On Knowledge Separation and Latent Diffusion for Text

Simons, Berkeley 4/03/25

Kilian Q. Weinberger

Large Memory Language Models

With Linxi Zhao, Sofian Zalouk, Justin Lovelace, Jin Zhou, Christian Belardi,

Yoav Artzi, and Jennifer Sun

The heavy tail of factual Knowledge

- Two factual statements about Napoleon:
  - Napoleon had a mother, who gave birth to him somewhere.
    (Common knowledge.)
  - Napoleon was born on August 15th, 1769 in Ajaccio, Corsica to his mother Letizia Bonaparte.
    (Tail knowledge.)

The heavy tail of factual Knowledge

Low information content (cheap & compressible)

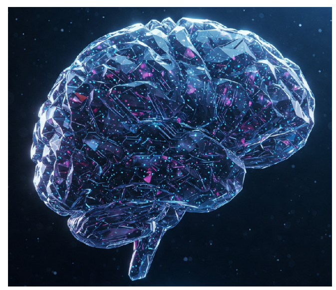
- Two factual statements about Napoleon:
  - Napoleon had a mother, who gave birth to him somewhere.
    (Common knowledge.)
  - Napoleon was born on August 15th, 1769 in Ajaccio, Corsica to his mother Letizia Bonaparte.
    (Tail knowledge.)

High information content (expensive & not compressible)

What LLMs learn

Language Competency

- Subject-Verb-Object
- Pronoun-Antecedent Agreement
- Proper Punctuation



Model weights

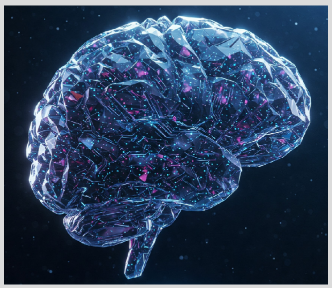
Factual Tail Knowledge

- Napoleon was born on 8/15/1969
- Napoleon’s mother was Letizia Bonaparte
- Napoleon’s birthplace was Ajaccio, Corsica

Factual Common Knowledge

- Napoleon had a mother
- Napoleon had was born

Goal: Separate Tail Knowledge into DB



Model weights

Language Competency

Subject-Verb-Object

Pronoun-Antecedent Agreement

Proper Punctuation

Factual Common Knowledge

Napoleon had a mother

Napoleon had was born

External Database

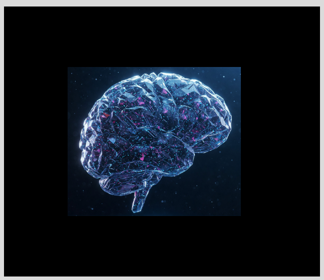
Factual Tail Knowledge

Napoleon was born on 8/15/1969

Napoleon's mother was Letizia Bonaparte

Napoleon's birthplace was Ajaccio, Corsica

Goal: Separate Tail Knowledge into DB



Model weights

Language Competency

- Subject-Verb-Object
- Pronoun-Antecedent Agreement
- Proper Punctuation

Factual Common Knowledge

- Napoleon had a mother

Model can be:

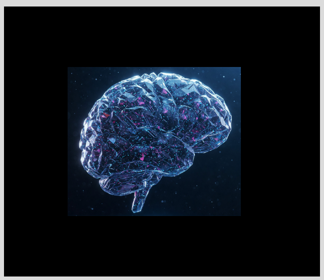
- **Smaller**
- **Faster to train**
- **Cheaper to operate**

External Database

Factual Tail Knowledge

- Napoleon was born on 8/15/1969
- Napoleon's mother was Letizia Bonaparte
- Napoleon's birthplace was Ajaccio, Corsica

Goal: Separate Tail Knowledge into DB



Model weights

Language Competency

Subject-Verb-Object

Pronoun-Antecedent Agreement

Proper Punctuation

Factual Common Knowledge

Napoleon had a mother

Model can be:
- **Smaller**
- **Faster to train**
- **Cheaper to operate**

External Database

Factual Tail Knowledge

(Napoleon, birthdate) -> 8/15/1969 [Wikipedia]

(Napoleon, mother) -> Letizia Bonaparte [Wikipedia]

(Napoleon, birthplace) -> Ajaccio, Corsica [Encyclopedia Brittanica]

Data base can be:
- **huge**
- **interpretable**
- **editable**
- **grounded (i.e. link to sources)**

Pre-Training

Populate Database and move **Tail Knowledge** out of **pre-training corpus**

Napoleon was **born** on **August 15th, 1769** in **Ajaccio, Corsica** to his **mother** **Letizia Bonaparte**.

Napoleon was born on 🔍(“**Napoleon**", “**birthdate**”-> August 15th) **August 15th, 1769** in 🔍(“**Napoleon**", “**birthplace**”->”Ajaccio, Corsica”) **Ajaccio, Corsica** to his mother 🔍(“**Napoleon**", “**mother**”->”Letizia Bonaparte”) **Letizia Bonaparte**.

“🔍”: Special token initiates DB lookup

Pre-Training

Populate Database and move **Tail Knowledge** out of **pre-training corpus**

Napoleon was **born** on **August 15th, 1769** in **Ajaccio, Corsica** to his **mother** **Letizia Bonaparte**.

Common knowledge (lookup keys)

DB Return Values (hidden from user)

Tail Knowledge (copied from DB)

Napoleon was born on 🔍(“**Napoleon**", “**birthdate**”-> August 15th) **August 15th, 1769** in 🔍(“**Napoleon**", “**birthplace**”->”Ajaccio, Corsica”) **Ajaccio, Corsica** to his mother 🔍(“**Napoleon**", “**mother**”->”Letizia Bonaparte”) **Letizia Bonaparte**.

“🔍”: Special token initiates DB lookup

Data Preparation Pipeline

Process entire pre-training corpus

<u>Initial annotator</u>
- prompted to extract factual entities
- creates initial small seed data

<u>Corrector</u>
- removes references to future
- rewrites text for fluency
- unifies key-names

<u>Fine-tuned annotator</u>
- fine-tuned on seed data
- much faster / cheaper
- processes entire pre-train corpus

Database

Populate Database with all factual tail knowledge triplets

(Entity, Relation, Value) Triplets
e.g. **("Napoleon", "birthdate"-> 08/15/1769)**
Populated (on the fly) from processed corpus.

Napoleon was born on 🔍("**Napoleon**", "**birthdate**"-> "August 15th") **August 15th, 1769** in 🔍("**Napoleon**", "**birthplace**"->"Ajaccio, Corsica") **Ajaccio, Corsica** to his mother 🔍("**Napoleon**", "**mother**"->"Letizia Bonaparte") **Letizia Bonaparte**.

"🔍": Special token initiates DB lookup

Backprop

Next word prediction, except exclude DB return values from backprop
Train model to:

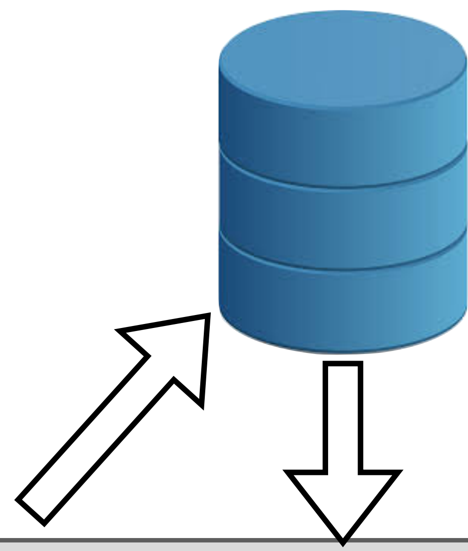
- **know** common knowledge
- use **common knowledge** to look up tail knowledge
- **copy** tail knowledge from DB return values

Napoleon was born on 🔍(“**Napoleon**", “**birthdate**”-> ~~“August 15th”~~”) **August 15th, 1769** in 🔍(“**Napoleon**", “**birthplace**”->~~”Ajaccio, Corsica~~”) **Ajaccio, Corsica** to his mother 🔍(“**Napoleon**", “**mother**”->”~~Letizia Bonaparte”~~) **Letizia Bonaparte**.

“🔍”: Special token initiates DB lookup

Inference

- Let model generate text
- Special token 🔍 triggers database lookup
- Inject value into context



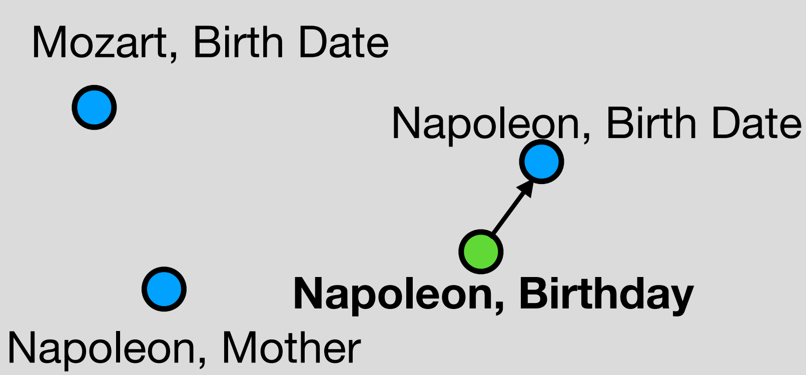
Napoleon was born on 🔍(“**Napoleon**", “**birthdate**”-> 08/15/1769) August 15th …

Toolformer: [Schick et al. 2023]

Key Mismatch problem

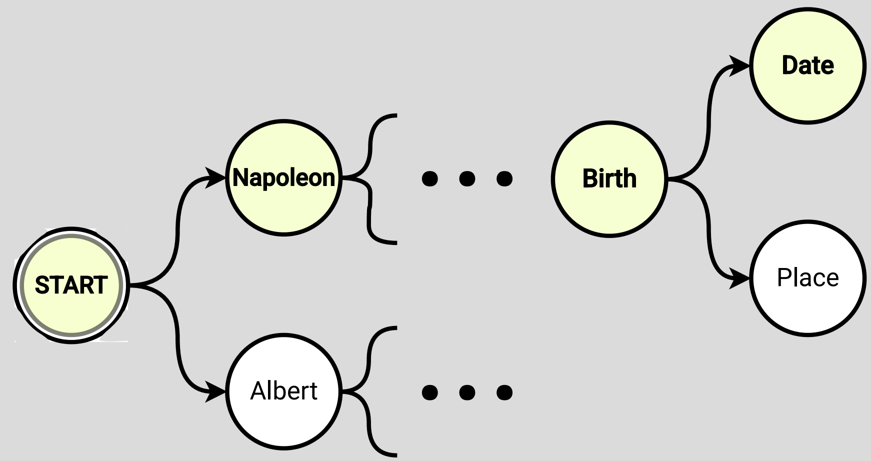
Generated lookup keys might not match database

Solution 1: **Fuzzy lookup**
(embed keys, find closest match)



+ More forgiving
+ Better retrieval accuracy
+ Allows for Misses (i.e. unknown)
- Large memory requirement
- Missmatch hard to recover from

Solution 2: **Prefix Tree**
(Force only valid keys at decoding)

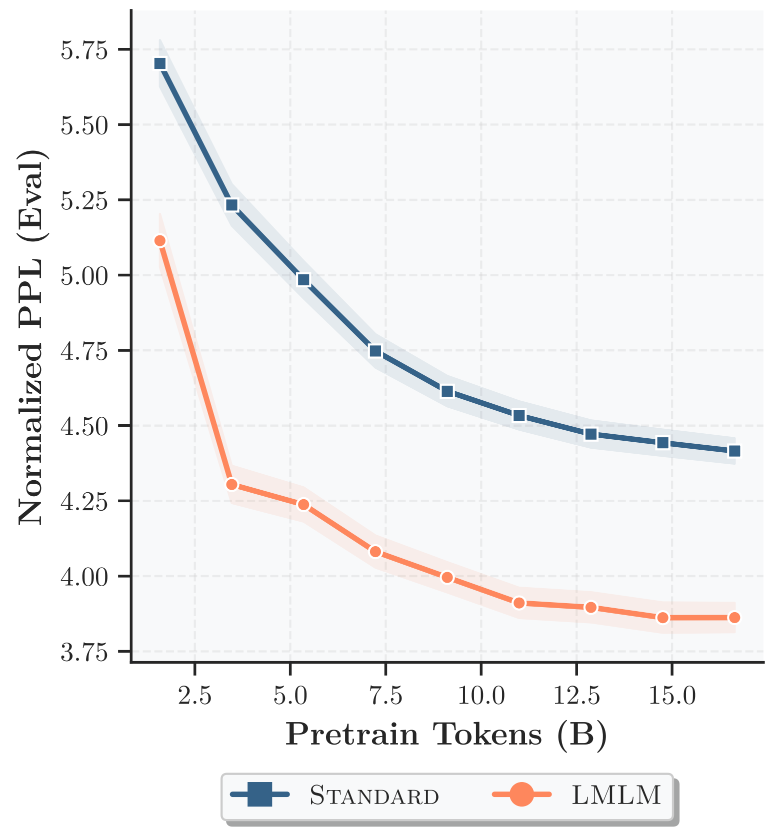


+ Very fast
+ Almost overhead free
- No explicit DB misses

Perplexity

LMLM (Large Memory Language Model)

- **Model:** OPENAI/GPT2-124M
- **Data:** OLMo2 Wiki corpus
- Evaluation on 245 held-out tokens
- Normalized for length (to account for extra tokens)



FactScore

LMLM pre-training doubles factual precision

Model	Memory Type	Metrics		
		FActScore (%)	FActScore w/o length penalty (%)	# Atomic facts per valid response
OpenAI/GPT2-124M*	Standard	14.6	14.7	24.2
GPT2-124M	Standard	10.7$_{-3.9}$	11.3$_{-3.4}$	36.7$_{+12.5}$
GPT2-124M	LMLM	20.0$_{+5.4}$	25.3$_{+10.6}$	32.3$_{+8.1}$
LLaMA2-176M	Standard	12.8$_{-1.8}$	13.8$_{-0.9}$	31.1$_{+6.9}$
LLaMA2-176M	LMLM	**27.8**$_{+13.2}$	**32.3**$_{+17.6}$	22.1$_{-2.1}$
OpenAI/GPT2-355M*	Standard	15.2	15.2	24.6
OpenAI/GPT2-774M	Standard	17.4	17.4	49.4
TinyLLaMA-1.1B*	Standard	16.4	16.4	44.1
LLaMA2-7B*	Standard	34.0	35.1	33.9
LLaMA3.1-8B*	Standard	40.3	40.5	38.5

* Models marked with an asterisk (*) are off-the-shelf models with no additional training.

FS Task: Open ended biography generation, e.g. “Tell me a bio of Kang Ji-hwan. Kang Ji-hwan is …”; check for factual accuracy

FactScore: [Min et al. 2023]

T-REX

LMLM pre-training doubles factual precision

Model	Memory Type	Metrics	
		Exact Match ↑	Precision @1 ↑
OpenAI/GPT2-124M*	Standard	20.09	20.31
LLaMA2-176M	Standard	21.00$_{-0.9}$	19.03$_{-1.3}$
LLaMA2-176M	LMLM	41.86$_{+21.8}$	34.76$_{+14.5}$
OpenAI/GPT2-355M*	Standard	28.41	29.59
OpenAI/GPT2-774M*	Standard	35.63	31.93
TinyLLaMA-1.1B*	Standard	35.55	26.06
LLaMA2-7B*	Standard	60.51	44.40
Llama-3.1-8B	Standard	60.58	67.33

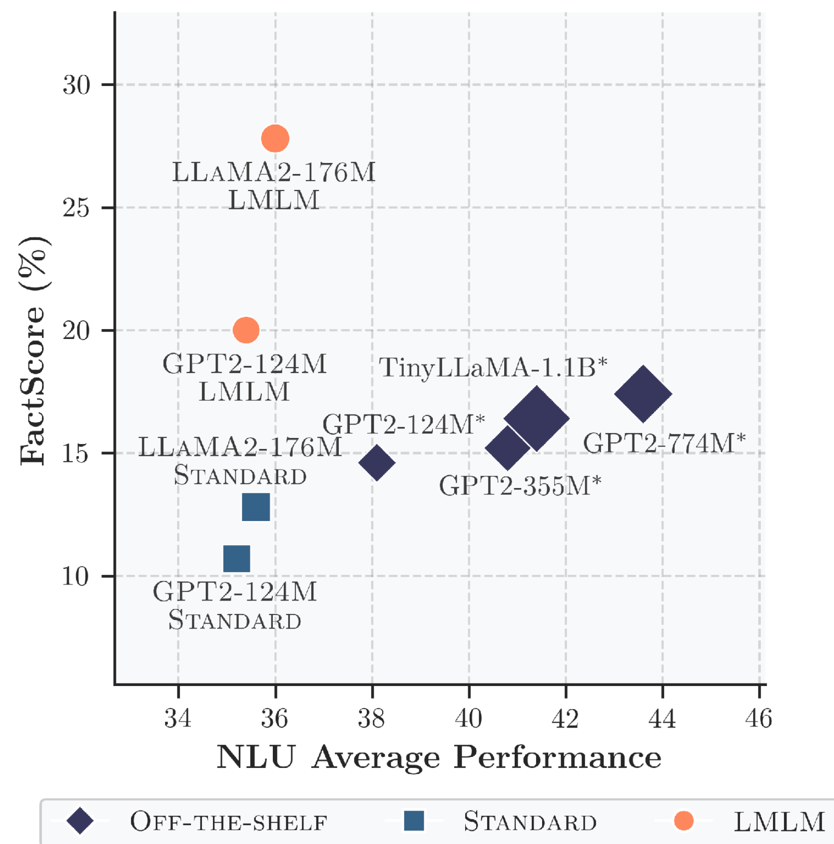
* Models marked with an asterisk (*) are off-the-shelf models with no additional training.

