Week 6: Multimodal Foundation Models

Generative AI
Saarland University – Winter Semester 2024/25

Adish Singla
genai-w24-tutors@mpi-sws.org

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

MAX-PLANCK-GESELLSCHAFT

Outline of the Lecture

- Organizational updates

- Token-based autoregressive generation

- Diffusion-based non-autoregressive generation

- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation

- Week 6 assignment

Outline of the Lecture

- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

Organizational Updates

- Next 2-3 lectures will take place in these two rooms (E1.5 029 + E1.4 024)
  - [26 Nov] Week 6
  - [03 Dec] Week 7
  - [10 Dec] Week 8?

- Week 6 assignment deadline: 05 Dec 2024 (Thursday) CET

Recap of the Course Timeline

[15 Oct] Week 1: Introduction

[22 Oct] Week 2: Background on Language Models and Transformers

[29 Oct] Week 3: Large Language Models and In-context Learning

[05 Nov] Week 4: Pre-training and Supervised Fine-tuning

[12 Nov] Week 5: Preference-based Fine-tuning for Alignment

[26 Nov] Week 6: Multimodal Foundation Models

[03 Dec] Week 7: Trustworthiness Aspects I

[10 Dec] Week 8: Trustworthiness Aspects II

[07 Jan] Week 9: GenAI-powered Programming Education I

[14 Jan] Week 10: GenAI-powered Programming Education II

[28 Jan] Week 11: Project Discussion

[04 Feb] Week 12: Examination Preparation

Outline of the Lecture

- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

[Recap Week 2] Predicting Next Tokens

Probability of a next token

- Probability of t-th token being x_t given history of $t-1$ tokens

$$P(x_t \mid x_1, x_2, x_3, \dots, x_{t-1})$$

or

$$P(x_t \mid x_{1:t-1})$$

- Models can generate content by predicting the next token over and over again
  - Called as *Autoregressive* models or token-based *autoregressive* generation
  - Chapter 10.1 and 10.2 of SLP book at https://web.stanford.edu/~jurafsky/slp3/

Types of Content and Tokens

- Content: Sentence → Tokens: Words
- Content: Code → Tokens: Programming Keywords
- Content: Image → Tokens: Pixel Values?

[Recap Week 3] Transformer-based Language Model

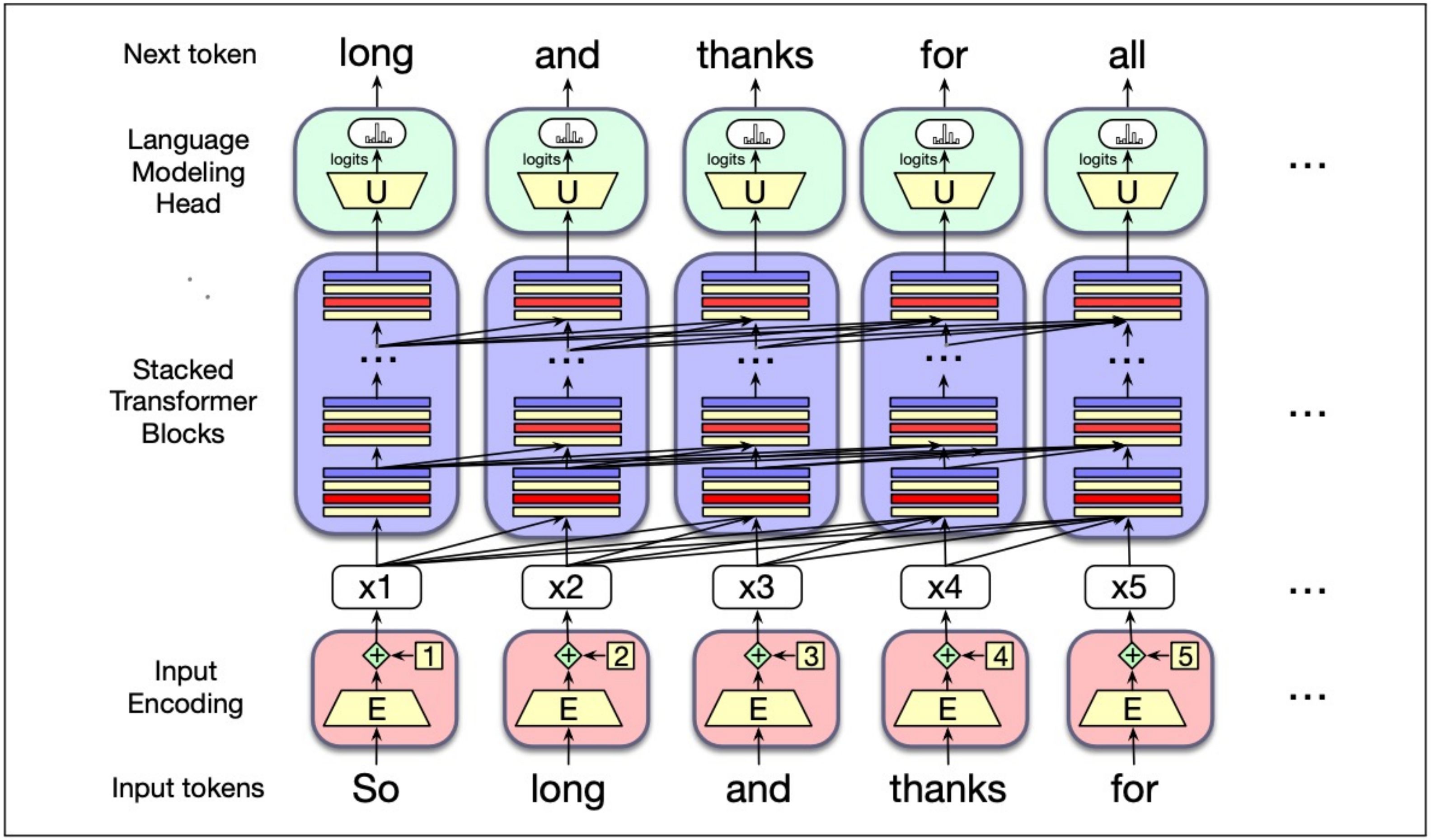


Figure 9.1 The architecture of a (left-to-right) transformer, showing how each input token get encoded, passed through a set of stacked transformer blocks, and then a language model head that predicts the next token.

Acknowledgements: Chapter 9.1 of SLP book at https://web.stanford.edu/~jurafsky/slp3/

