}$ obeying a forward process $q(\mathbf{z}|\mathbf{x})$ starting at data $\mathbf{x} \sim p(\mathbf{x})$. The forward process is a Gaussian process that satisfies the Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}) \tag{1}$$

where $0 \le s < t \le 1$, $\sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$, and α_t, σ_t specify a differentiable noise schedule whose log signal-to-noise-ratio $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ decreases with t until $q(\mathbf{z}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

*Equal contribution

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML
https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [66, 67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

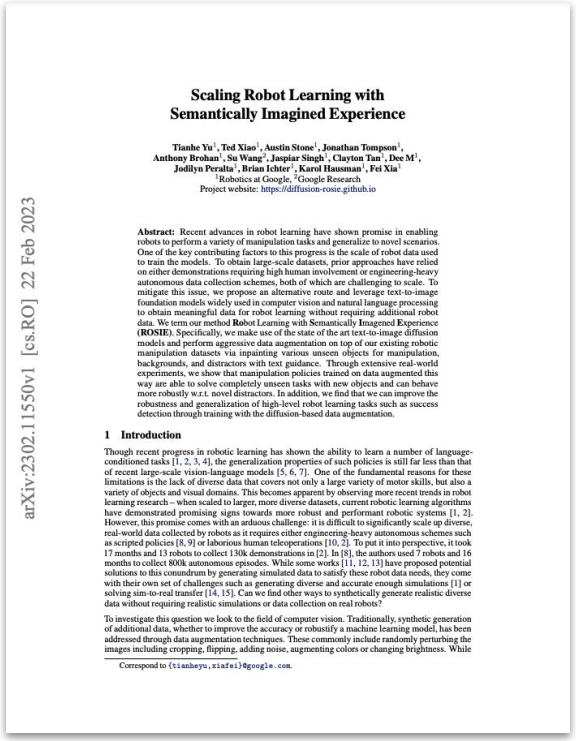
*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [30, 85] and beyond [7, 45, 48, 57]. and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

1



Scaling Robot Learning with Semantically Imagined Experience

Tianhe Yu[1], Ted Xiao[1], Austin Stone[1], Jonathan Tompson[1], Anthony Brohan[1], Su Wang[2], Jaspiar Singh[1], Clayton Tan[1], Dee M[1], Jodilyn Peralta[1], Brian Ichter[1], Karol Hausman[1], Fei Xia[1]
[1]Robotics at Google, [2]Google Research
Project website: https://diffusion-rosie.github.io

Abstract: Recent advances in robot learning have shown promise in enabling robots to perform a variety of manipulation tasks and generalize to novel scenarios. One of the key contributing factors to this progress is the scale of robot data used to train the models. To obtain large-scale datasets, prior approaches have relied on either demonstrations requiring high human involvement or engineering-heavy autonomous data collection schemes, both of which are challenging to scale. To mitigate this issue, we propose an alternative route and leverage text-to-image foundation models widely used in computer vision and natural language processing to obtain meaningful data for robot learning without requiring additional robot data. We term our method **Ro**bot Learning with **S**emantically Imagened **E**xperience (**ROSIE**). Specifically, we make use of the state of the art text-to-image diffusion models and perform aggressive data augmentation on top of our existing robotic manipulation datasets via inpainting various unseen objects for manipulation, backgrounds, and distractors with text guidance. Through extensive real-world experiments, we show that manipulation policies trained on data augmented this way are able to solve completely unseen tasks with new objects and can behave more robustly w.r.t. novel distractors. In addition, we find that we can improve the robustness and generalization of high-level robot learning tasks such as success detection through training with the diffusion-based data augmentation.

1 Introduction

Though recent progress in robotic learning has shown the ability to learn a number of language-conditioned tasks [1, 2, 3, 4], the generalization properties of such policies is still far less than that of recent large-scale vision-language models [5, 6, 7]. One of the fundamental reasons for these limitations is the lack of diverse data that covers not only a large variety of motor skills, but also a variety of objects and visual domains. This becomes apparent by observing more recent trends in robot learning research – when scaled to larger, more diverse datasets, current robotic learning algorithms have demonstrated promising signs towards more robust and performant robotic systems [1, 2]. However, this promise comes with an arduous challenge: it is difficult to significantly scale up diverse, real-world data collected by robots as it requires either engineering-heavy autonomous schemes such as scripted policies [8, 9] or laborious human teleoperations [10, 2]. To put it into perspective, it took 17 months and 13 robots to collect 130k demonstrations in [2]. In [8], the authors used 7 robots and 16 months to collect 800k autonomous episodes. While some works [11, 12, 13] have proposed potential solutions to this conundrum by generating simulated data to satisfy these robot data needs, they come with their own set of challenges such as generating diverse and accurate enough simulations [1] or solving sim-to-real transfer [14, 15]. Can we find other ways to synthetically generate realistic diverse data without requiring realistic simulations or data collection on real robots?

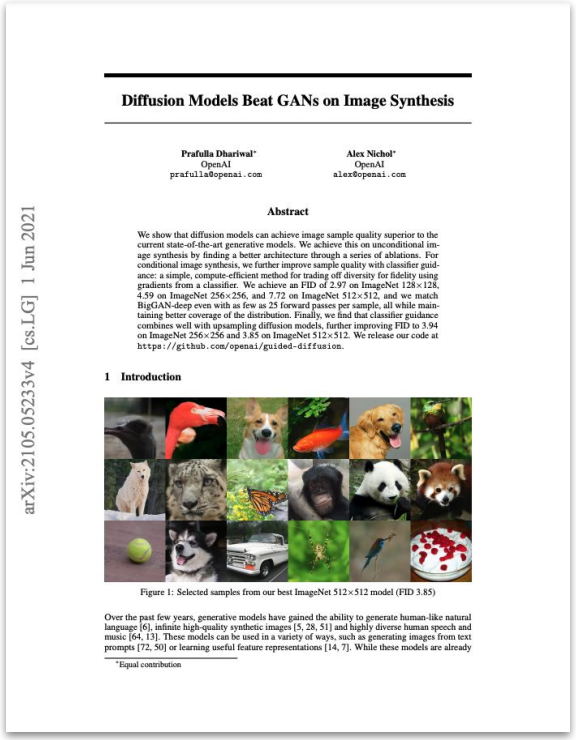
To investigate this question we look to the field of computer vision. Traditionally, synthetic generation of additional data, whether to improve the accuracy or robustify a machine learning model, has been addressed through data augmentation techniques. These commonly include randomly perturbing the images including cropping, flipping, adding noise, augmenting colors or changing brightness. While

Correspond to {tianheyu,xiaf ei}@google.com.

Classic diffusion models for videos

Today

After: Fourth batch of four **paper presentations**.



Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com
Alex Nichol* OpenAI alex@openai.com

Abstract

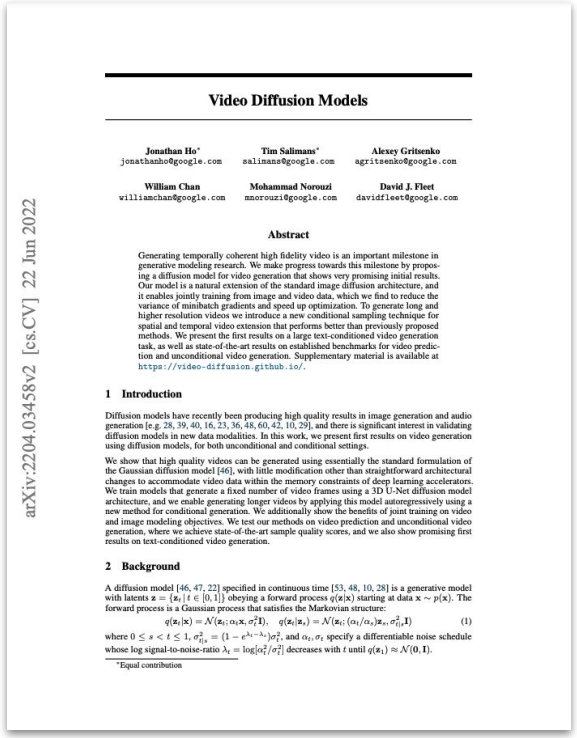
We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 28, 51] and highly diverse human speech and music [64, 13]. These models can be used in a variety of ways, such as generating images from text prompts [72, 50] or learning useful feature representations [14, 7]. While these models are already

*Equal contribution



Video Diffusion Models

Jonathan Ho* jonathanho@google.com
Tim Salimans* salimans@google.com
Alexey Gritsenko agritsenko@google.com
William Chan williamchan@google.com
Mohammad Norouzi mnorouzi@google.com
David J. Fleet davidfleet@google.com

Abstract

Generating temporally coherent high fidelity video is an important milestone in generative modeling research. We make progress towards this milestone by proposing a diffusion model for video generation that shows very promising initial results. Our model is a natural extension of the standard image diffusion architecture, and it enables jointly training from image and video data, which we find to reduce the variance of minibatch gradients and speed up optimization. To generate long and higher resolution videos we introduce a new conditional sampling technique for spatial and temporal video extension that performs better than previously proposed methods. We present the first results on a large text-conditioned video generation task, as well as state-of-the-art results on established benchmarks for video prediction and unconditional video generation. Supplementary material is available at https://video-diffusion.github.io/.

1 Introduction

Diffusion models have recently been producing high quality results in image generation and audio generation [e.g. 28, 39, 40, 16, 23, 36, 48, 60, 42, 10, 29], and there is significant interest in validating diffusion models in new data modalities. In this work, we present first results on video generation using diffusion models, for both unconditional and conditional settings.

We show that high quality videos can be generated using essentially the standard formulation of the Gaussian diffusion model [46], with little modification other than straightforward architectural changes to accommodate video data within the memory constraints of deep learning accelerators. We train models that generate a fixed number of video frames using a 3D U-Net diffusion model architecture, and we enable generating longer videos by applying this model autoregressively using a new method for conditional generation. We additionally show the benefits of joint training on video and image modeling objectives. We test our methods on video prediction and unconditional video generation, where we achieve state-of-the-art sample quality scores, and we also show promising first results on text-conditioned video generation.

2 Background

A diffusion model [46, 47, 22] specified in continuous time [53, 48, 10, 28] is a generative model with latents $\mathbf{z} = \{\mathbf{z}_t | t \in [0, 1]\}$ obeying a forward process $q(\mathbf{z}|\mathbf{x})$ starting at data $\mathbf{x} \sim p(\mathbf{x})$. The forward process is a Gaussian process that satisfies the Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}) \tag{1}$$

where $0 \leq s < t \leq 1$, $\sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$, and α_t, σ_t specify a differentiable noise schedule whose log signal-to-noise-ratio $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ decreases with t until $q(\mathbf{z}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

*Equal contribution

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML

https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

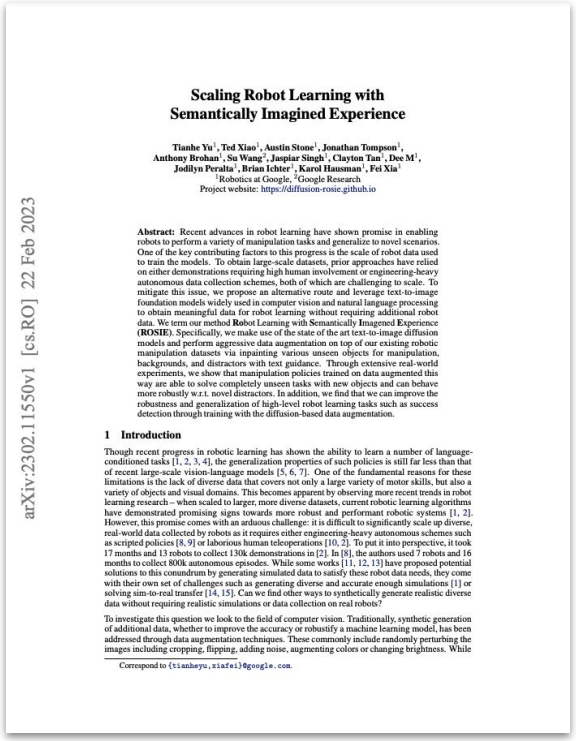
Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [66, 67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [30, 85] and beyond [7, 45, 48, 57], and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,



Scaling Robot Learning with Semantically Imagined Experience

Tianhe Yu[1], Ted Xiao[1], Austin Stone[1], Jonathan Tompson[1], Anthony Brohan[1], Su Wang[2], Jaspiar Singh[1], Clayton Tan[1], Dee M[1], Jodilyn Peralta[1], Brian Ichter[1], Karol Hausman[1], Fei Xia[1]
[1]Robotics at Google, [2]Google Research
Project website: https://diffusion-rosie.github.io

Abstract: Recent advances in robot learning have shown promise in enabling robots to perform a variety of manipulation tasks and generalize to novel scenarios. One of the key contributing factors to this progress is the scale of robot data used to train the models. To obtain large-scale datasets, prior approaches have relied on either demonstrations requiring high human involvement or engineering-heavy autonomous data collection schemes, both of which are challenging to scale. To mitigate this issue, we propose an alternative route and leverage text-to-image foundation models widely used in computer vision and natural language processing to obtain meaningful data for robot learning without requiring additional robot data. We term our method **Ro**bot Learning with **S**emantically Imagened **E**xperience (**ROSIE**). Specifically, we make use of the state of the art text-to-image diffusion models and perform aggressive data augmentation on top of our existing robotic manipulation datasets via inpainting various unseen objects for manipulation, backgrounds, and distractors with text guidance. Through extensive real-world experiments, we show that manipulation policies trained on data augmented this way are able to solve completely unseen tasks with new objects and can behave more robustly w.r.t. novel distractors. In addition, we find that we can improve the robustness and generalization of high-level robot learning tasks such as success detection through training with the diffusion-based data augmentation.

1 Introduction

Though recent progress in robotic learning has shown the ability to learn a number of language-conditioned tasks [1, 2, 3, 4], the generalization properties of such policies is still far less than that of recent large-scale vision-language models [5, 6, 7]. One of the fundamental reasons for these limitations is the lack of diverse data that covers not only a large variety of motor skills, but also a variety of objects and visual domains. This becomes apparent by observing more recent trends in robot learning research – when scaled to larger, more diverse datasets, current robotic learning algorithms have demonstrated promising signs towards more robust and performant robotic systems [1, 2]. However, this promise comes with an arduous challenge: it is difficult to significantly scale up diverse, real-world data collected by robots as it requires either engineering-heavy autonomous schemes such as scripted policies [8, 9] or laborious human teleoperations [10, 2]. To put it into perspective, it took 17 months and 13 robots to collect 130k demonstrations in [2]. In [8], the authors used 7 robots and 16 months to collect 800k autonomous episodes. While some works [11, 12, 13] have proposed potential solutions to this conundrum by generating simulated data to satisfy these robot data needs, they come with their own set of challenges such as generating diverse and accurate enough simulations [1] or solving sim-to-real transfer [14, 15]. Can we find other ways to synthetically generate realistic diverse data without requiring realistic simulations or data collection on real robots?

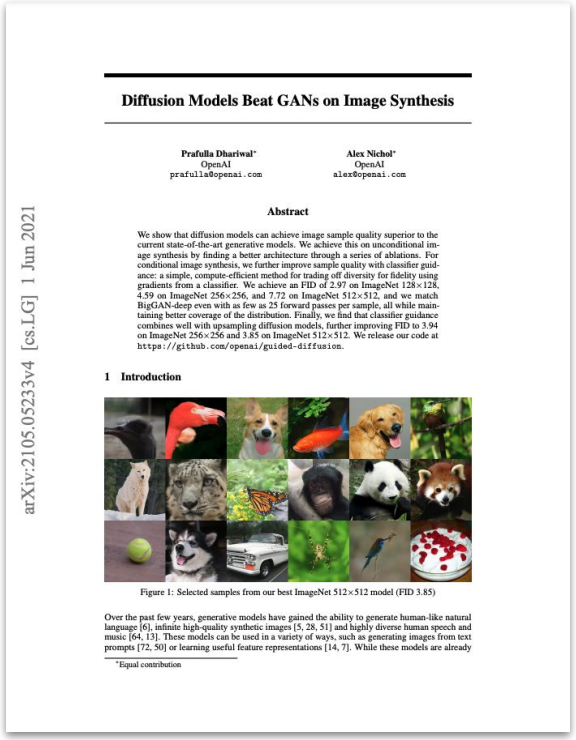
To investigate this question we look to the field of computer vision. Traditionally, synthetic generation of additional data, whether to improve the accuracy or robustify a machine learning model, has been addressed through data augmentation techniques. These commonly include randomly perturbing the images including cropping, flipping, adding noise, augmenting colors or changing brightness. While

Correspond to {tianheyu,xiaf ei}@google.com.

Latent diffusion! Another classic in diffusion.

Today

After: Fourth batch of four **paper presentations.**



arXiv:2105.05233v4 [cs.LG] 1 Jun 2021

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com
Alex Nichol* OpenAI alex@openai.com

Abstract

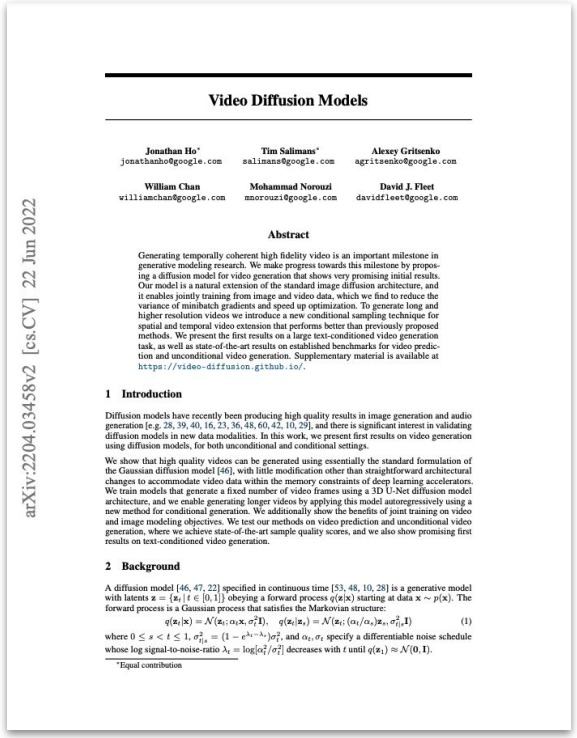
We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 28, 51] and highly diverse human speech and music [64, 13]. These models can be used in a variety of ways, such as generating images from text prompts [72, 50] or learning useful feature representations [14, 7]. While these models are already

*Equal contribution



arXiv:2204.03458v2 [cs.CV] 22 Jun 2022

Video Diffusion Models

Jonathan Ho* jonathanho@google.com
Tim Salimans* salimans@google.com
Alexey Gritsenko agritsenko@google.com
William Chan williamchan@google.com
Mohammad Norouzi mnorouzi@google.com
David J. Fleet davidfleet@google.com

Abstract

Generating temporally coherent high fidelity video is an important milestone in generative modeling research. We make progress towards this milestone by proposing a diffusion model for video generation that shows very promising initial results. Our model is a natural extension of the standard image diffusion architecture, and it enables jointly training from image and video data, which we find to reduce the variance of minibatch gradients and speed up optimization. To generate long and higher resolution videos we introduce a new conditional sampling technique for spatial and temporal video extension that performs better than previously proposed methods. We present the first results on a large text-conditioned video generation task, as well as state-of-the-art results on established benchmarks for video prediction and unconditional video generation. Supplementary material is available at https://video-diffusion.github.io/.

1 Introduction

Diffusion models have recently been producing high quality results in image generation and audio generation [e.g. 28, 39, 40, 16, 23, 36, 48, 60, 42, 10, 29], and there is significant interest in validating diffusion models in new data modalities. In this work, we present first results on video generation using diffusion models, for both unconditional and conditional settings.

We show that high quality videos can be generated using essentially the standard formulation of the Gaussian diffusion model [46], with little modification other than straightforward architectural changes to accommodate video data within the memory constraints of deep learning accelerators. We train models that generate a fixed number of video frames using a 3D U-Net diffusion model architecture, and we enable generating longer videos by applying this model autoregressively using a new method for conditional generation. We additionally show the benefits of joint training on video and image modeling objectives. We test our methods on video prediction and unconditional video generation, where we achieve state-of-the-art sample quality scores, and we also show promising first results on text-conditioned video generation.

2 Background

A diffusion model [46, 47, 22] specified in continuous time [53, 48, 10, 28] is a generative model with latents $\mathbf{z} = \{\mathbf{z}_t | t \in [0, 1]\}$ obeying a forward process $q(\mathbf{z}|\mathbf{x})$ starting at data $\mathbf{x} \sim p(\mathbf{x})$. The forward process is a Gaussian process that satisfies the Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t\mathbf{x}, \sigma_t^2\mathbf{I}), \quad q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2\mathbf{I}) \tag{1}$$

where $0 \leq s < t \leq 1$, $\sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$, and α_t, σ_t specify a differentiable noise schedule whose log signal-to-noise-ratio $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ decreases with t until $q(\mathbf{z}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

*Equal contribution

arXiv:2112.10752v2 [cs.CV] 13 Apr 2022

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML

https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [66, 67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

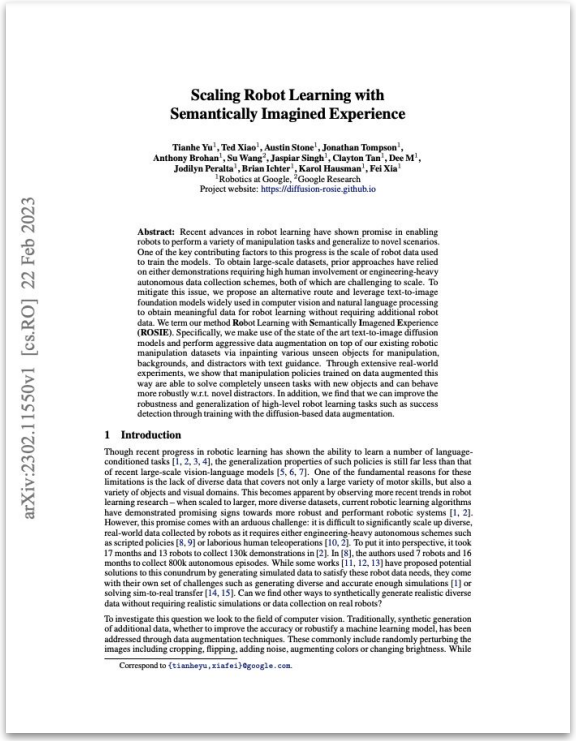
*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [30, 85] and beyond [7, 45, 48, 57], and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

1



arXiv:2302.11550v1 [cs.RO] 22 Feb 2023

Scaling Robot Learning with Semantically Imagined Experience

Tianhe Yu[1], Ted Xiao[1], Austin Stone[1], Jonathan Tompson[1], Anthony Brohan[1], Su Wang[2], Jaspiar Singh[1], Clayton Tan[1], Dee M[1], Jodilyn Peralta[1], Brian Ichter[1], Karol Hausman[1], Fei Xia[1]
[1]Robotics at Google, [2]Google Research
Project website: https://diffusion-rosie.github.io

Abstract: Recent advances in robot learning have shown promise in enabling robots to perform a variety of manipulation tasks and generalize to novel scenarios. One of the key contributing factors to this progress is the scale of robot data used to train the models. To obtain large-scale datasets, prior approaches have relied on either demonstrations requiring high human involvement or engineering-heavy autonomous data collection schemes, both of which are challenging to scale. To mitigate this issue, we propose an alternative route and leverage text-to-image foundation models widely used in computer vision and natural language processing to obtain meaningful data for robot learning without requiring additional robot data. We term our method **Ro**bot Learning with **S**emantically Imagened **E**xperience (**ROSIE**). Specifically, we make use of the state of the art text-to-image diffusion models and perform aggressive data augmentation on top of our existing robotic manipulation datasets via inpainting various unseen objects for manipulation, backgrounds, and distractors with text guidance. Through extensive real-world experiments, we show that manipulation policies trained on data augmented this way are able to solve completely unseen tasks with new objects and can behave more robustly w.r.t. novel distractors. In addition, we find that we can improve the robustness and generalization of high-level robot learning tasks such as success detection through training with the diffusion-based data augmentation.

1 Introduction

Though recent progress in robotic learning has shown the ability to learn a number of language-conditioned tasks [1, 2, 3, 4], the generalization properties of such policies is still far less than that of recent large-scale vision-language models [5, 6, 7]. One of the fundamental reasons for these limitations is the lack of diverse data that covers not only a large variety of motor skills, but also a variety of objects and visual domains. This becomes apparent by observing more recent trends in robot learning research – when scaled to larger, more diverse datasets, current robotic learning algorithms have demonstrated promising signs towards more robust and performant robotic systems [1, 2]. However, this promise comes with an arduous challenge: it is difficult to significantly scale up diverse, real-world data collected by robots as it requires either engineering-heavy autonomous schemes such as scripted policies [8, 9] or laborious human teleoperations [10, 2]. To put it into perspective, it took 17 months and 13 robots to collect 130k demonstrations in [2]. In [8], the authors used 7 robots and 16 months to collect 800k autonomous episodes. While some works [11, 12, 13] have proposed potential solutions to this conundrum by generating simulated data to satisfy these robot data needs, they come with their own set of challenges such as generating diverse and accurate enough simulations [1] or solving sim-to-real transfer [14, 15]. Can we find other ways to synthetically generate realistic diverse data without requiring realistic simulations or data collection on real robots?

To investigate this question we look to the field of computer vision. Traditionally, synthetic generation of additional data, whether to improve the accuracy or robustify a machine learning model, has been addressed through data augmentation techniques. These commonly include randomly perturbing the images including cropping, flipping, adding noise, augmenting colors or changing brightness. While

Correspond to {tianheyu,xiaf ei}@google.com.

Text-to-image diffusion for robotics 🤖

Today — Reminder: Midway Report

Midway report due date is today (by **EoD, anywhere on earth**).

- Midway report — we are expecting a ~4 page report, using the LaTeX template (Overleaf) shared with the class.
- Each group should have one teammate upload a **single PDF file** on Brightspace.