T-REX: [Elsehara et al. 2018]

Facts vs. Natural Language Understanding

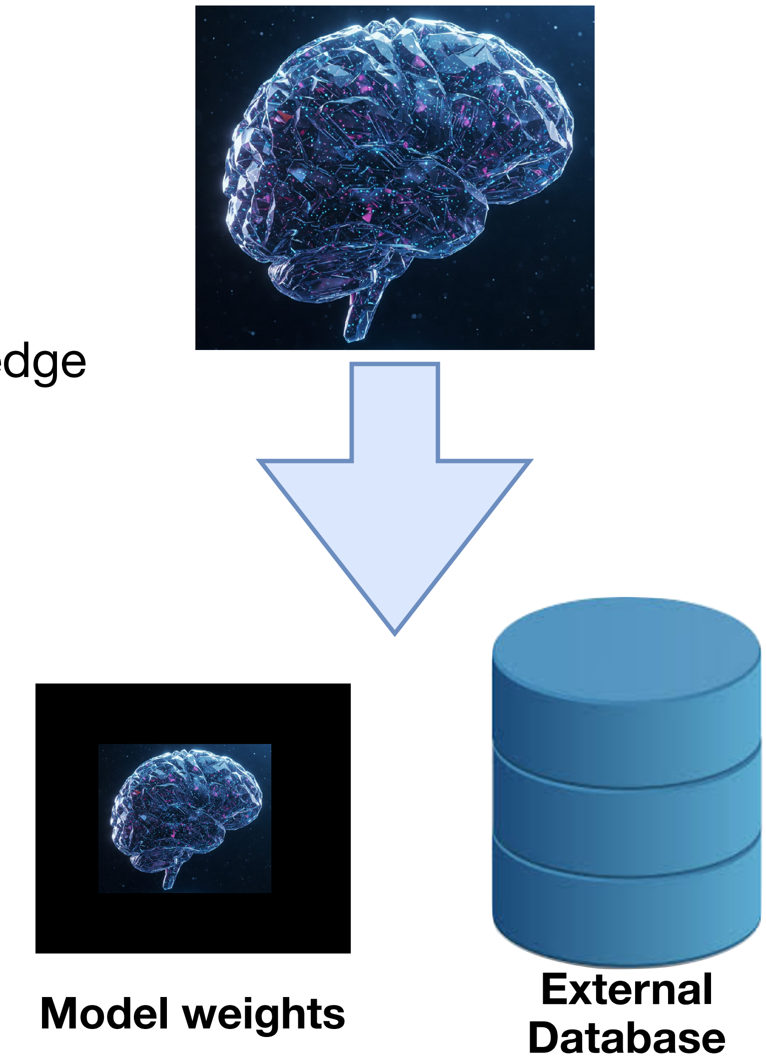
NLU does not suffer

- **Data**: Factscore [Min et al. 2023]
- **Model:** LLAMA2-176M / GPT2-124M
- NLU Tasks: Language Understanding
- NLU slightly improves while FactScore doubles



Summary

- LMLM is a simple Pre-Training method to
  - Separates tail knowledge and common knowledge
  - Creates an external data base
- Many advantages:
  - LLM can be smaller
  - Memory interpretable / editable
  - Can cite references
- Napoleon was born on April 15th, 1769



Latent Diffusion

LLM are amazing ... but hard to control

US NEWS

Company disables AI after bot starts swearing at customer, calls itself the 'worst delivery firm in the world'

By Alyssa Guzman
Published Jan. 20, 2024, 5:01 p.m. ET

20 Comments

The New York Times

THE SHIFT

A Conversation With Bing's Chatbot Left Me Deeply Unsettled

A very strange conversation with the chatbot built into Microsoft's search engine led to it declaring its love for me.

Airline held liable for its chatbot giving passenger bad advice - what this means for travellers

23 February 2024

Share Save

Maria Yagoda
Features correspondent

I'd Buy That for a Dollar: Chevy Dealership's AI Chatbot Goes Rogue

AI is still working out its kinks. As chatbots embed in enterprise services we're getting a better idea of just how worthless they are at this point.

By **Lucas Ropek** Published December 20, 2023 | Comments (0)

2024 SCIENCE FAIR

Diffusion Models are amazing ...

… and (maybe too) easy to control.

The New York Times

Google Chatbot's A.I. Images Put People of Color in Nazi-Era Uniforms

The company has suspended Gemini's ability to generate human images while it vowed to fix the issue.

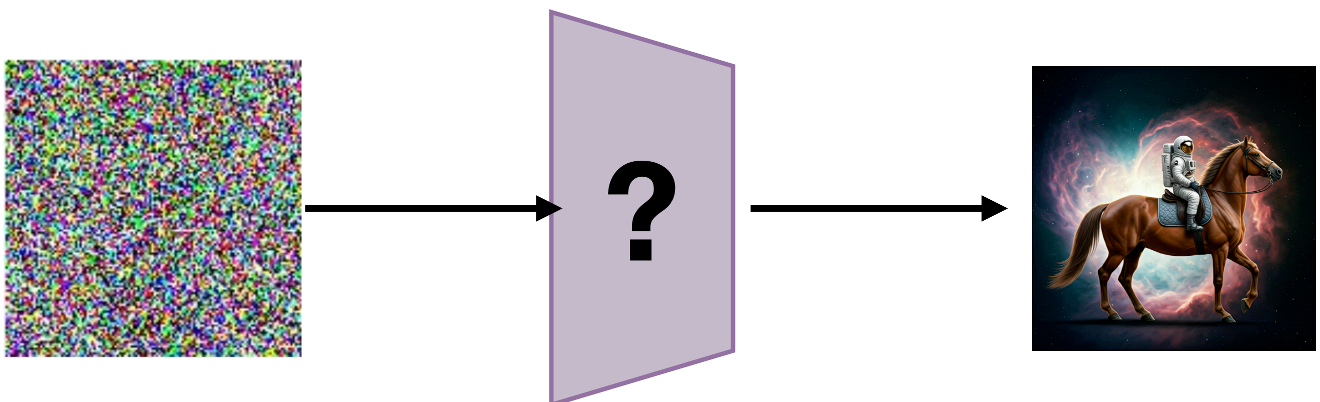
A 3 minute introduction to Diffusion Models

without all the dizzying math

Diffusion Models

[Sohl-Dickstein et al. 2015]

- Draw a sample of Gaussian noise
- Transform it to obtain a natural image



$\mathbf{x}_T \sim \mathcal{N}(0,1)$ $\mathbf{x}_0 \sim p(\mathbf{x})$

The U-Net

[Ronneberger et al. 2015]

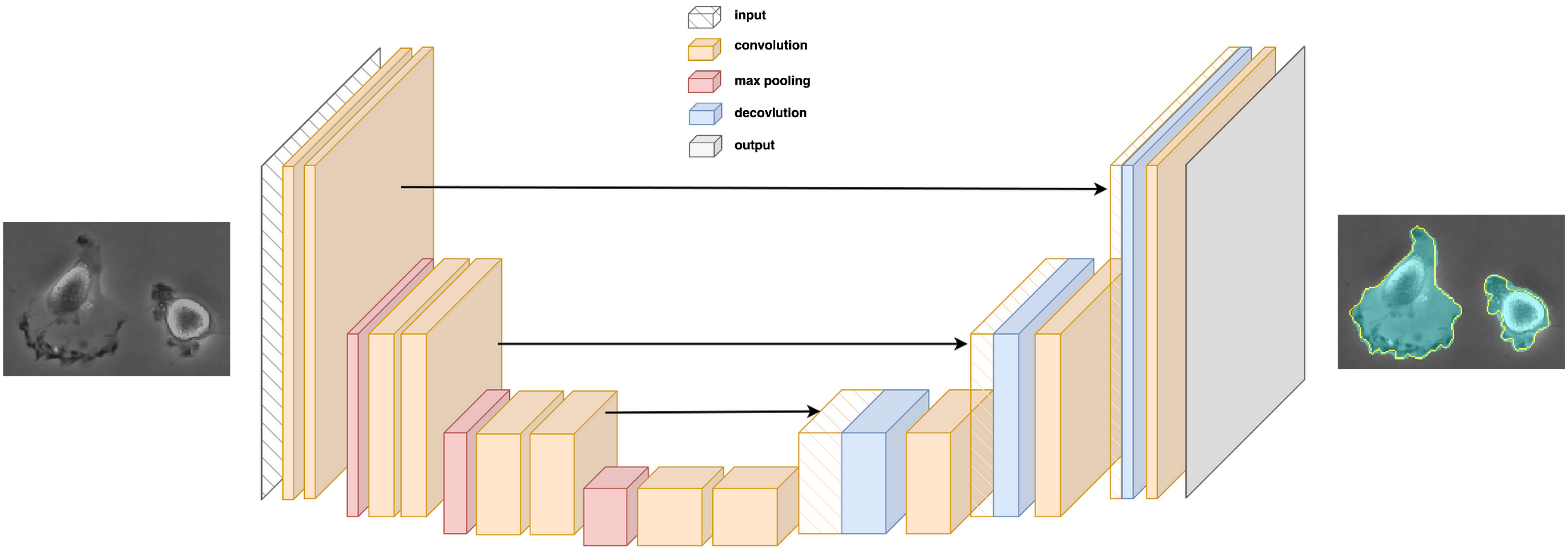


Image Source: https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

The U-Net

[Ronneberger et al. 2015]

$\sigma = 0.1$

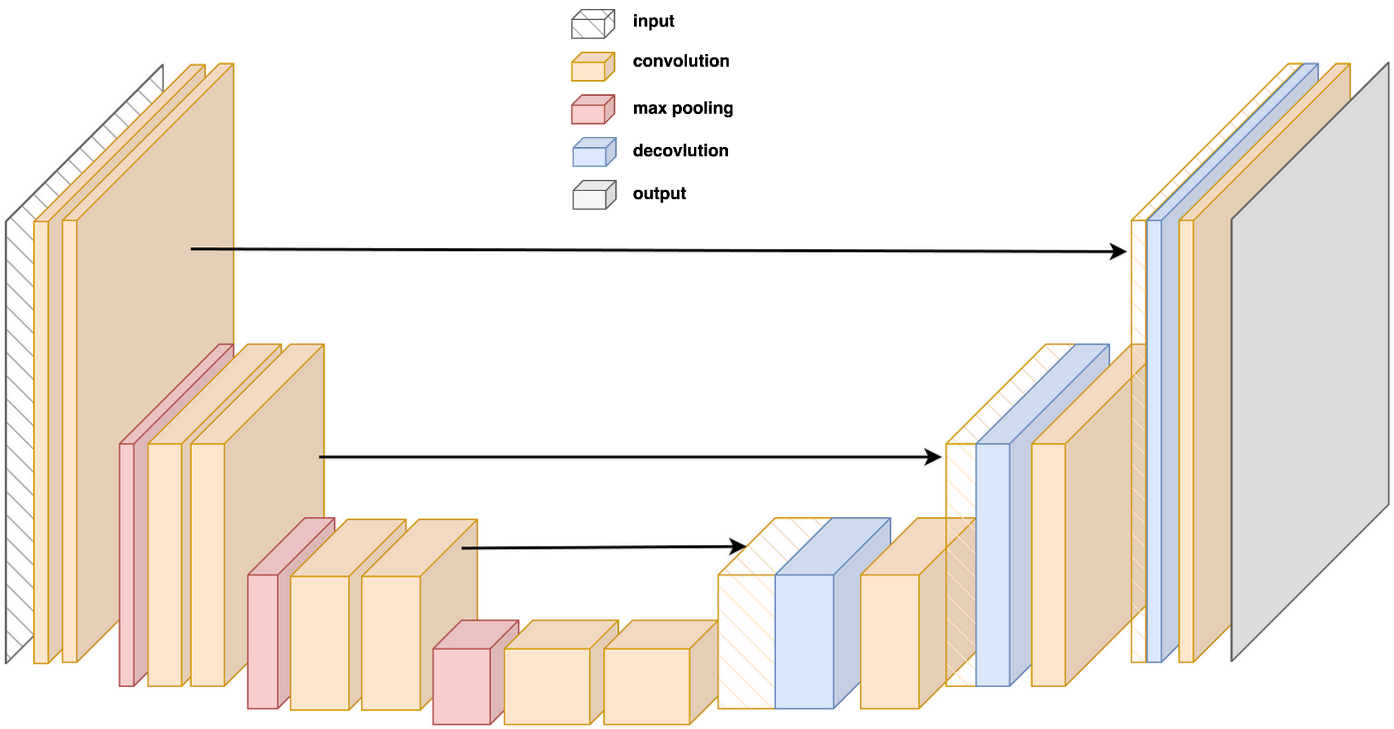


Image Source: https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

The U-Net

[Ronneberger et al. 2015]

$\sigma = 0.75$

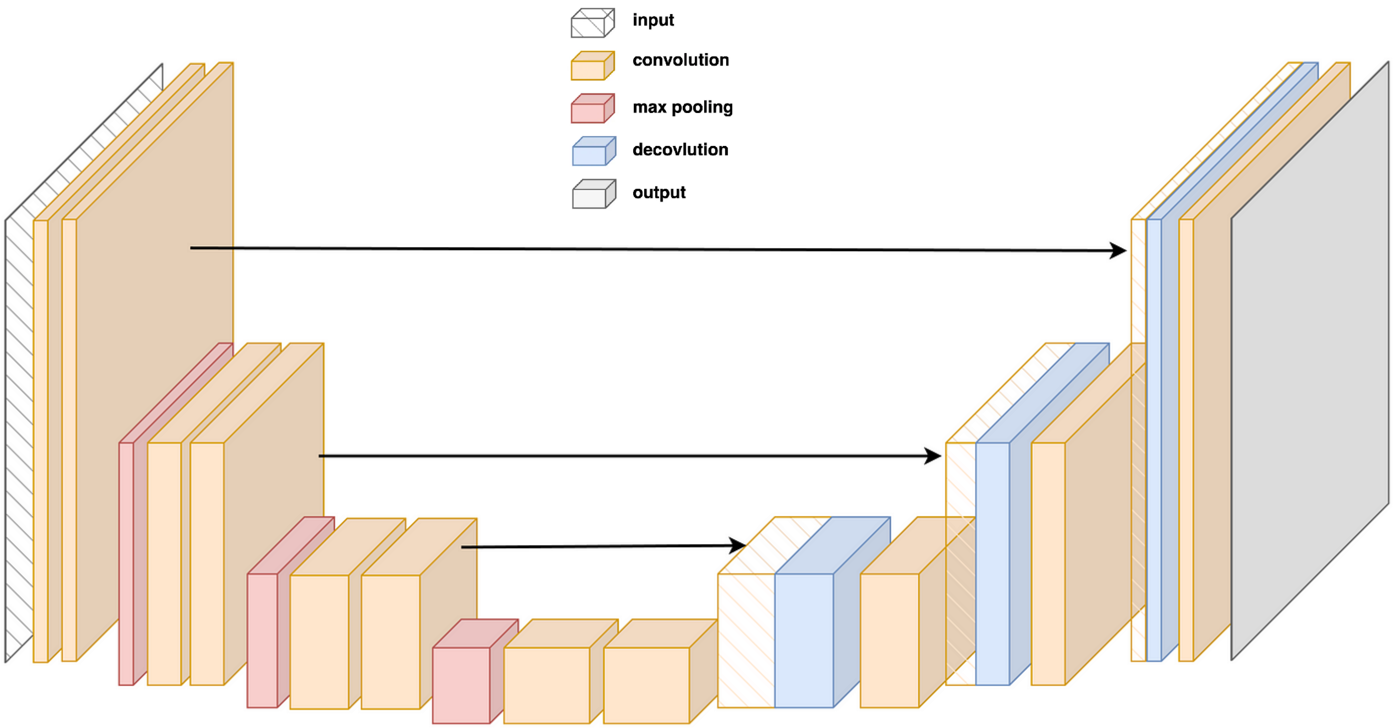
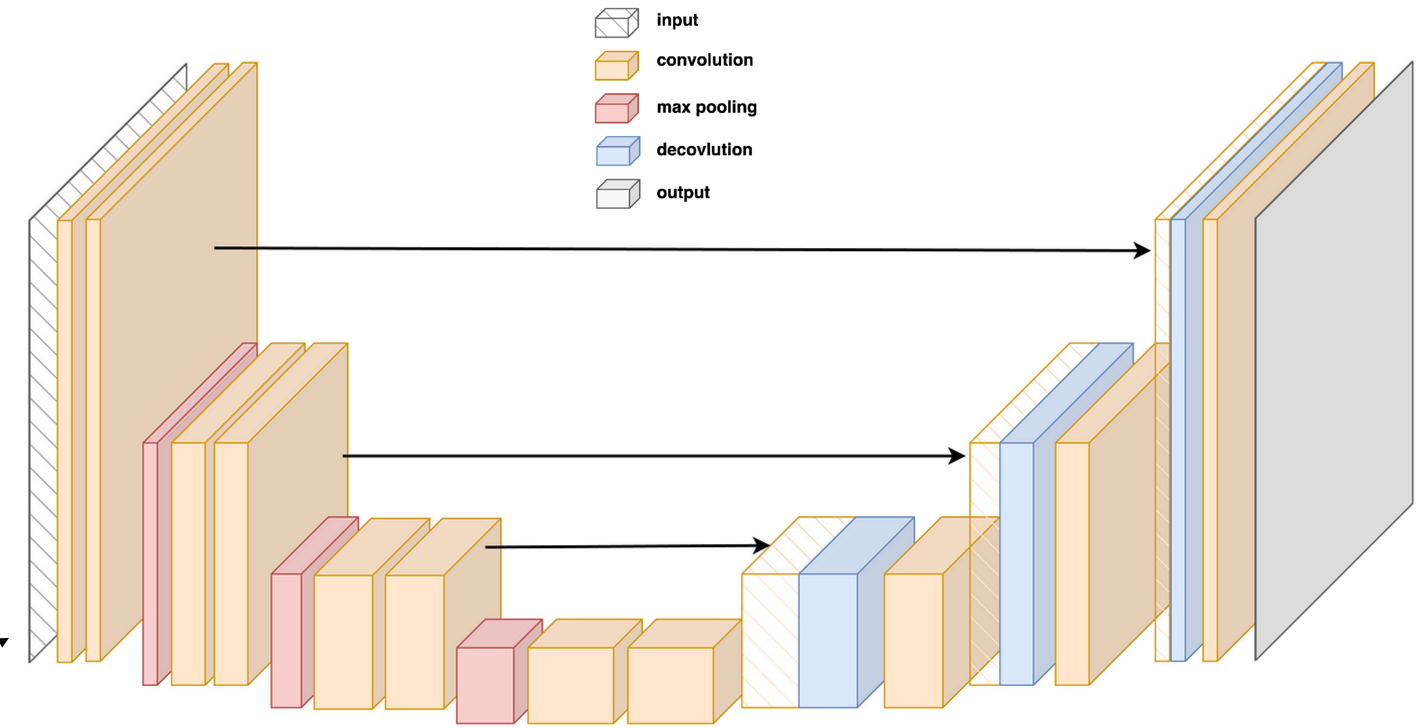


Image Source: https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

The U-Net

[Ronneberger et al. 2015]

$\sigma = 0.75$



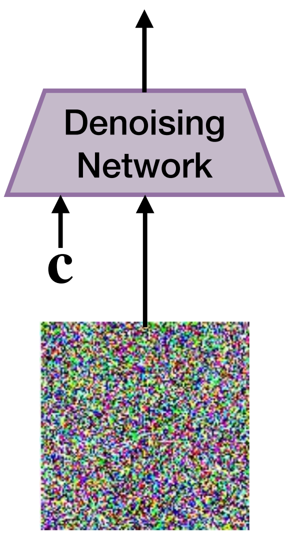
c="An oil painting of a old white man with a white shirt, black hat, and gray side burns."