[Recap Week 3] Transformer-based Language Model

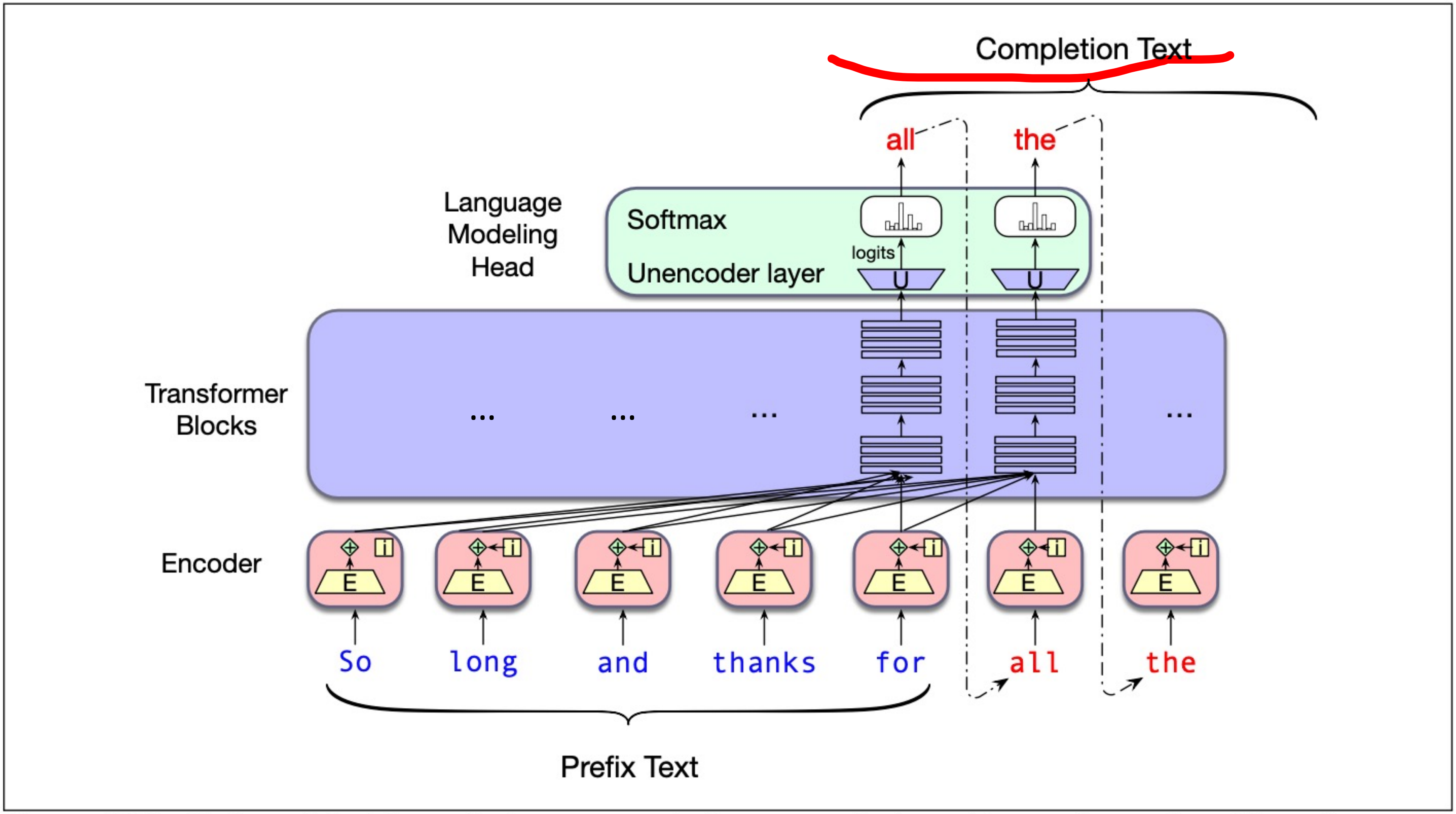


Figure 10.1 Left-to-right (also called autoregressive) text completion with transformer-based large language models. As each token is generated, it gets added onto the context as a prefix for generating the next token.

Acknowledgements: Chapter 10.1 of SLP book at https://web.stanford.edu/~jurafsky/slp3/

[Recap Week 3] Transformer-based Language Model

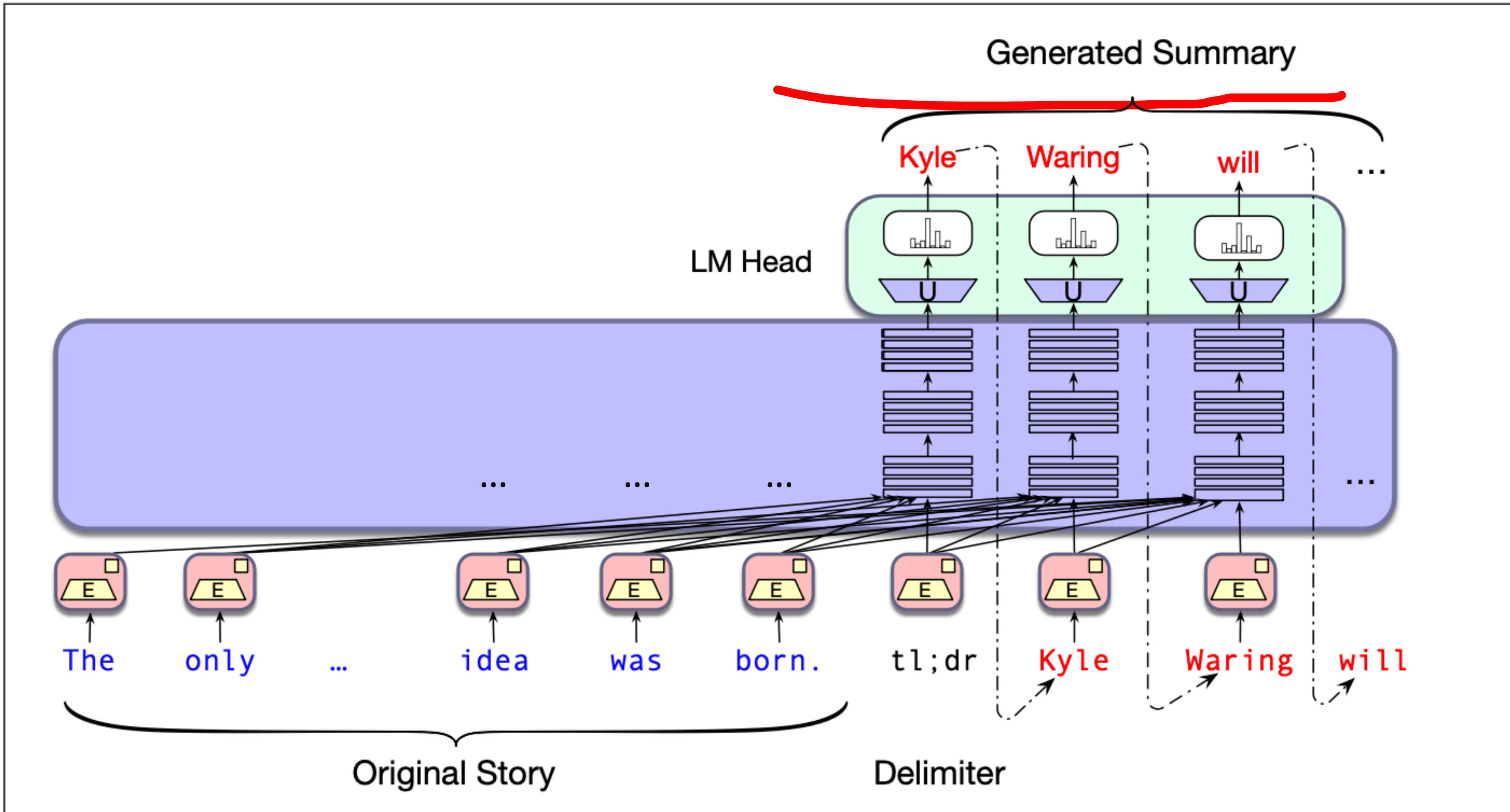


Figure 10.3 Summarization with large language models using the `tl;dr` token and context-based autoregressive generation.