Today — Reminder: Also, Scribe and Discussion Lead PDFs

Reminder about submitting scribe and discussion lead assignments:

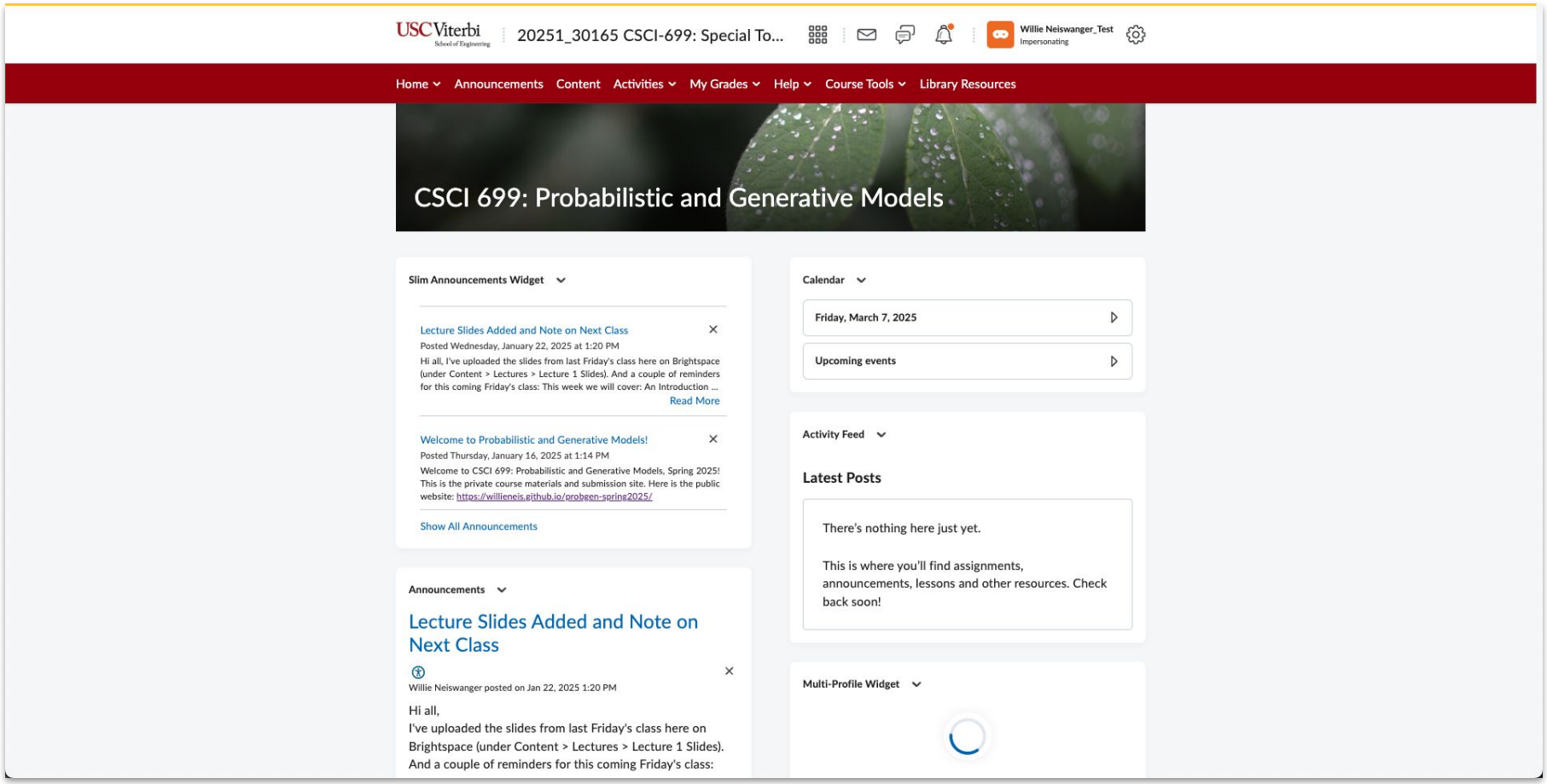
- As you know, discussion leads bring ~5 questions on their assigned paper, and scribes write up ~1 page document.
- Both assignments should be uploaded as a **PDF file** on Brightspace.
- Ideally within a ~week of your scribe/discussion lead duties.

Today — Reminder

Here’s how to upload:

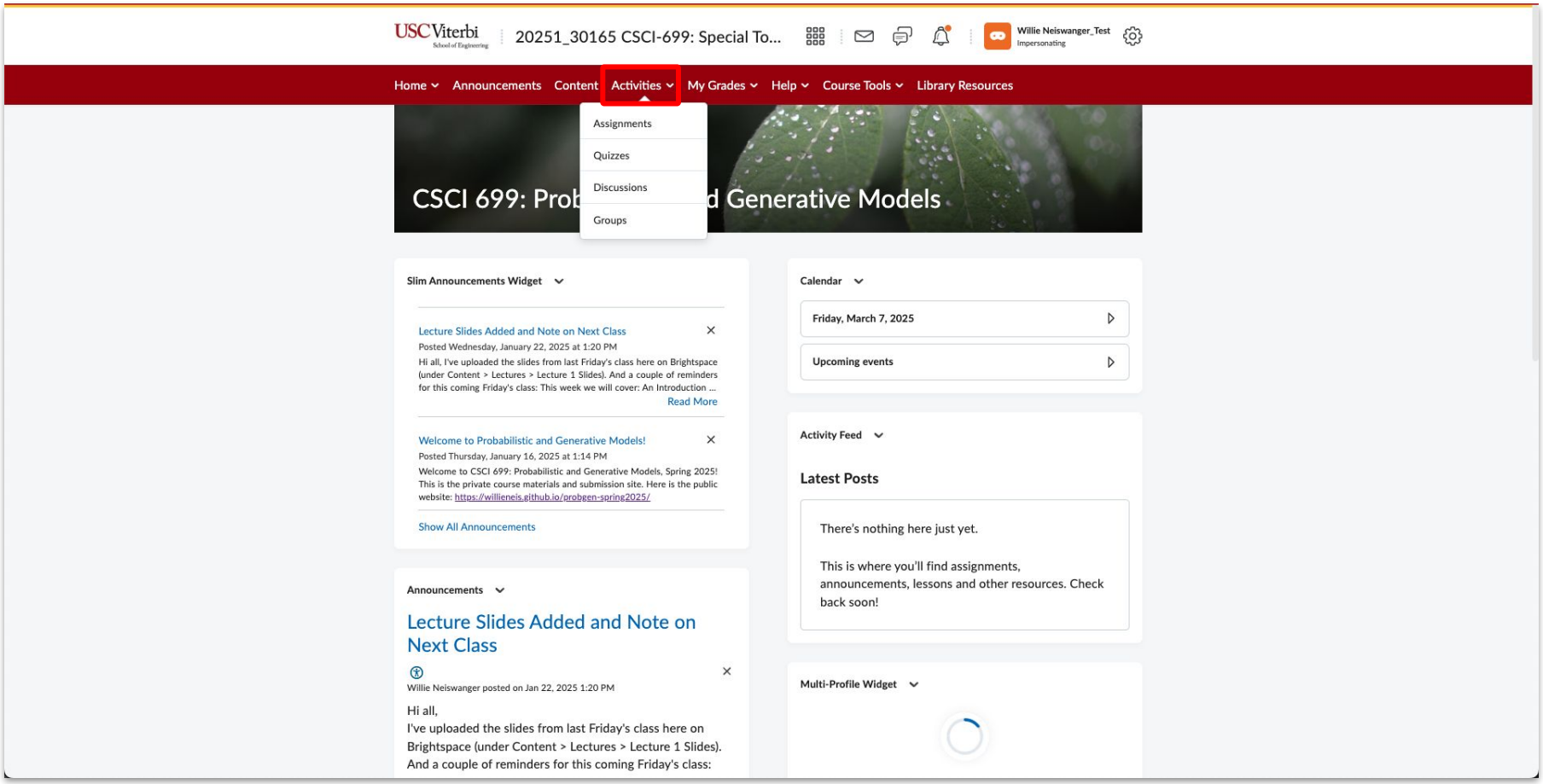
Today — Reminder

Here's how to upload: (1) Go to main course homepage on Brightspace.



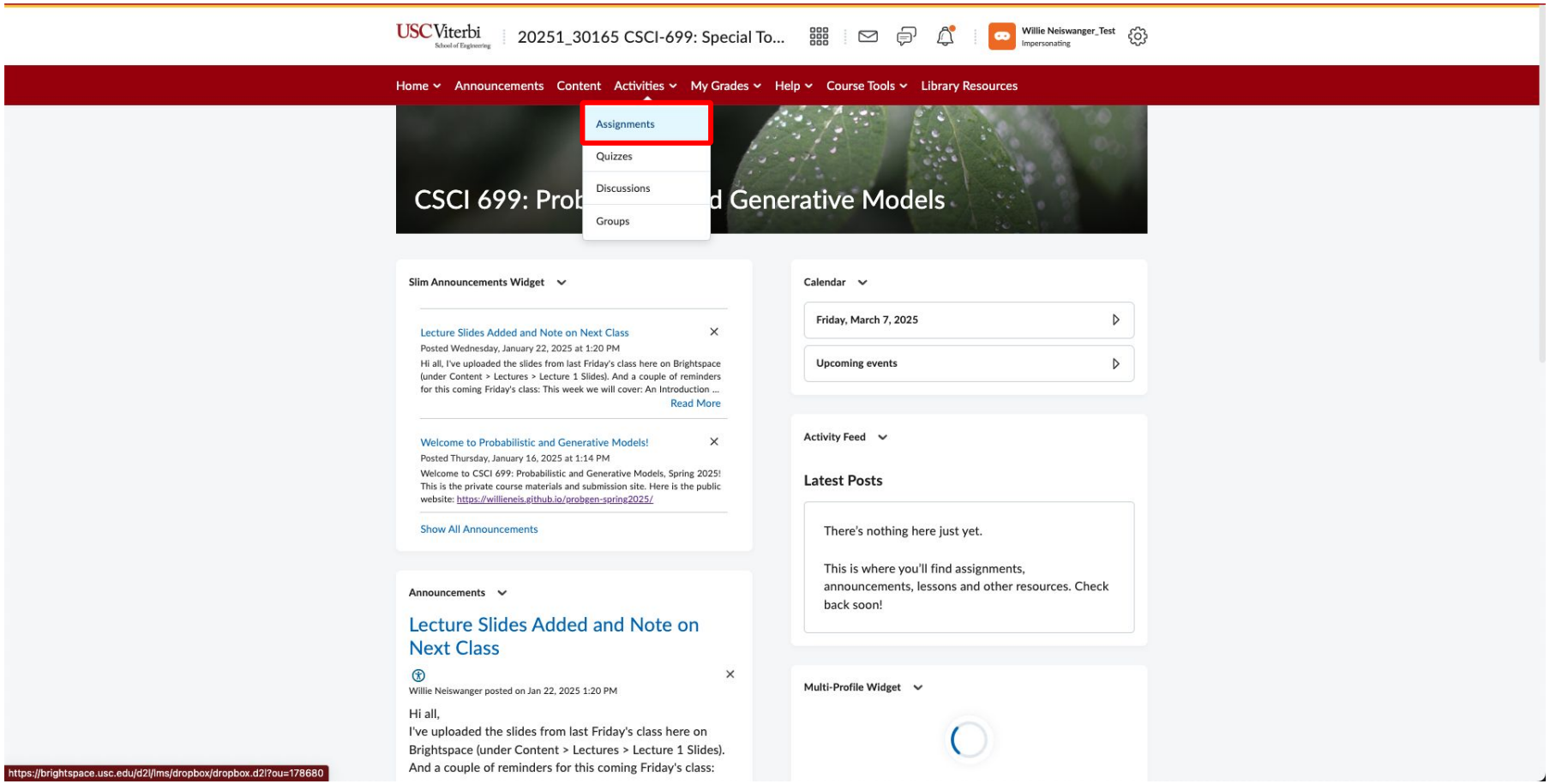
Today — Reminder

Here's how to upload: (2) Click on *Activities* > *Assignments* in menu bar at top.



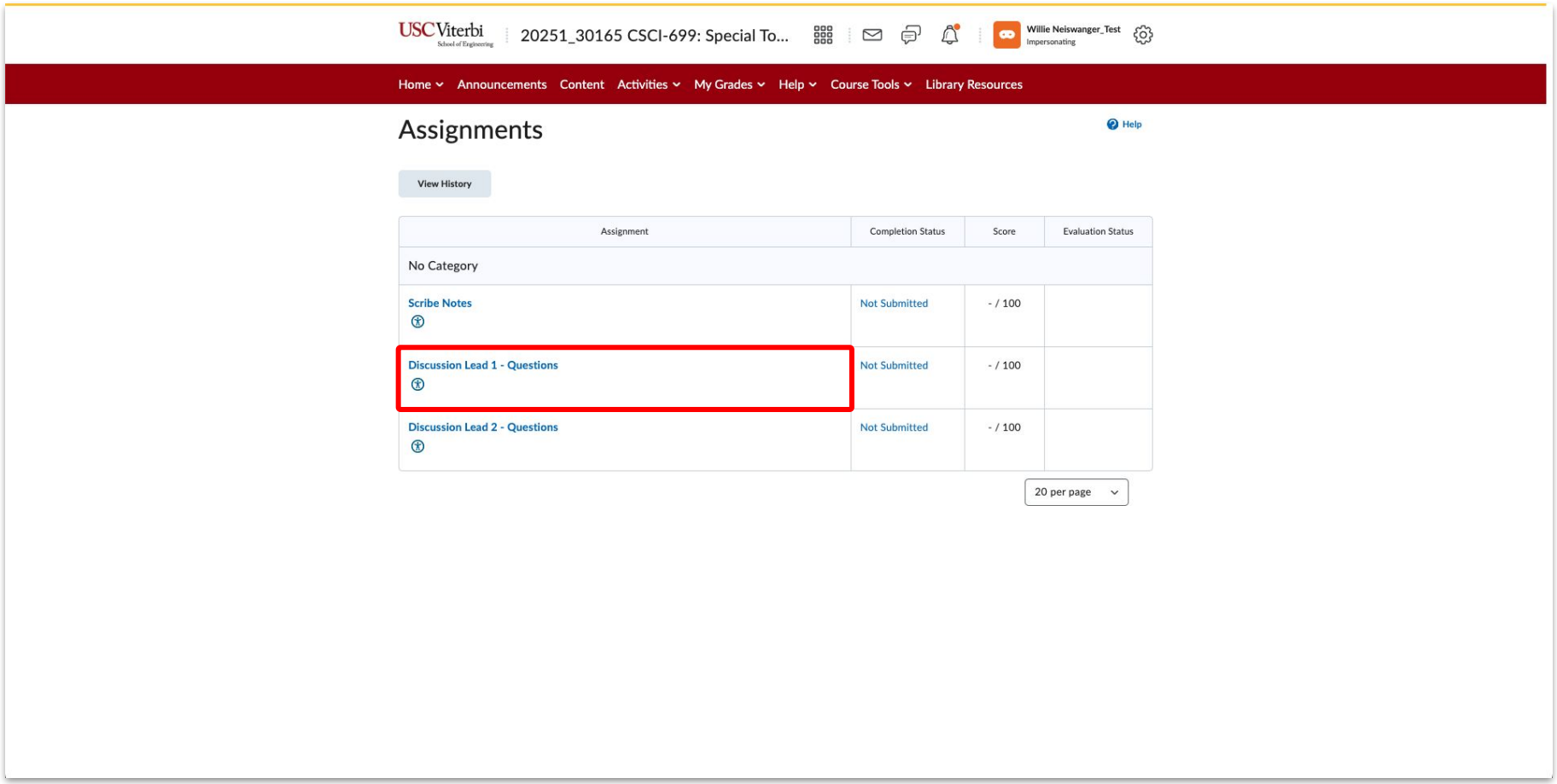
Today — Reminder

Here's how to upload: (2) Click on *Activities* > *Assignments* in menu bar at top.



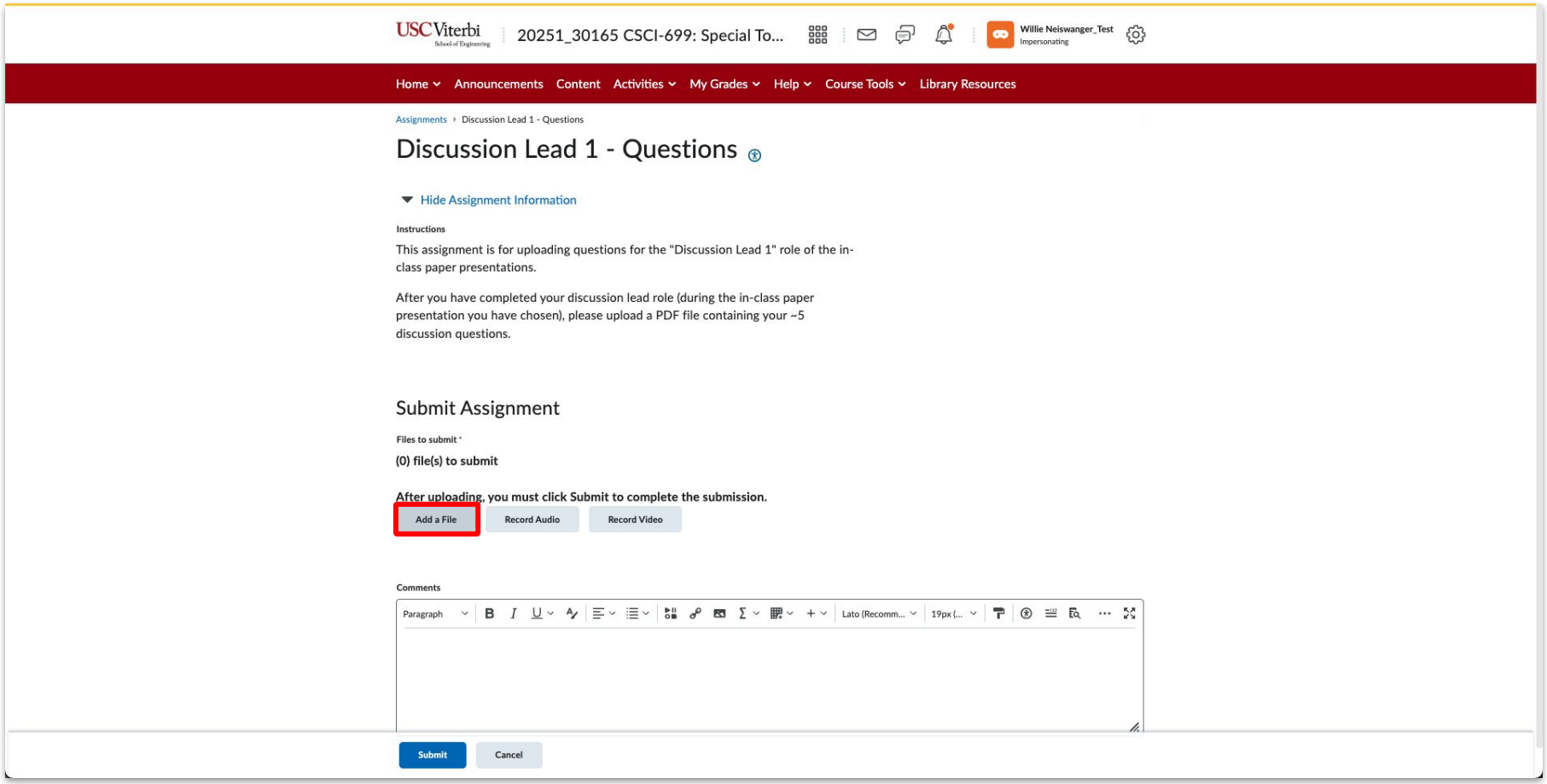
Today — Reminder

Here's how to upload: (3) Click on any of the listed assignments.



Today — Reminder

Here's how to upload: (4) Upload PDF (under *Add a File*)



Goals for Final DGM Lecture

Goals for Final DGM Lecture

(1) Make sure score matching (+ noise-conditional score matching) is clear.

Goals for Final DGM Lecture

(1) Make sure score matching (+ noise-conditional score matching) is clear.

(2) Explain the connections/equivalence to diffusion models.

Goals for Final DGM Lecture

(1) Make sure score matching (+ noise-conditional score matching) is clear.

(2) Explain the connections/equivalence to diffusion models.

(3) Discuss relation between: score-matching, diffusion, and VAE generative models.

- And "*implicit*" probabilistic models (score-based, flow) vs likelihood-based models (diffusion, VAE).

Review/Finish: Score-based Generative Models

Score-based Generative Models — Setup

Score-based Generative Models — Setup

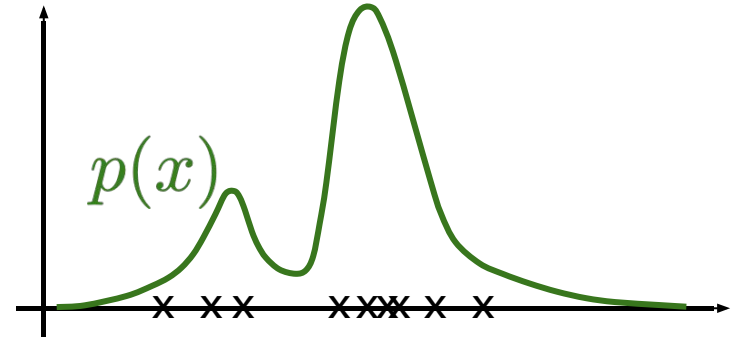
Suppose that we have a dataset: $D = \{x_1, x_2, \ldots, x_n\}$

Where each data point is assumed to be drawn from some unknown distribution, *i.e.*,

$$x_i \sim p(x)$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup



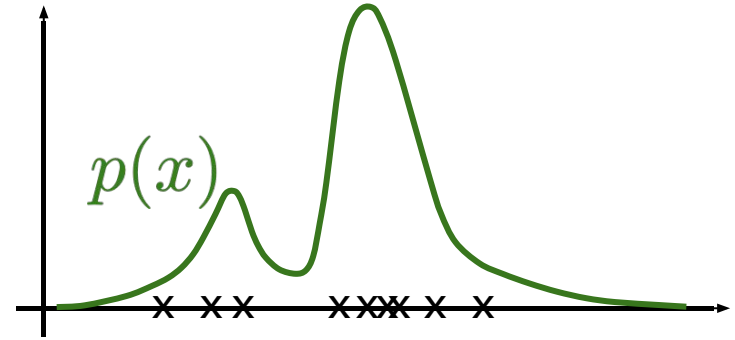
Suppose that we have a dataset: $D = \{x_1, x_2, \ldots, x_n\}$

Where each data point is assumed to be drawn from some unknown distribution, *i.e.*,

$$x_i \sim p(x)$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup



Suppose that we have a dataset: $D = \{x_1, x_2, \ldots, x_n\}$

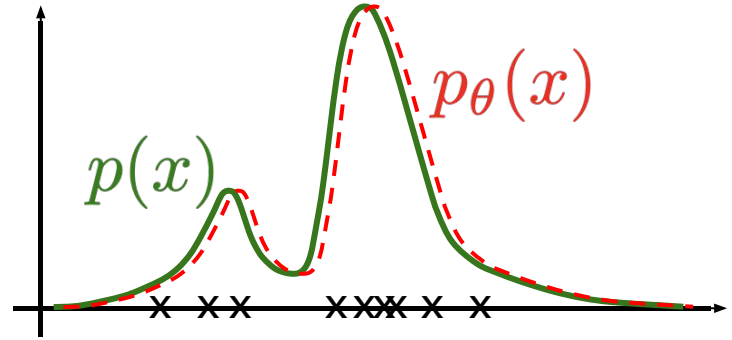
Where each data point is assumed to be drawn from some unknown distribution, *i.e.*,

$$x_i \sim p(x)$$

Suppose we want to model this distribution using a model $p_\theta(x)$, where we assume this model is parameterized by a $\theta \in \mathbb{R}^d$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup



Suppose that we have a dataset: $D = \{x_1, x_2, \ldots, x_n\}$

Where each data point is assumed to be drawn from some unknown distribution, *i.e.*,

$$x_i \sim p(x)$$

Suppose we want to model this distribution using a model $p_\theta(x)$, where we assume this model is parameterized by a $\theta \in \mathbb{R}^d$.

⇒ We want to learn θ such that $p_\theta(x) \approx p(x)$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup

We can write this model $p_\theta(x)$ in a general way as:

Score-based Generative Models — Setup

We can write this model $p_\theta(x)$ in a general way as:

$$p_\theta(x) = \frac{1}{Z(\theta)} \tilde{p}_\theta(x) = \frac{1}{Z(\theta)} e^{-f_\theta(x)}$$

Where, as before,

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup

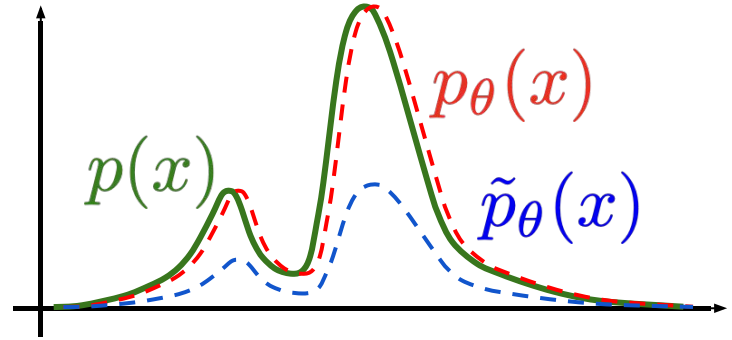
We can write this model $p_\theta(x)$ in a general way as:

$$p_\theta(x) = \frac{1}{Z(\theta)} \tilde{p}_\theta(x) = \frac{1}{Z(\theta)} e^{-f_\theta(x)}$$

Where, as before, $Z(\theta)$ is the normalization constant.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup



We can write this model $p_\theta(x)$ in a general way as:

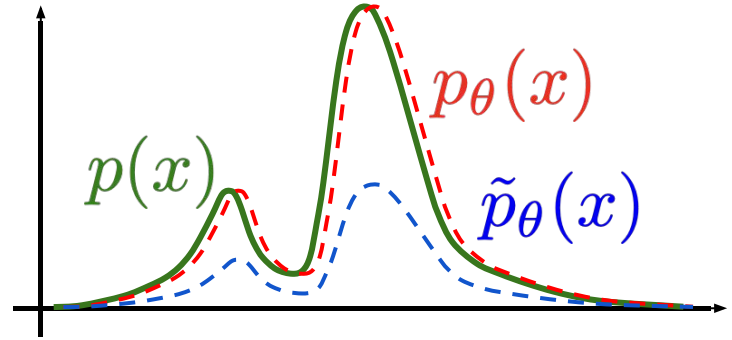
$$p_\theta(x) = \frac{1}{Z(\theta)} \tilde{p}_\theta(x) = \frac{1}{Z(\theta)} e^{-f_\theta(x)}$$

Where, as before, $Z(\theta)$ is the normalization constant.

And $\tilde{p}_\theta(x)$ is an unnormalized PDF.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Setup



We can write this model $p_\theta(x)$ in a general way as:

$$p_\theta(x) = \frac{1}{Z(\theta)} \tilde{p}_\theta(x) = \frac{1}{Z(\theta)} e^{-f_\theta(x)}$$

Where, as before, $Z(\theta)$ is the normalization constant.

And $\tilde{p}_\theta(x)$ is an unnormalized PDF.

And $f_\theta(x)$ is equal to the negative log of the unnormalized PDF, *i.e.*,

$$f_\theta(x) = -\log \tilde{p}_\theta(x)$$

Sometimes called an *energy-based model*.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Difficulties with MLE

Goal: to perform generative modeling, we want to learn model parameters θ .

Score-based Generative Models — Difficulties with MLE

Goal: to perform generative modeling, we want to learn model parameters θ .