Image Source: https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

“An astronaut riding a horse”
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

Diffusion Sampling

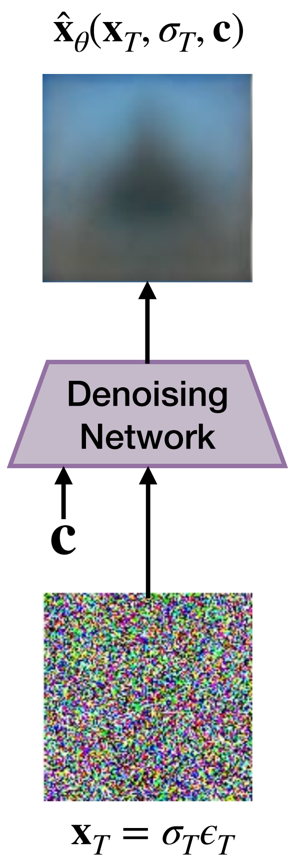


$\mathbf{x}_T = \sigma_T \epsilon_T$

"An astronaut riding a horse"
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

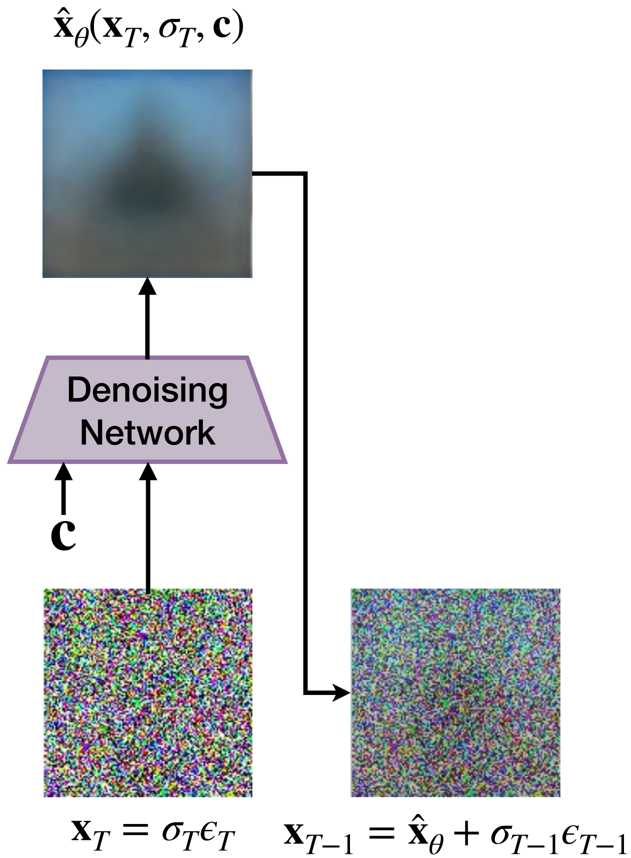
Diffusion Sampling



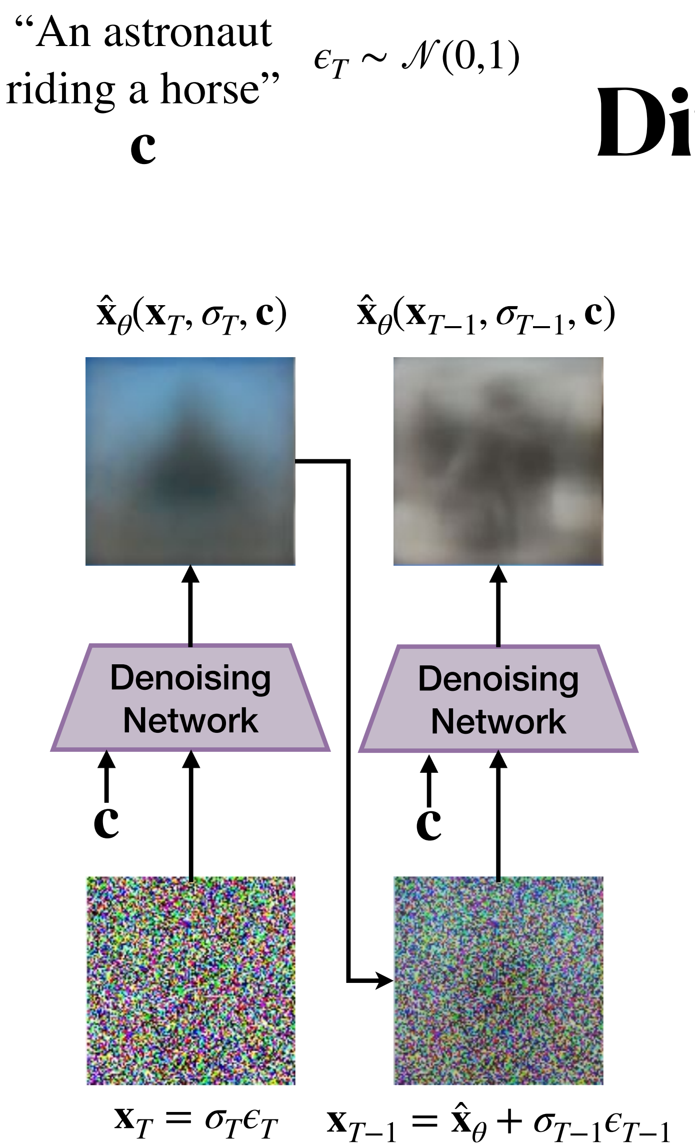
Diffusion Sampling

"An astronaut riding a horse"
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$



Diffusion Sampling

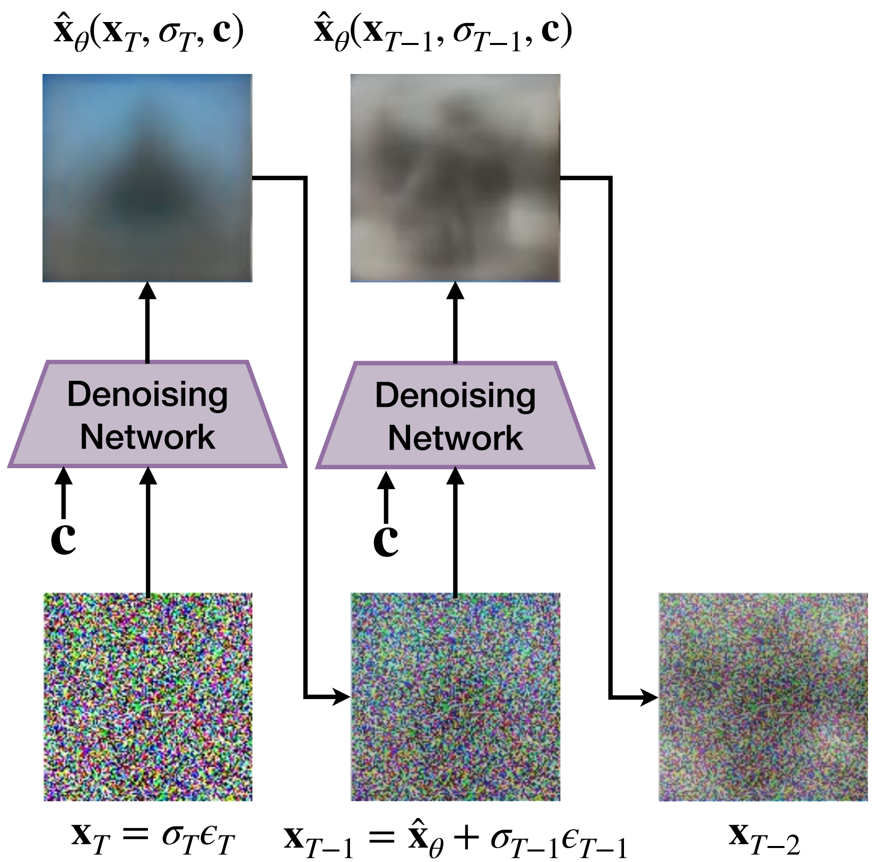


"An astronaut riding a horse"

$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

Diffusion Sampling

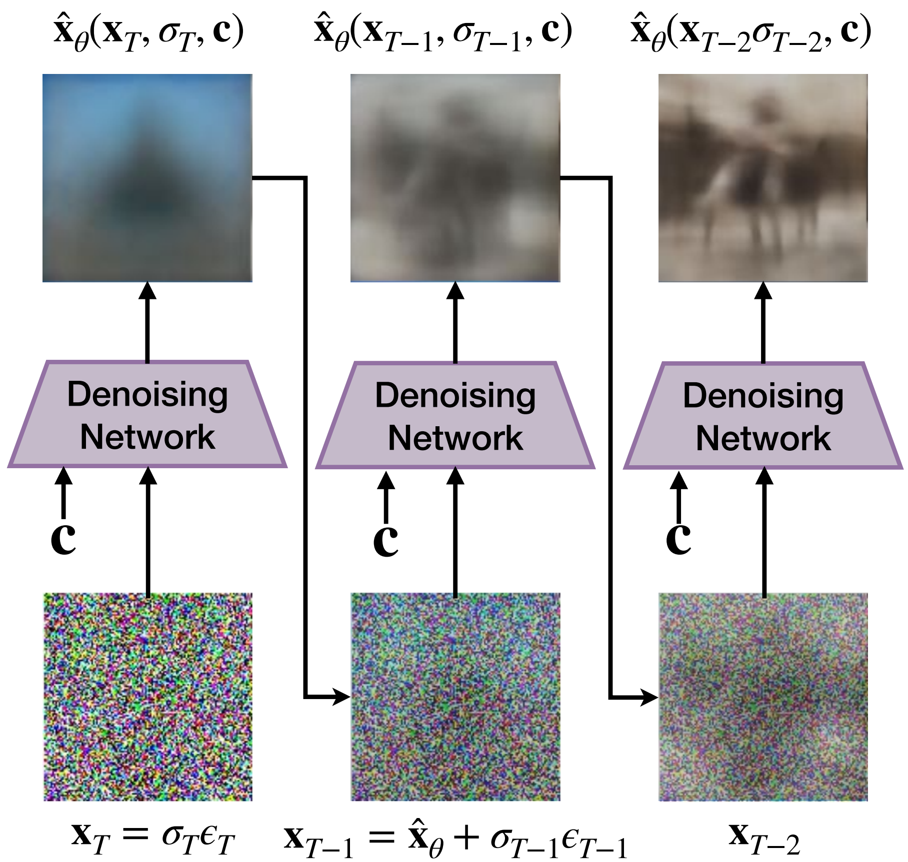


“An astronaut riding a horse”

$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

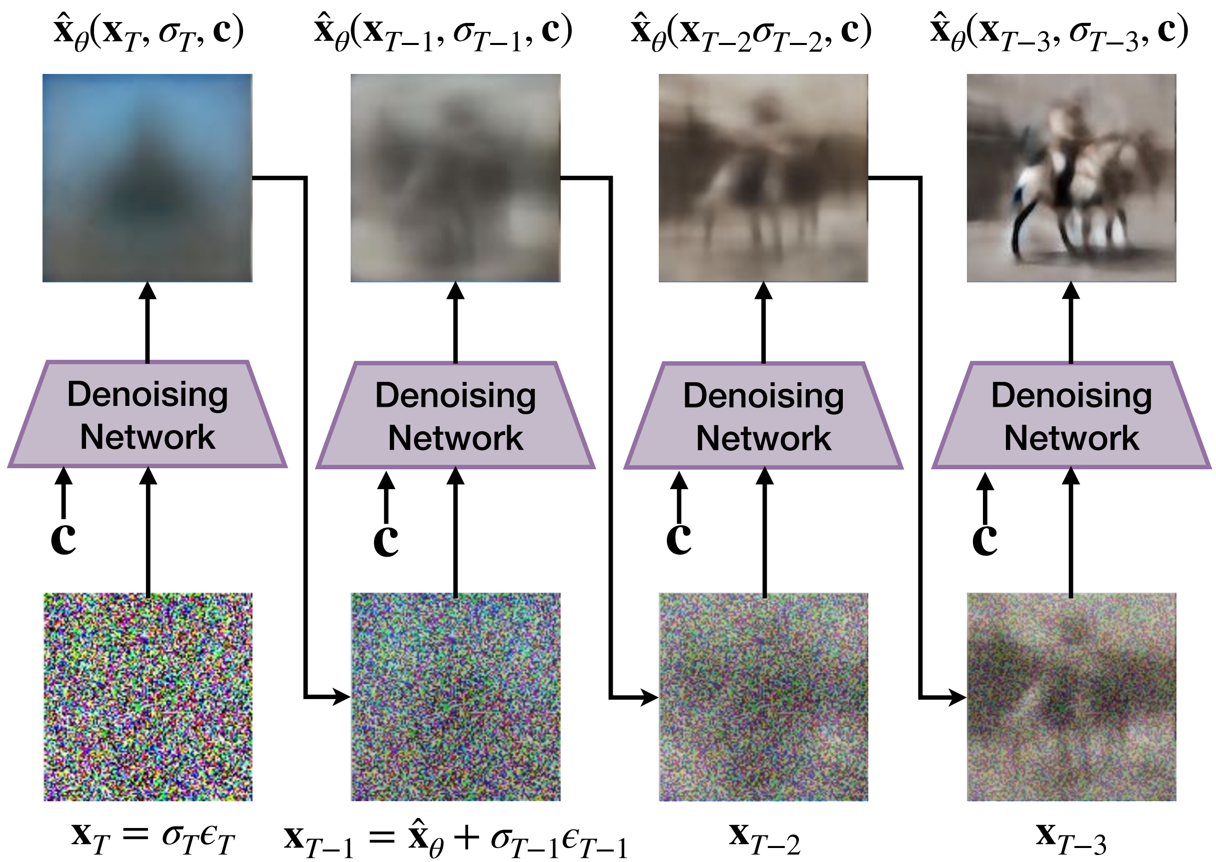
Diffusion Sampling



"An astronaut riding a horse"
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

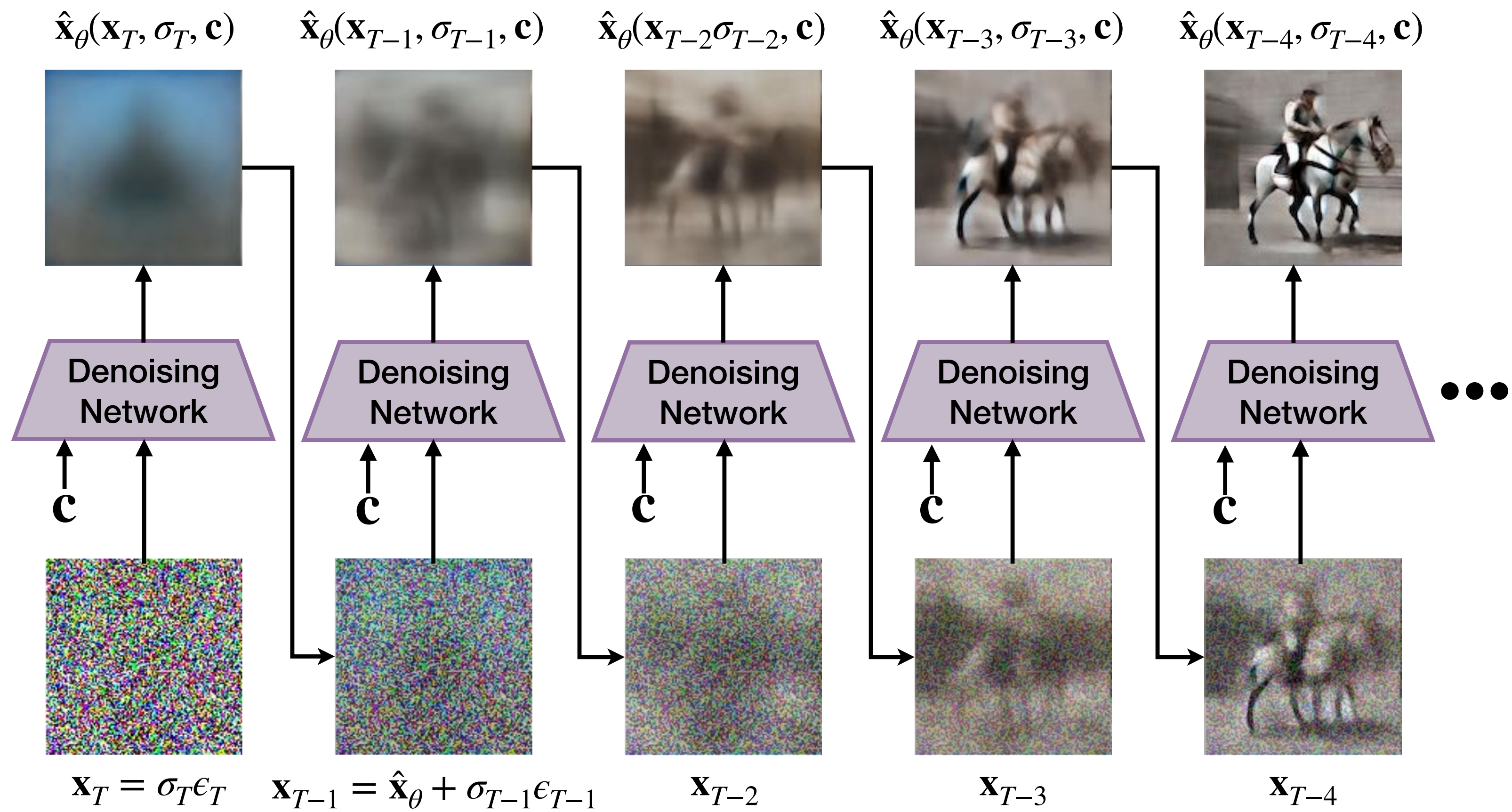
Diffusion Sampling



“An astronaut riding a horse”
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