Acknowledgements: Chapter 10.1 of SLP book at https://web.stanford.edu/~jurafsky/slp3/

Transformer-based Image Model: Pixel Values as Tokens

Transformer-based image model

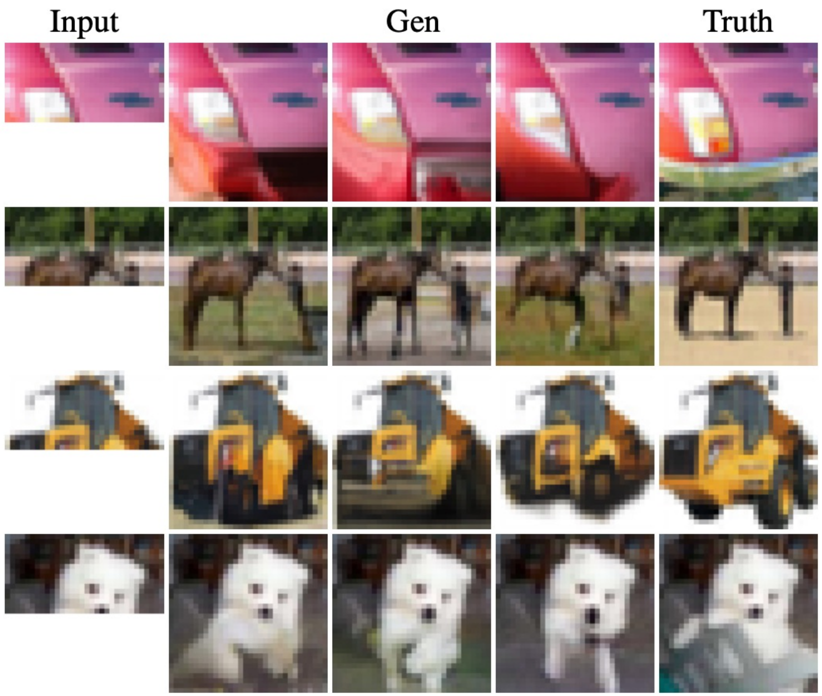
- Introduced in 2018: Image Transformer [Parmar et al., ICML'18]
- Partly same team who introduced Transformer [Vaswani et al., NeurIPS'17]

Key ingredients for image modeling

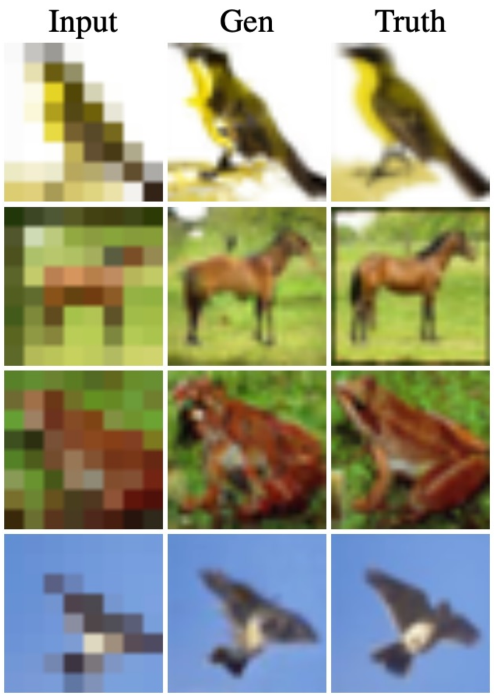
- Tokens and vocabulary:
  - ~~Next pixel~~ in terms of RGB intensity values
  - 256 vocabulary size when treating intensity values as discrete categories
- Context:
  - Part of the image (e.g., first half of the image)
  - Low resolution version of the image (e.g., 8x8 instead of 32x32)
  - Class of the image (e.g., plane, car)
- Attention: Attend all tokens or attend tokens in local neighborhood
- Positional encoding: 2-dimensional coordinates of a pixel

Transformer-based Image Model: Pixel Values as Tokens

Context: Half of the image (16x32 pixels)

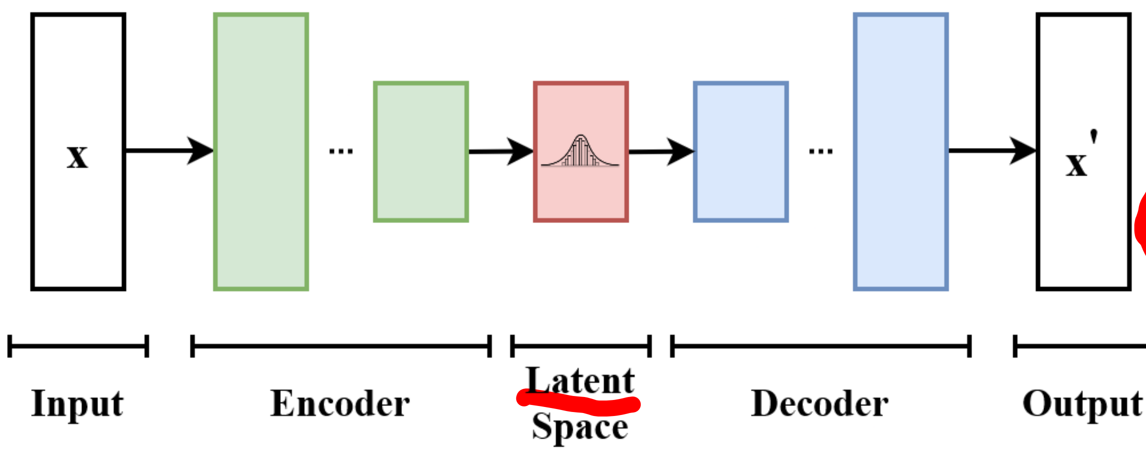


Context: 8x8 low-resolution image



Transformer-based Image Model: Latent Visual Tokens

- Pixel values as tokens is not scalable for high-resolution image generation
- Use a latent space of tokens for representing and generating images
- These latent visual tokens take values from a fixed visual codebook
  - Visual codebook is the vocabulary for the latent visual tokens, e.g., patterns
- Visual codebook learnt, e.g., with discrete Variational Autoencoder (dVAE)
- References for additional background
  - Neural Discrete Representation Learning [Oord et al., NeurIPS'17]
  - Generating Diverse High-Fidelity Images with VQ-VAE-2 [Razavi et al., NeurIPS'19]



视觉 Codebook（Visual Code
）
这个 Codebook 可以被离散变
编码器（dVAE）训练得到。
这些 Token 表示更高级的视
，比如：
纹理（patterns）
形状（shapes）
颜色块（color patches）

Transformer-based Image Model: OpenAI's DALL-E 1

DALL-E 1 Model

- Introduced by OpenAI in January 2021: https://openai.com/index/dall-e/
- Week 6 reading: Zero-Shot Text-to-Image Generation [Ramesh et al., ICML'21]
- DALL-E 1 is a transformer model: 12 billion parameter version of GPT-3

Key ideas: Visual codebook

- Learn a fixed visual codebook using dVAE
  - Represent a 256x256 RGB image with 32x32 grid of latent visual tokens
  - Each latent visual token is one of 8,192 possible categories (size of visual codebook)
- dVAE will also be used to encode or decode images

Key ideas: Unified stream of text and visual tokens

- Text tokens with vocabulary size of 16,384
- Visual tokens with vocabulary size of 8,192
- DALL-E 1 is trained with 250 million text-images pairs

Transformer-based Image Model: OpenAI’s DALL-E 1

Context: Text

- Input
  - Up to 256 text tokens with vocabulary size of 16,384
- Output
  - Latent output: 32x32 latent visual tokens with vocabulary size of 8,192
  - Final output: 256x256 RGB image

Text Prompt a store front that has the word ‘openai’ written on it. . . .