Previous strategies involve trying to learning parameters θ via maximum likelihood:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta(x_i)$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Difficulties with MLE

Goal: to perform generative modeling, we want to learn model parameters θ .

Previous strategies involve trying to learning parameters θ via maximum likelihood:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta(x_i)$$

I.e., maximum (marginal) likelihood, *aka* MLE

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Difficulties with MLE

Goal: to perform generative modeling, we want to learn model parameters θ.

Previous strategies involve trying to learning parameters θ via maximum likelihood:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta(x_i)$$

I.e., maximum (marginal) likelihood, *aka* MLE

However:

- To optimize (or evaluate) $p_\theta(x)$, we need the normalization constant $Z(\theta)$.
- …which is intractable in general for most $\tilde{p}_\theta(x)$.
- In VAEs, we perform MLE on θ via the use of an approximate posterior q.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

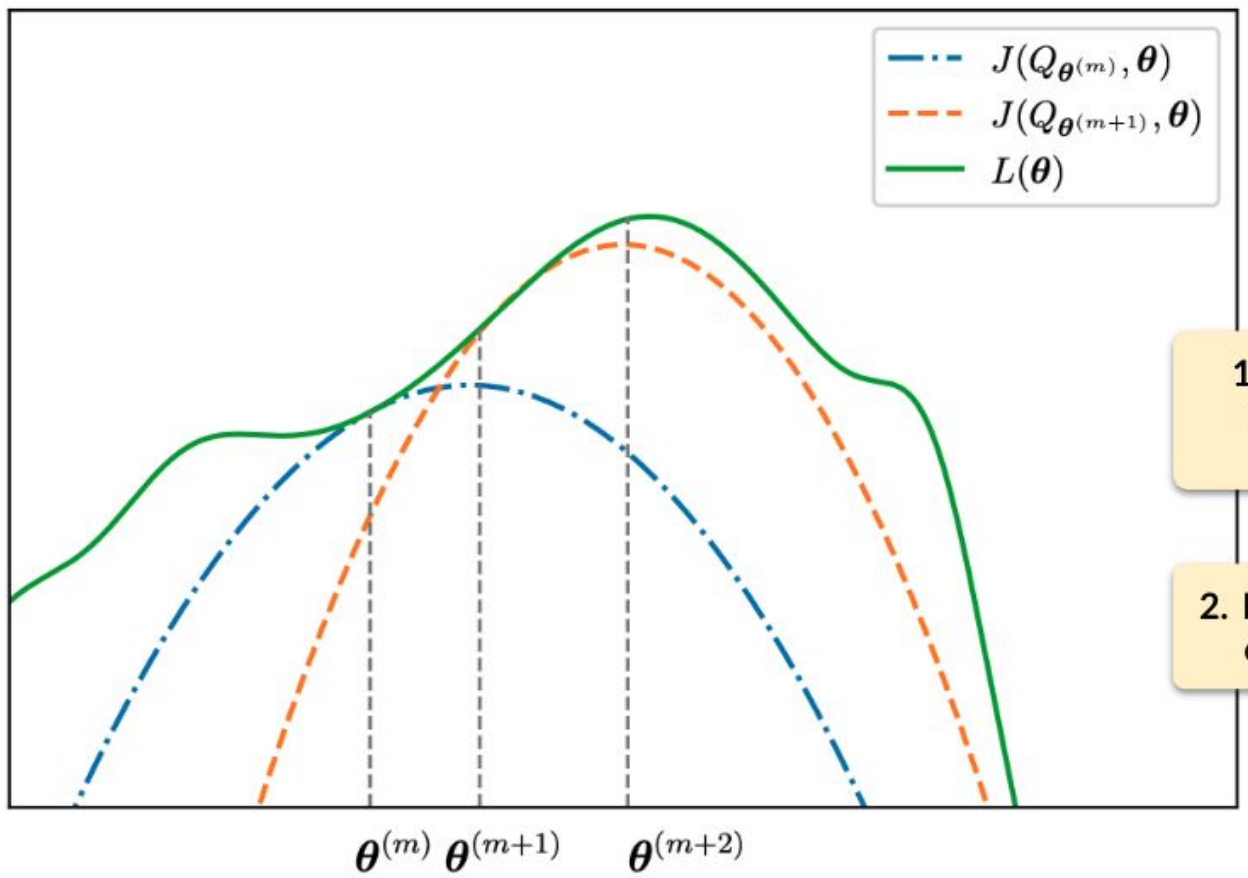
Score-based Generative Models — Difficulties with MLE

Goa

Prev ihood:

Hov

- $\theta)$.
- In VAEs, we perform MLE on θ via the use of an approximate posterior q.



Iteratively forming then optimizing lower bounds to the (*marginal) likelihood* in latent-variable models.

1. **E-Step:** do inference to compute next objective (next lower bound)

2. **M-Step:** maximize this new objective (lower bound).

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Main Strategy

So instead, in score-based models, we will take the following strategy:

Score-based Generative Models — Main Strategy

So instead, in score-based models, we will take the following strategy:

(1) **First**, we will aim to learn what is referred to as the **(Stein) score function** of the data density.

(2) **Second**, Given this score function, we will be able to draw samples from $p_\theta(x)$ using Langevin Monte Carlo.

Score-based Generative Models — Main Strategy

So instead, in score-based models, we will take the following strategy:

(1) **First**, we will aim to learn what is referred to as the **(Stein) score function** of the data density.

(2) **Second**, Given this score function, we will be able to draw samples from $p_\theta(x)$ using Langevin Monte Carlo.

Going through these two steps…

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

Gradient of the PDF with respect to x

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

Gradient of the PDF with respect to x

Note that:

- The term *score* is also sometimes used to refer to the gradient of the log-likelihood function with respect to the parameter θ.
- We've seen this term before! (Recall in LMC/HMC…).

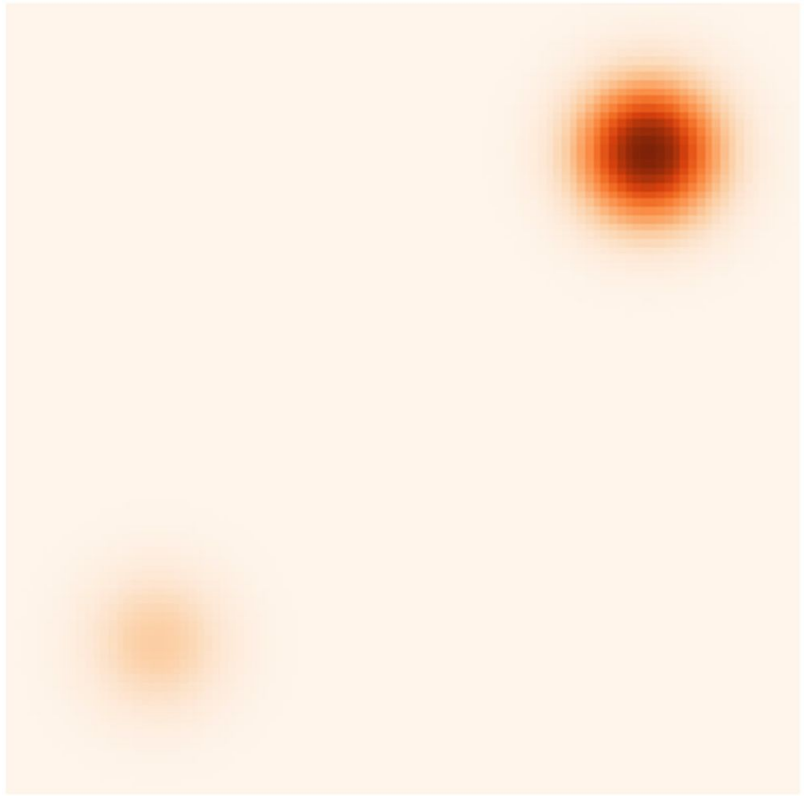
Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

Illustrating this!

Score-based Generative Models

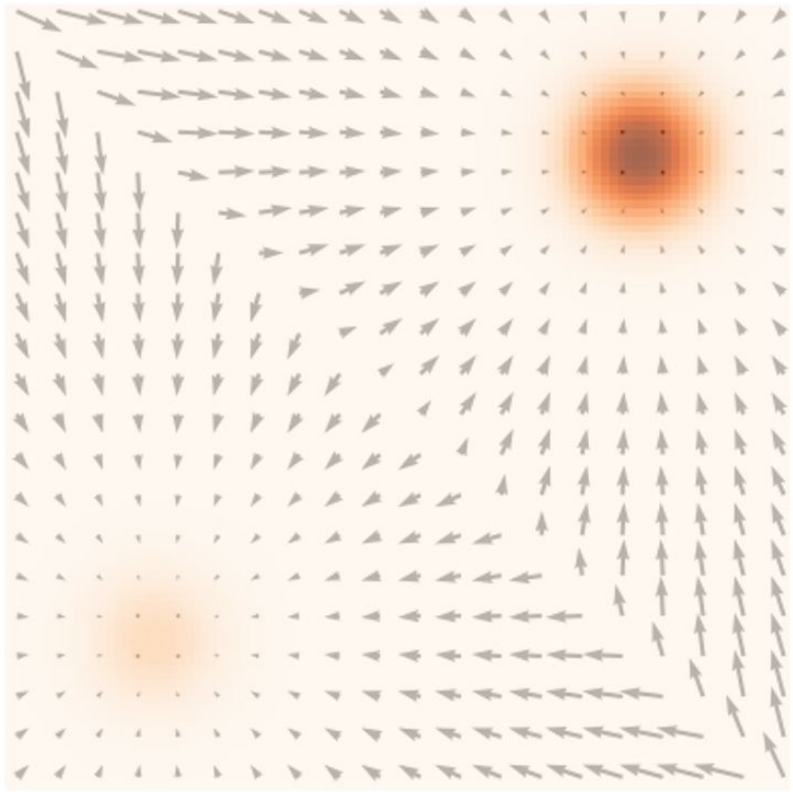
First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,



PDF $p(x)$

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

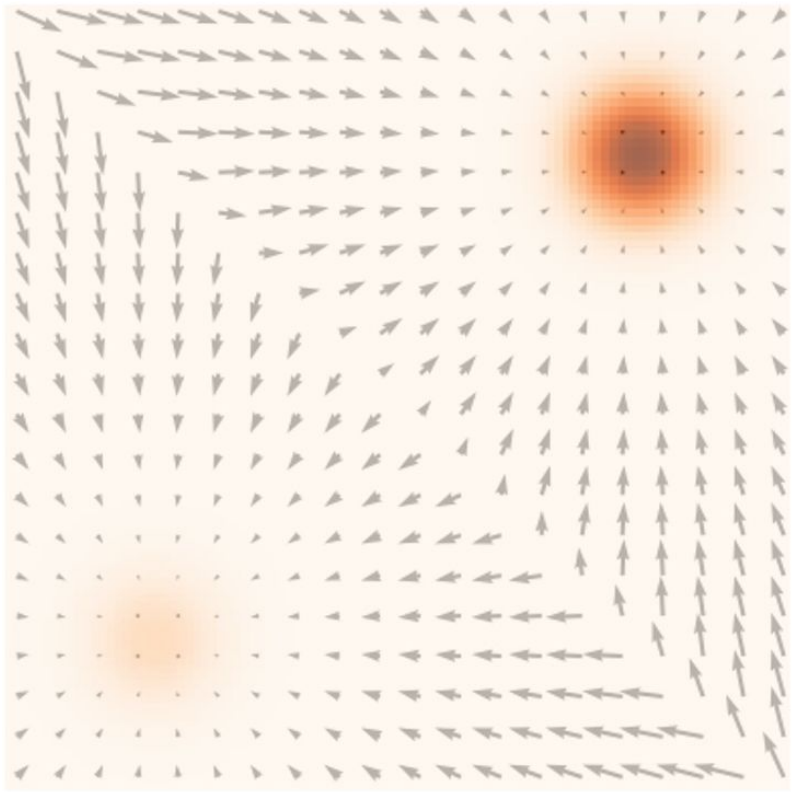


Score fn $s(x) = \nabla_x \log p(x)$

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

"Points" to the high-density regions of the PDF

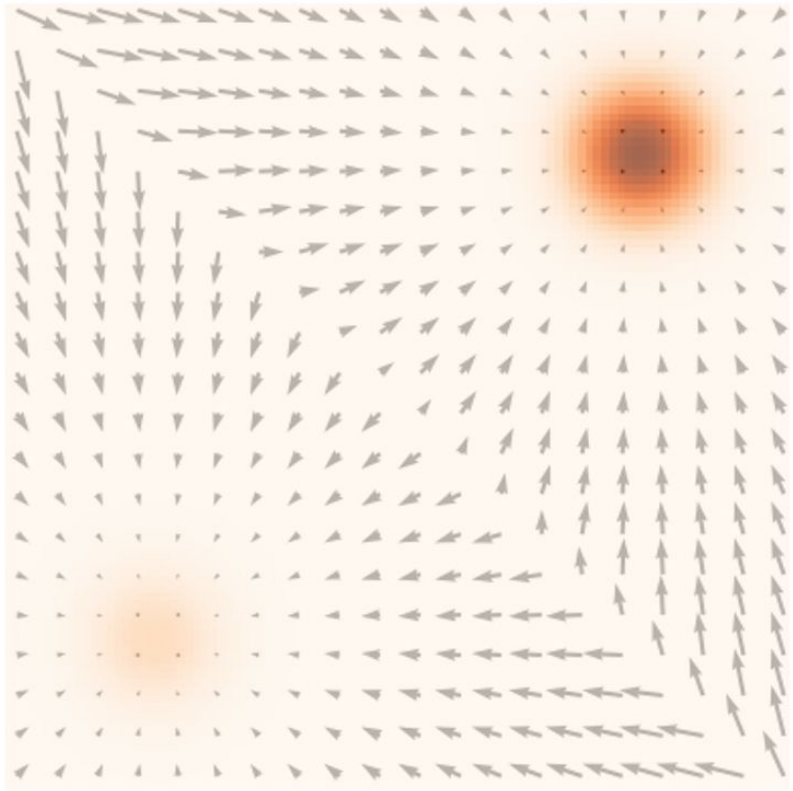


Score fn $s(x) = \nabla_x \log p(x)$

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

"Points" to the high-density regions of the PDF



Score fn $s(x) = \nabla_x \log p(x)$

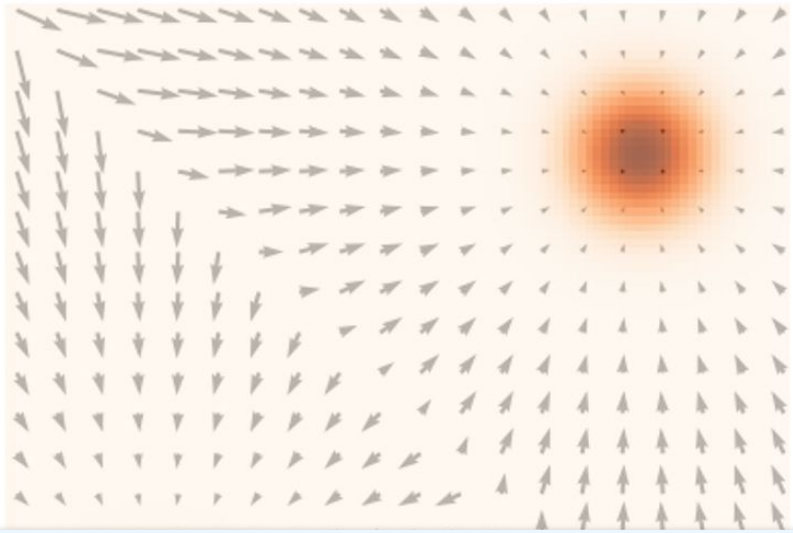
Also note:

$$\nabla_x \log p(x) = \nabla_x \log \tilde{p}(x)$$

Score-based Generative Models

First, we will aim to learn the **(Stein) score function** of the data density $p(x)$, *i.e.*,

"Points" to the high-density regions of the PDF



Score fn $s(x) = \nabla_x \log p(x)$

Also note:

$$\nabla_x \log p(x) = \nabla_x \log \tilde{p}(x)$$

Recall that in Langevin Monte Carlo (LMC):

- In order to sample from a PDF, you just need access to the gradient of the unnormalized PDF!

Score-based Generative Models

Therefore (after learning the score function):

Score-based Generative Models

Therefore (after learning the score function):

We can simply run LMC with our learned score function $s_\theta(x)$:

$$x_{i+1} \leftarrow x_i + \epsilon s_\theta(x) + \sqrt{2\epsilon}\,\mathcal{N}(0, I), \quad i = 0, 1, \cdots, K,$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

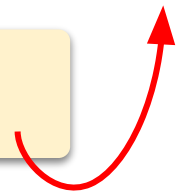
Score-based Generative Models

Therefore (after learning the score function):

We can simply run LMC with our learned score function $s_\theta(x)$:

$$x_{i+1} \leftarrow x_i + \epsilon s_\theta(x) + \sqrt{2\epsilon}\,\mathcal{N}(0, I), \quad i = 0, 1, \cdots, K,$$

Note the learned score function is swapped in.



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models

Therefore (after learning the score function):

We can simply run LMC with our learned score function $s_\theta(x)$:

$$x_{i+1} \leftarrow x_i + \epsilon s_\theta(x) + \sqrt{2\epsilon}\,\mathcal{N}(0, I), \quad i = 0, 1, \cdots, K,$$

Note the learned score function is swapped in.

This is equivalent to previous LMC, but step size is written slightly different here.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — An illustration

An illustration:

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — An illustration

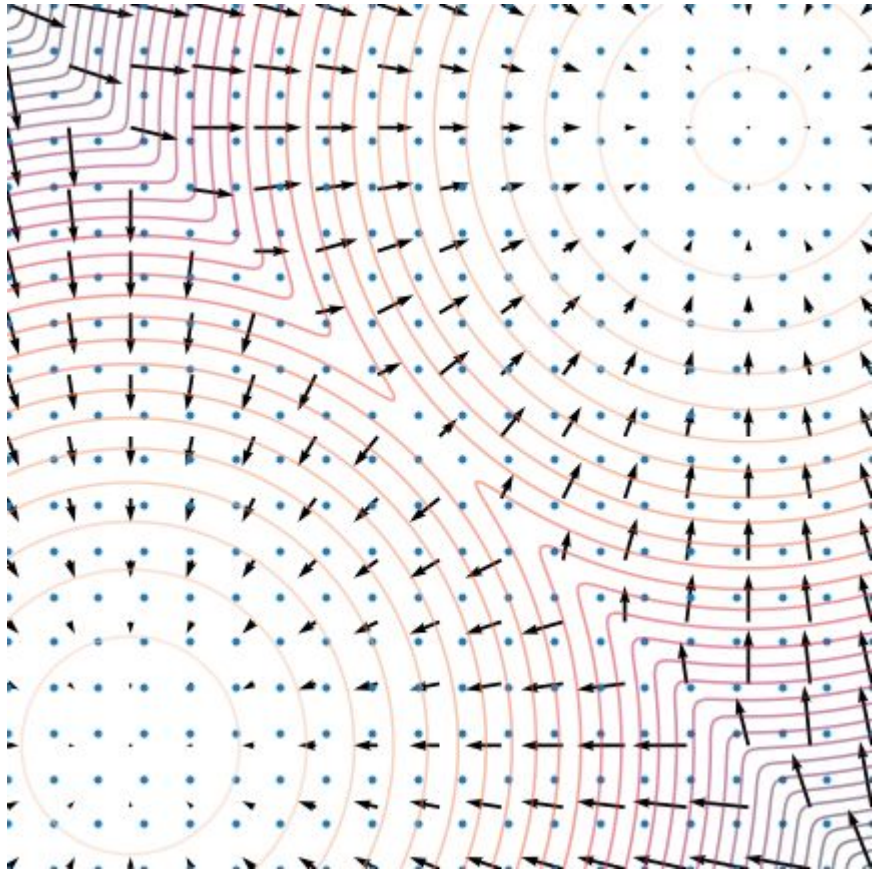
An illustration:

Contour lines denote the PDF $p_\theta(x)$.

Black arrows denote the score function $s_\theta(x)$.

Blue dots denote the samples produced via LMC.

(Many chains, randomly initialized).



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

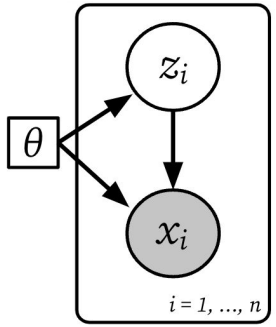
Score-based Generative Models — Intuition on Learning

What are we learning here (*i.e.*, what is neural network doing) versus in VAEs?

Score-based Generative Models — Intuition on Learning

What are we learning here (*i.e.*, what is neural network doing) versus in VAEs?

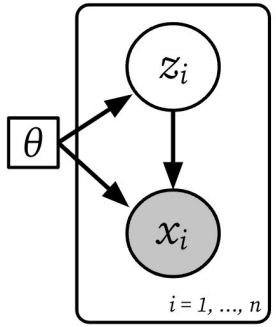
- In VAEs, we aim to learn a **direct map** from noise to data.
  - *I.e.*, directly learn the PDF $p_\theta(x)$ of interest.
  - ⇒ a neural network that generates a sample from this PDF.



Score-based Generative Models — Intuition on Learning

What are we learning here (*i.e.*, what is neural network doing) versus in VAEs?

- In VAEs, we aim to learn a **direct map** from noise to data.
  - *I.e.*, directly learn the PDF $p_\theta(x)$ of interest.
  - ⇒ a neural network that generates a sample from this PDF.

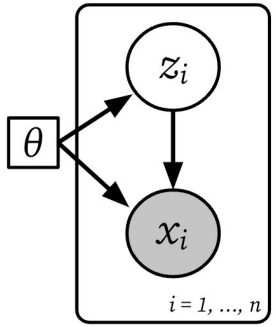


- In score-based models, we aim to learn a **score function** $s_\theta(x)$.

Score-based Generative Models — Intuition on Learning

What are we learning here (*i.e.*, what is neural network doing) versus in VAEs?

- In VAEs, we aim to learn a **direct map** from noise to data.
  - *I.e.*, directly learn the PDF $p_\theta(x)$ of interest.
  - ⇒ a neural network that generates a sample from this PDF.



- In score-based models, we aim to learn a **score function** $s_\theta(x)$.
  - Which we can view as pointing us in the direction of steepest ascent of PDF.
    ⇒ Use neural network *repeatedly* to take gradient steps (along with noise) to sample from this PDF.
  - The associated probabilistic model $p_\theta(x)$ is defined *implicitly*.

Score-based Generative Models

So the main **question** is: How do we learn an approximation $s_\theta(x)$ to the score function $s(x) = \nabla_x \log p(x)$?

Score-based Generative Models

So the main **question** is: How do we learn an approximation $s_\theta(x)$ to the score function $s(x) = \nabla_x \log p(x)$?

⇒ **Answer**: we will minimize the (so called) *Fisher divergence* between the model and data distributions, written as:

$$\mathcal{L}(\theta) = \mathbb{E}_{p(x)} \left[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2 \right]$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models

So the main **question** is: How do we learn an approximation $s_\theta(x)$ to the score function $s(x) = \nabla_x \log p(x)$?

⇒ **Answer**: we will minimize the (so called) *Fisher divergence* between the model and data distributions, written as:

$$\mathcal{L}(\theta) = \mathbb{E}_{p(x)} \left[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2 \right]$$

Expectation taken over the data density.

Intuitively: this compares the squared L2 distance between the ground-truth data score function and the score-based model.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models

So the main **question** is: How do we learn an approximation $s_\theta(x)$ to the score function $s(x) = \nabla_x \log p(x)$?

⇒ **Answer**: we will minimize the (so called) *Fisher divergence* between the model and data distributions, written as:

$$\mathcal{L}(\theta) = \mathbb{E}_{p(x)} \left[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2 \right]$$

Expectation taken over the data density.

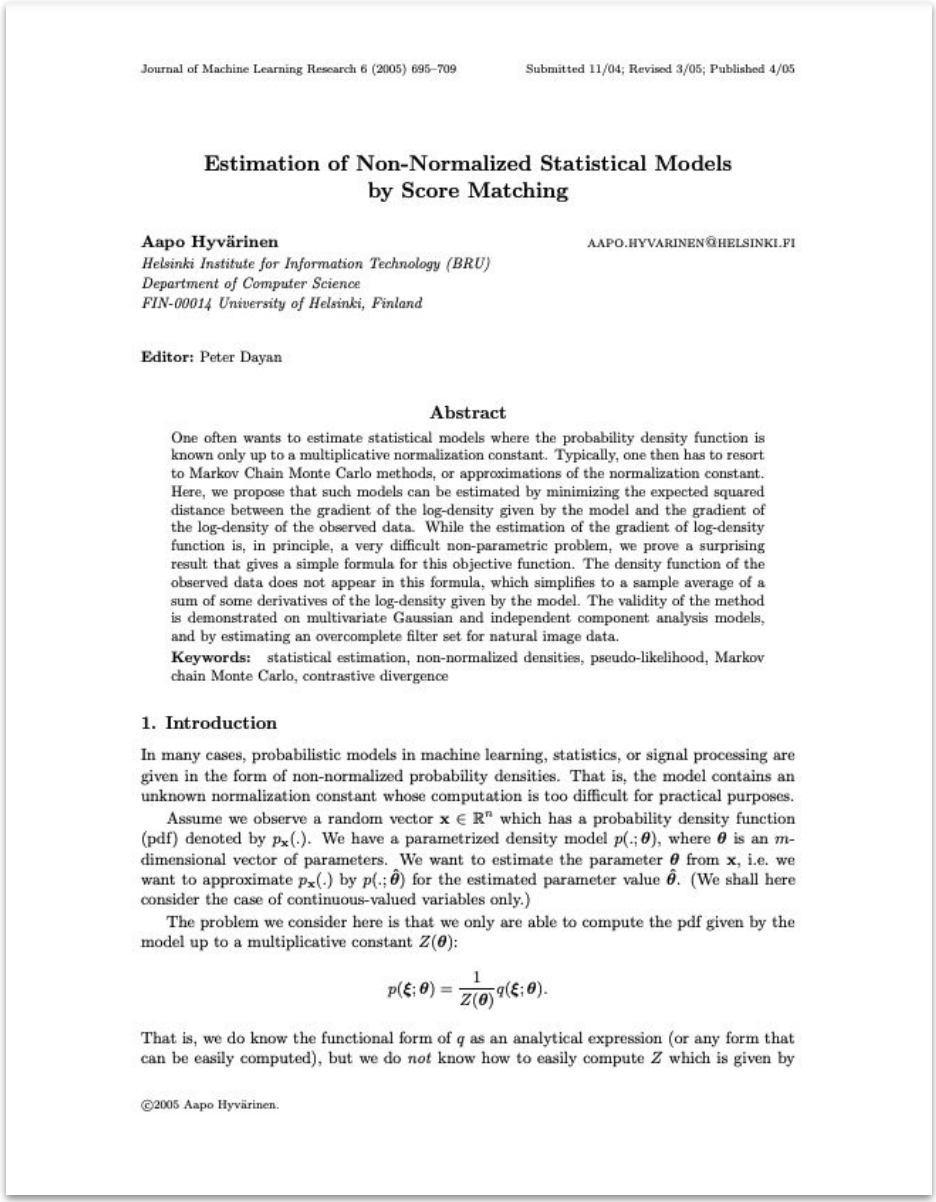
However, there is an **issue**: we don't know the ground truth PDF $p(x)$ nor ground truth data score $\nabla_x \log p(x)$).