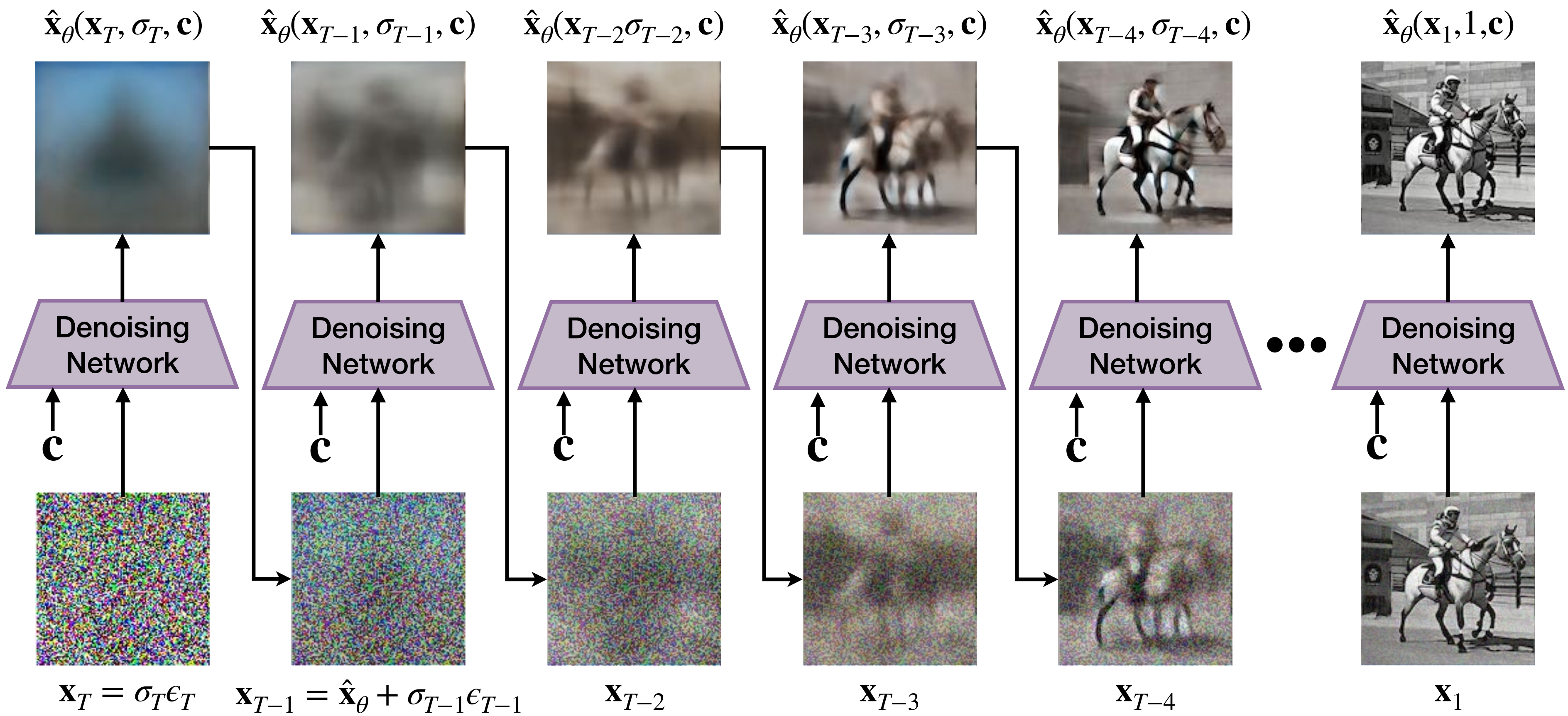
Diffusion Sampling



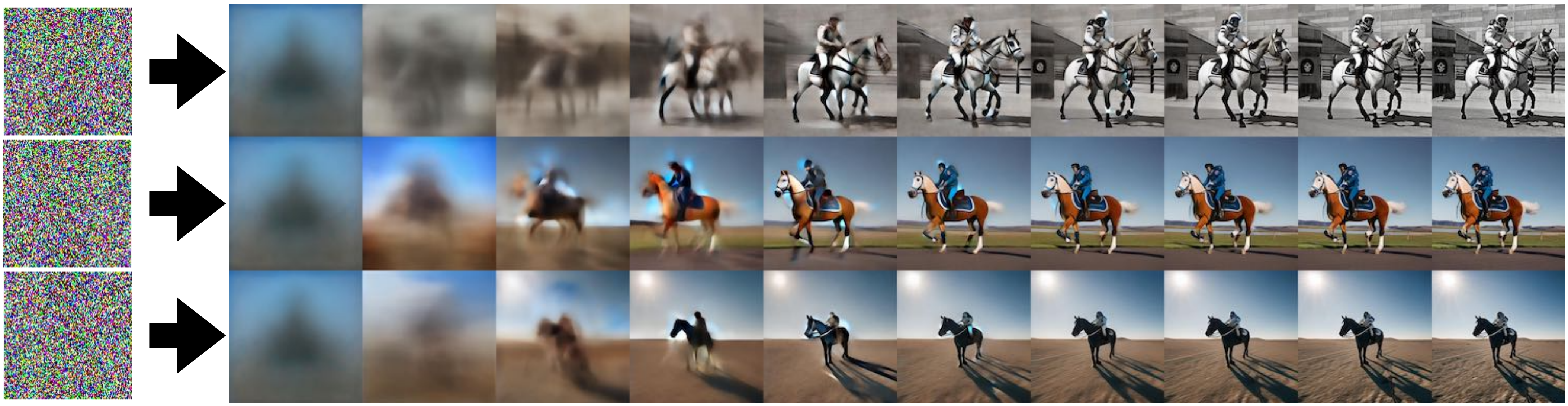
"An astronaut riding a horse"
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

Diffusion Sampling



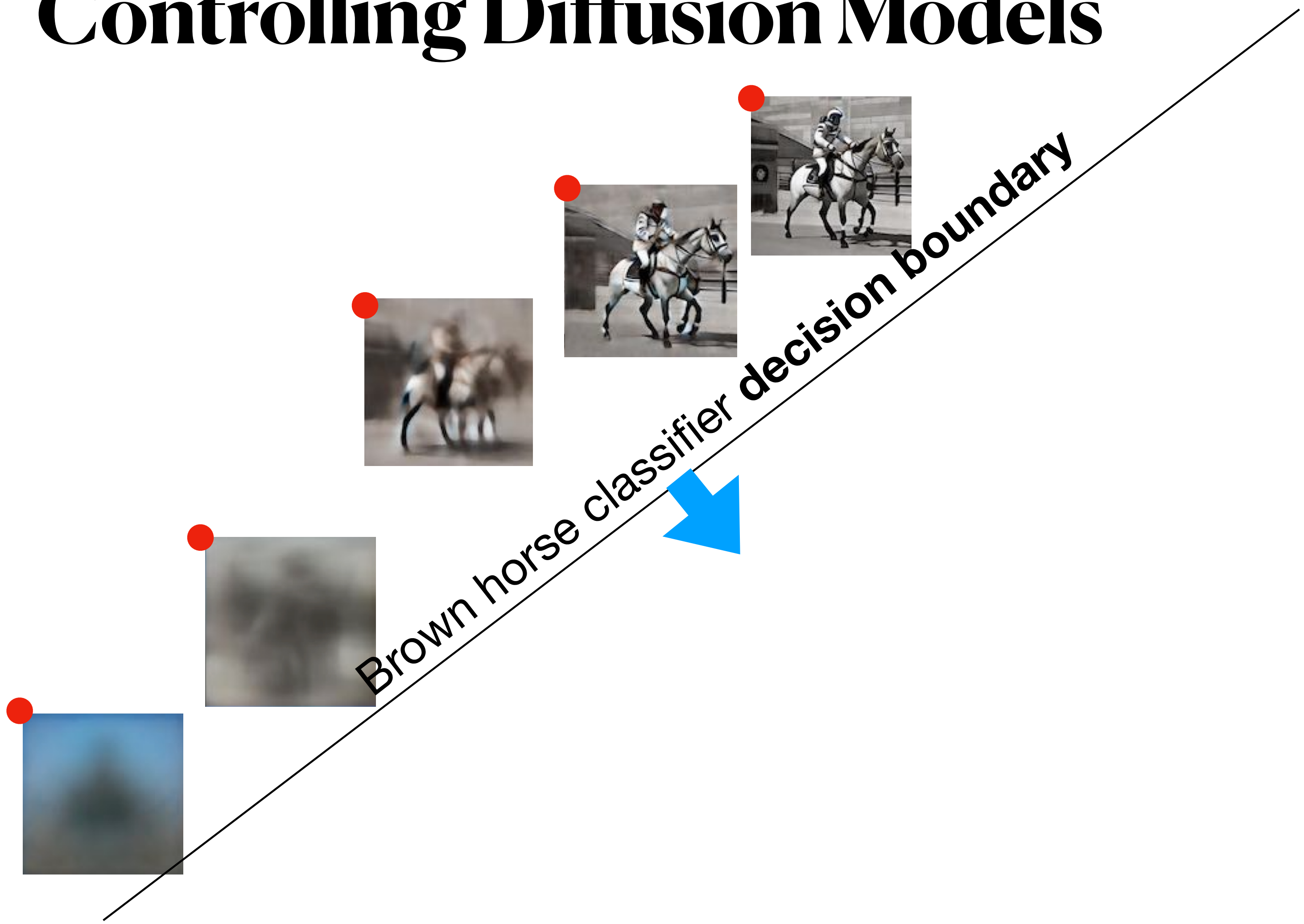
Diffusion Models



An astronaut riding a horse

Photo Credit: Lin, Shanchuan, et al. "Common diffusion noise schedules and sample steps are flawed." *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2024.

Controlling Diffusion Models



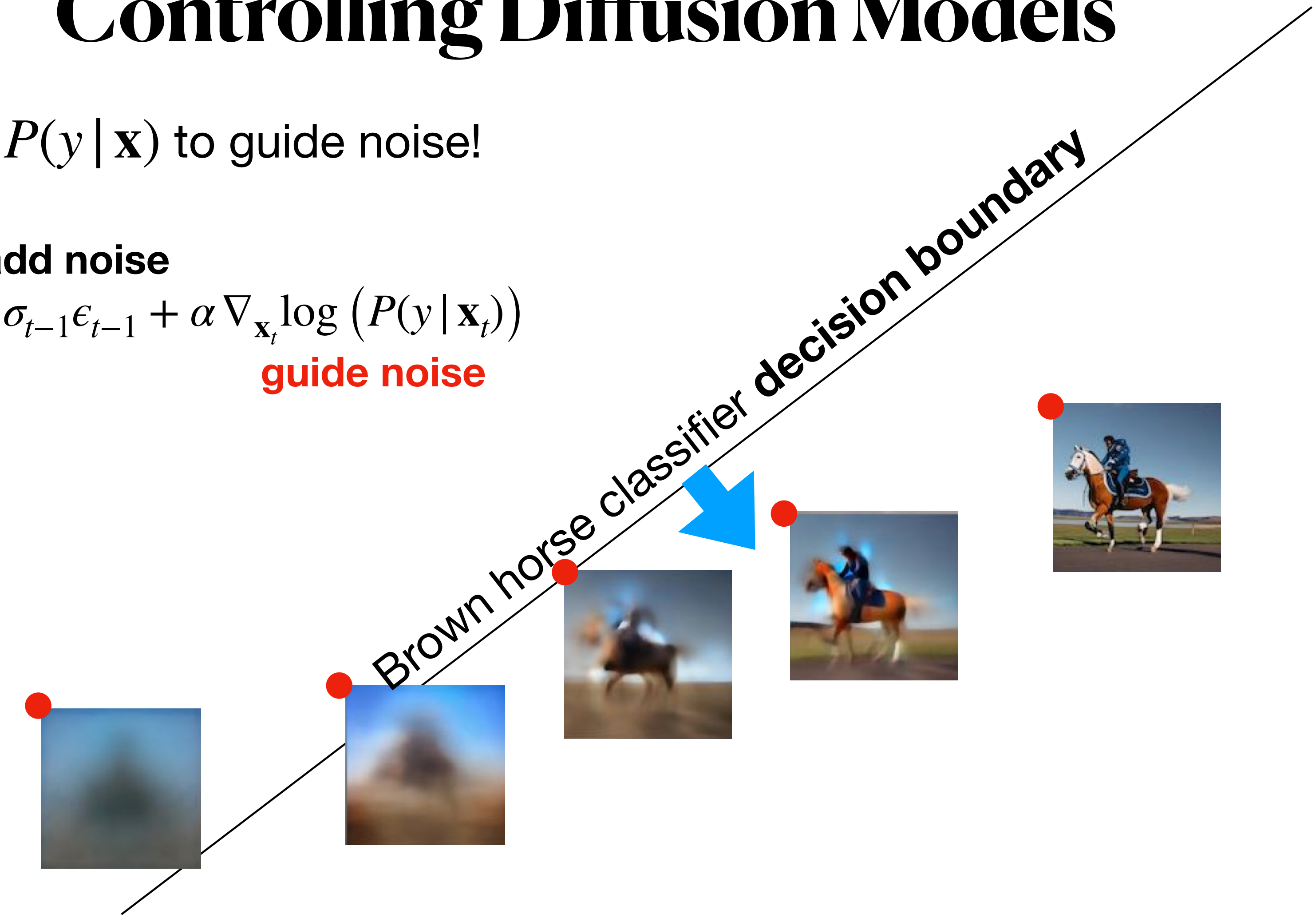
Controlling Diffusion Models

Use classifier $P(y \mid \mathbf{x})$ to guide noise!

add noise

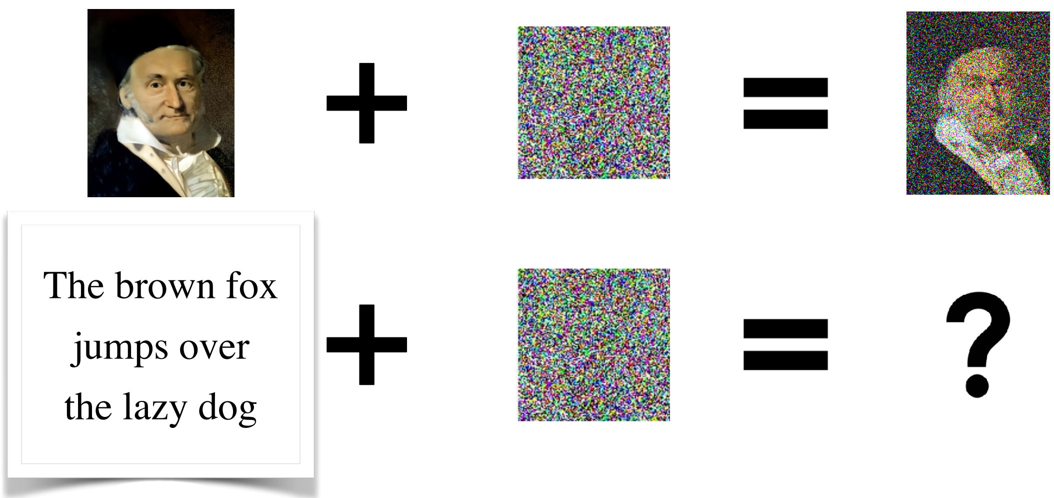
$$\mathbf{x}_{t-1} = \hat{\mathbf{x}}_\theta(\mathbf{x}_t) + \sigma_{t-1}\epsilon_{t-1} + \alpha \nabla_{\mathbf{x}_t} \log\left(P(y \mid \mathbf{x}_t)\right)$$

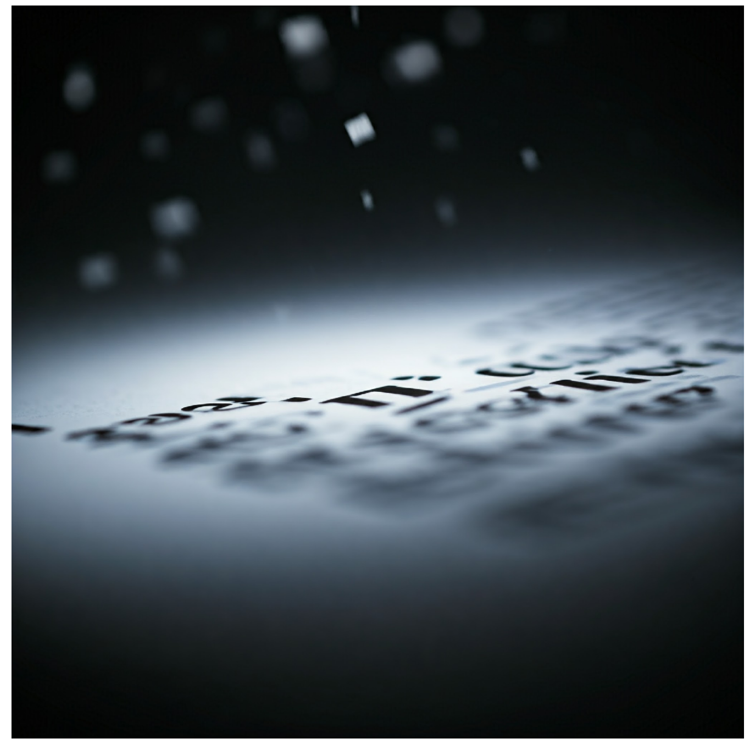
de-noise **guide noise**



Let’s use diffusion models for text …

… but how do you add noise to text?