AI Generated images

Transformer-based Image Model: OpenAI's DALL-E 1

Context: (Text, Image)

- Input
  - Up to 256 text tokens with vocabulary size of 16,384
  - 256x256 RGB image ➔ 32x32 latent visual tokens with vocabulary size of 8,192
- Output
  - Latent output: 32x32 latent visual tokens with vocabulary size of 8,192
  - Final output: 256x256 RGB image

Shortcomings of Token-based Autoregressive Generation

Example application scenarios

- Improve the quality of given text
- Repair a given buggy code
- Refine the quality of an image

Potential shortcomings of token-based generation

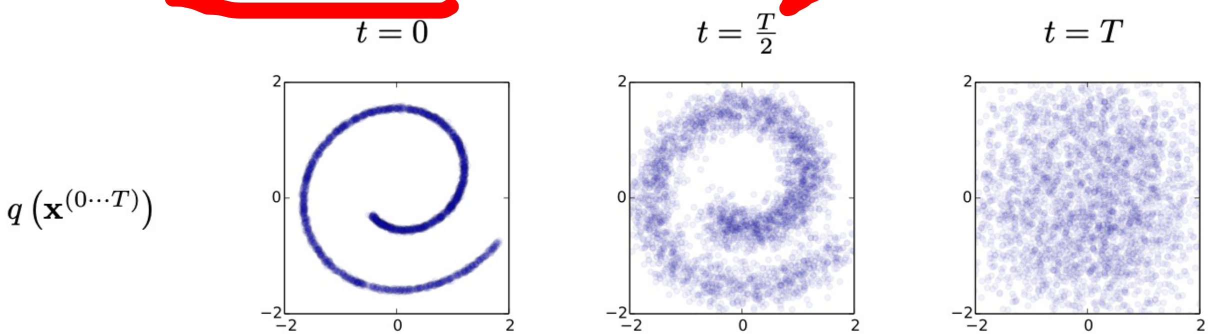
- Generation process would begin from scratch
  ➔ Lack an intuitive notion of iterative refinement
- Generation process goes token by token
  ➔ Output may not satisfy desired structural constraints

Outline of the Lecture

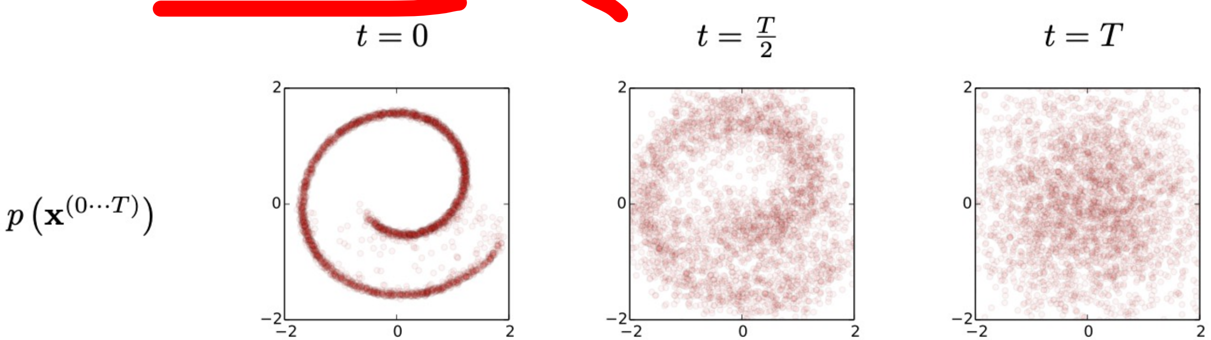
- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

Diffusion Models: Background

- Recently introduced in 2015: Deep Unsupervised Learning using Nonequilibrium Thermodynamics [Sohl-Dickstein et al., ICML'15]
- A diffusion model is characterized by two processes:
  - Forward process: Data → Noise



  - Reverse process: Data ← Noise



Diffusion Models: Background

- Got popular after a follow-up paper demonstrated high-quality image generation: Denoising Diffusion Probabilistic Models [Jo et al., NeurIPS'20]

Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

Diffusion Models: Background

- Got popular after a follow-up paper demonstrated high-quality image generation: Denoising Diffusion Probabilistic Models [Jo et al., NeurIPS'20]

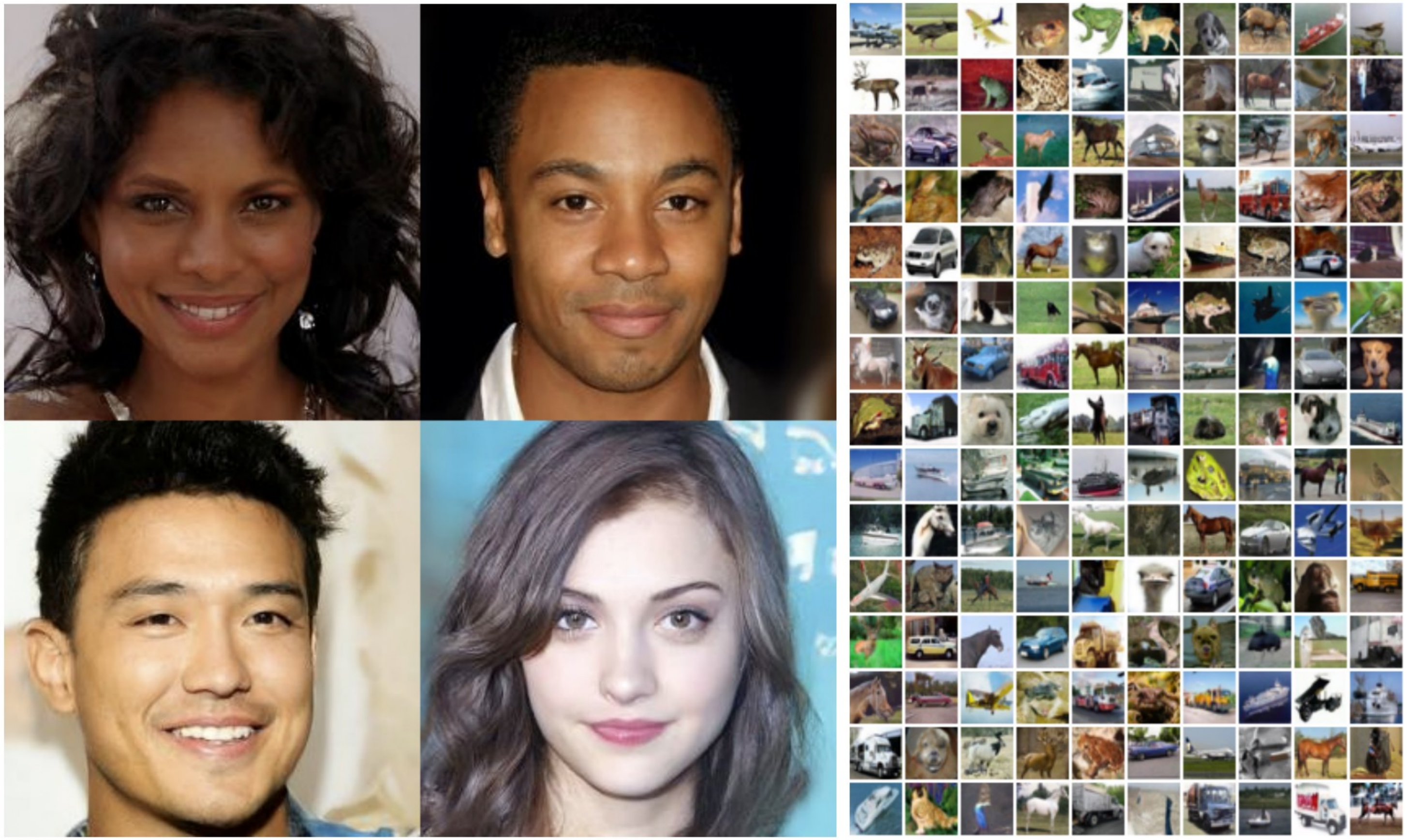


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Diffusion Models: Forward Process