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models

Solution: Use a technique called **Score Matching.**

This will let us minimize the Fisher divergence without knowledge of the ground-truth data score.



Journal of Machine Learning Research 6 (2005) 695–709 Submitted 11/04; Revised 3/05; Published 4/05

Estimation of Non-Normalized Statistical Models by Score Matching

Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Helsinki Institute for Information Technology (BRU)
Department of Computer Science
FIN-00014 University of Helsinki, Finland

Editor: Peter Dayan

Abstract

One often wants to estimate statistical models where the probability density function is known only up to a multiplicative normalization constant. Typically, one then has to resort to Markov Chain Monte Carlo methods, or approximations of the normalization constant. Here, we propose that such models can be estimated by minimizing the expected squared distance between the gradient of the log-density given by the model and the gradient of the log-density of the observed data. While the estimation of the gradient of log-density function is, in principle, a very difficult non-parametric problem, we prove a surprising result that gives a simple formula for this objective function. The density function of the observed data does not appear in this formula, which simplifies to a sample average of a sum of some derivatives of the log-density given by the model. The validity of the method is demonstrated on multivariate Gaussian and independent component analysis models, and by estimating an overcomplete filter set for natural image data.

Keywords: statistical estimation, non-normalized densities, pseudo-likelihood, Markov chain Monte Carlo, contrastive divergence

1. Introduction

In many cases, probabilistic models in machine learning, statistics, or signal processing are given in the form of non-normalized probability densities. That is, the model contains an unknown normalization constant whose computation is too difficult for practical purposes.

Assume we observe a random vector $\mathbf{x} \in \mathbb{R}^n$ which has a probability density function (pdf) denoted by $p_{\mathbf{x}}(.)$. We have a parametrized density model $p(.;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is an m-dimensional vector of parameters. We want to estimate the parameter $\boldsymbol{\theta}$ from $\mathbf{x}$, i.e. we want to approximate $p_{\mathbf{x}}(.)$ by $p(.;\hat{\boldsymbol{\theta}})$ for the estimated parameter value $\hat{\boldsymbol{\theta}}$. (We shall here consider the case of continuous-valued variables only.)

The problem we consider here is that we only are able to compute the pdf given by the model up to a multiplicative constant $Z(\boldsymbol{\theta})$:

$$p(\boldsymbol{\xi};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} q(\boldsymbol{\xi};\boldsymbol{\theta}).$$

That is, we do know the functional form of q as an analytical expression (or any form that can be easily computed), but we do *not* know how to easily compute Z which is given by

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

And then we went through a long derivation to get an alternative expression for the Fisher divergence objective, which came out to be …

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Can write the Fisher divergence as follows:

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\operatorname{tr}\left(\nabla_x s_\theta(x)\right) + \frac{1}{2}\|s_\theta(x)\|_2^2\right]$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Can write the Fisher divergence as follows:

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\text{tr}\left(\nabla_x s_\theta(x)\right) + \frac{1}{2}\|s_\theta(x)\|_2^2\right]$$

"Expectation of: *trace–of-gradient-of-score-model, and norm-of-score-model*"

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Can write the Fisher divergence as follows:

(equivalently…)

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\operatorname{tr}\left(\nabla_x^2 \log p_\theta(x)\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x)\|_2^2\right]$$

"Expectation of: *trace–of-gradient-of-score-model, and norm-of-score-model*"

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Can write the Fisher divergence as follows:

Note: does not involve/need $p(x)$

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\text{tr}\left(\nabla_x^2 \log p_\theta(x)\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x)\|_2^2\right]$$

"Expectation of: *trace–of-gradient-of-score-model, and norm-of-score-model*"

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Can write the Fisher divergence as follows:

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\operatorname{tr}\left(\nabla_x^2 \log p_\theta(x)\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x)\|_2^2\right]$$

⇒ Monte Carlo estimate!

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Can write the Fisher divergence as follows:

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\operatorname{tr}\left(\nabla_x^2 \log p_\theta(x)\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x)\|_2^2\right]$$

⇒ Monte Carlo estimate!

$$\approx \frac{1}{M}\sum_{m=1}^{M} \operatorname{tr}\left(\nabla_x^2 \log p_\theta(x^{(m)})\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x^{(m)})\|_2^2,$$

where $x^{(m)} \sim p(x)$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

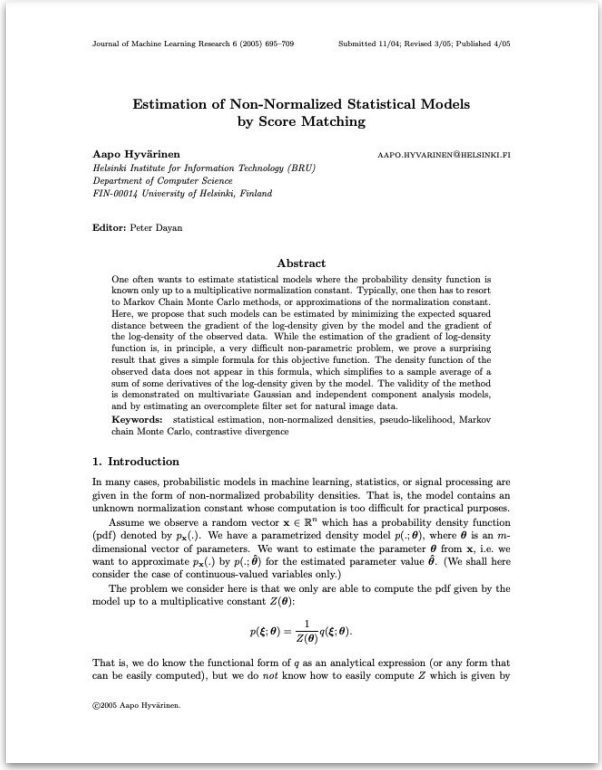
Can write the Fisher divergence as follows:

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\operatorname{tr}\left(\nabla_x^2 \log p_\theta(x)\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x)\|_2^2\right]$$

⇒ Monte Carlo estimate!

$$\approx \frac{1}{M}\sum_{m=1}^{M} \operatorname{tr}\left(\nabla_x^2 \log p_\theta(x^{(m)})\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x^{(m)})\|_2^2,$$

where $x^{(m)} \sim p(x)$



Journal of Machine Learning Research 6 (2005) 695–709 Submitted 11/04; Revised 3/05; Published 4/05

Estimation of Non-Normalized Statistical Models by Score Matching

Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Helsinki Institute for Information Technology (BRU)
Department of Computer Science
FIN-00014 University of Helsinki, Finland

Editor: Peter Dayan

Abstract

One often wants to estimate statistical models where the probability density function is known only up to a multiplicative normalization constant. Typically, one then has to resort to Markov Chain Monte Carlo methods, or approximations of the normalization constant. Here, we propose that such models can be estimated by minimizing the expected squared distance between the gradient of the log-density given by the model and the gradient of the log-density of the observed data. While the estimation of the gradient of log-density function is, in principle, a very difficult non-parametric problem, we prove a surprising result that gives a simple formula for this objective function. The density function of the observed data does not appear in this formula, which simplifies to a sample average of a sum of some derivatives of the log-density given by the model. The validity of the method is demonstrated on multivariate Gaussian and independent component analysis models, and by estimating an overcomplete filter set for natural image data.

Keywords: statistical estimation, non-normalized densities, pseudo-likelihood, Markov chain Monte Carlo, contrastive divergence

1. Introduction

In many cases, probabilistic models in machine learning, statistics, or signal processing are given in the form of non-normalized probability densities. That is, the model contains an unknown normalization constant whose computation is too difficult for practical purposes.

Assume we observe a random vector $\mathbf{x} \in \mathbb{R}^n$ which has a probability density function (pdf) denoted by $p_{\mathbf{x}}(.)$. We have a parametrized density model $p(.;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is an m-dimensional vector of parameters. We want to estimate the parameter $\boldsymbol{\theta}$ from $\mathbf{x}$, i.e. we want to approximate $p_{\mathbf{x}}(.)$ by $p(.;\hat{\boldsymbol{\theta}})$ for the estimated parameter value $\hat{\boldsymbol{\theta}}$. (We shall here consider the case of continuous-valued variables only.)

The problem we consider here is that we only are able to compute the pdf given by the model up to a multiplicative constant $Z(\boldsymbol{\theta})$:

$$p(\boldsymbol{\xi};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} q(\boldsymbol{\xi};\boldsymbol{\theta}).$$

That is, we do know the functional form of q as an analytical expression (or any form that can be easily computed), but we do *not* know how to easily compute Z which is given by

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching

Important: scalability of method, and how to actually optimize this loss to learn score network $s_\theta(x)$:

- Hard to scale up equation below to high dimensions.
- More scalable methods: *sliced score matching*, and *denoising score matching*.

Can write the Fisher divergence as follows:

$$\mathcal{L}(\theta) \propto \mathbb{E}_{p(x)}\left[\text{tr}\left(\nabla_x^2 \log p_\theta(x)\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x)\|_2^2\right]$$

⇒ Monte Carlo estimate!

$$\approx \frac{1}{M}\sum_{m=1}^{M} \text{tr}\left(\nabla_x^2 \log p_\theta(x^{(m)})\right) + \frac{1}{2}\|\nabla_x \log p_\theta(x^{(m)})\|_2^2,$$

where $x^{(m)} \sim p(x)$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Summary

To summarize the (naive/vanilla) procedure:

Summary

To summarize the (naive/vanilla) procedure:

(1) Use Monte Carlo estimate from previous slide to learn a score network $s_\theta(x)$.

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Summary

To summarize the (naive/vanilla) procedure:

(1) Use Monte Carlo estimate from previous slide to learn a score network $s_\theta(x)$.

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

(2) Generate samples via Langevin Monte Carlo

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Summary

To summarize the (naive/vanilla) procedure:

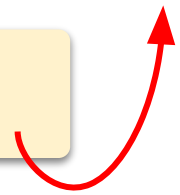
(1) Use Monte Carlo estimate from previous slide to learn a score network $s_\theta(x)$.

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

(2) Generate samples via Langevin Monte Carlo

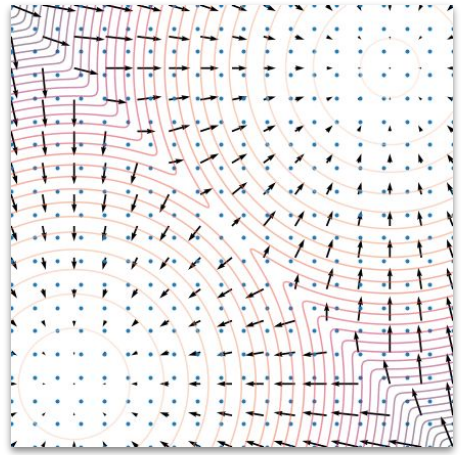
$$x_{i+1} \leftarrow x_i + \epsilon s_\theta(x) + \sqrt{2\epsilon}\,\mathcal{N}(0, I), \quad i = 0, 1, \cdots, K,$$

Note the learned score function is swapped in.



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Summary



To summarize the (naive/vanilla) procedure:

(1) Use Monte Carlo estimate from previous slide to learn a score network $s_\theta(x)$.

$$s_\theta(x) = \nabla_x \log p_\theta(x)$$

(2) Generate samples via Langevin Monte Carlo

$$x_{i+1} \leftarrow x_i + \epsilon s_\theta(x) + \sqrt{2\epsilon}\,\mathcal{N}(0, I), \quad i = 0, 1, \cdots, K,$$

Note the learned score function is swapped in.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score Matching → Noise-conditional Score Matching

So far this does not look super similar to diffusion (going from noise to data).

But it will when we go from *score matching* to *noise-conditional score-matching*!

Score Matching → Noise-conditional Score Matching

So far this does not look super similar to diffusion (going from noise to data).

But it will when we go from *score matching* to *noise-conditional score-matching*!

So next we will cover:

- The **difficulties** with the previous procedure.
- And the **solution** → sample from a sequence of PDFs, from noise to data distribution.

Difficulties to Solve in Practice

Unfortunately, this naive/vanilla procedure has some pitfalls in practice!

⇒ Running this without modification does not work great.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Unfortunately, this naive/vanilla procedure has some pitfalls in practice!

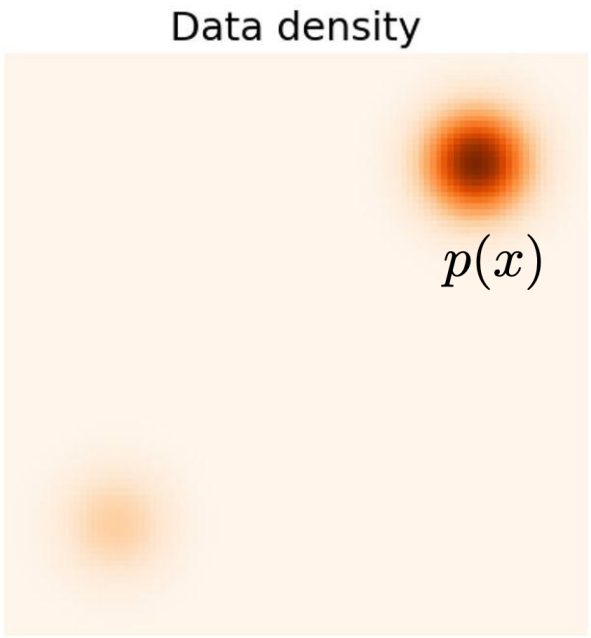
⇒ Running this without modification does not work great.

Main issue: estimate score function is inaccurate in low-density regions of $p(x)$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

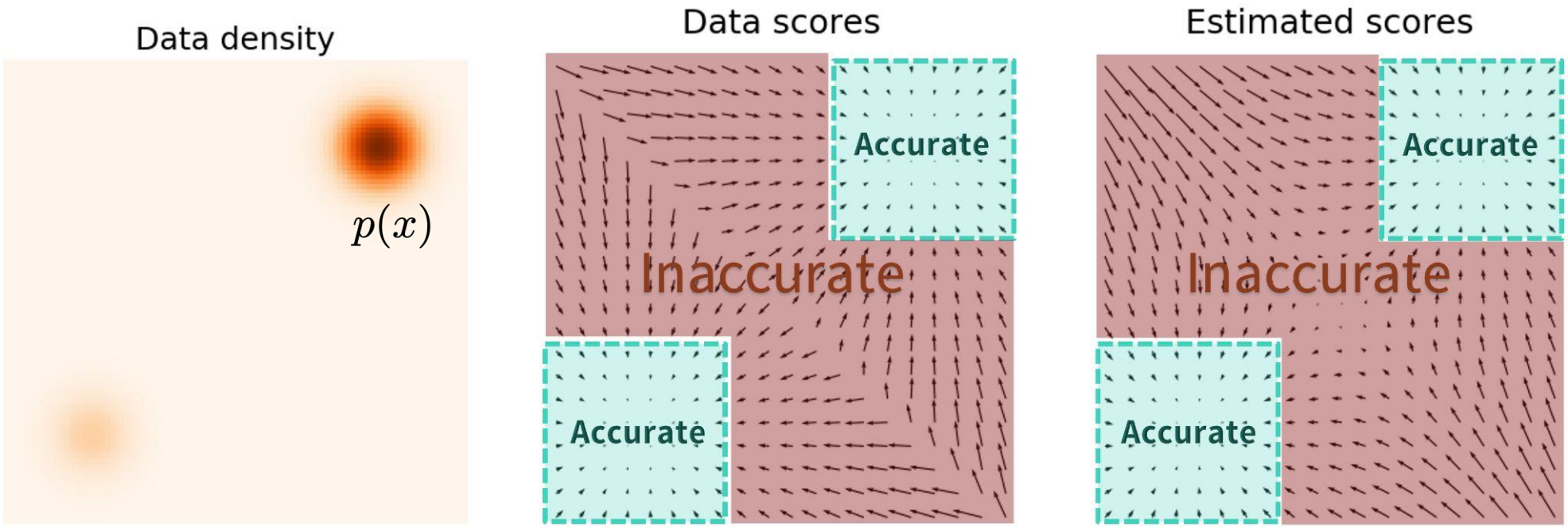
To illustrate, suppose we have samples from data density $p(x)$ below:



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

To illustrate, suppose we have samples from data density $p(x)$ below:



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Consequences of this:

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Consequences of this:

- Our initial samples will likely be in low-density regions (especially when our data is high dimensional, *e.g.*, images.)

Difficulties to Solve in Practice

Consequences of this:

- Our initial samples will likely be in low-density regions (especially when our data is high dimensional, *e.g.*, images.)
- ⇒ Langevin Monte Carlo becomes “derailed” at the beginning of the procedure.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Consequences of this:

- Our initial samples will likely be in low-density regions (especially when our data is high dimensional, *e.g.*, images.)
- ⇒ Langevin Monte Carlo becomes “derailed” at the beginning of the procedure.
- ⇒ Unable to generate high quality samples.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

However…

- Suppose that we perturb the data points with noise, and train score-based models on the noisy data points instead.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

However…

- Suppose that we perturb the data points with noise, and train score-based models on the noisy data points instead.

- When the noise is large enough, it can populate low-density regions of $p(x)$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

However…

- Suppose that we perturb the data points with noise, and train score-based models on the noisy data points instead.
- When the noise is large enough, it can populate low-density regions of $p(x)$.
- ⇒ This improves the accuracy of estimated scores (for a slightly different/biased distribution).

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

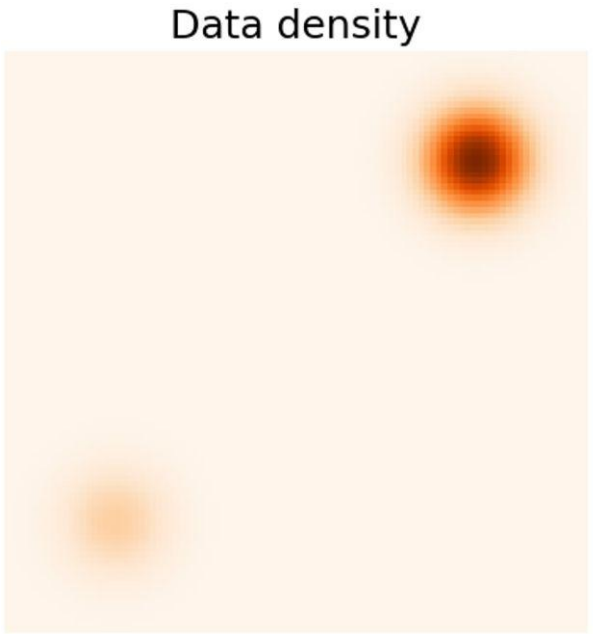
Difficulties to Solve in Practice

Visually, using previous figure:

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Visually, using previous figure:



Before: **No perturbation.**

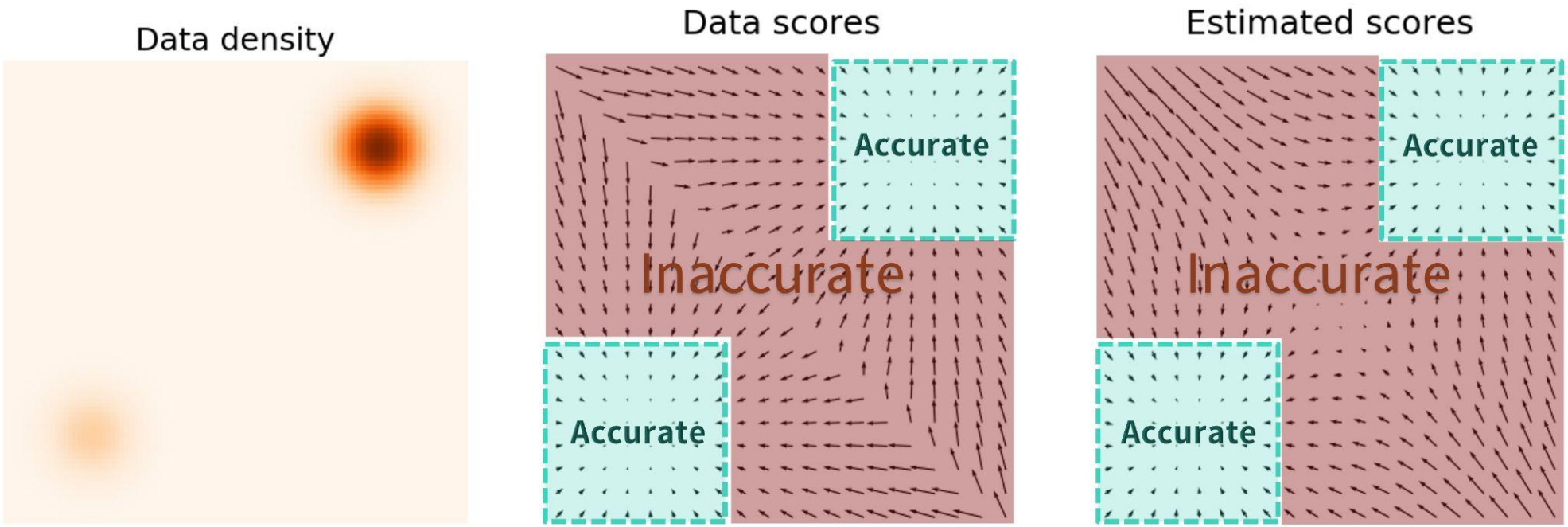
⇒ Estimated score is inaccurate in low-density regions.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Visually, using previous figure:

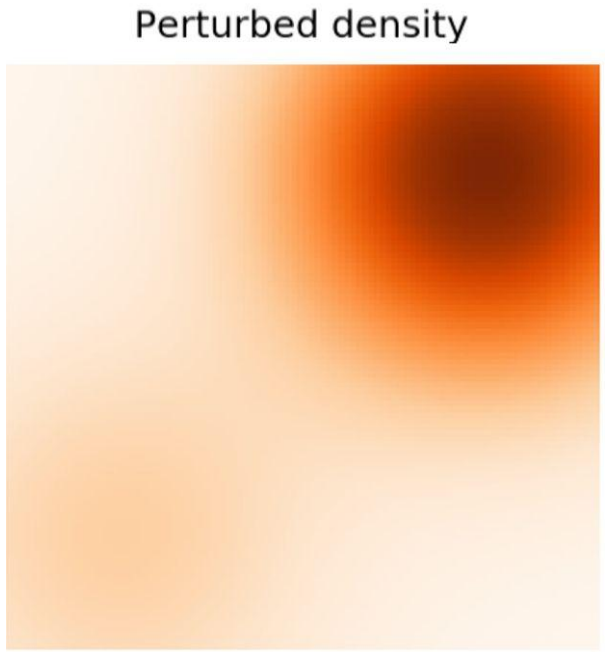
Before: **No perturbation.**
⇒ Estimated score is inaccurate in low-density regions.



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Visually, using previous figure:



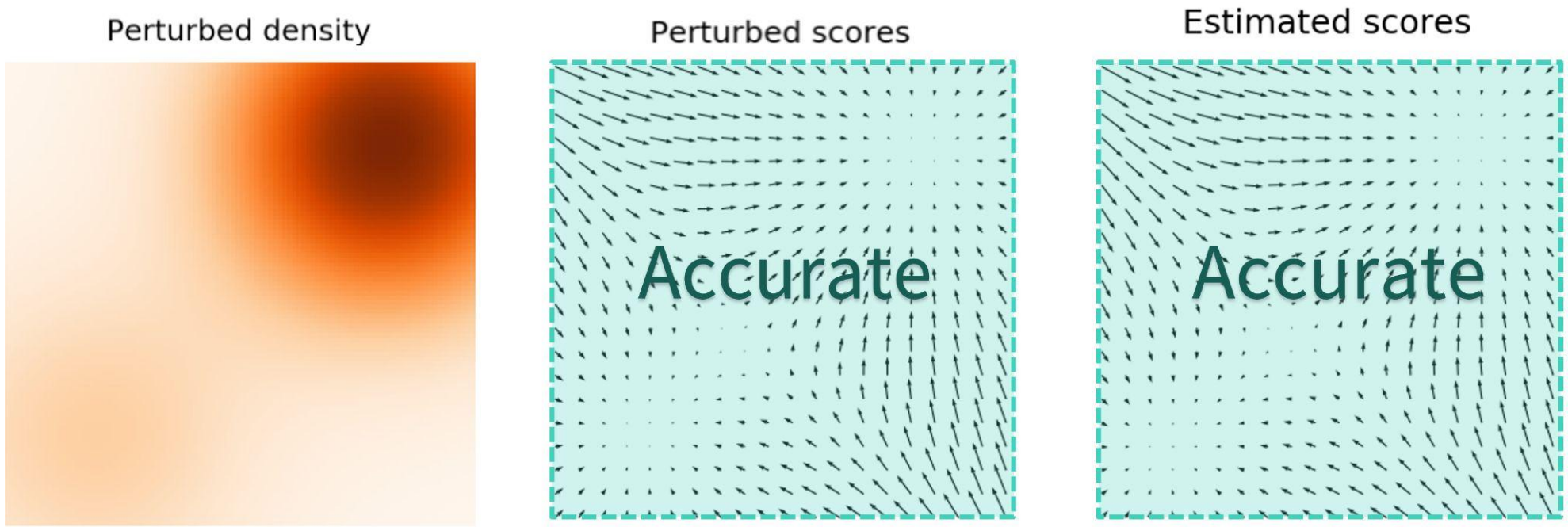
Instead: **Perturb (add noise to)** $p(x)$**.**
⇒ Produces more samples in low-density regions, thus making score more accurate.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Difficulties to Solve in Practice

Visually, using previous figure:

Instead: **Perturb (add noise to)** $p(x)$**.**
⇒ Produces more samples in low-density regions, thus making score more accurate.



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

So we have a tradeoff:

Score-based Generative Models — Multiple Noise Perturbations

So we have a tradeoff:

Less noise makes score matching **harder** (especially at start of LMC), but yields more-accurate (*unbiased*) scores.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

So we have a tradeoff:

Less noise makes score matching **harder** (especially at start of LMC), but yields more-accurate (*unbiased*) scores.

More noise makes score matching **easier** (samples cover low-density regions), but yields less-accurate (*biased*) scores.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

So we have a tradeoff:

Less noise makes score matching **harder** (especially at start of LMC), but yields more-accurate (*unbiased*) scores.

More noise makes score matching **easier** (samples cover low-density regions), but yields less-accurate (*biased*) scores.

Solution:

Perform score matching (learn the score function) for *multiple noise levels*!