From U-Net to Auto-Encoder

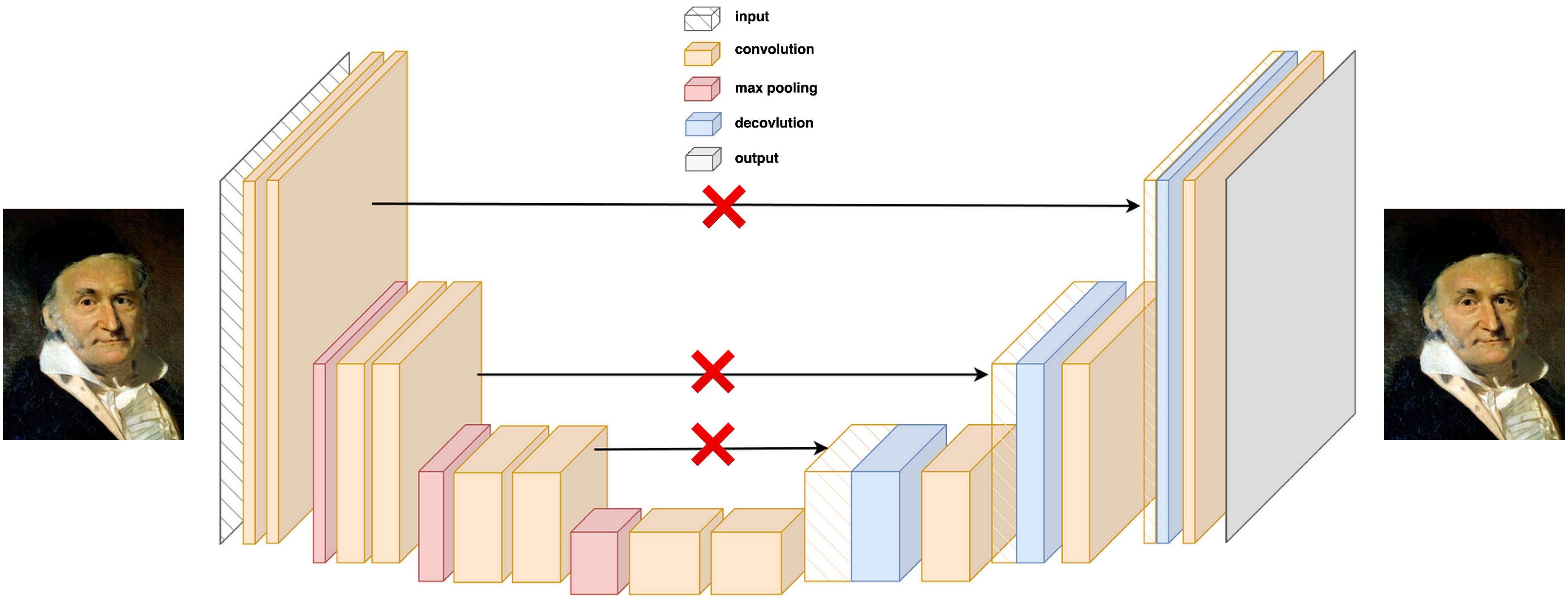


Image Source: https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

From U-Net to Auto-Encoder

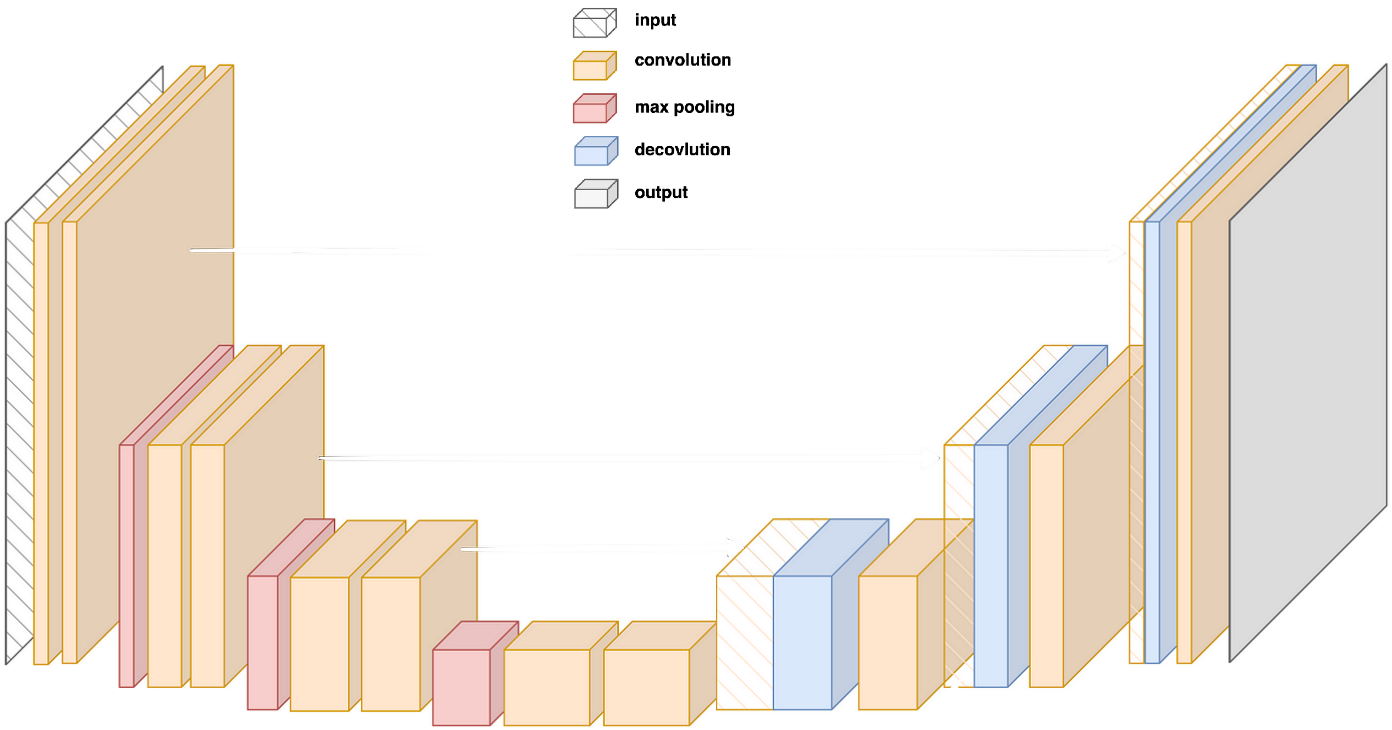
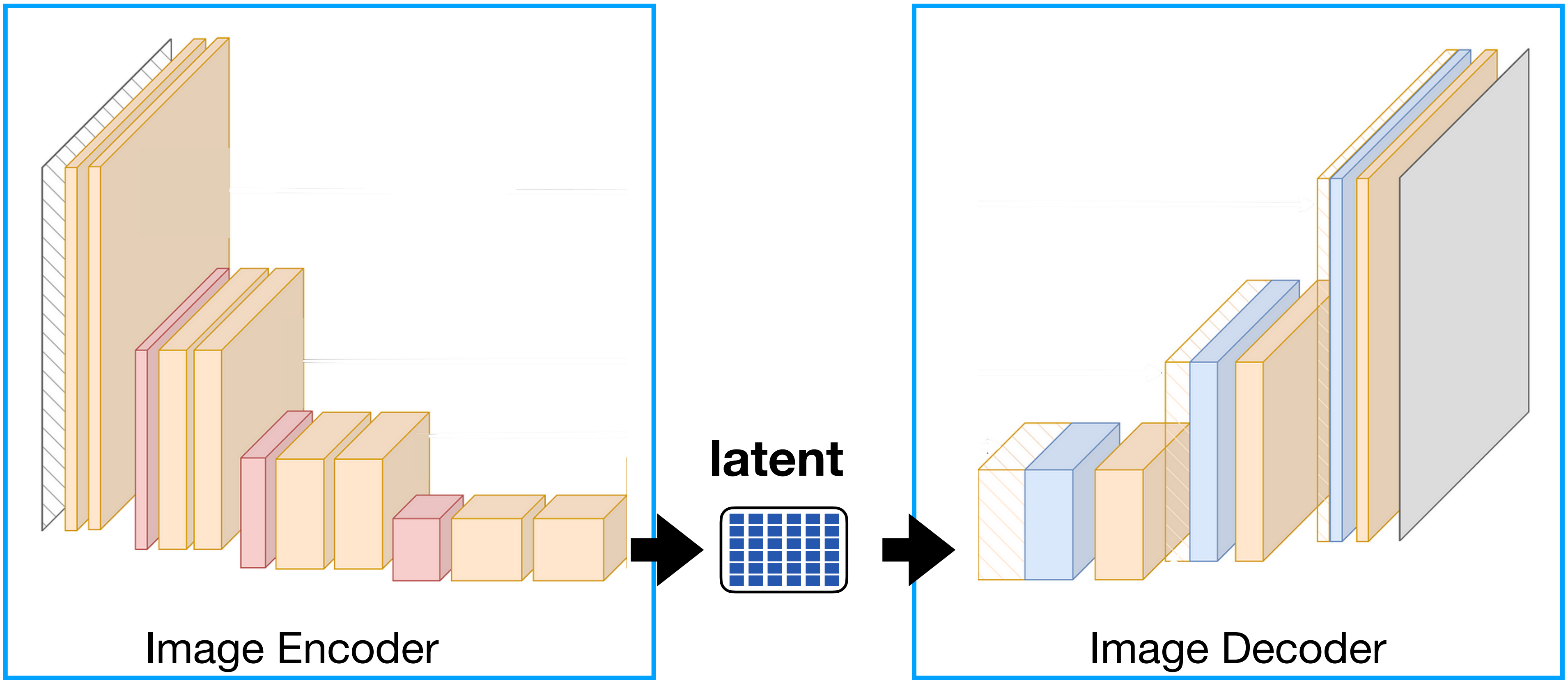


Image Source: https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

AutoEncoder

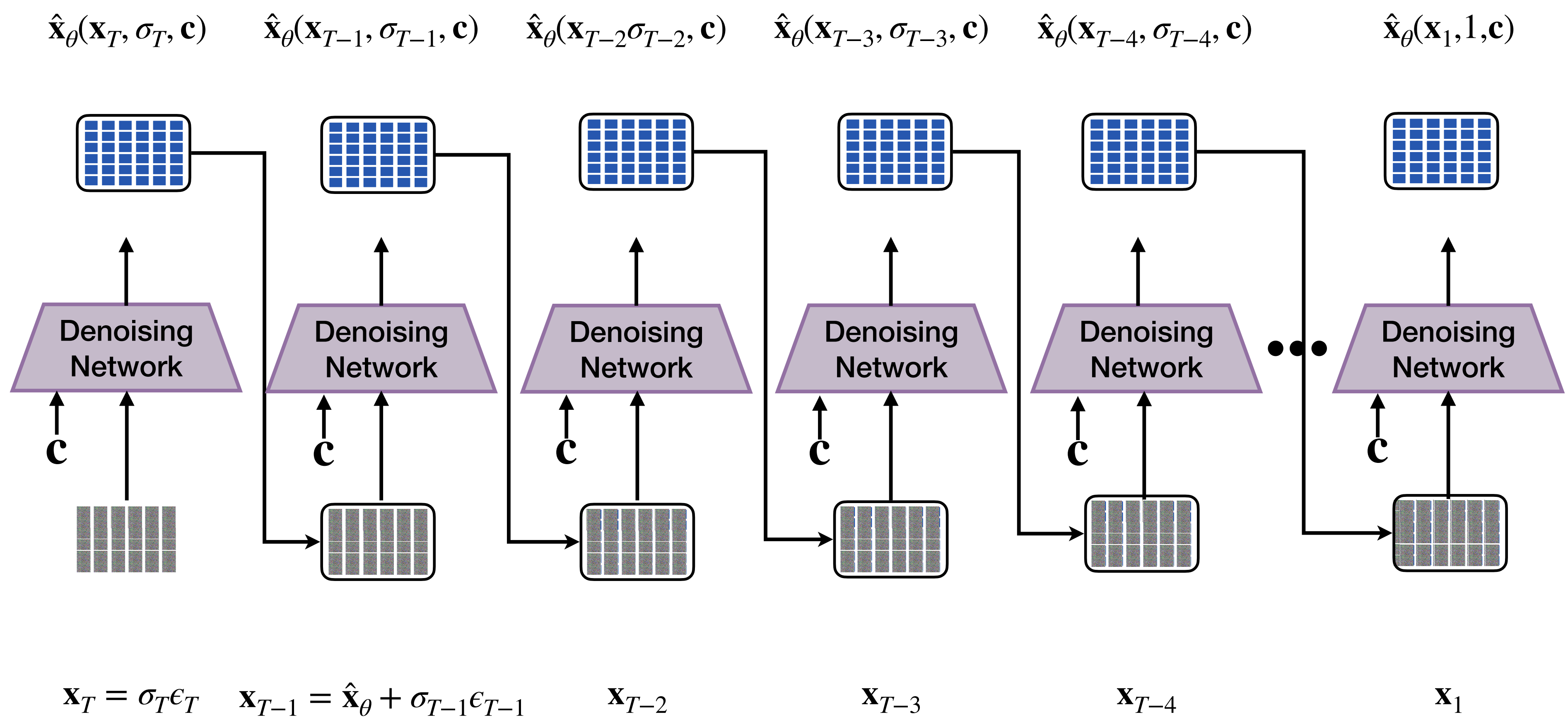


"An astronaut riding a horse"
$\mathbf{c}$

$\epsilon_T \sim \mathcal{N}(0,1)$

Stable Diffusion

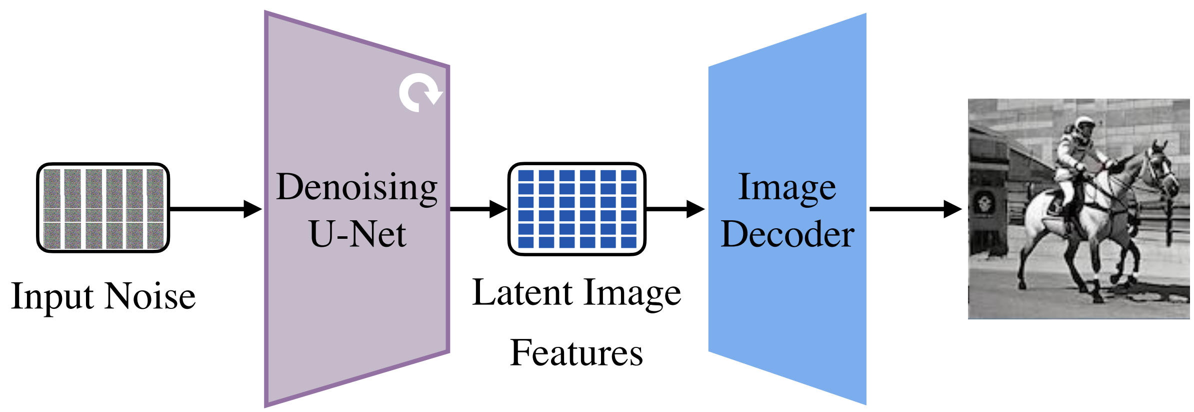
[Rombach et al. 2022]



Stable Diffusion

[Rombach et al. 2022]

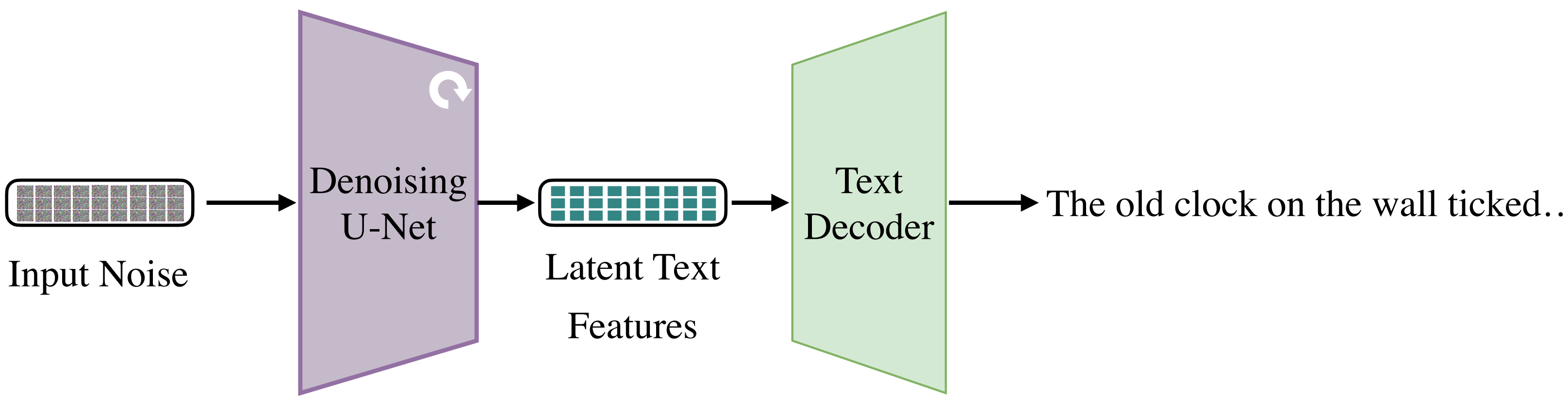
1. Generate latent image features with diffusion model
2. Decode to pixel space with pre-trained decoder