- Based on Denoising Diffusion Probabilistic Models [Jo et al., NeurIPS’20]

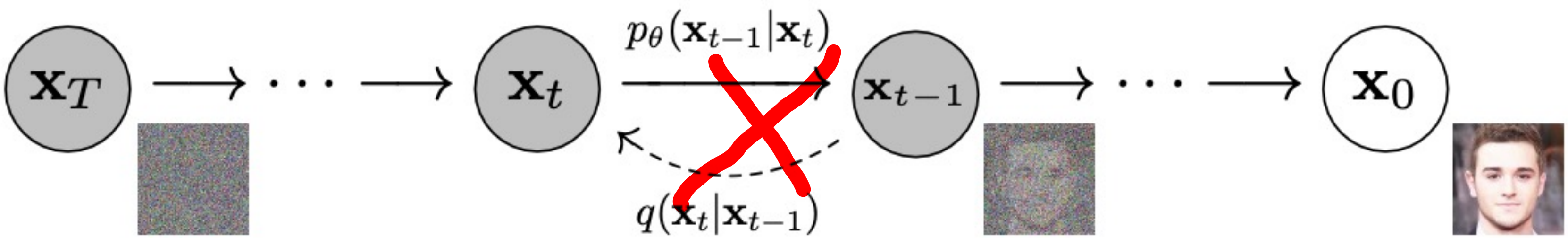


Figure 2: The directed graphical model considered in this work.

小，仅添加少量噪声，
数据仍然接近原始数据。
大，添加更多噪声，
数据变得更加随机。

- Starting at $\mathbf{x}_0 \sim q(\mathbf{x}_0) \coloneqq \text{data distribution}$

- A Markov chain gradually adds Gaussian noise with fixed parameters β_t

$$q(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

均值

协方差矩阵

数据保留部分

噪音部分

Diffusion Models: Reverse Process

- Based on Denoising Diffusion Probabilistic Models [Jo et al., NeurIPS'20]

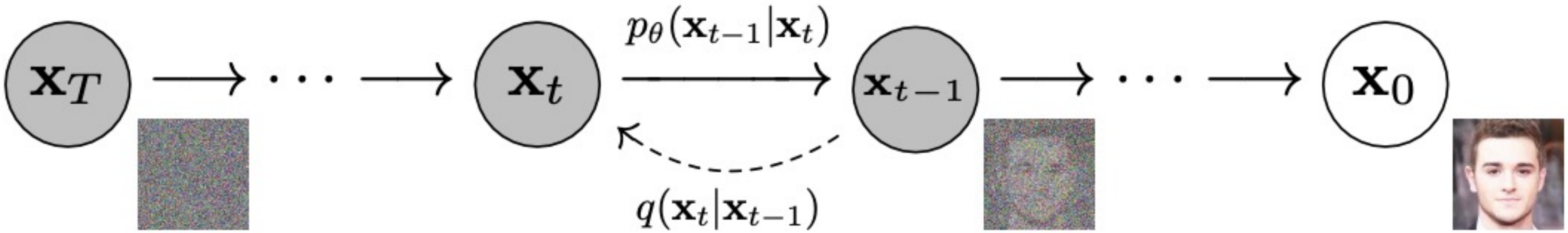


Figure 2: The directed graphical model considered in this work.

- Starting at $\mathbf{x}_T \sim p(\mathbf{x}_T) \coloneqq \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

- A Markov chain having Gaussian transitions with learnable parameters θ

$$p_\theta(\mathbf{x}_{t-1}|\,\mathbf{x}_t) \coloneqq \mathcal{N}\big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\big)$$

均值，去噪的方向，朝向真实数据逼近；方差，
去噪时添加多少随机性，防止模式崩溃

Diffusion Models: Simpler Parameterization and Objective

- [Jo et al., NeurIPS'20] proposed a simpler model parameterization and objective
- Simplifications with following fixed parameters:
  - Forward process uses fixed parameters for β_t (i.e., hyperparameters, not learnt)
  - Reverse process uses fixed parameters for $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) \coloneqq \sigma_t^2 \mathbf{I}$, e.g., $\sigma_t^2 = \beta_t$
- Loss term at step t for sample $\mathbf{x}_t$ corresponds to minimizing the KL divergence:
  - $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$: $\mathbf{x}_{t-1}$ distribution from forward process posteriors
  - $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$: $\mathbf{x}_{t-1}$ distribution from reverse process modeled by θ

- Consider data point $\mathbf{x}_0$ and step t. We can sample $\mathbf{x}_t$ at any step t in closed form:
  - We get $q(\mathbf{x}_t | \mathbf{x}_0) \coloneqq \mathcal{N}\left(\mathbf{x}_t; \sqrt{\overline{\alpha_t}}\, \mathbf{x}_0, (1 - \overline{\alpha_t})\, \mathbf{I}\right)$ where $\alpha_t \coloneqq 1 - \beta_t$ and $\overline{\alpha_t} \coloneqq \prod_s^t \alpha_s$
  - For noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we can write $\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\overline{\alpha_t}}\, \mathbf{x}_0 + (1 - \overline{\alpha_t})\boldsymbol{\epsilon}$

- Proposed simpler model parameterization and training objective:
  - $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ intends to predict noise $\boldsymbol{\epsilon}$ from $\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon})$
  - $\text{loss for step } t \coloneqq \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\overline{\alpha_t}}\, \mathbf{x}_0 + \sqrt{1 - \overline{\alpha_t}}\, \boldsymbol{\epsilon}, t\right) \right\|$

Diffusion Models: Training and Sampling Procedure

- Based on simpler parameterization and objective by [Jo et al., NeurIPS'20]

Algorithm 1 Training

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

Algorithm 2 Sampling

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

Diffusion-based Image Model: Diffusion in Pixel Space

- [Jo et al., NeurIPS'20] operates directly in the pixel space
  - 32x32 RGB images for CIFAR10 dataset
  - 256x256 RGB images for CelebA-HQ dataset
- $\epsilon_\theta(\mathbf{x}_t, t)$ intends to predict noise ϵ from $\mathbf{x}_t(\mathbf{x}_0, \epsilon)$
  - Input: Image of dimensionality d
  - Output: Predicted noise of same dimensionality d
- CNN-based model architecture with parameters shared across steps
  - Step t is encoded via positional embedding

Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

Diffusion-based Image Model: Diffusion in Latent Space

- Pixel space is not scalable for high-resolution image generation
- Use a latent space for representation/generation, introduced by High-Resolution Image Synthesis with Latent Diffusion Models [Rombach et al., CVPR'22]
  - E.g., use Variational Autoencoder (VAE) for learning a fixed latent space
- Moreover, reverse process parameters are conditioned on context (e.g., prompt)
- This work forms the basis of Stable Diffusion family of models