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

So we have a tradeoff:

Less noise makes score matching **harder** (especially at start of LMC), but yields more-accurate (*unbiased*) scores.

More noise makes score matching **easier** (samples cover low-density regions), but yields less-accurate (*biased*) scores.

Solution:

Perform score matching (learn the score function) for *multiple noise levels*!

And then, to sample in LMC, we will traverse through the noise levels (from high to low noise).

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

In particular:

Score-based Generative Models — Multiple Noise Perturbations

In particular:

- Suppose we perturb the data with isotropic Gaussian noise.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

In particular:

- Suppose we perturb the data with isotropic Gaussian noise.
- And that we have L increasing noise levels: $i = 1, \ldots, L$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

In particular:

- Suppose we perturb the data with isotropic Gaussian noise.
- And that we have L increasing noise levels: $i = 1, \ldots, L$.
- For each noise level i, we can add Gaussian noise $\mathcal{N}(0, \sigma_i^2 I)$ to $p(x)$ to form the *noise-perturbed distribution* $p_{\sigma_i}(x)$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

Next, we can estimate the score function for each noise-perturbed distribution *i*, which we can denote $s_\theta(x, i)$, using score matching.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

Next, we can estimate the score function for each noise-perturbed distribution *i*, which we can denote $s_\theta(x, i)$, using score matching.

At the end of this procedure, our score network should approximate:

$$s_\theta(x, i) \approx \nabla_x \log p_{\sigma_i}(x), \quad i = 1, 2, \ldots, L$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

Next, we can estimate the score function for each noise-perturbed distribution *i*, which we can denote $s_\theta(x, i)$, using score matching.

At the end of this procedure, our score network should approximate:

$$s_\theta(x, i) \approx \nabla_x \log p_{\sigma_i}(x), \quad i = 1, 2, \ldots, L$$

We will call this model the **noise conditional score network (NCSN)**.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

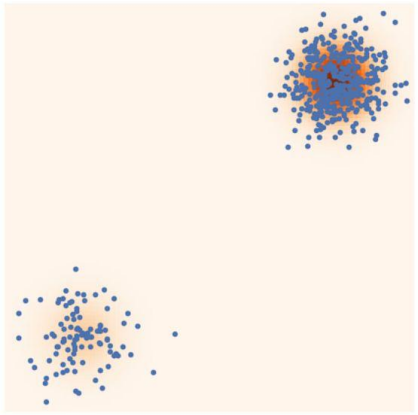
Visualizing this:

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

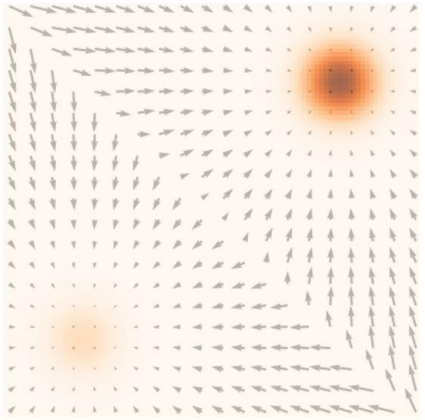
Visualizing this:

σ_1



$p_{\sigma_i}(x)$

Perturbed density, and samples



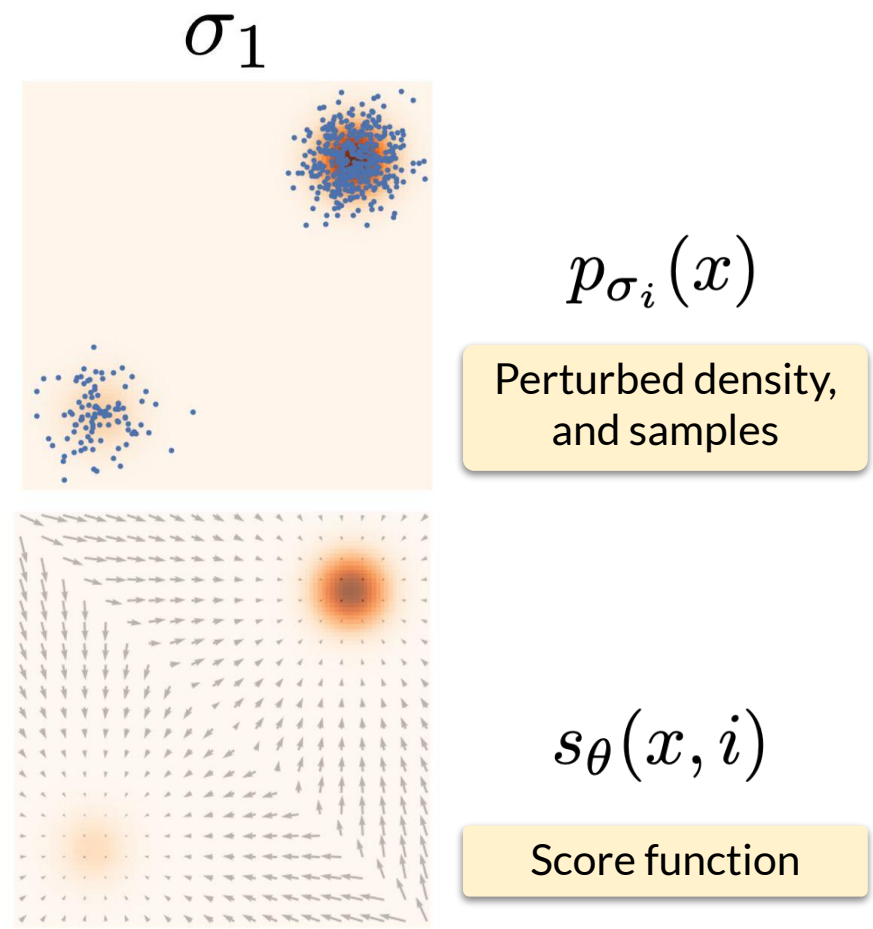
$s_\theta(x, i)$

Score function

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

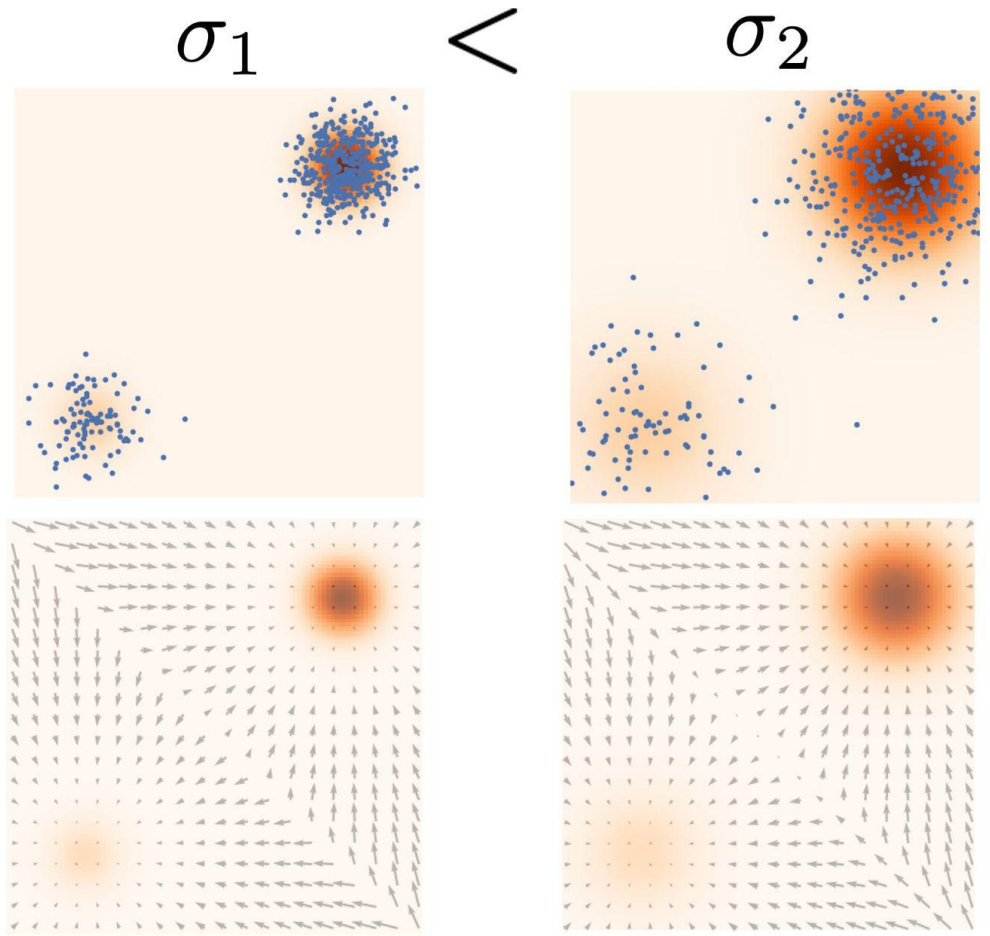
Visualizing this:



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

Visualizing this:



$p_{\sigma_i}(x)$

Perturbed density, and samples

$s_\theta(x, i)$

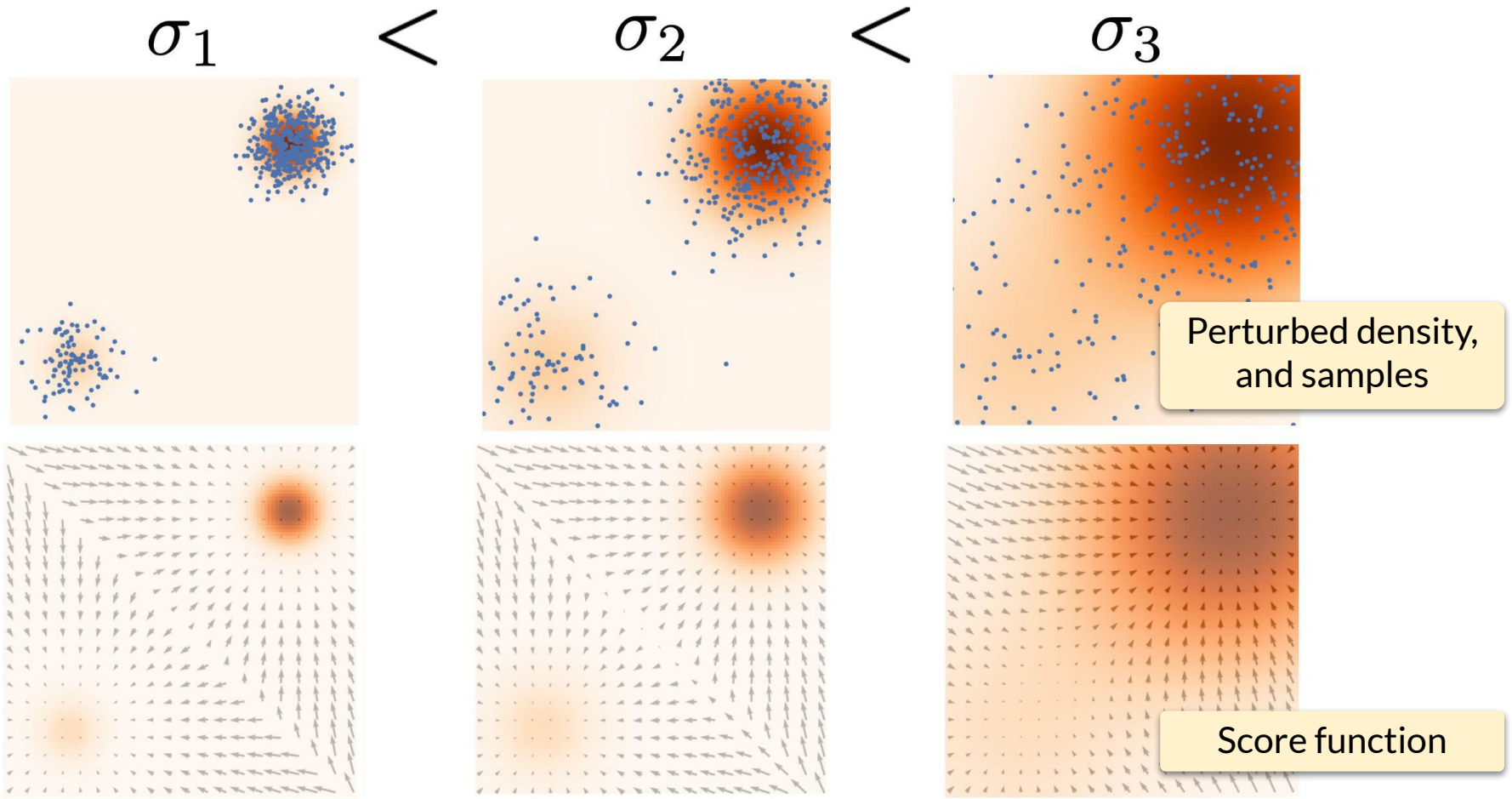
Score function

Increasing noise level →

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

Visualizing this:



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Multiple Noise Perturbations

If we perturb a data with multiple scales of Gaussian noise, we see an increasingly noisy data point.

An image example:

Note: it is easy for us to sample this perturbed data (given our original dataset).

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Learning the NCSN

To learn the noise-conditional score network (NCSN) $s_\theta(x, i)$:

Score-based Generative Models — Learning the NCSN

To learn the noise-conditional score network (NCSN) $s_\theta(x, i)$:

We use a modified training objective — a weighted sum of Fisher divergences over all of the noise scales:

$$\mathcal{L}(\theta) = \sum_{i=1}^{L} \lambda(i) \mathbb{E}_{p_{\sigma_i}(x)} \left[\| \nabla_x \log p_{\sigma_i}(x) - s_\theta(x, i) \|_2^2 \right]$$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Learning the NCSN

To learn the noise-conditional score network (NCSN) $s_\theta(x, i)$:

We use a modified training objective — a weighted sum of Fisher divergences over all of the noise scales:

$$\mathcal{L}(\theta) = \sum_{i=1}^{L} \lambda(i) \mathbb{E}_{p_{\sigma_i}(x)} \left[\| \nabla_x \log p_{\sigma_i}(x) - s_\theta(x, i) \|_2^2 \right]$$

Where the weighting function is often taken to be $\lambda(i) = \sigma_i^2$.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Generation from NCSN

Finally, to sample from this model:

Score-based Generative Models — Generation from NCSN

Finally, to sample from this model:

Similar to before, run Langevin Monte Carlo (LMC) using the learned (noise-conditional) score network $s_\theta(x, i)$...

While sequentially stepping (backwards) through the noise levels: $i = L, L\text{-}1, L\text{-}2, \ldots, 1$

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Generation from NCSN

Finally, to sample from this model:

Similar to before, run Langevin Monte Carlo (LMC) using the learned (noise-conditional) score network $s_\theta(x, i)$...

While sequentially stepping (backwards) through the noise levels: $i = L, L\text{-}1, L\text{-}2, \ldots, 1$

⇒ This is LMC with an annealed noise level/step size! *(Mentioned in MCMC lecture).*

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Generation from NCSN

Altogether, how does this look?

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Generation from NCSN

Altogether, how does this look?

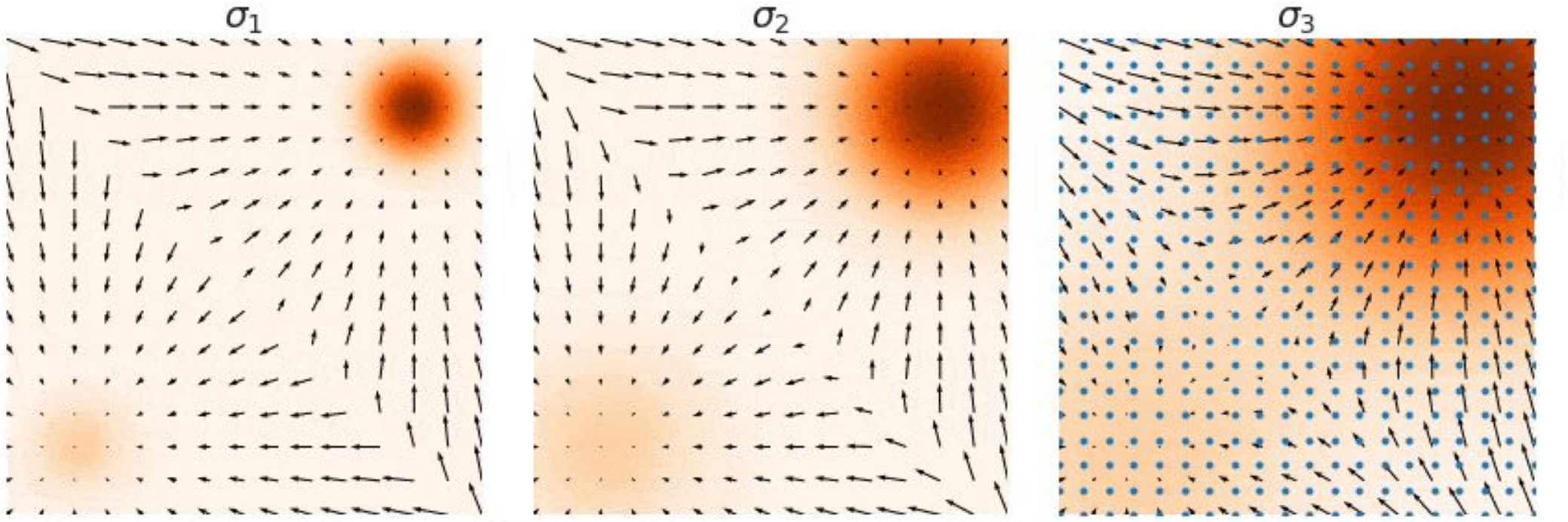
← Proceeding from **right** to **left**.

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Generation from NCSN

Altogether, how does this look?

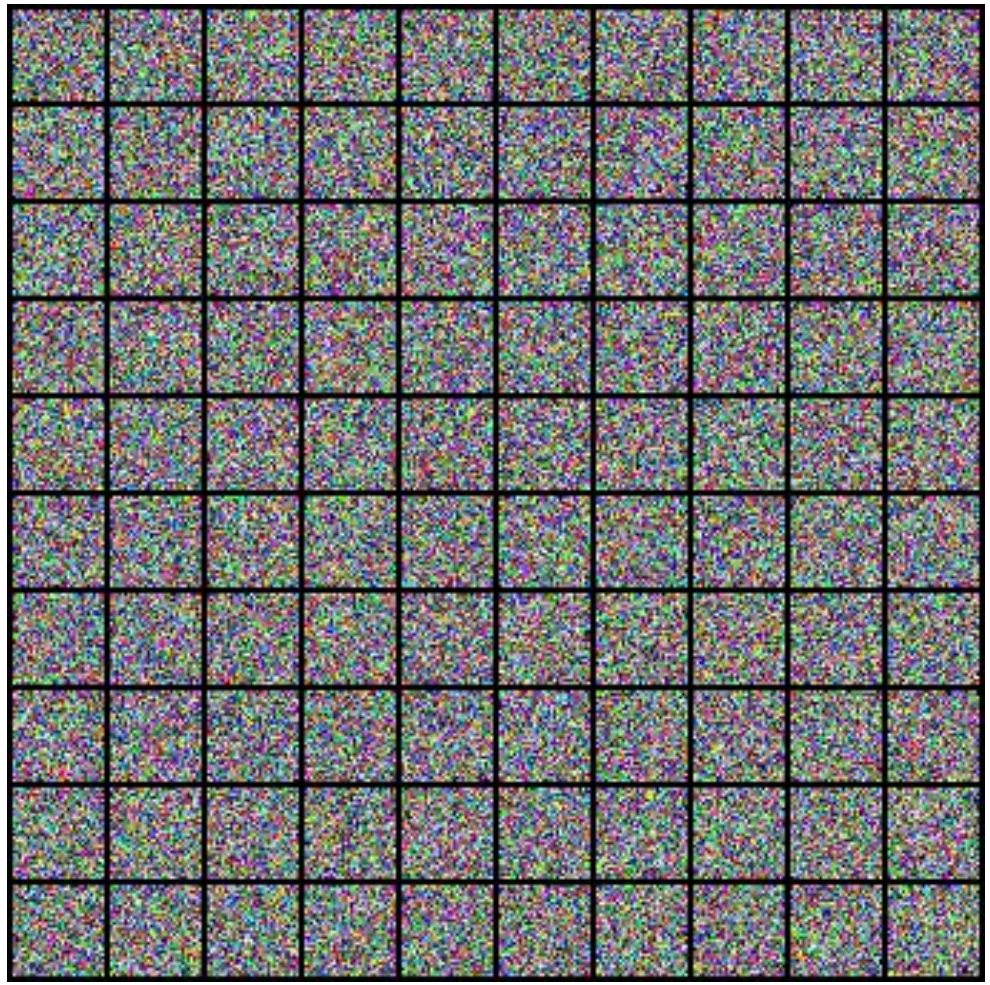
← Proceeding from **right** to **left**.



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Score-based Generative Models — Generation from NCSN

And on higher dimensional data (face images):



Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Connections to Diffusion Models and VAEs

Connections to Diffusion Models

Source: Yang Song, Generative Modeling by Estimating Gradients of the Data Distribution

Connections to Diffusion Models

You can see how score-based models start to look like what we know of as **diffusion models**.

Connections to Diffusion Models

You can see how score-based models start to look like what we know of as **diffusion models**.

Diffusion models can be viewed in the following way:

Connections to Diffusion Models

You can see how score-based models start to look like what we know of as **diffusion models**.

Diffusion models can be viewed in the following way:

- Iteratively add noise to data points (via a *forward diffusion process*).

Connections to Diffusion Models

You can see how score-based models start to look like what we know of as **diffusion models**.

Diffusion models can be viewed in the following way:

- Iteratively add noise to data points (via a *forward diffusion process*).
- Then learn a *reverse diffusion process*, which iteratively denoises data.

Connections to Diffusion Models

You can see how score-based models start to look like what we know of as **diffusion models**.

Diffusion models can be viewed in the following way:

- Iteratively add noise to data points (via a *forward diffusion process*).
- Then learn a *reverse diffusion process*, which iteratively denoises data.
- By sampling from the reverse diffusion process, can generate new data samples.

Connections to Diffusion Models

A popular example is the **denoising diffusion probabilistic model (DDPM)**, *NeurIPS 2019* → presented in class!

(We will hear presentations of multiple classic diffusion papers later today :-))

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at https://github.com/hojonathanho/diffusion.

1 Introduction

Deep generative models of all kinds have recently exhibited high quality samples in a wide variety of data modalities. Generative adversarial networks (GANs), autoregressive models, flows, and variational autoencoders (VAEs) have synthesized striking image and audio samples [14, 27, 3, 58, 38, 25, 10, 32, 44, 57, 26, 33, 45], and there have been remarkable advances in energy-based modeling and score matching that have produced images comparable to those of GANs [11, 55].

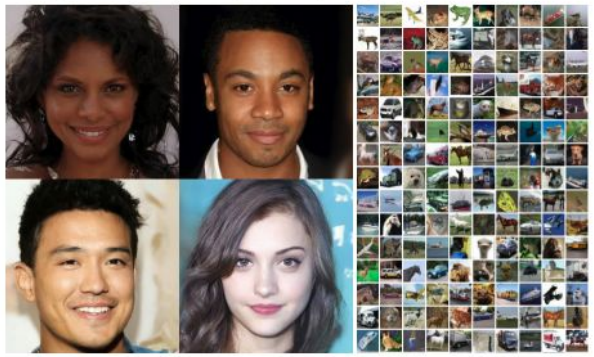


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Connections to Diffusion Models

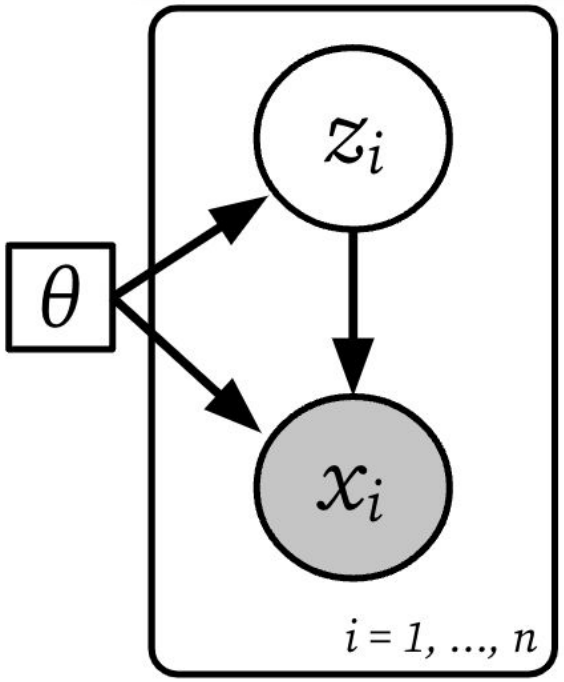
A brief overview:

Connections to Diffusion Models

A brief overview $\rightarrow$ First, recall VAEs:

Connections to Diffusion Models

A brief overview → First, recall VAEs:



First, we define our model (Bayesian network)

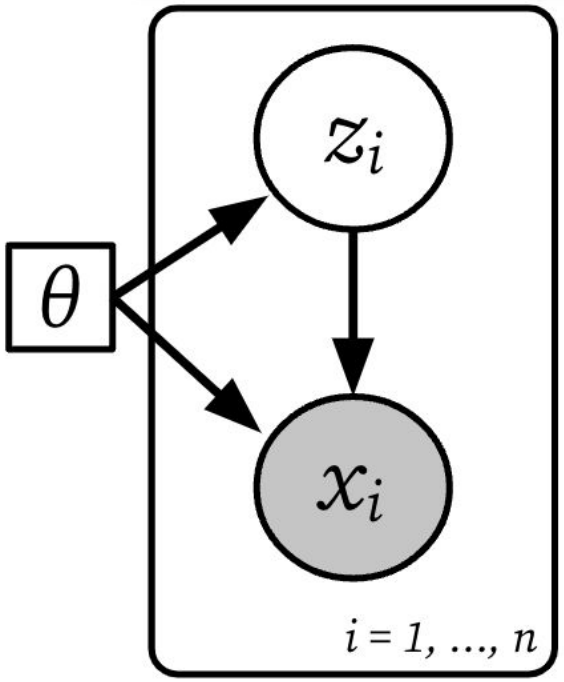
Joint Model

Prior: $p_\theta(z_i)$

Likelihood: $p_\theta(x_i \mid z_i)$

Connections to Diffusion Models

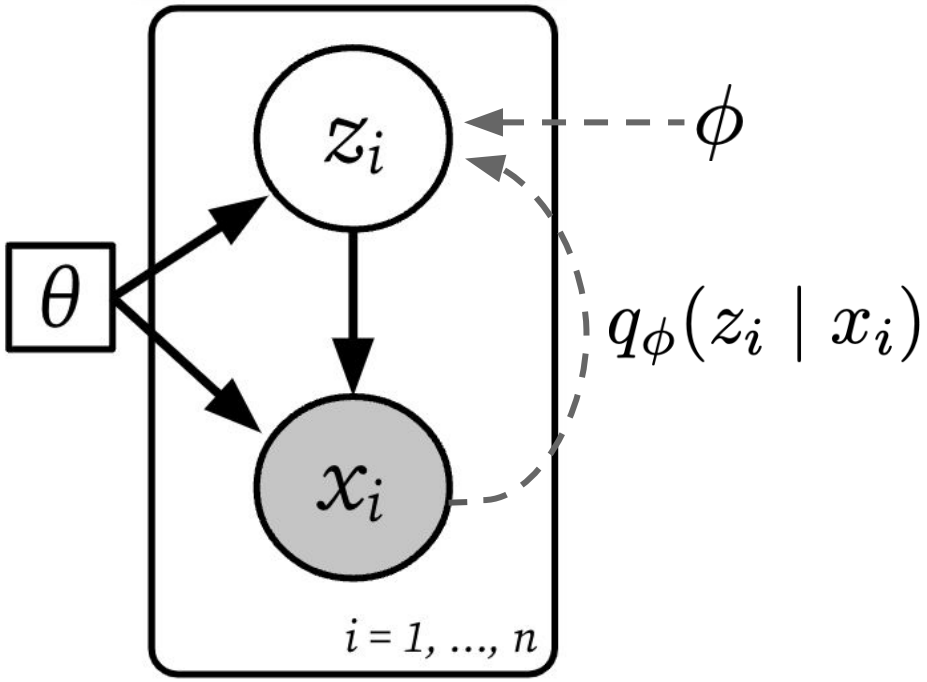
A brief overview → First, recall VAEs:



- You have a *decoder* (model likelihood $p_\theta(x \mid z)$)

Connections to Diffusion Models

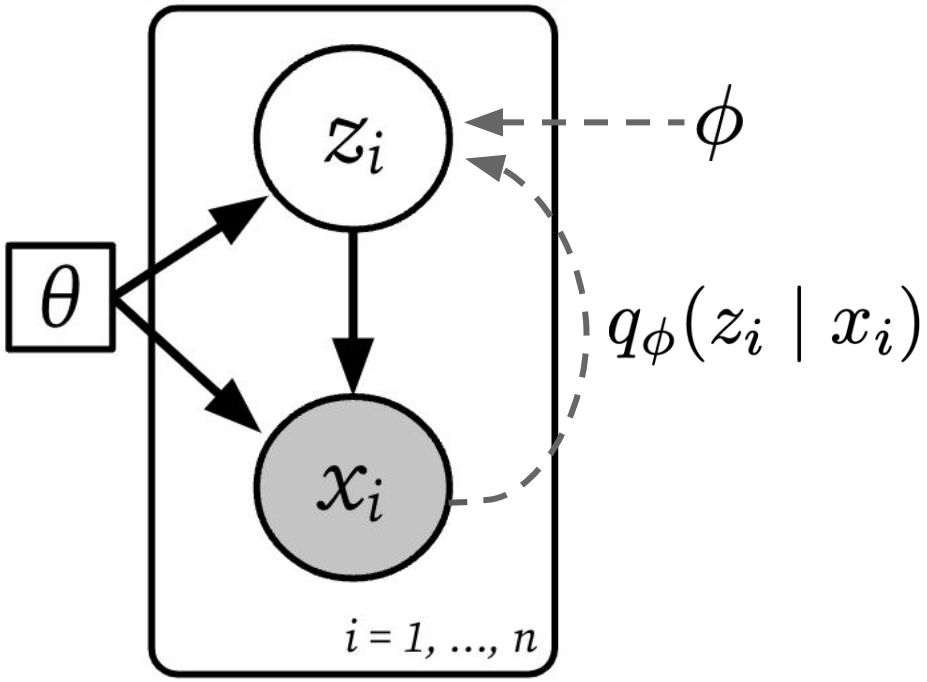
A brief overview → First, recall VAEs:



- You have a *decoder* (model likelihood $p_\theta(x \mid z)$)
- And an *encoder* (variational approx $q_\phi(z \mid x)$)

Connections to Diffusion Models

A brief overview → First, recall VAEs:



- You have a *decoder* (model likelihood $p_\theta(x \mid z)$)
- And an *encoder* (variational approx $q_\phi(z \mid x)$)
- And we perform MLE to learn p_θ (while simultaneously inferring q_ϕ)

Connections to Diffusion Models

A brief overview $\rightarrow$ Now, back to DDPMS

Connections to Diffusion Models

A brief overview:

- Define the DDPM as a probabilistic model.
- Where we view the *forward diffusion process* (adding noise) as a variational approximation to the posterior.

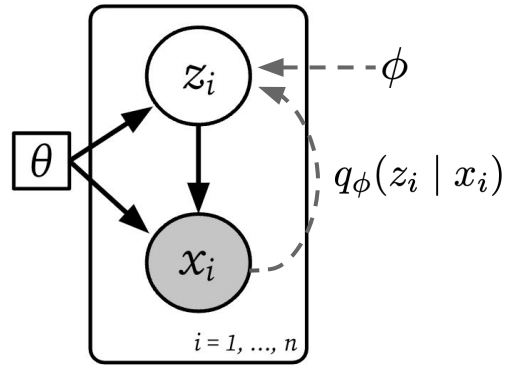
Connections to Diffusion Models

A brief overview:

- Define the DDPM as a probabilistic model.
- Where we view the *forward diffusion process* (adding noise) as a variational approximation to the posterior.
- And we view the *reverse diffusion process* (denoising) as a Markov chain, with generative model parameters θ that we'd like to learn.

Connections to Diffusion Models

Somewhat similar in this way to VAEs

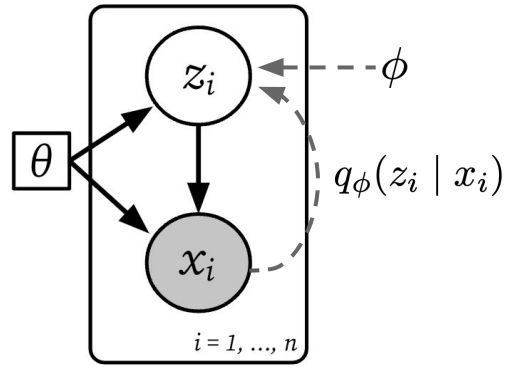


A brief overview:

- Define the DDPM as a probabilistic model.
- Where we view the *forward diffusion process* (adding noise) as a variational approximation to the posterior.
- And we view the *reverse diffusion process* (denoising) as a Markov chain, with generative model parameters θ that we'd like to learn.

Connections to Diffusion Models

Somewhat similar in this way to VAEs



A brief overview:

- Define the DDPM as a probabilistic model.
- Where we view the *forward diffusion process* (adding noise) as a variational approximation to the posterior.
- And we view the *reverse diffusion process* (denoising) as a Markov chain, with generative model parameters θ that we'd like to learn.
- And similar to VAE models, we maximize the ELBO while learning θ.

Connections to Diffusion Models

A brief overview:

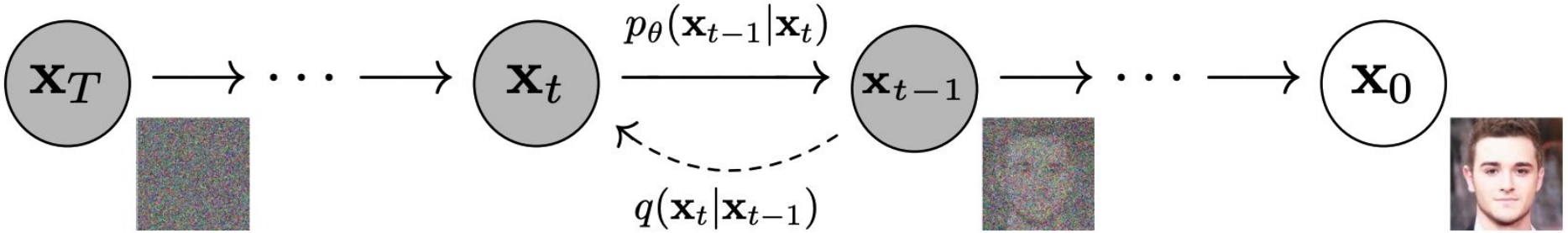
The graphical model for the DDPM is popularly illustrated as:

Source: “Denoising Diffusion Probabilistic Models”, Ho et al., 2019

Connections to Diffusion Models

A brief overview:

The graphical model for the DDPM is popularly illustrated as:

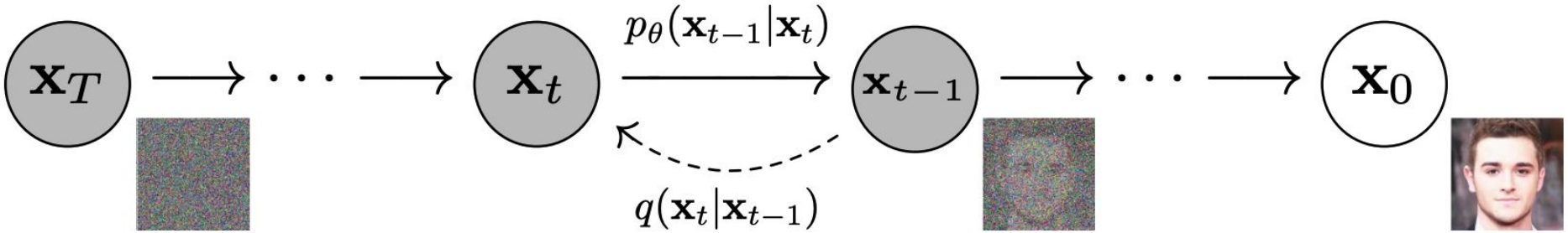


Source: “Denoising Diffusion Probabilistic Models”, Ho et al., 2019

Connections to Diffusion Models

A brief overview:

The graphical model for the DDPM is popularly illustrated as:



Notice q is not learned (like in VAEs), but is a fixed process

… except for noise level parameters, in some cases.

Source: "Denoising Diffusion Probabilistic Models", Ho et al., 2019

Connections to Diffusion Models

So, to be explicit, the probabilistic model (joint PDF / Bayesian network) is:

$$p_\theta(x_0, \ldots, x_T) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

Connections to Diffusion Models

So, to be explicit, the probabilistic model (joint PDF / Bayesian network) is:

$$p_\theta(x_0, \ldots, x_T) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

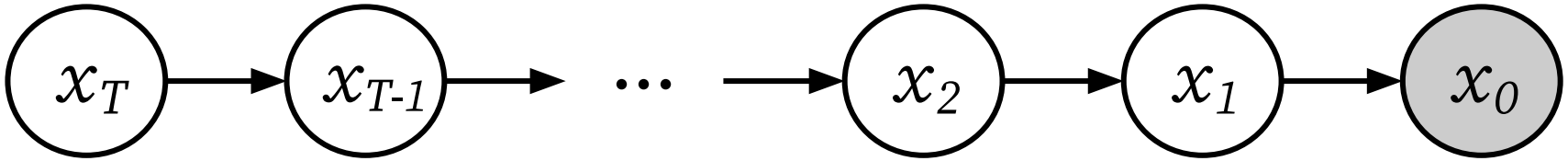
A Markov chain (with one node observed)

Connections to Diffusion Models

So, to be explicit, the probabilistic model (joint PDF / Bayesian network) is:

$$p_\theta(x_0, \ldots, x_T) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

A Markov chain (with one node observed)



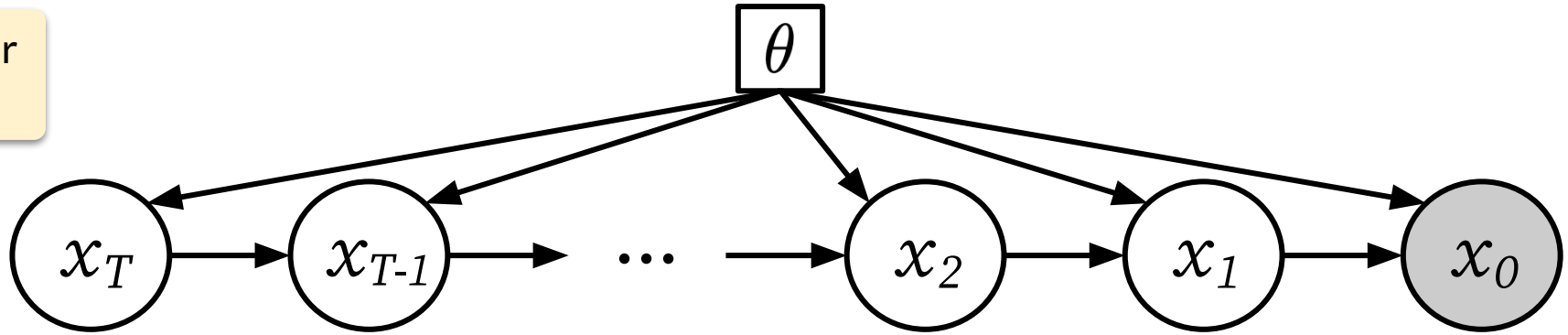
Connections to Diffusion Models

So, to be explicit, the probabilistic model (joint PDF / Bayesian network) is:

$$p_\theta(x_0, \ldots, x_T) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

A Markov chain (with one node observed)

With parameter θ shown



Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

(Another Markov chain!)

Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

(Another Markov chain!)

The graphical model for this distribution might be drawn as:

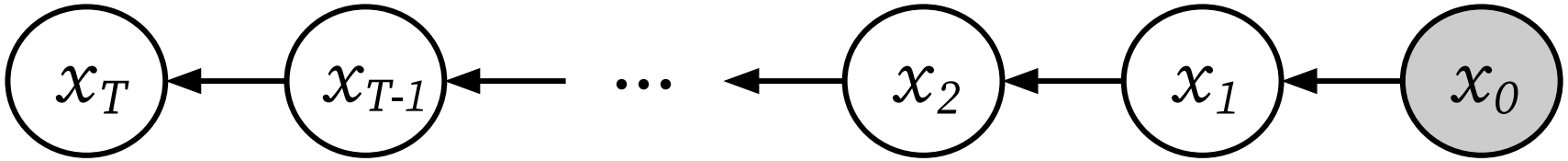
Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

(Another Markov chain!)

The graphical model for this distribution might be drawn as:



Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

(Another Markov chain!)

And sticking both together we get:

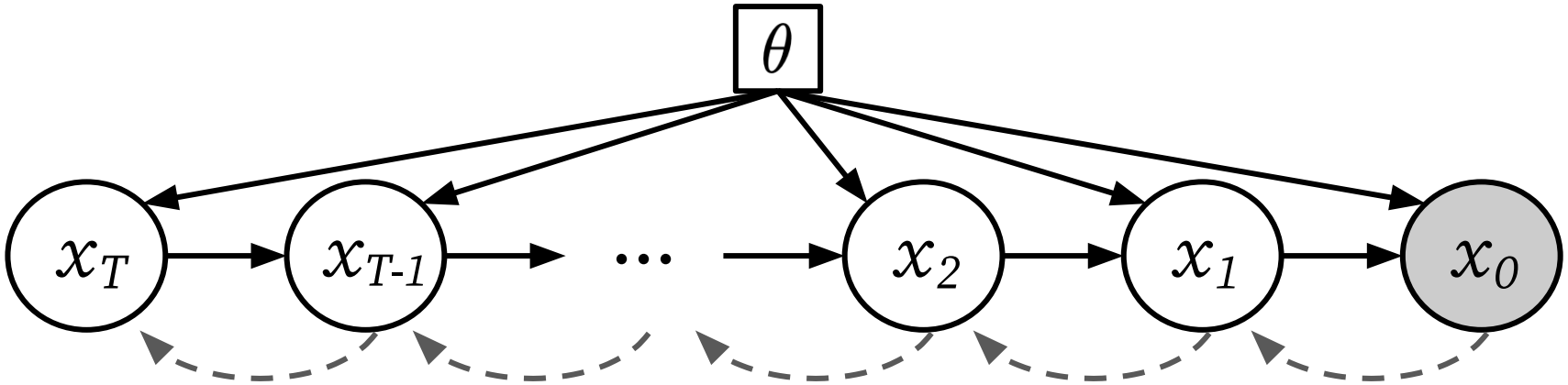
Connections to Diffusion Models

While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

(Another Markov chain!)

And sticking both together we get:



Connections to Diffusion Models

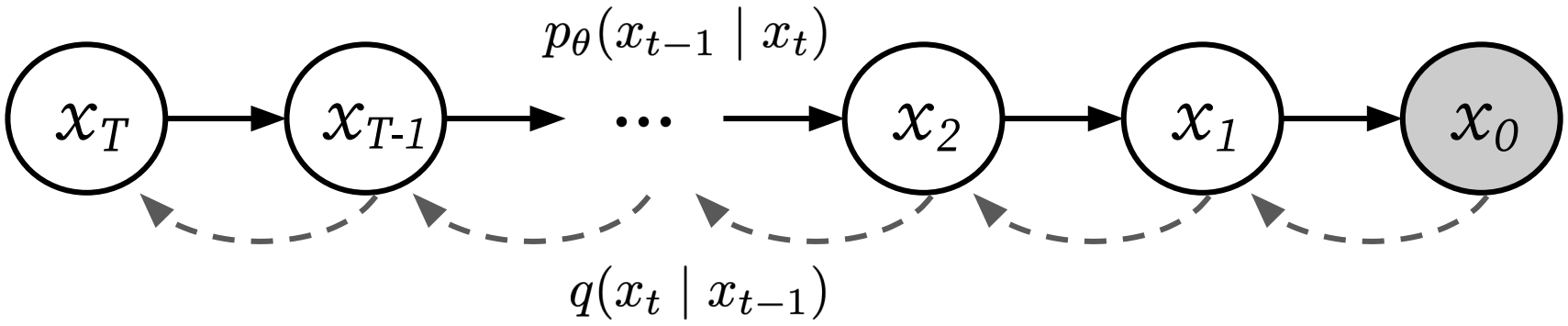
While the variational family of *posterior approximations* can be written:

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

(Another Markov chain!)

And sticking both together we get:

Similar to the "usual" PGM diagram for diffusion



Connections to Diffusion Models

Similar to VAE, you can:

- Write out the ELBO — has a particular form due to the structure of p and q.
- Then try to optimize it with respect to θ (and/or any VI parameters).

From the DDPM paper:

$$\mathbb{E}\left[\log p_\theta(\mathbf{x}_0)\right] \geq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right]$$

$$= \mathbb{E}_q\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\right]$$

Connections to Diffusion Models

Before the DDPM paper, diffusion models as described here (roughly) already existed in prior work.

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However, these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $\phi(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff–for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective[1].

1.1. Diffusion probabilistic models

We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

[1]Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

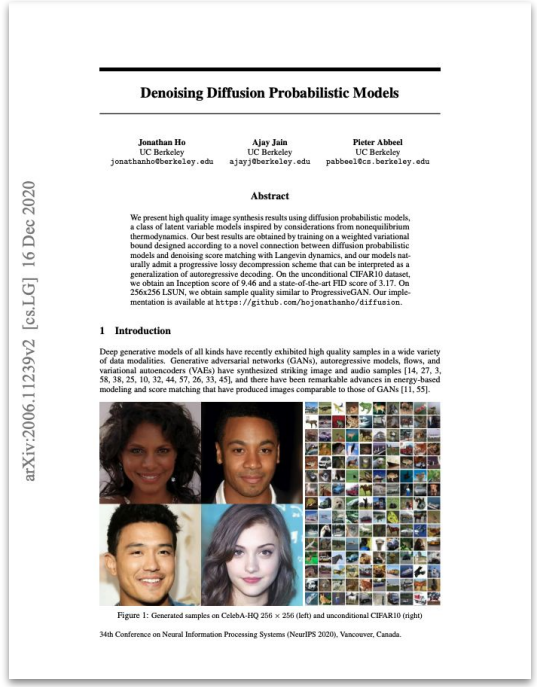
Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

Source: “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”, Sohl-Dickstein et al., 2015

Connections to Diffusion Models

Before the DDPM paper, diffusion models as described here (roughly) already existed in prior work.

However, the DDPM paper does a couple of things:



arXiv:2006.11239v2 [cs.LG] 16 Dec 2020

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at https://github.com/hojonathanho/diffusion.

1 Introduction

Deep generative models of all kinds have recently exhibited high quality samples in a wide variety of data modalities. Generative adversarial networks (GANs), autoregressive models, flows, and variational autoencoders (VAEs) have synthesized striking image and audio samples [14, 27, 3, 58, 38, 25, 10, 32, 44, 57, 26, 33, 45], and there have been remarkable advances in energy-based modeling and score matching that have produced images comparable to those of GANs [11, 55].

Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

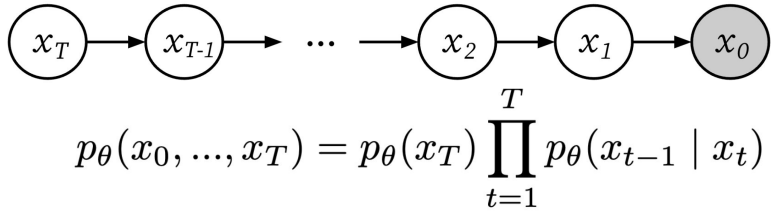
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Source: “Denoising Diffusion Probabilistic Models”, Ho et al., 2019

Connections to Diffusion Models

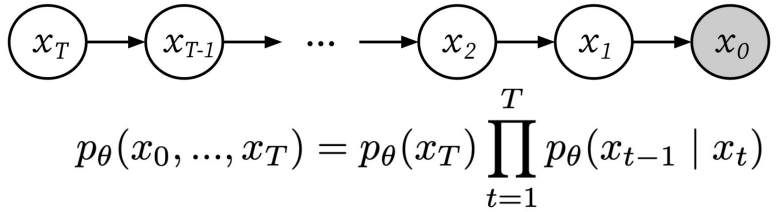
(1) Shows that under a certain parameterization of the model, their algorithm is *equivalent to* denosing score matching.

Connections to Diffusion Models



(1) Shows that under a certain parameterization of the model, their algorithm is *equivalent to* denosing score matching.

Connections to Diffusion Models



(1) Shows that under a certain parameterization of the model, their algorithm is *equivalent to* denosing score matching.

- Shows optimizing the ELBO ⇒ equivalent to the "*sum of Fisher divergences*" loss from before!
- And that sampling from the forward model ⇒ equivalent to Langevin dynamics on a learned score function!

Connections to Diffusion Models

DDPM Samples

(1) Shows that under a certain parameterization of the model, their algorithm is *equivalent to* denosing score matching.

- Shows optimizing the ELBO ⇒ equivalent to the "*sum of Fisher divergences*" loss from before!
- And that sampling from the forward model ⇒ equivalent to Langevin dynamics on a learned score function!

(2) Shows that under this parameterization (along with a few additional tricks to better handle image data), sample quality is can be very good.

(Possibly better than most/all previous generative models up until this point).

Explicit vs Implicit Probabilistic Generative Models

Explicit vs Implicit Probabilistic Generative Models

Explicit

Defines a probability model

Then learns parameter of model.

Explicit vs Implicit Probabilistic Generative Models

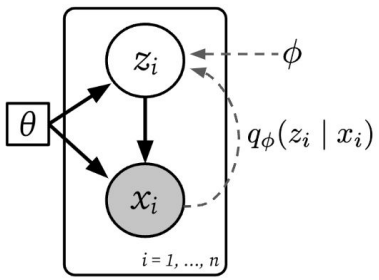
Explicit

Defines a probability model

Then learns parameter of model.

E.g.,

VAEs



Explicit vs Implicit Probabilistic Generative Models

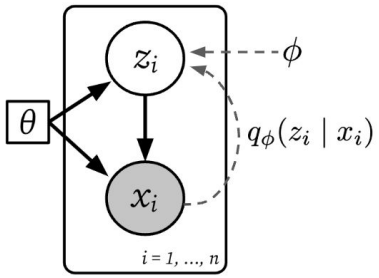
<u>Explicit</u>

Defines a probability model

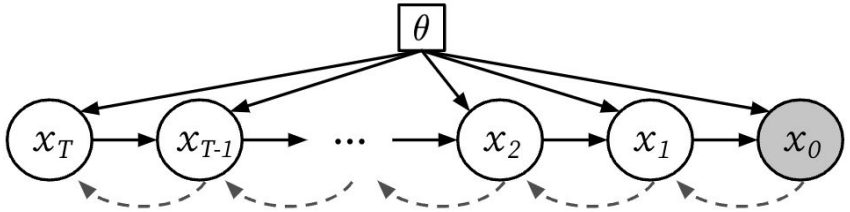
Then learns parameter of model.

E.g.,

VAEs



Diffusion Models



Explicit vs Implicit Probabilistic Generative Models

Explicit

Defines a probability model

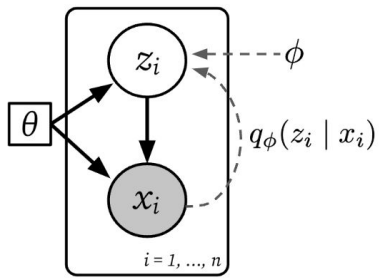
Then learns parameter of model.

Implicit

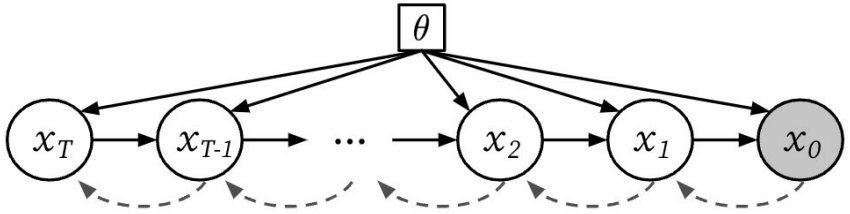
Learns some quantity that implies a probabilistic model and allows for sampling…

E.g.,

VAEs



Diffusion Models



Explicit vs Implicit Probabilistic Generative Models

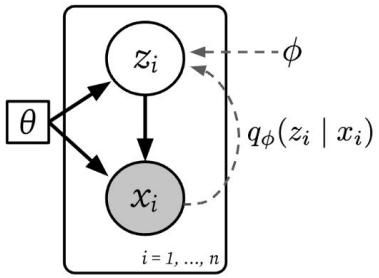
Explicit

Defines a probability model

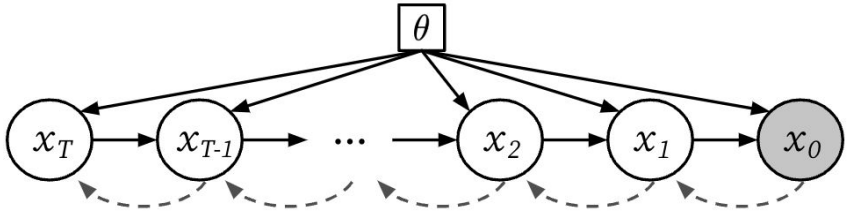
Then learns parameter of model.