Latent Diffusion for Language

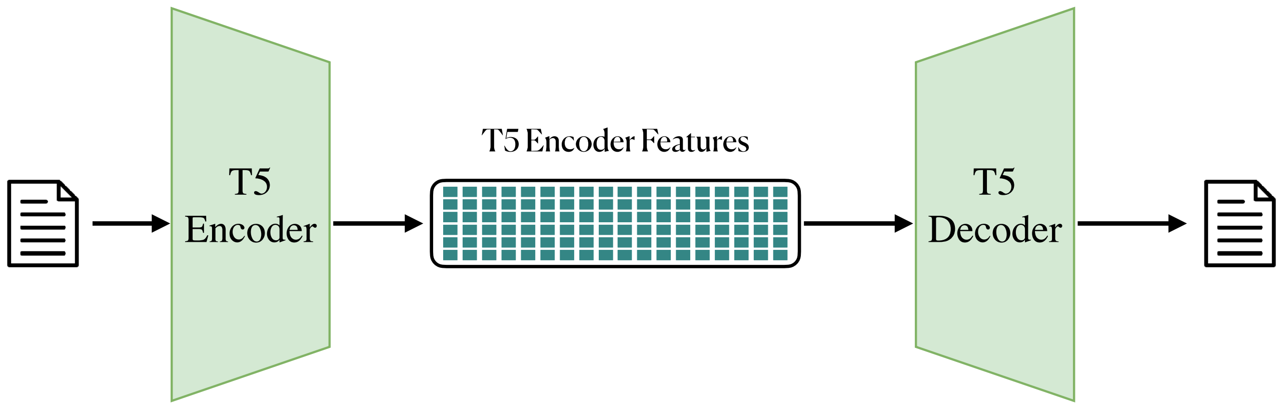
Adapt latent diffusion to language:

1. Generate continuous language representation with diffusion
2. Decode latent features to discrete tokens autoregressively



Language Autoencoder

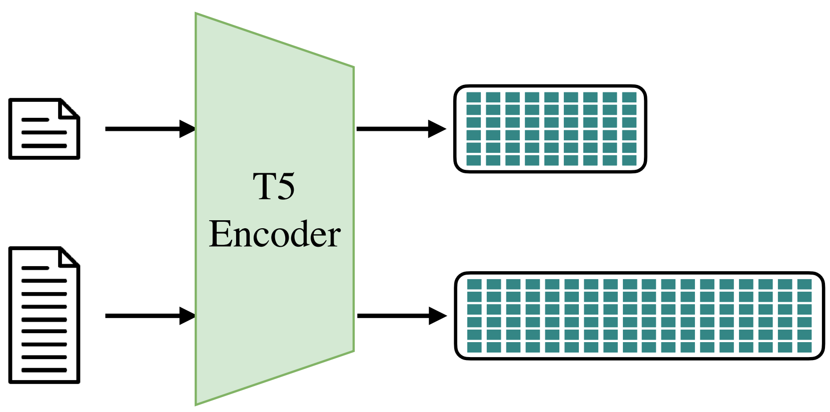
- Start with pre-trained encoder-decoder language models (T5, BART, etc.)
  - Gives us a strong text encoder and decoder
- Doesn't quite get us there…



Language Autoencoder

Main Challenges

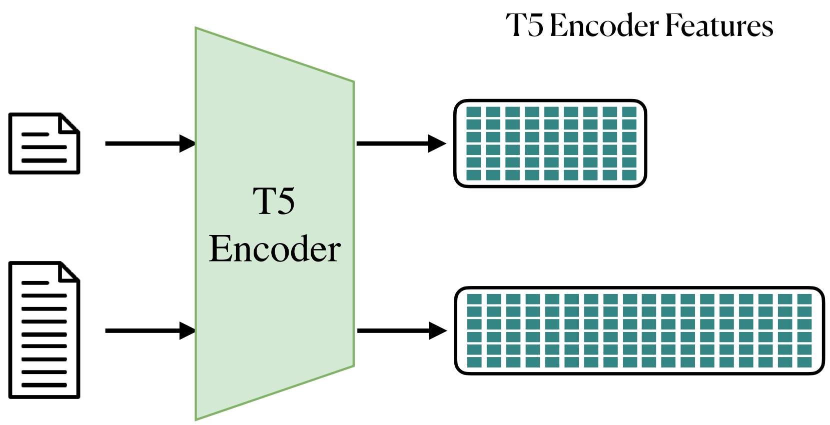
1. Length of encoder features varies with the input sequence length
    - Don't know this at generation time!
2. Language model features are very high-dimensional



Language Autoencoder

Compression

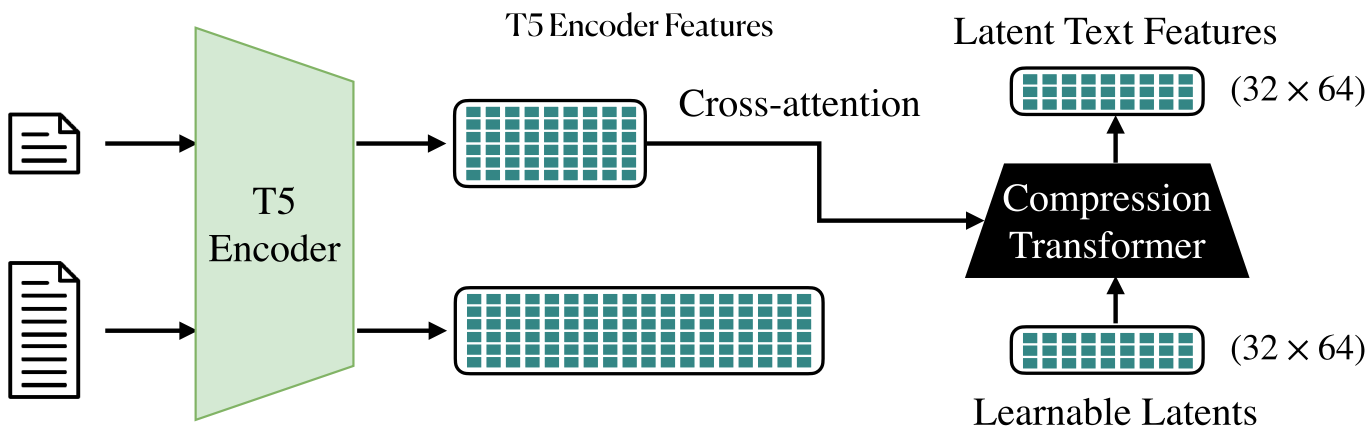
- Compress T5 encoder features with a small transformer
- Read variable-length encoder features into a fixed set of latent vectors



Language Autoencoder

Compression

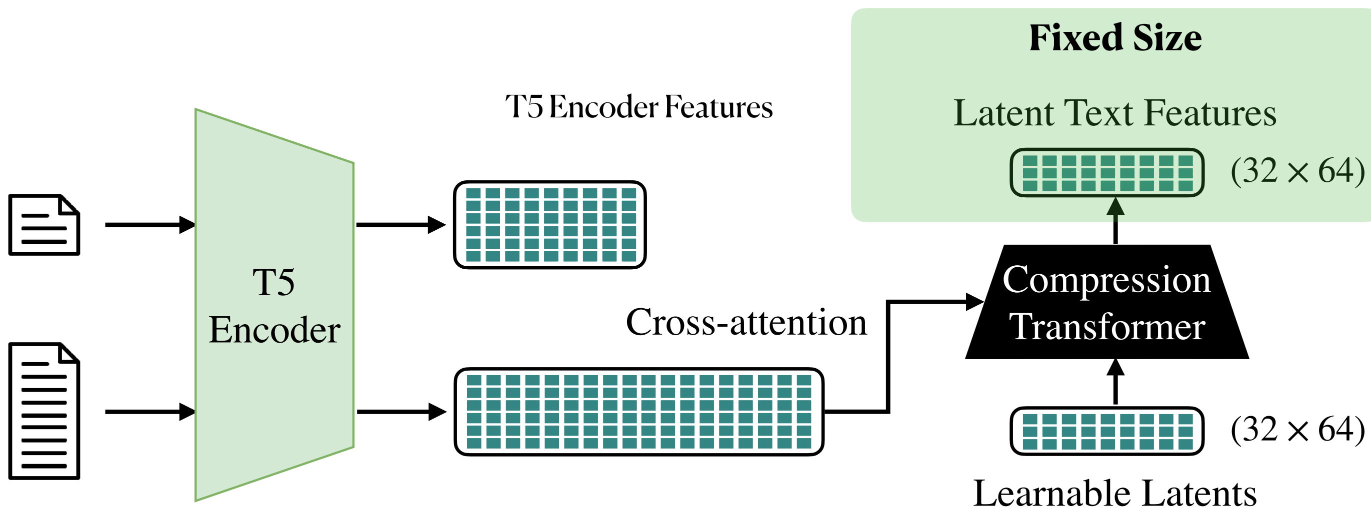
- Compress T5 encoder features with a small transformer
- Read variable-length encoder features into a fixed set of latent vectors



Language Autoencoder

Compression

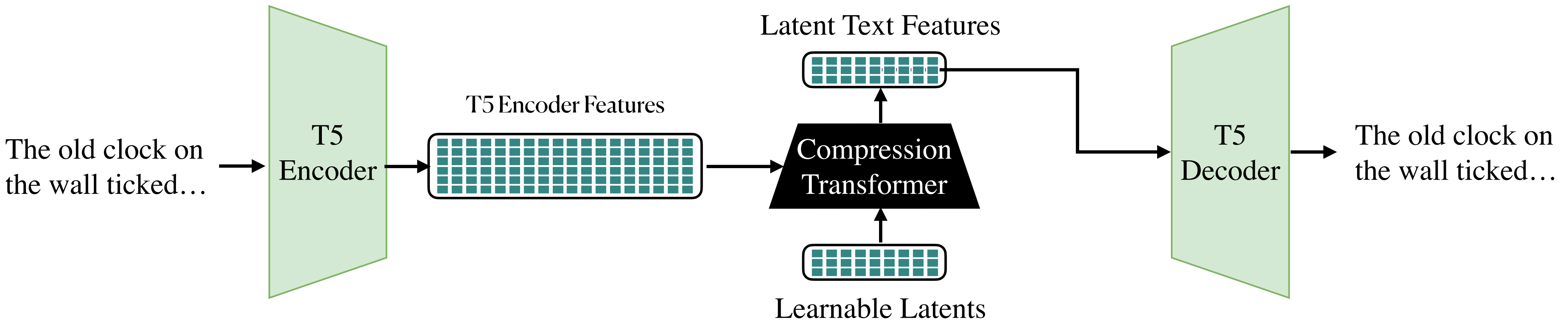
- Compress T5 encoder features with a small transformer
- Read variable-length encoder features into a fixed set of latent vectors



Language Autoencoder

Final Design

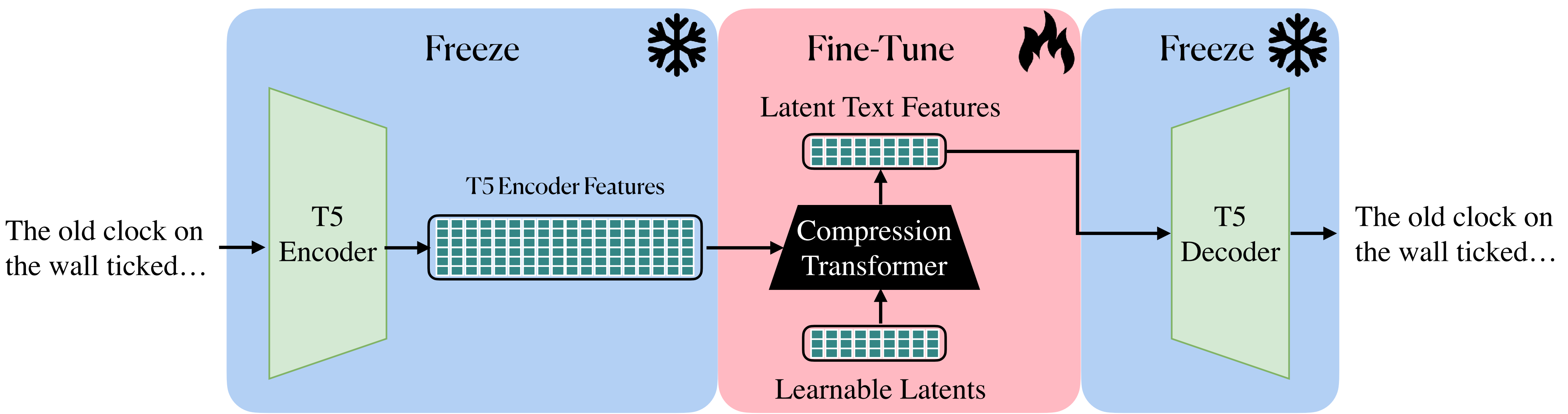
- Compress variable-length encoder features to a fixed-size latent space
- Train decoder to reconstruct the input text
    - Achieve near-perfect reconstruction



Language Autoencoder

Final Design

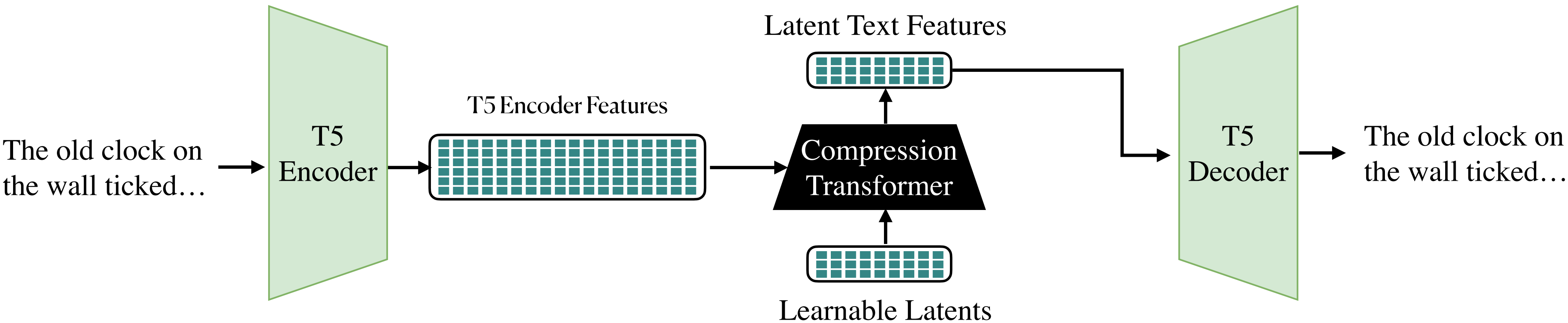
- Freeze pre-trained Encoder/Decoder
- Only learn compression transformer



Language Autoencoder

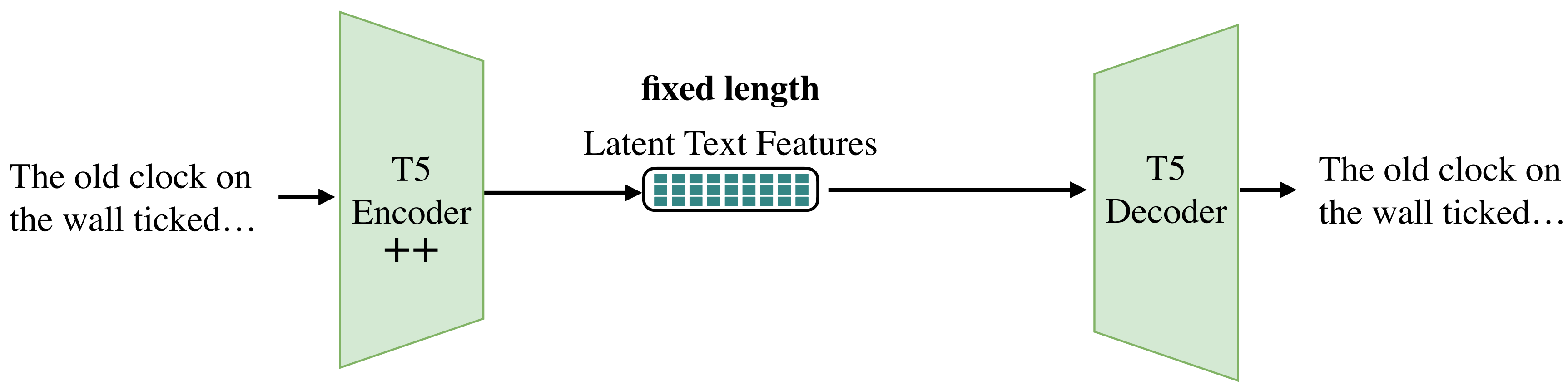
Final Design

- Gives us a compact latent space well-suited for diffusion



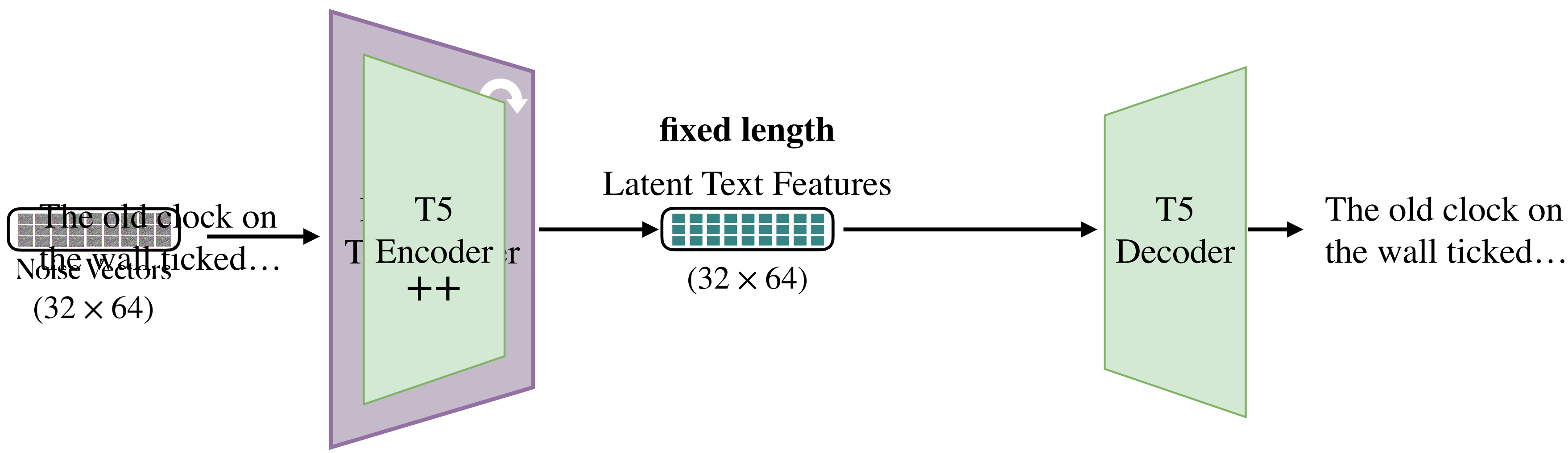
Final Design

- Gives us a compact latent space well-suited for diffusion



Latent Diffusion for Language Generation

1. Generate continuous language representation with diffusion
2. Decode latent features to discrete tokens autoregressively



Unconditional Language Generation

- ROCStories Dataset [Mostafazadeh et al. 2016]
    - Simple 5-sentence stories

The boy went to a video arcade. He played his favorite machine. His games didn't go very well. He told the owner about his experience. The owner explained that he had made the game settings harder.