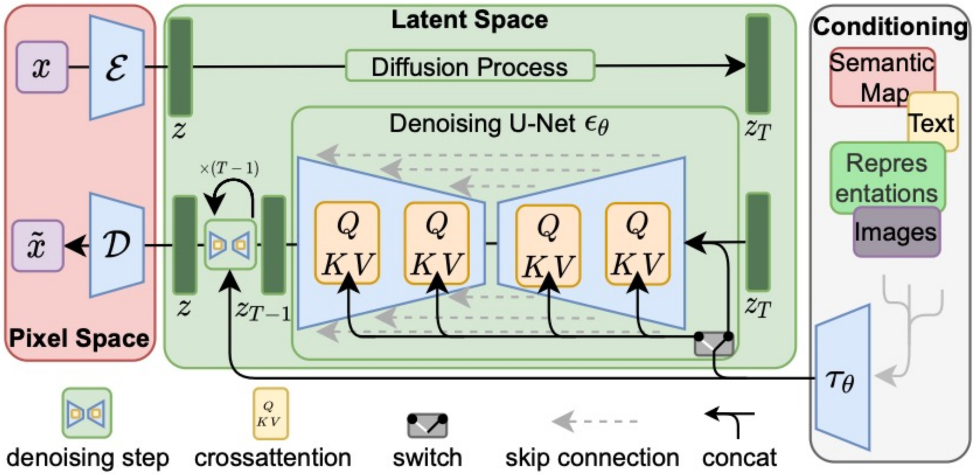


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism. See Sec. 3.3

Diffusion Models: Background and Recent Advances

- Recently introduced in 2015: Deep Unsupervised Learning using Nonequilibrium Thermodynamics [Sohl-Dickstein et al., ICML'15]
- Got popular after a follow-up paper demonstrated high-quality image generation: Denoising Diffusion Probabilistic Models [Jo et al., NeurIPS'20]
- Scaling up to high-resolution images with latent space: High-Resolution Image Synthesis with Latent Diffusion Models [Rombach et al., CVPR'22]

- State-of-the-art image generation is based on diffusion-based models
  - Stable Diffusion
  - DALL-E 2 / DALL-E 3 (Note: DALL-E 1 was token-based autoregressive model)

- Recent interest in using diffusion-based models for text generation
  - Diffusion-LM Improves Controllable Text Generation [Li et al., NeurIPS'22]
  - Text Generation with Diffusion Language Models: A Pre-training Approach with Continuous Paragraph Denoise [Lin et al., ICML'23]
  - CODEFUSION: A Pre-trained Diffusion Model for Code Gen. [Singh et al., EMNLP'23]

Diffusion-based Language Model: Controllable Generation

- Based on Diffusion-LM Improves Controllable Text Generation [Li et al., NeurIPS'22]

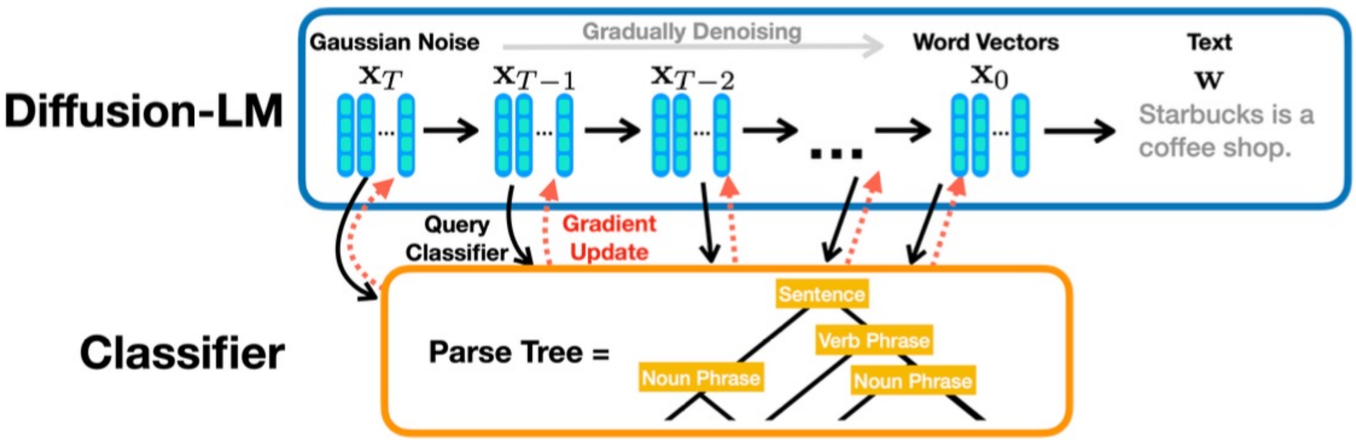


Figure 1: Diffusion-LM iteratively denoises a sequence of Gaussian vectors into word vectors, yielding a intermediate latent variables of decreasing noise level $\mathbf{x}_T \cdots \mathbf{x}_0$. For controllable generation, we iteratively perform gradient updates on these continuous latents to optimize for fluency (parametrized by Diffusion-LM) and satisfy control requirements (parametrized by a classifier).

Summary of Token-based and Diffusion-based Generation

- Token-based autoregressive models dominate text generation
- Diffusion-based non-autoregressive models dominate image and video generation
- Emergent multimodal foundation models seek to leverage both architectures

Outline of the Lecture

- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

Multimodal Joint Representation: Background

Multimodal generative models

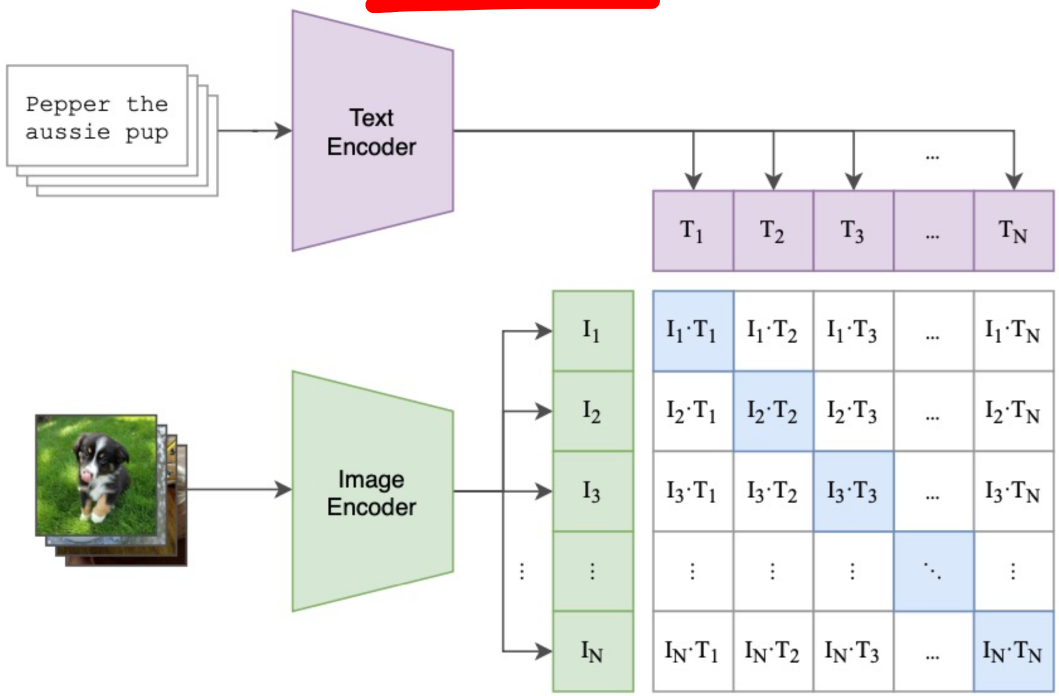
- Models for text generation
  - Input: (text, image)
  - Output: text
- Models for image generation
  - Input: (text, image)
  - Output: image

Intermediate joint representation for multimodal data

- Input (text, image) → Joint representation → Output
- Recall DALL-E 1: Zero-Shot Text-to-Image Generation [Ramesh et al., ICML'21]
  - Transformer-based model on unified stream of text and visual tokens
  - Implicitly learning a joint representation for (text, image) data
- Can we explicitly learn a joint representation for (text, image) data?