E.g.,

VAEs



Diffusion Models



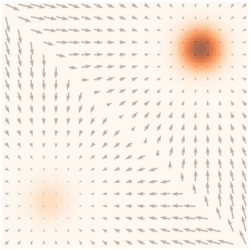
Implicit

Learns some quantity that implies a probabilistic model and allows for sampling…

E.g.,

Score-based Models

$$s(x) = \nabla_x \log p(x)$$



Source: Yang Song

Explicit vs Implicit Probabilistic Generative Models

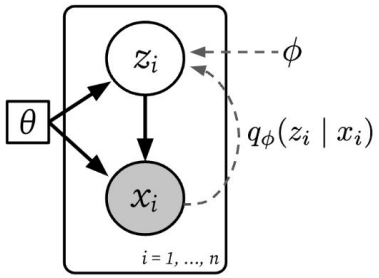
Explicit

Defines a probability model

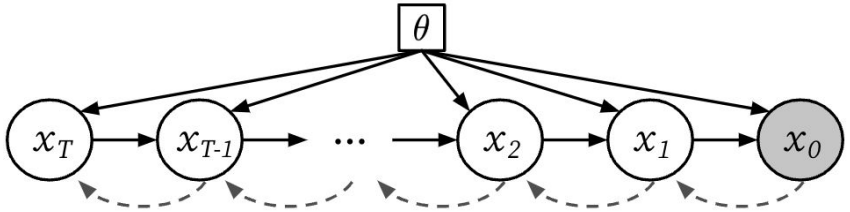
Then learns parameter of model.

E.g.,

VAEs



Diffusion Models



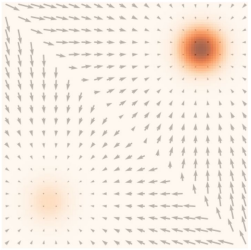
Implicit

Learns some quantity that implies a probabilistic model and allows for sampling…

E.g.,

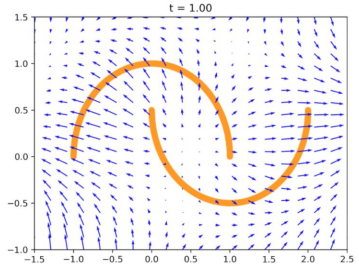
Score-based Models

$$s(x) = \nabla_x \log p(x)$$



Source: Yang Song

Flow-matching



Velocity field

Source: Jakub Tomczak

Connections to Diffusion Models

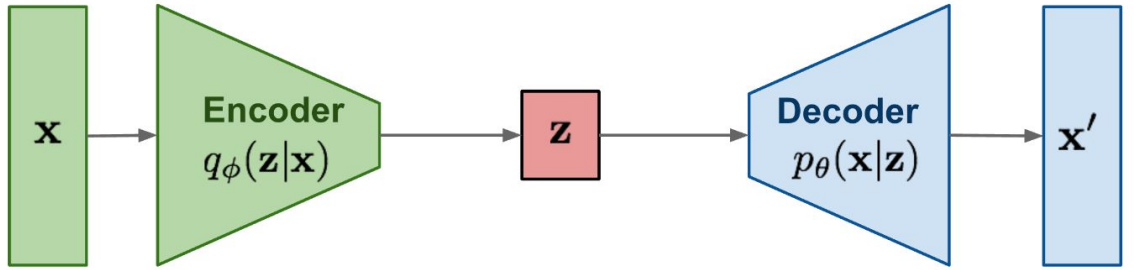
A comparison of generative modeling paradigms (from Lil'Log blog):

Source: "What are Diffusion Models?", Lilian Weng., 2021

Connections to Diffusion Models

A comparison of generative modeling paradigms (from Lil'Log blog):

VAE: maximize variational lower bound

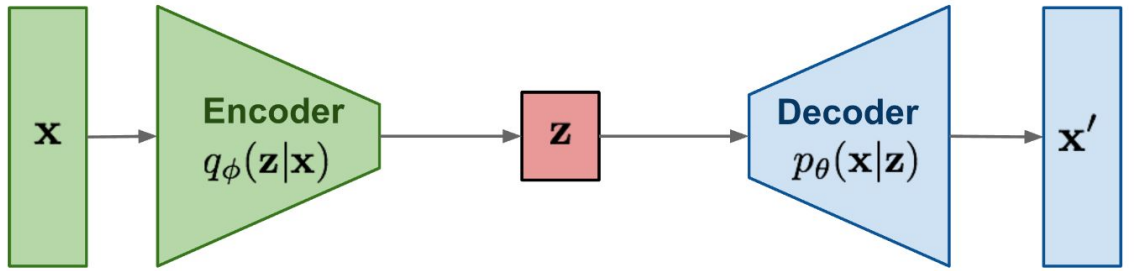


Source: "What are Diffusion Models?", Lilian Weng., 2021

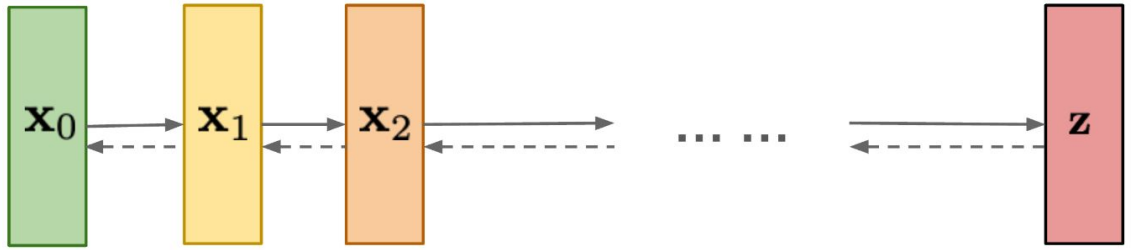
Connections to Diffusion Models

A comparison of generative modeling paradigms (from Lil'Log blog):

VAE: maximize variational lower bound



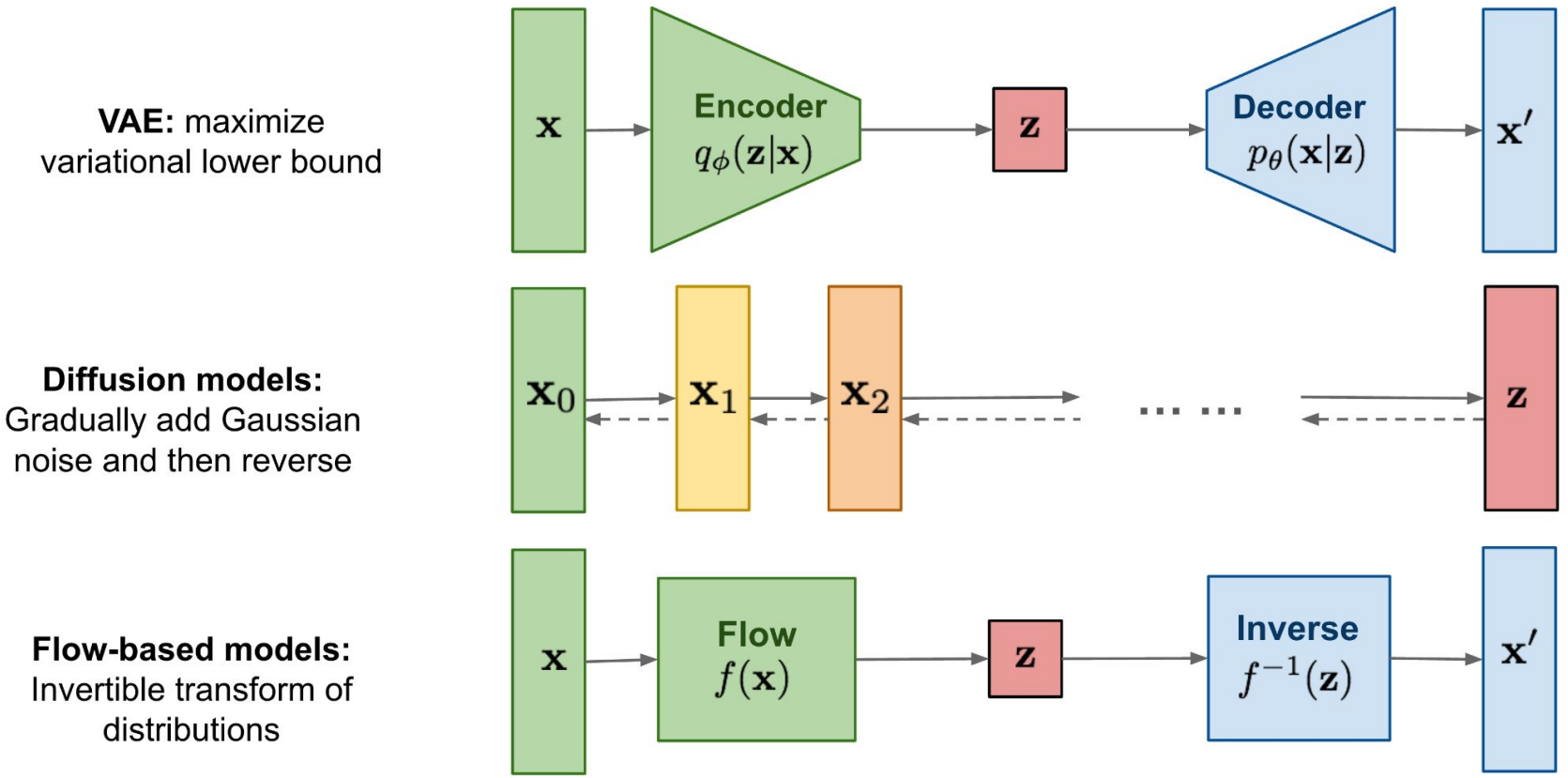
Diffusion models: Gradually add Gaussian noise and then reverse



Source: "What are Diffusion Models?", Lilian Weng., 2021

Connections to Diffusion Models

A comparison of generative modeling paradigms (from Lil'Log blog):



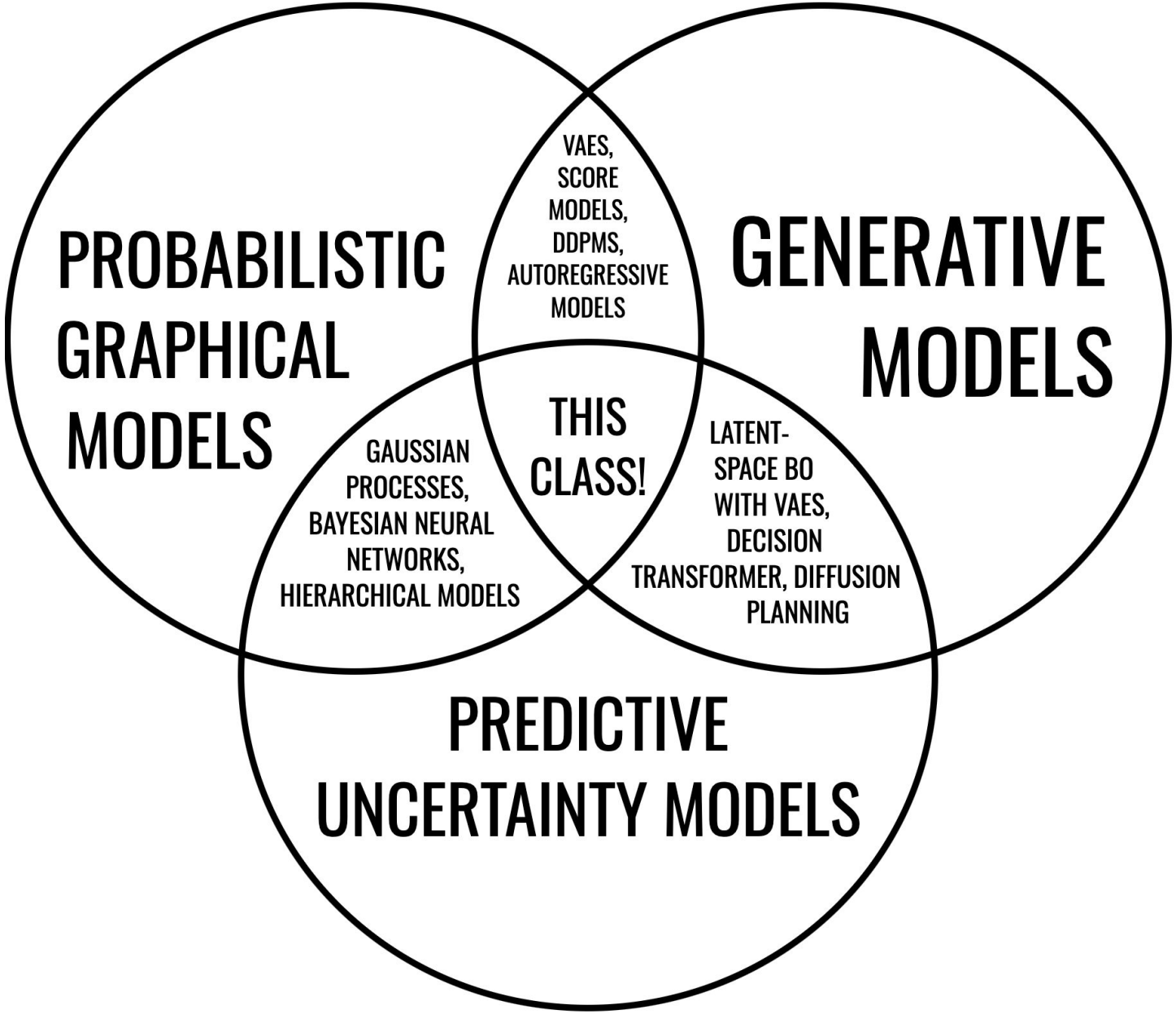
Source: "What are Diffusion Models?", Lilian Weng., 2021

Next: Predictive Uncertainty Quantification (UQ)

Class Outline

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

This course is inspired by some previous classes…



Deep Uncertainty Models

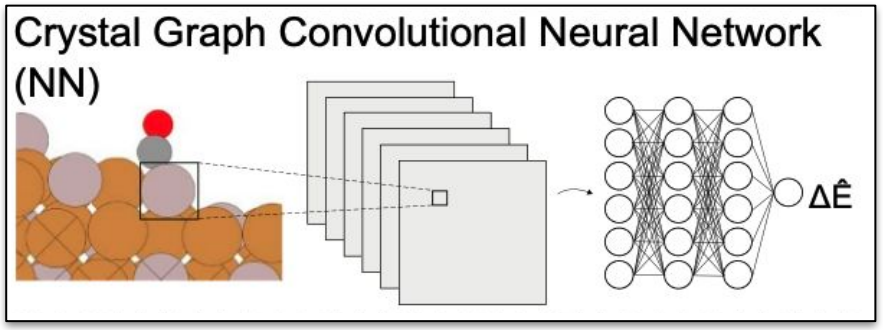
For example, we'll study methods for incorporating predictive uncertainty into neural networks:

Deep Uncertainty Models —Example: Computational Catalyst Design

For example, we'll study methods for incorporating predictive uncertainty into neural networks:

Deep Uncertainty Models —Example: Computational Catalyst Design

For example, we'll study methods for incorporating predictive uncertainty into neural networks:



*Xie, Tian, and Jeffrey C. Grossman. "**Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties.**" Physical review letters, 2018.*

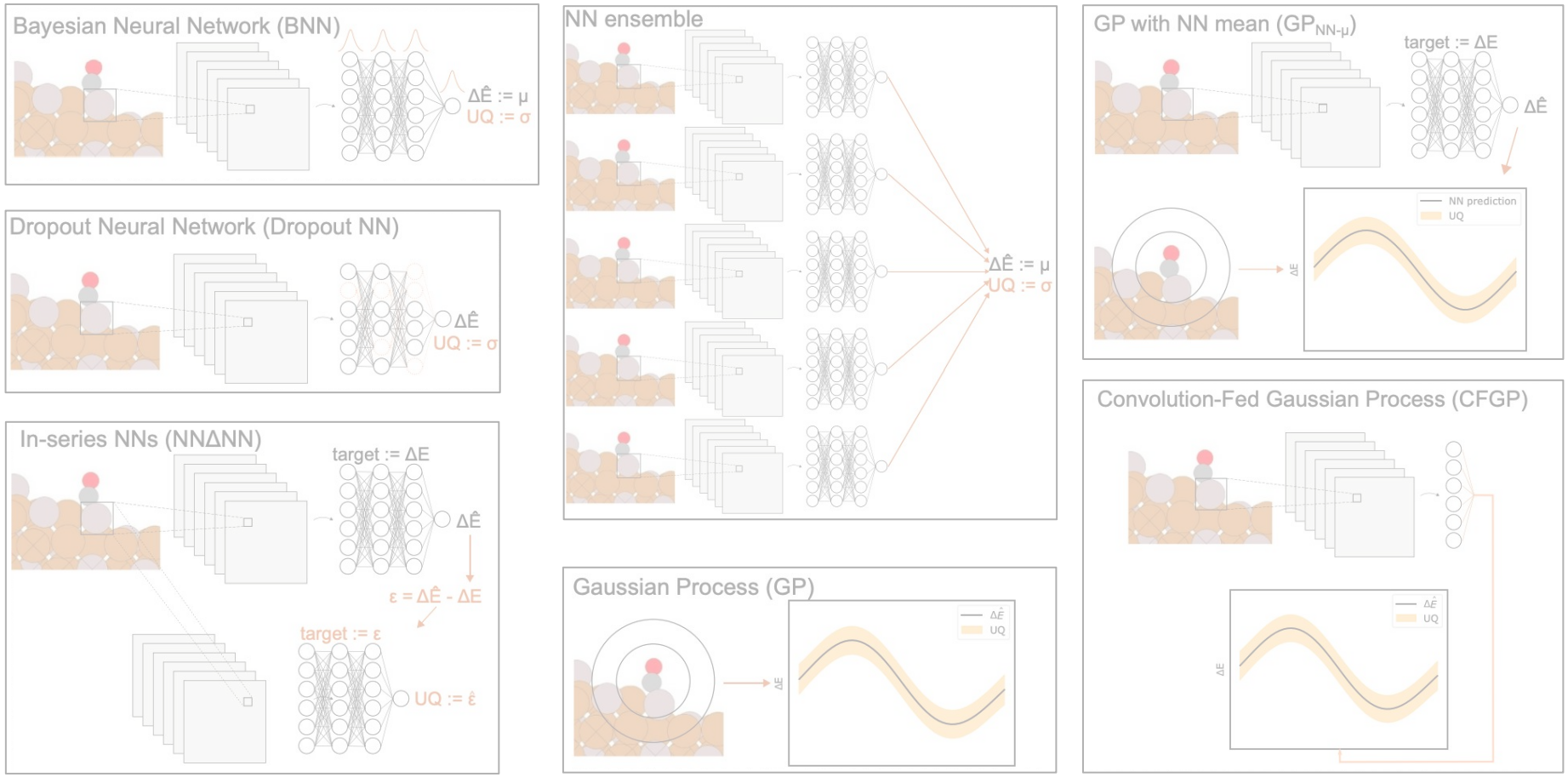
CGCNN Model (a graph convolutional neural network model)

- *Inputs*: three-dimensional atomic structure (a graph).
- *Outputs*: DFT-calculated site adsorption energies ΔE (*regression*).

"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

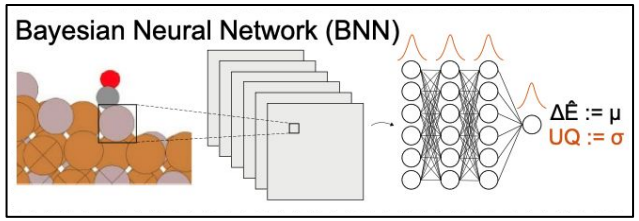
Deep Uncertainty Models —Example: Computational Catalyst Design

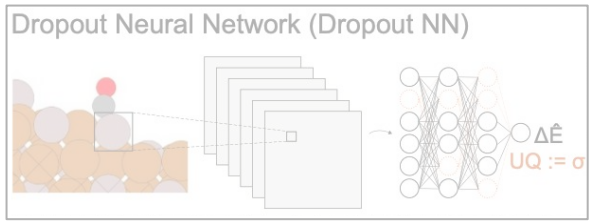


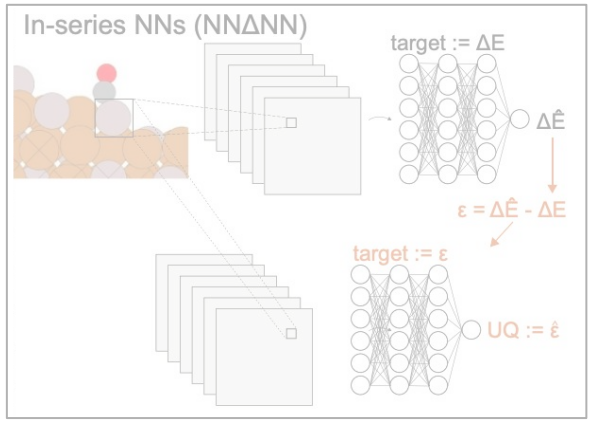
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

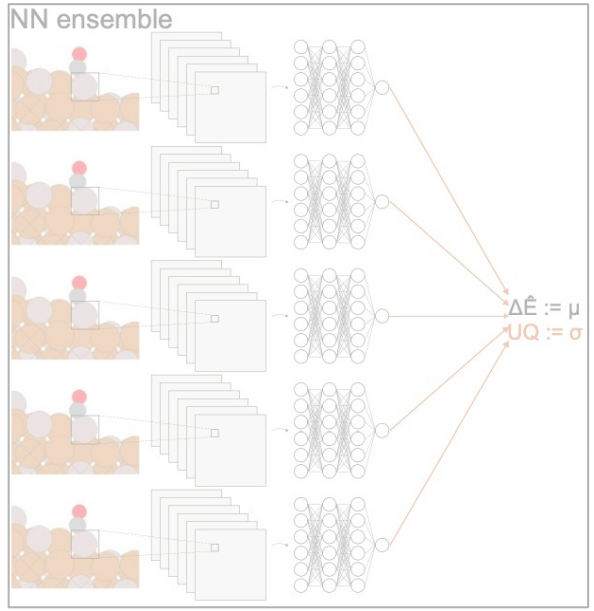
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

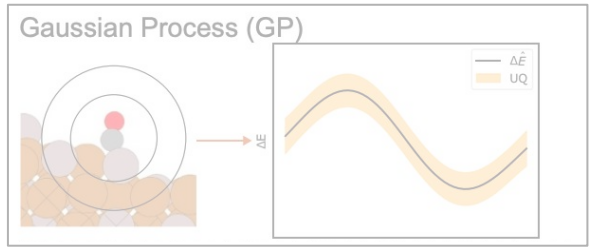
Deep Uncertainty Models —Example: Computational Catalyst Design

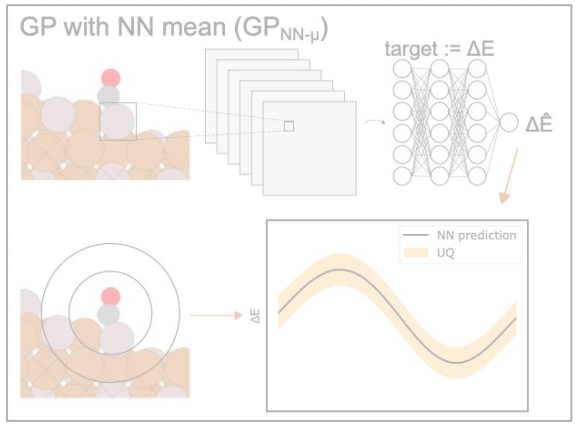


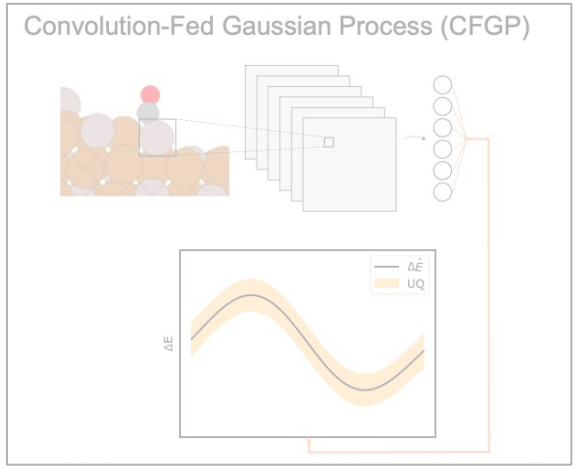








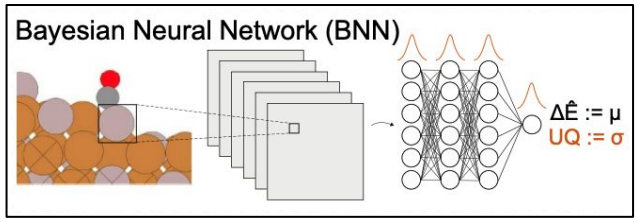


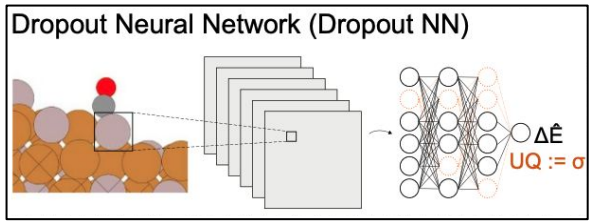


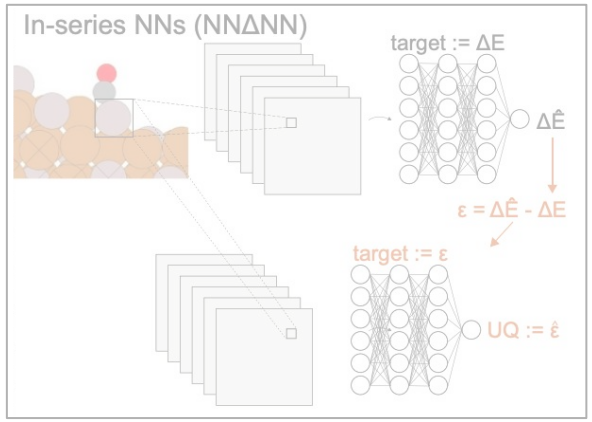
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

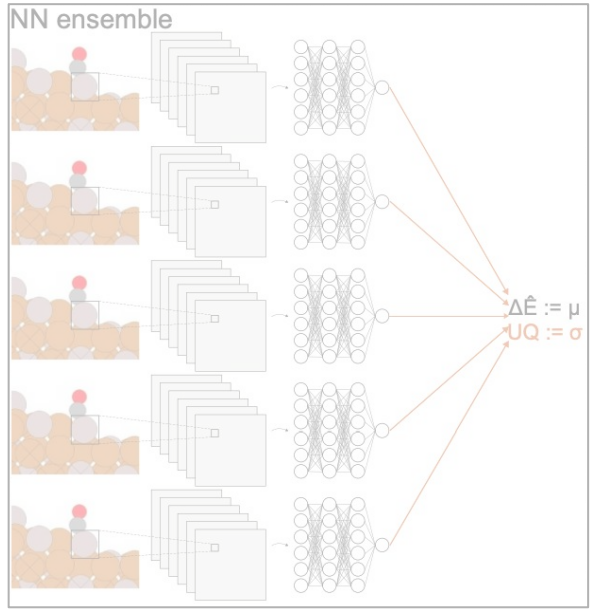
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