Dataset Sample

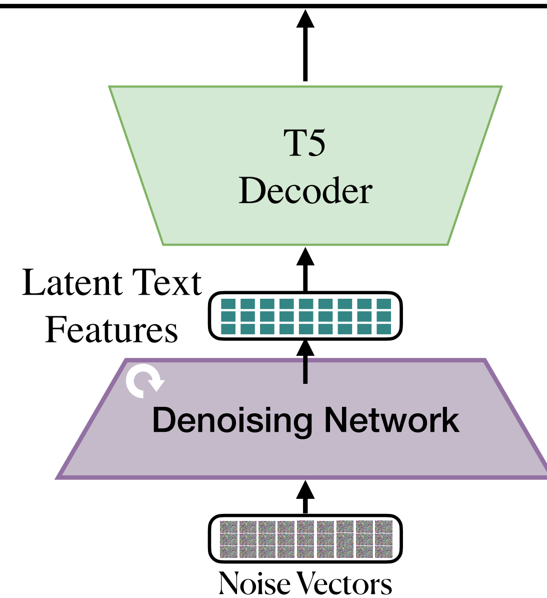
Unconditional Language Generation

- ROCStories Dataset [Mostafazadeh et al. 2016]
  - Simple 5-sentence stories

The boy went to a video arcade.
He played his favorite machine.
His games didn't go very well.
He told the owner about his experience.
The owner explained that he had made the game settings harder.

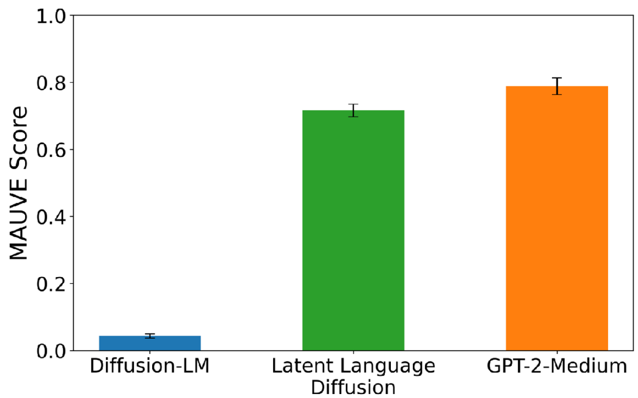
Dataset Sample

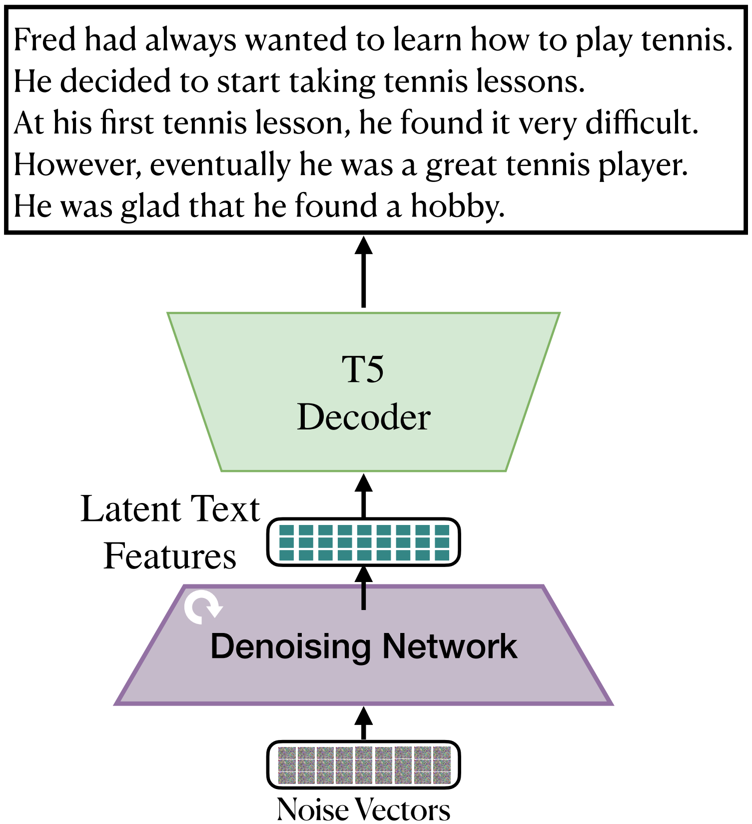
Fred had always wanted to learn how to play tennis.
He decided to start taking tennis lessons.
At his first tennis lesson, he found it very difficult.
However, eventually he was a great tennis player.
He was glad that he found a hobby.



Unconditional Language Generation

- ROCStories Dataset
    - Simple 5-sentence stories
- More effective than prior text diffusion approaches
    - Diffusion-LM: word embedding diffusion

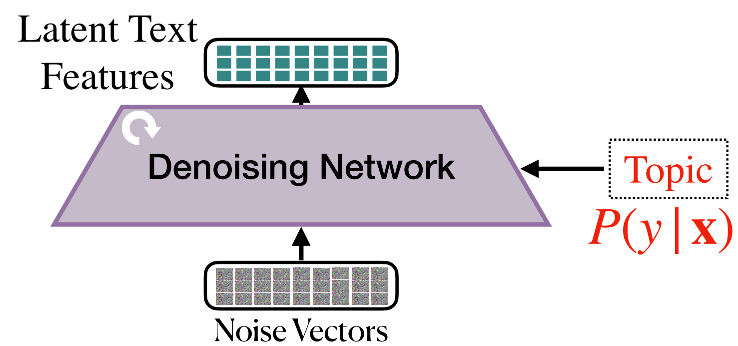




Li, Xiang, et al. "Diffusion-lm improves controllable text generation." *Advances in Neural Information Processing Systems* 35 (2022): 4328-4343.
Pillutla, Krishna, et al. "Mauve: Measuring the gap between neural text and human text using divergence frontiers." *Advances in Neural Information Processing Systems* 34 (2021): 4816-4828.

Class-Conditional Language Generation

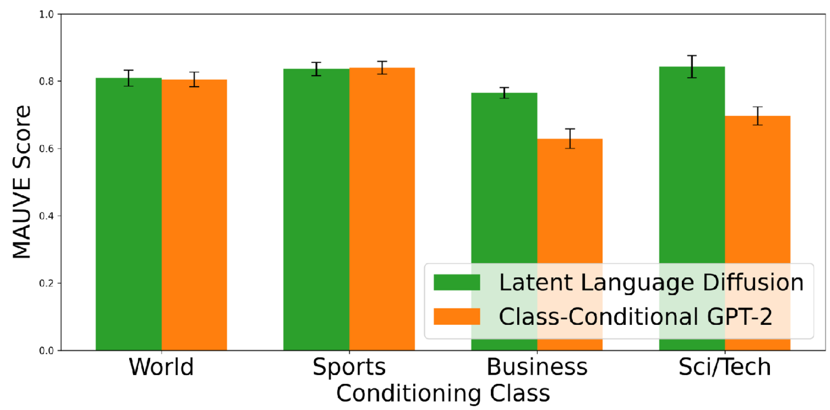
- AGNews Dataset
  - Descriptions from news articles across four classes: "World", "Sports", "Business", "Sci/Tech"

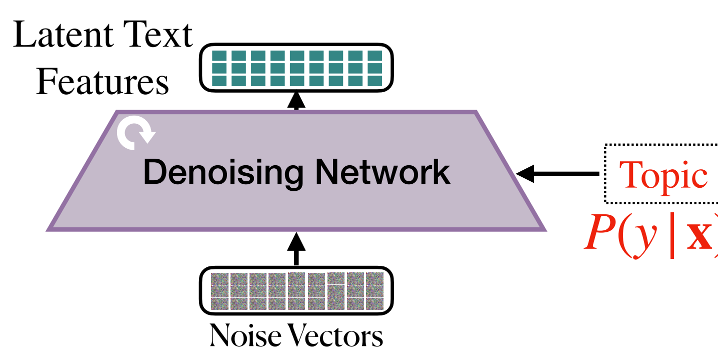


Pillutla, Krishna, et al. "Mauve: Measuring the gap between neural text and human text using divergence frontiers." *Advances in Neural Information Processing Systems* 34 (2021): 4816-4828.

Class-Conditional Language Generation

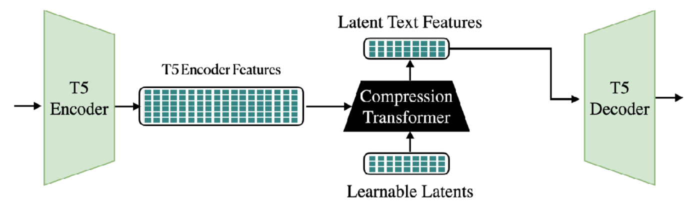
- AGNews Dataset
  - Descriptions from news articles across four classes: "World", "Sports", "Business", "Sci/Tech"
- Better alignment with topic distributions than class-conditional GPT-2





Pillutla, Krishna, et al. "Mauve: Measuring the gap between neural text and human text using divergence frontiers." *Advances in Neural Information Processing Systems* 34 (2021): 4816-4828.

Takeaways



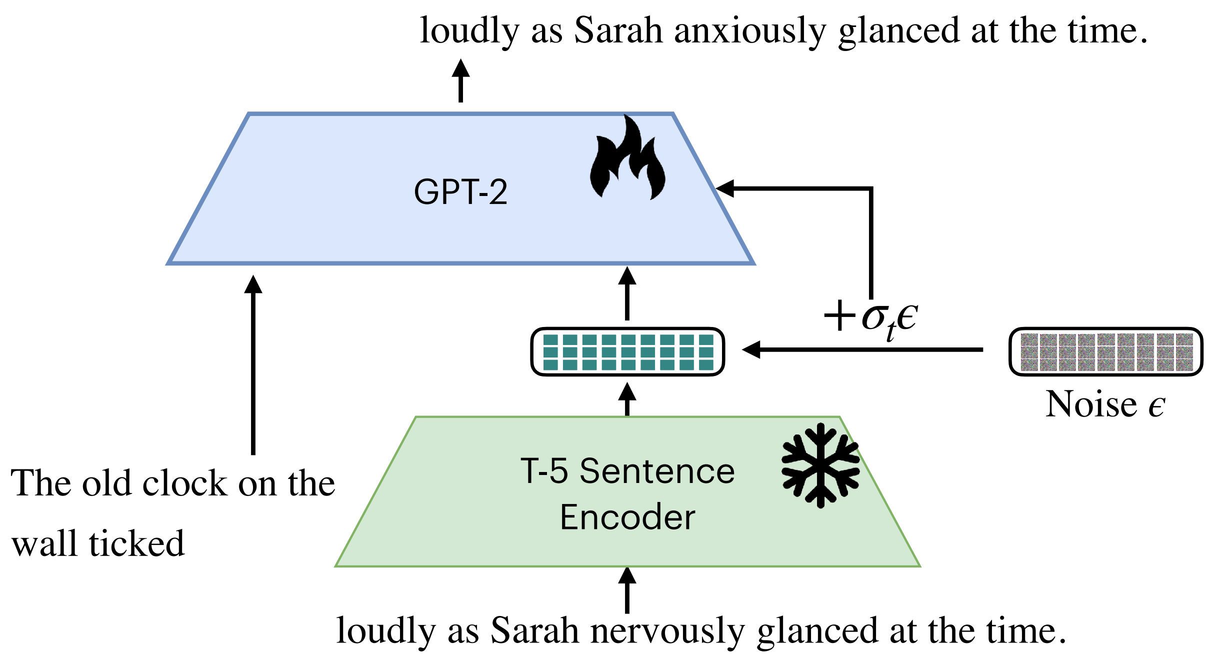
- Latent diffusion is viable for language generation
  - Generate continuous text features with diffusion
  - Offload modeling a discrete distribution to an autoregressive decoder
- Caveat: challenging to match quality of autoregressive LMs
- Some evidence for advantages of diffusion for controllable generation
- **Next question**: **Can we use DMs to control AutoRegressive Models?**

How to control an elephant?

Diffusion Guided Language Modeling

[ACL 2024]

During Training of GPT-2:

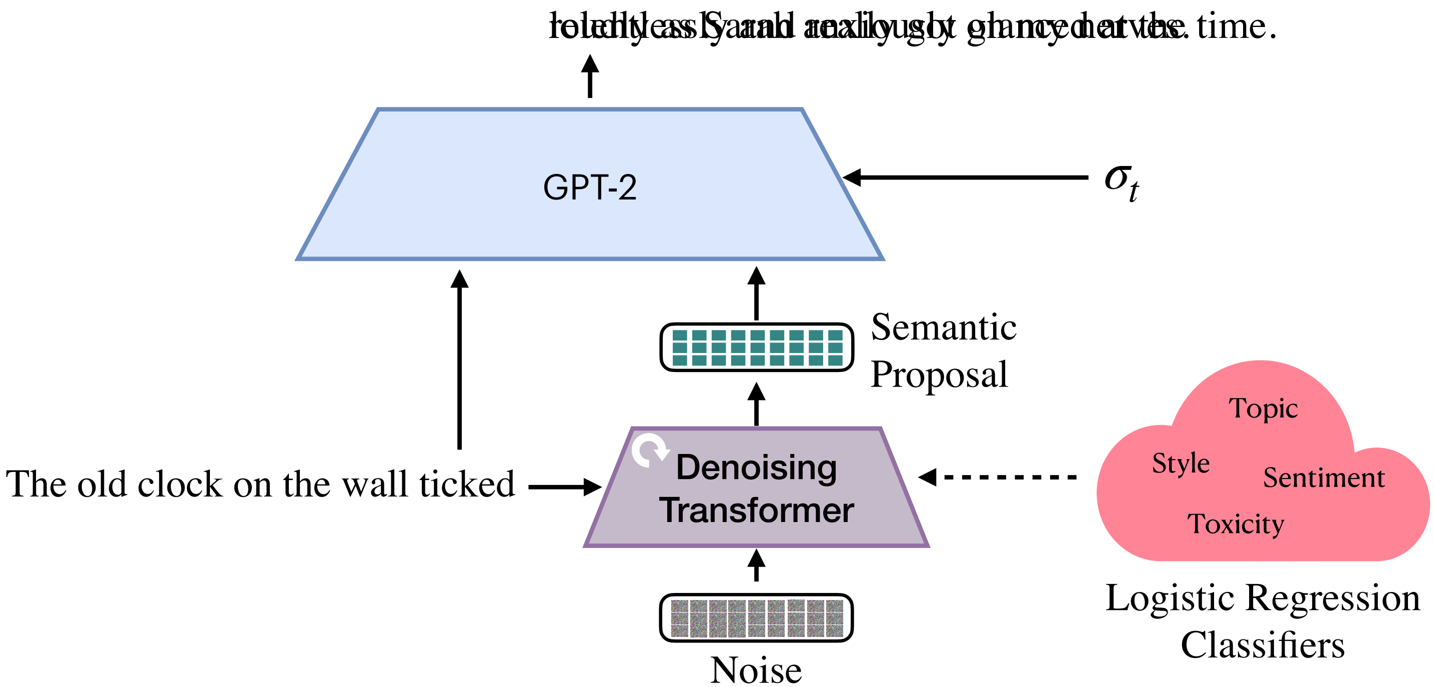


T5 Sentence encoder [Ni et al. 2021]

Diffusion Guided Language Modeling

[ACL 2024]

At Inference:



T5 Sentence encoder [Ni et al. 2021]

Diffusion Guided Language Modeling

Sentiment Control

Prefix:

`Cycle, published by the CTC, is running…`

Control Condition (Sentiment=Positive)

… its 10th edition and it is getting **better every time** I see the contents! It's also very…

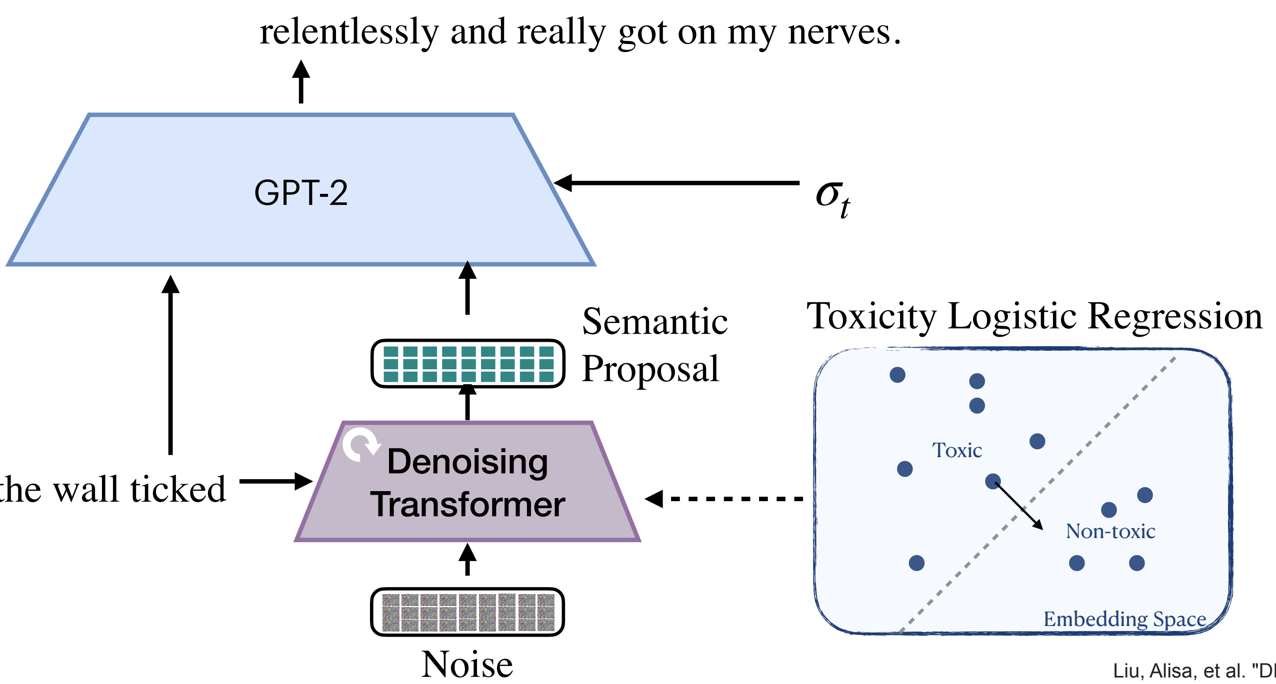
Control Condition (Sentiment=Negative)