Multimodal Joint Representation: OpenAI's CLIP

- Introduced by OpenAI in January 2021: https://openai.com/index/clip/
- Week 6 reading: Learning Transferable Visual Models From Natural Language Supervision [Radford et al., ICML'21]
- Architecture and training
    - Text encoder: Transformer; Image encoder: Vision Transformer or ResNet
    - Trained over 400 million text-images pairs



训练目标：最大化匹配文本-图
的相似度，最小化错误匹配的
度（对比学习）。

Multimodal Joint Representation: OpenAI's CLIP

- CLIP can be readily applied for zero-shot image classification to any domain
  - CLIP's performance matches state-of-the-art models trained for specific domains
- Trained encoders later used for multimodal generative models (e.g., DALL-E 2)

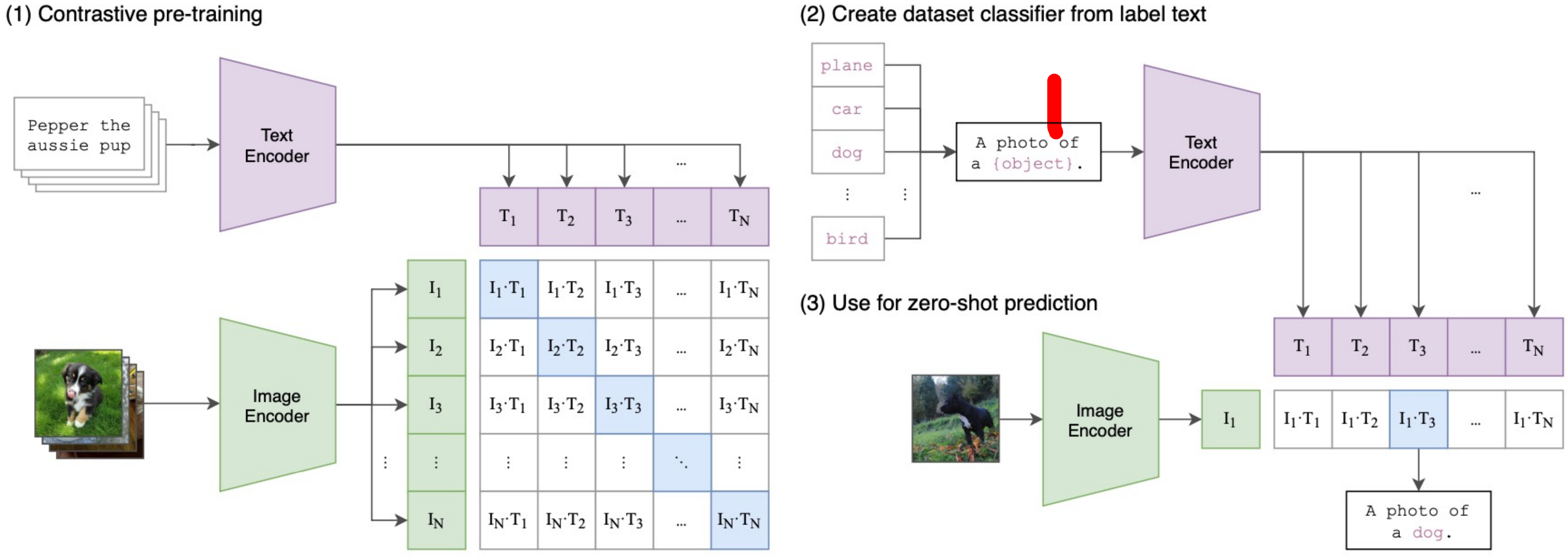


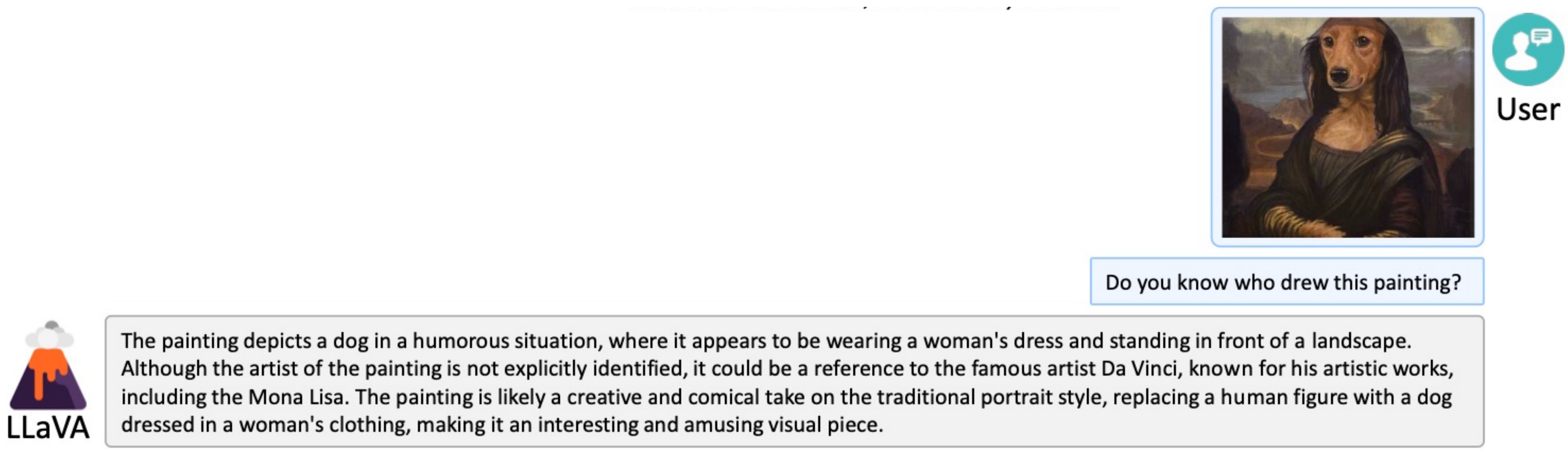
Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Outline of the Lecture

- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

Multimodal Text Generation: LLaVA

- LLaVA, i.e., Large Language and Vision Assistant
  - connects CLIP’s image encoder with transformer-based language model
  - trained end-to-end on synthetic data: (text, image)→ text
- Week 6 reading: Visual Instruction Tuning [Liu et al., NeurIPS’23]



Multimodal Text Generation: LLaVA

- LLaVA, i.e., Large Language and Vision Assistant
  - connects CLIP's image encoder with transformer-based language model
  - trained end-to-end on synthetic data: (text, image)→ text
- Week 6 reading: Visual Instruction Tuning [Liu et al., NeurIPS'23]