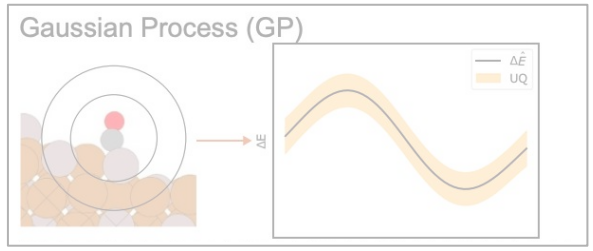
Deep Uncertainty Models —Example: Computational Catalyst Design

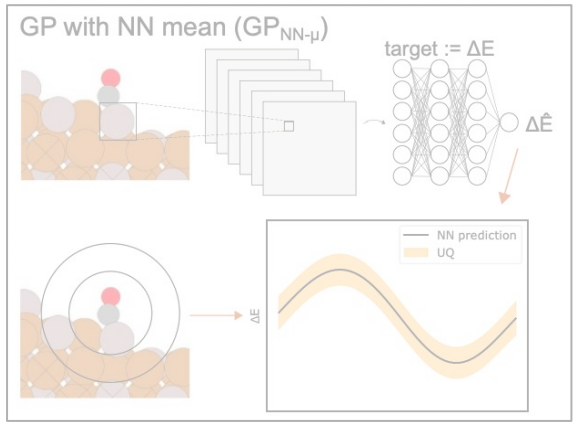


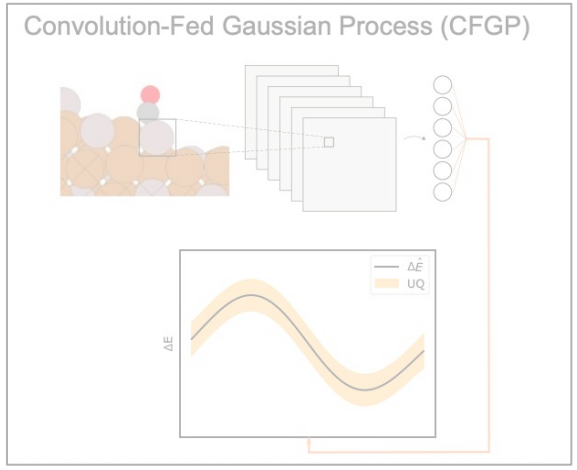








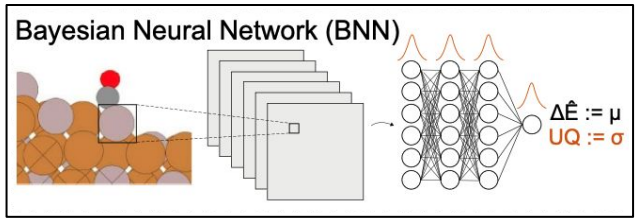


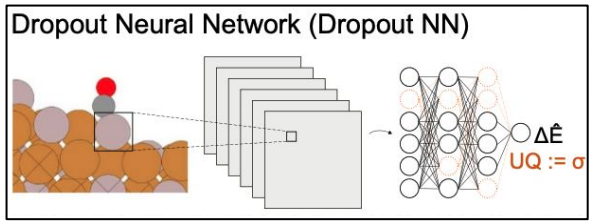


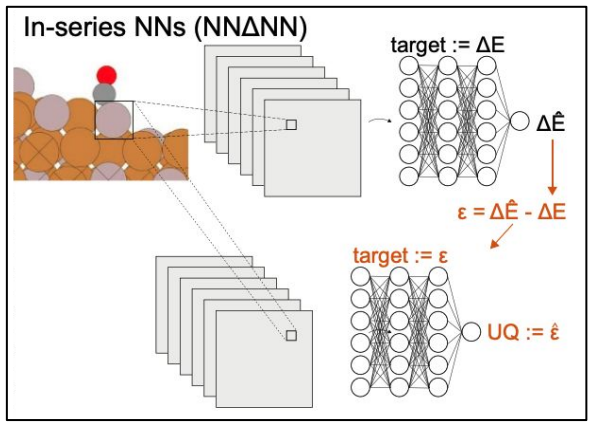
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

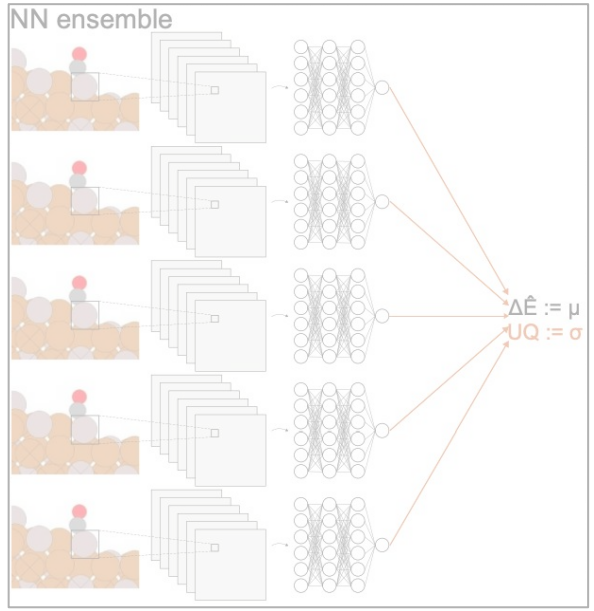
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

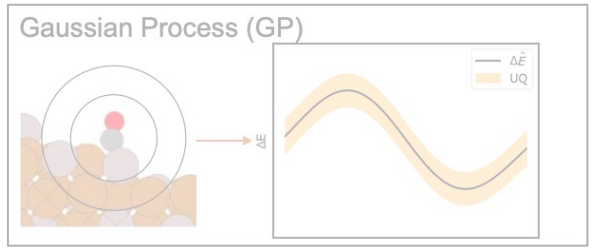
Deep Uncertainty Models —Example: Computational Catalyst Design

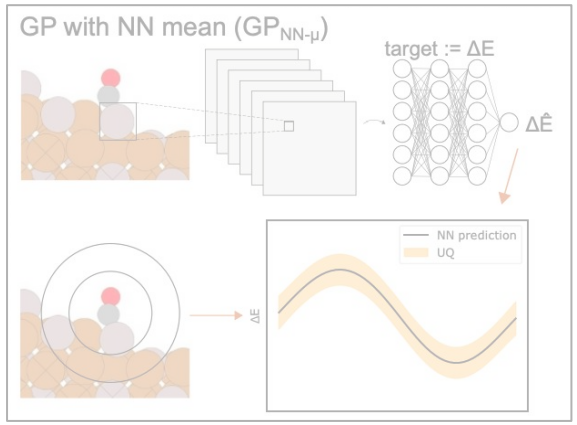


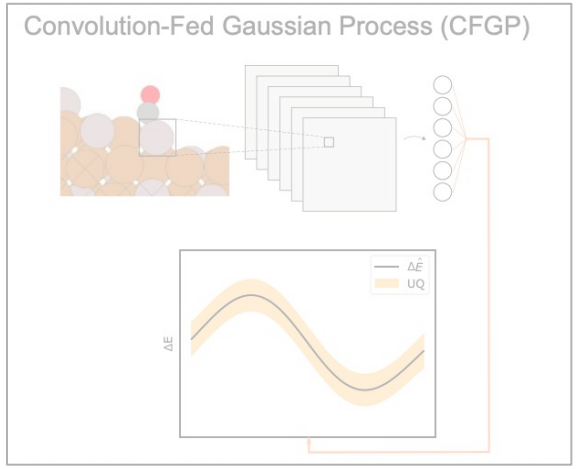








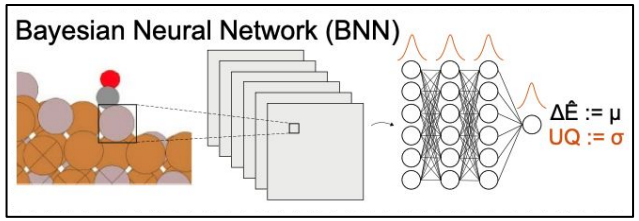


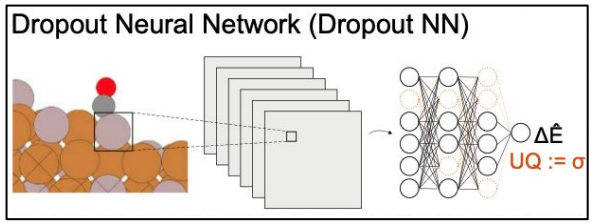


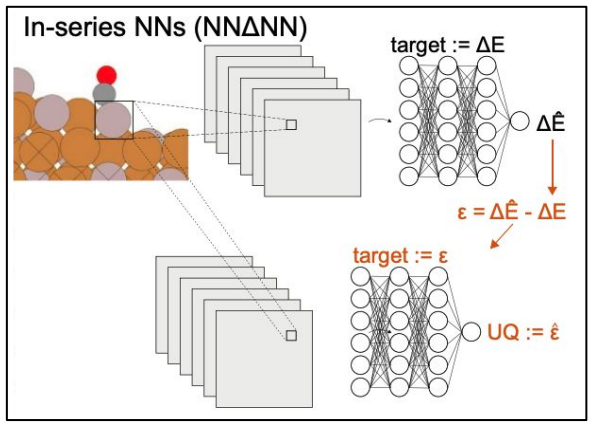
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

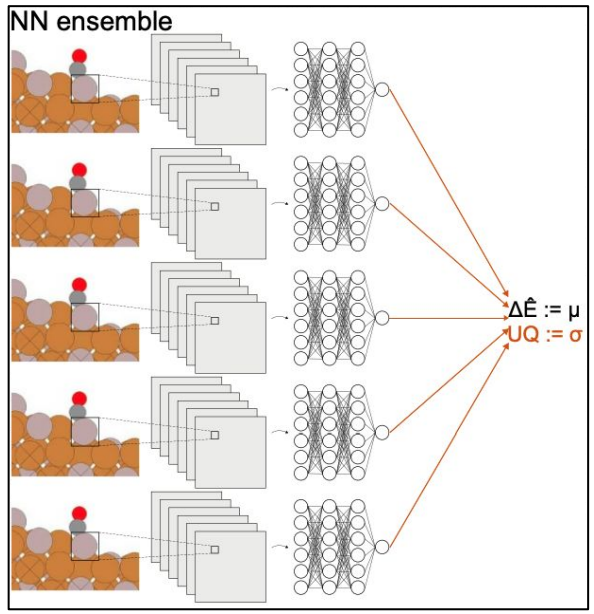
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

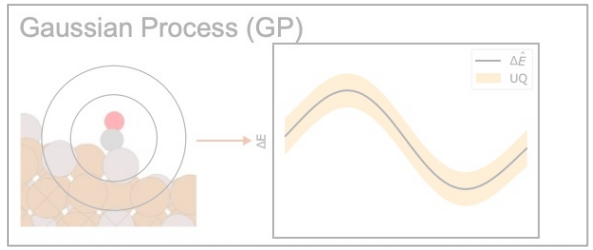
Deep Uncertainty Models —Example: Computational Catalyst Design

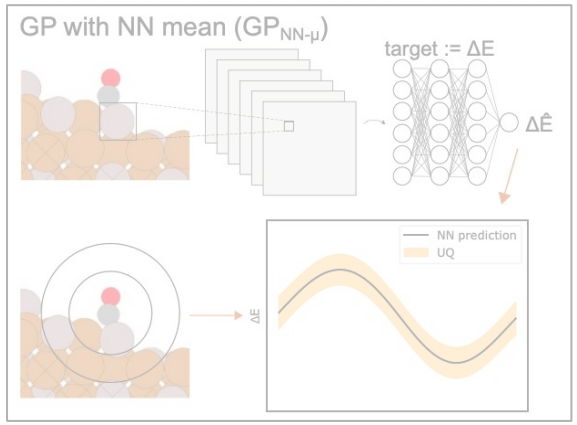


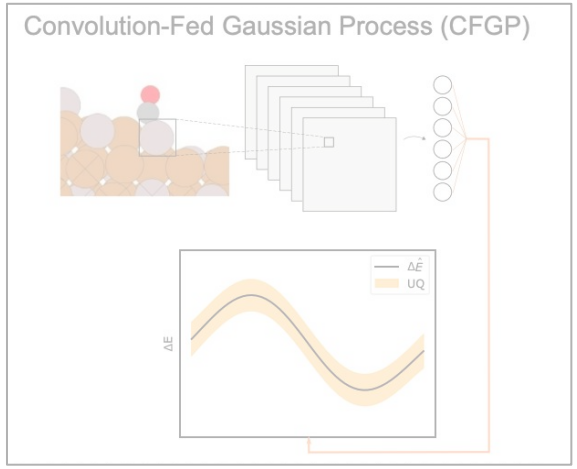








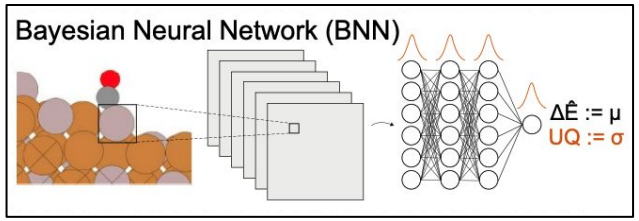


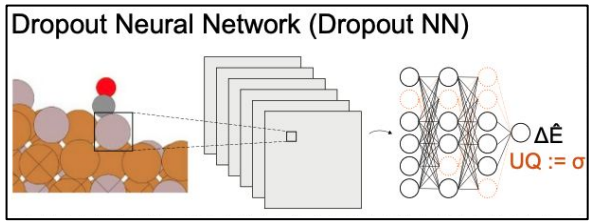


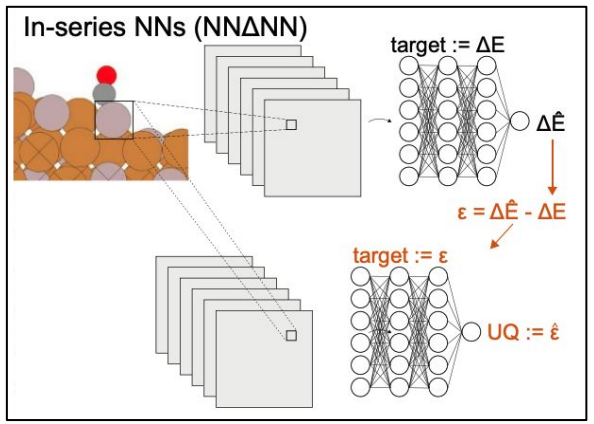
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

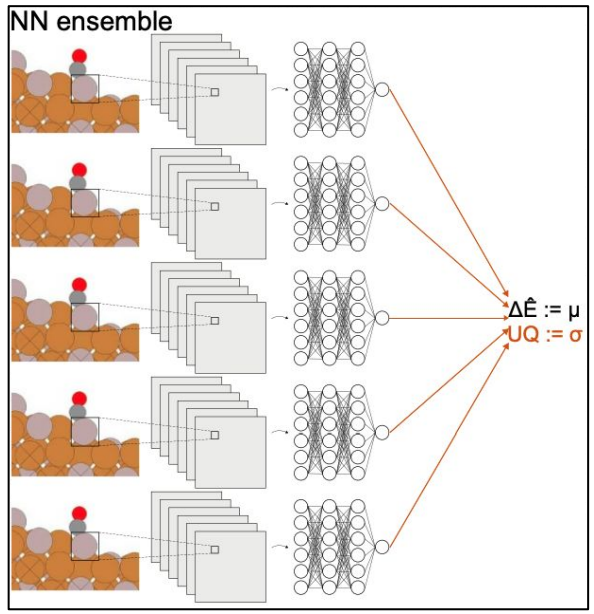
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

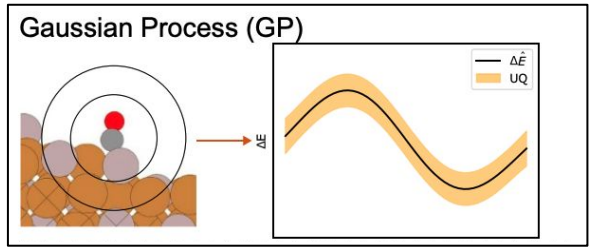
Deep Uncertainty Models —Example: Computational Catalyst Design

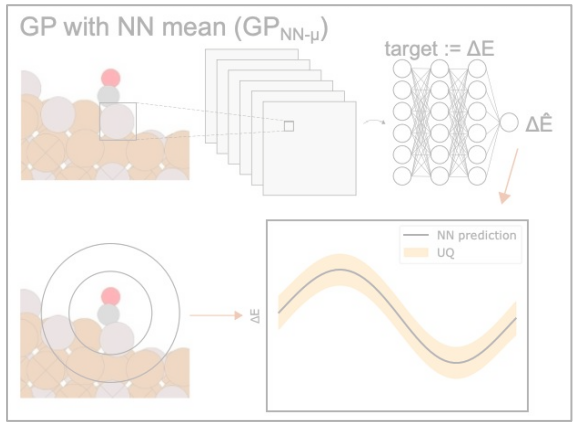


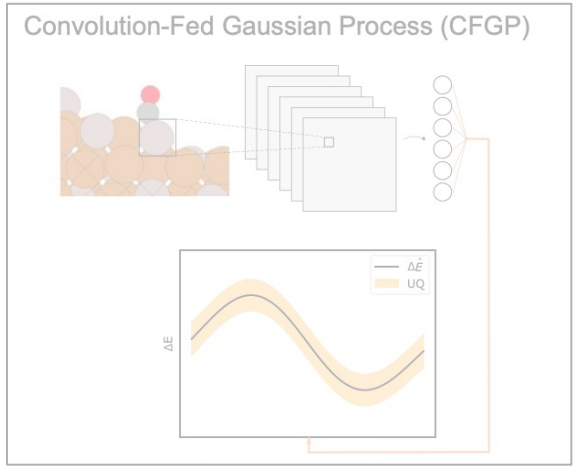








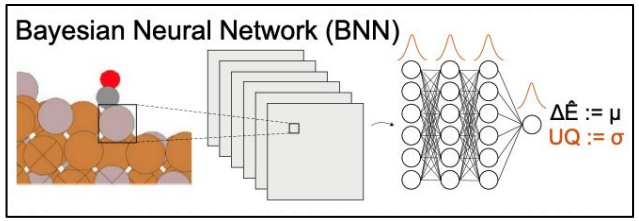


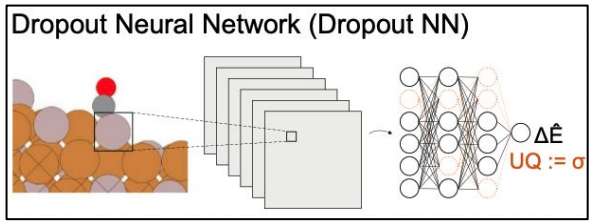


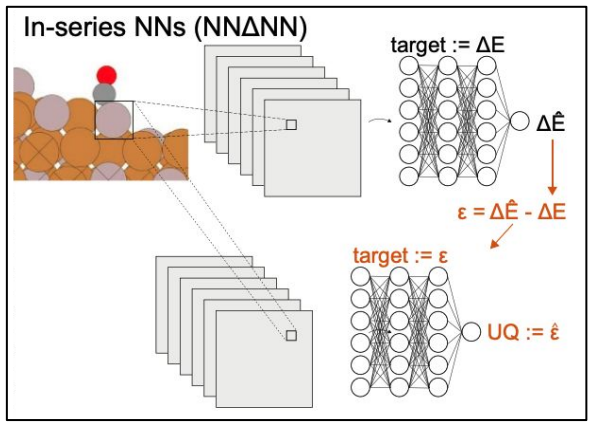
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

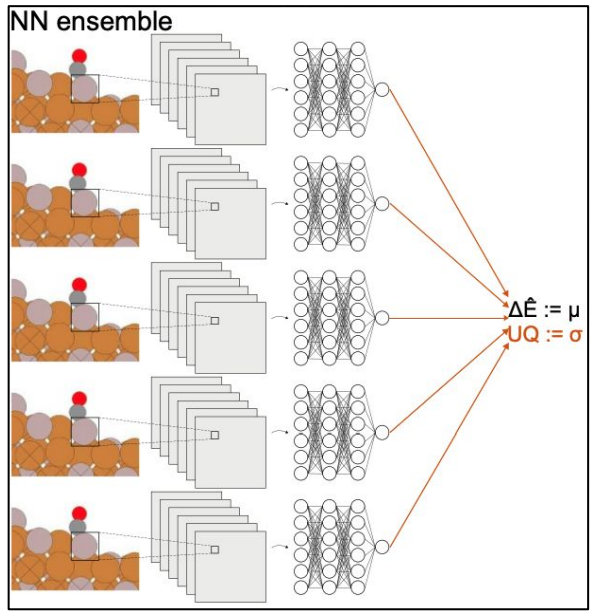
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

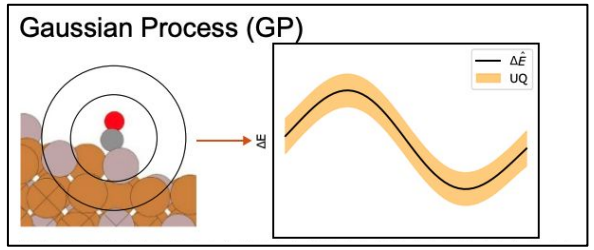
Deep Uncertainty Models —Example: Computational Catalyst Design

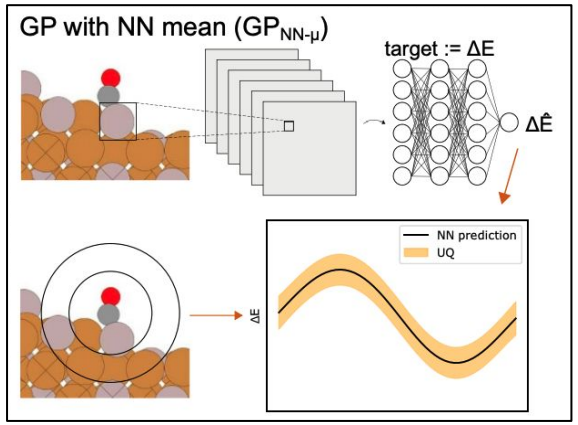


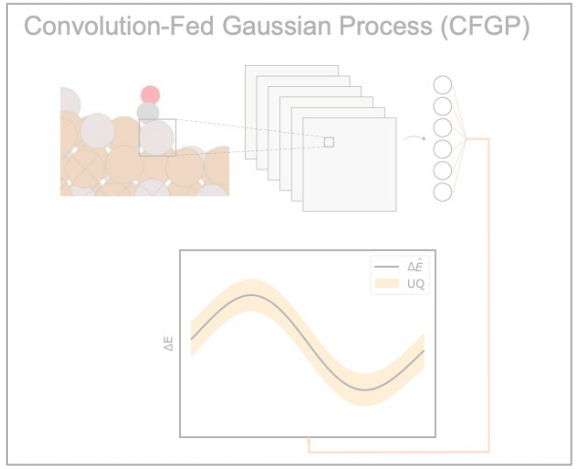








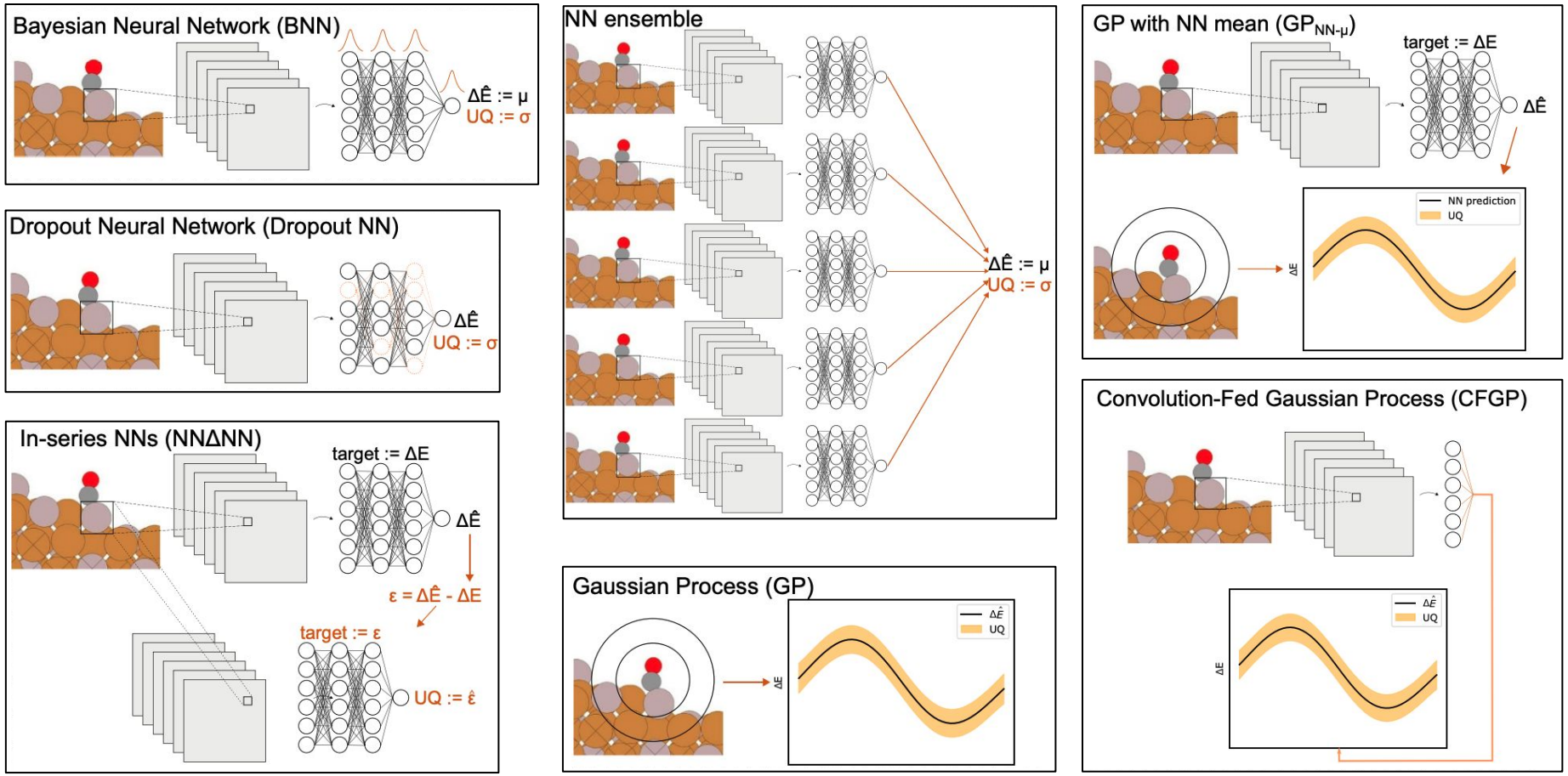




"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

Deep Uncertainty Models —Example: Computational Catalyst Design



"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021