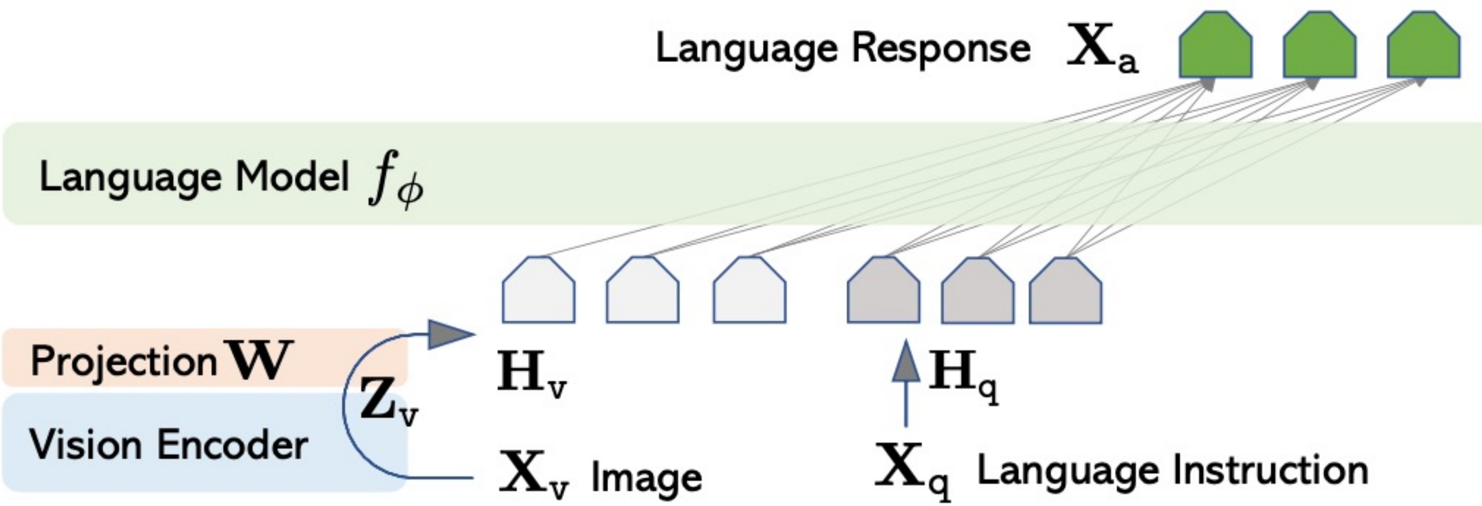


Figure 1: LLaVA network architecture.

Multimodal Text Generation: OpenAI's GPT-1 → GPT-4o

- **OpenAI models:** https://platform.openai.com/docs/models

- **GPT-1 to GPT-3.5:** Introduced between 2018 and 2022
  - text → text

- **GPT-4:** Introduced in March 2023: https://openai.com/index/gpt-4/
  - (text, image) → text
  - First model in the GPT series with capabilities for image understanding
  - Exact architecture and training details are not available

- **GPT-4o:** Introduced in May 2024: https://openai.com/index/hello-gpt-4o/
  - (text, image) → text when using OpenAI's API for GPT-4o
  - Improvements focused on multimodal data, including text, image, and speech
  - Exact architecture and training details are not available
  - Week 6 implementation: Experiment with GPT-4o model using OpenAI's API

Outline of the Lecture

- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal joint representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

Multimodal Image Generation: OpenAI's DALL-E 2

- Introduced by OpenAI in April 2022: https://openai.com/index/dall-e-2/
- Hierarchical Text-Conditional Image Gen. with CLIP Latents [Ramesh et al., '22]

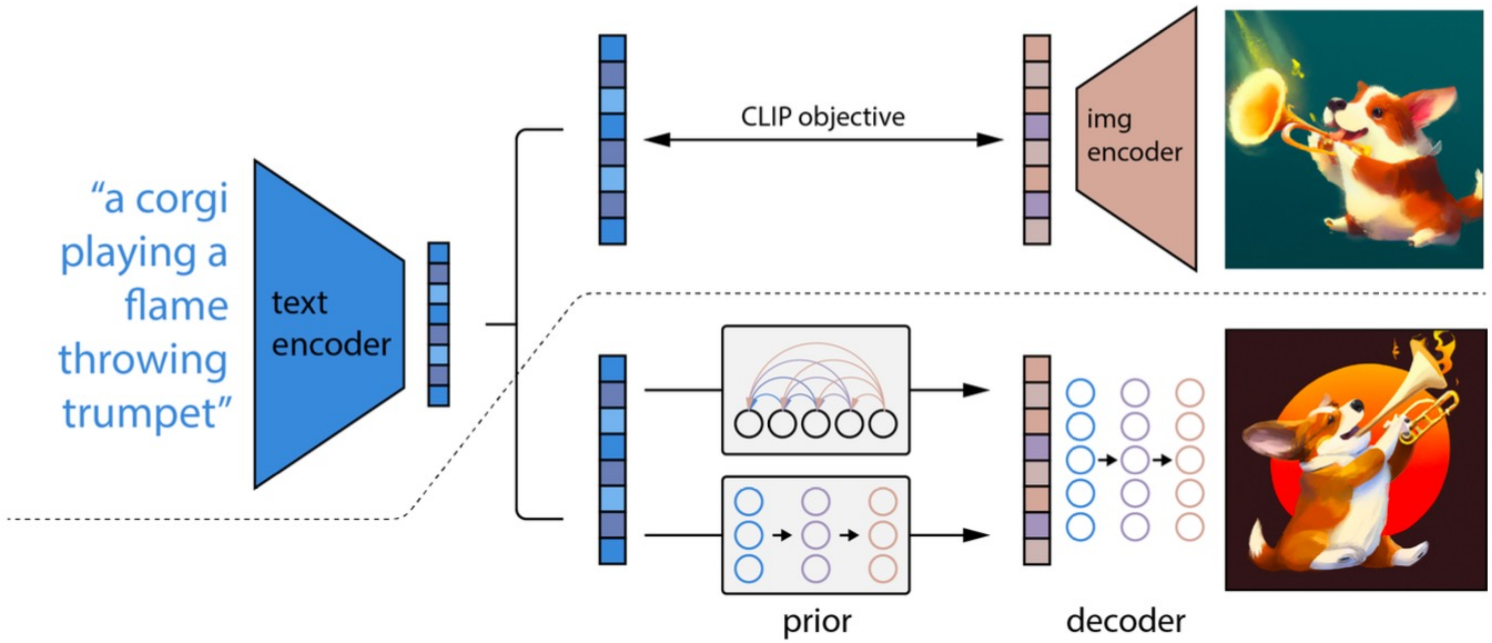


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

Multimodal Image Generation: OpenAI's DALL-E 1/2/3

- **DALL-E 1:** Introduced in January 2021: https://openai.com/index/dall-e/
  - Week 6 reading: Zero-Shot Text-to-Image Generation [Ramesh et al., ICML'21]
  - DALL-E 1 is a transformer model (a smaller version of GPT-3)

- **DALL-E 2:** Introduced in April 2022: https://openai.com/index/dall-e-2/
  - Week 6 optional reading: Hierarchical Text-Conditional Image Generation with CLIP Latents [Ramesh et al., '22]
  - DALL-E 2 uses diffusion model for image generation via CLIP embeddings

- **DALL-E 3:** Introduced in September 2023: https://openai.com/index/dall-e-3/
  - Week 6 optional reading: Improving Image Generation with Better Captions [Betker et al., '23]
  - DALL-E 3 uses diffusion model and improves on DALL-E 2
  - Exact architecture and training details are not available
  - Week 6 implementation: Experiment with DALL-E 3 model using OpenAI's API

Outline of the Lecture

- Organizational updates
- Token-based autoregressive generation
- Diffusion-based non-autoregressive generation
- Multimodal representation and generation
  - (text, image) joint representation
  - (text, image) → text generation
  - (text, image) → image generation
- Week 6 assignment

Week 6 Assignment

- Week 6 assignment deadline: 05 Dec 2024 (Thursday) CET
- Assignment PDF will be available online by tonight

- You will get to experiment with OpenAI's GPT-4o and DALL-E 3 models using APIs
  - OpenAI API uses API keys for authentication
  - https://platform.openai.com/docs/api-reference/introduction

- Your API key is available in the folder where you upload content
  - {id}_{name}/organization/{id}_{name}_openai.txt

- Please use the provided API key only for the assignment purposes
  - Expected cost for queries needed for this assignment is under 1 USD
  - Each API key has a hard budget limit beyond which queries will be blocked

- We will cover the expenses associated with your API key for the assignment ☺