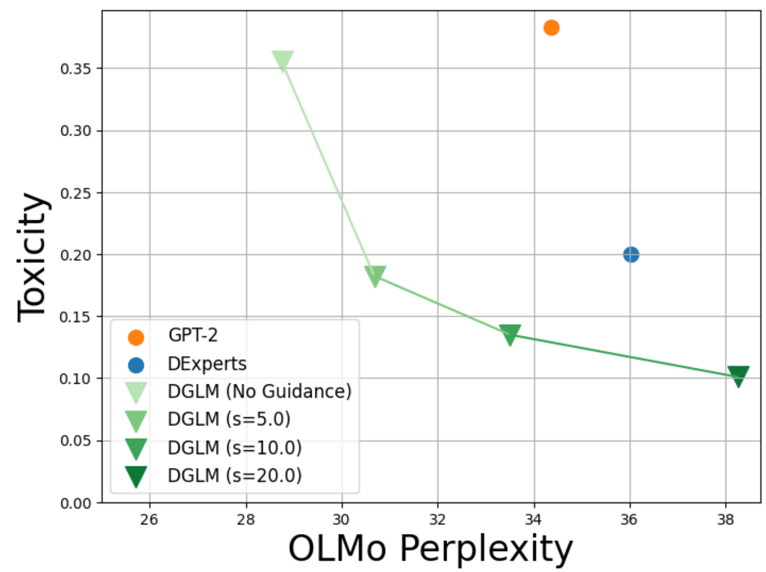
… its **'news'** section, with **no substance at all** and **zero interest** in the subject it...

Diffusion Guided Language Modeling

Toxicity Mitigation

- Reduce toxicity with minimal loss of fluency
- Outperform prior plug-and-play methods (DExperts)

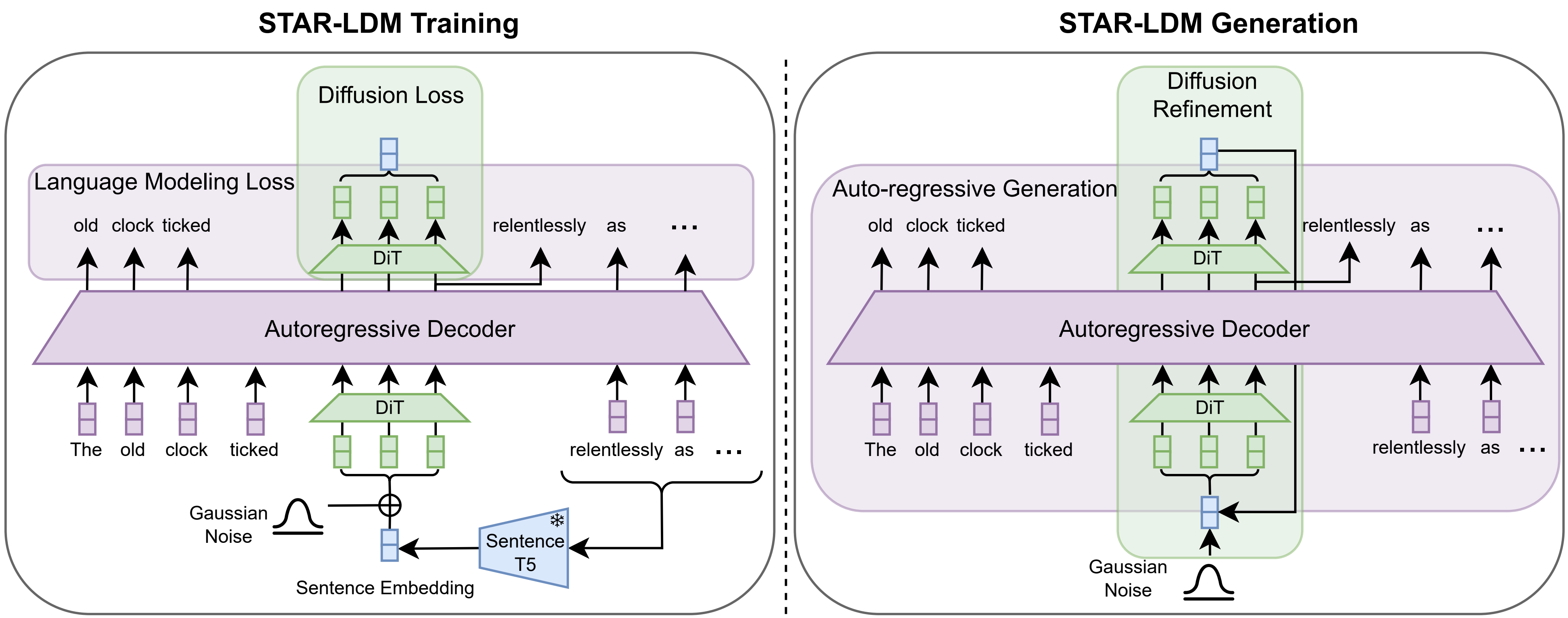




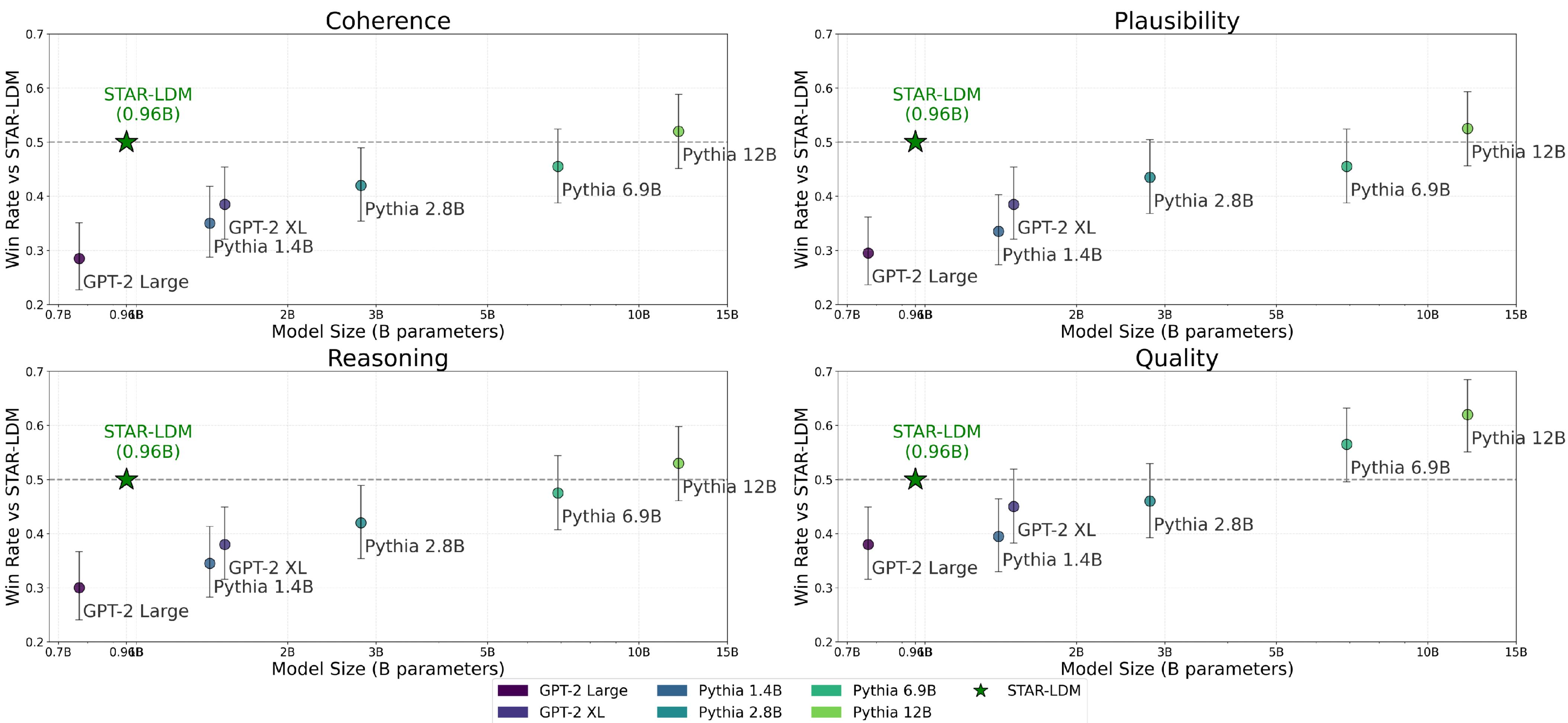
Liu, Alisa, et al. "DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts." *ACL*, 2021.

Stop Think AutoRegress (STAR)

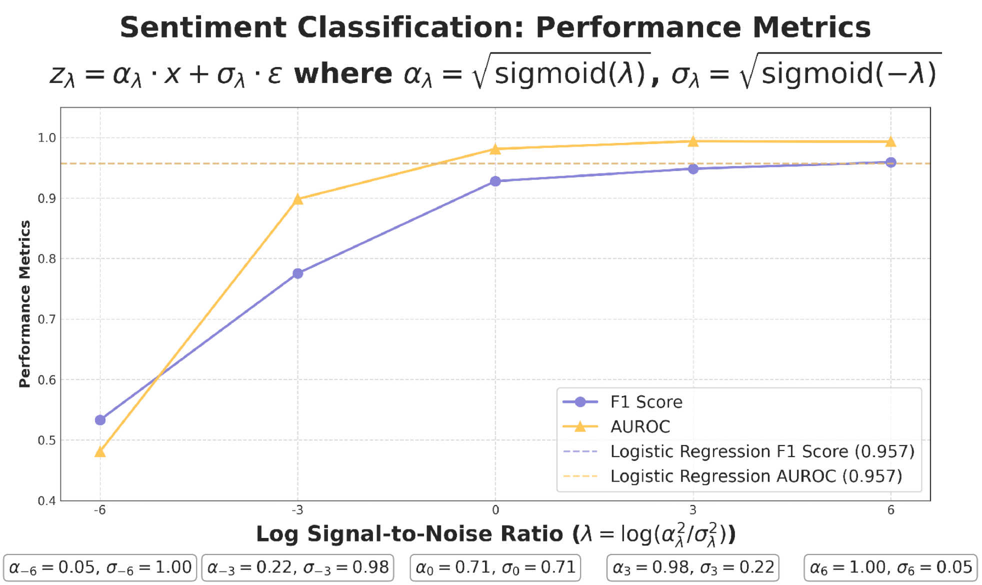
Stop Think AutoRegress (STAR): Transfusion

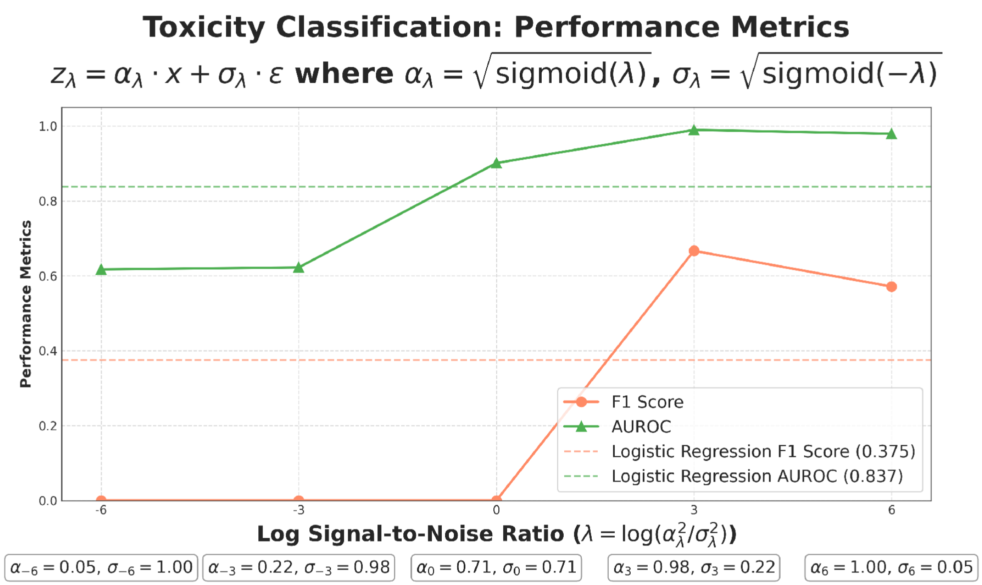


Under submission at COLM, inspired by [Zhou et al. 2024]   DiT: [Peebles & Xie, 2022)]



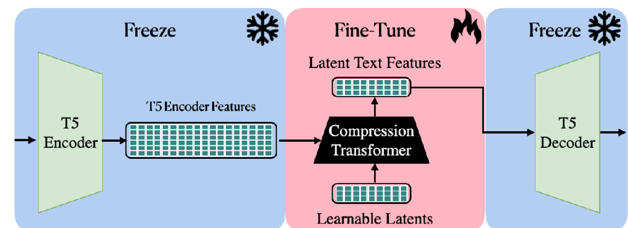
Data: StoryCloze [Mostafazadeh et al., 2016] 5th sentence completion of 4 sentence stories. LLM judge Claude 3.7

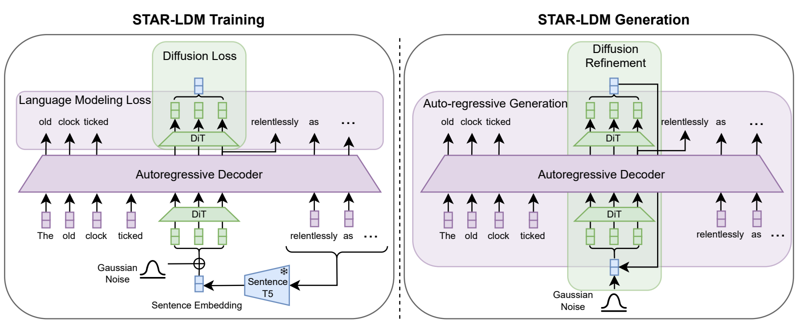

Sentiment Classification: Performance Metrics
$z_\lambda = \alpha_\lambda \cdot x + \sigma_\lambda \cdot \varepsilon$ where $\alpha_\lambda = \sqrt{\text{sigmoid}(\lambda)}$, $\sigma_\lambda = \sqrt{\text{sigmoid}(-\lambda)}$
Performance Metrics
F1 Score
AUROC
Logistic Regression F1 Score (0.957)
Logistic Regression AUROC (0.957)
Log Signal-to-Noise Ratio ($\lambda = \log(\alpha_\lambda^2/\sigma_\lambda^2)$)
$\alpha_{-6} = 0.05$, $\sigma_{-6} = 1.00$
$\alpha_{-3} = 0.22$, $\sigma_{-3} = 0.98$
$\alpha_0 = 0.71$, $\sigma_0 = 0.71$
$\alpha_3 = 0.98$, $\sigma_3 = 0.22$
$\alpha_6 = 1.00$, $\sigma_6 = 0.05$

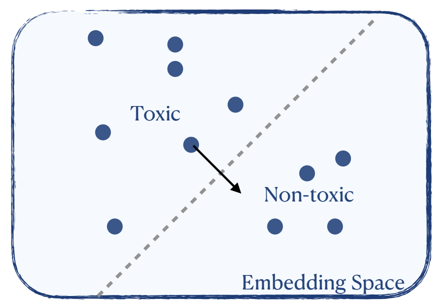

Toxicity Classification: Performance Metrics
$z_\lambda = \alpha_\lambda \cdot x + \sigma_\lambda \cdot \varepsilon$ where $\alpha_\lambda = \sqrt{\text{sigmoid}(\lambda)}$, $\sigma_\lambda = \sqrt{\text{sigmoid}(-\lambda)}$
Performance Metrics
F1 Score
AUROC
Logistic Regression F1 Score (0.375)
Logistic Regression AUROC (0.837)
Log Signal-to-Noise Ratio ($\lambda = \log(\alpha_\lambda^2/\sigma_\lambda^2)$)
$\alpha_{-6} = 0.05$, $\sigma_{-6} = 1.00$
$\alpha_{-3} = 0.22$, $\sigma_{-3} = 0.98$
$\alpha_0 = 0.71$, $\sigma_0 = 0.71$
$\alpha_3 = 0.98$, $\sigma_3 = 0.22$
$\alpha_6 = 1.00$, $\sigma_6 = 0.05$

Conclusion

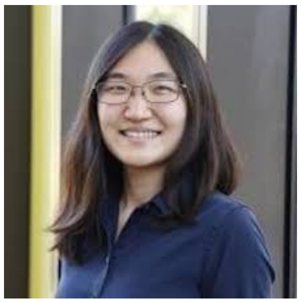
- DMs are awesome and easy to control.
- Autoencoder with fixed-length latent enables Latent Diffusion
- Allows **control** of **text generation** with simple classifier
- We can **guide** Auto-Regressive models with DMs
- **STAR Transfusion** integrates diffusion into autoregressive process







Thanks to


Jennifer Sun


Linxi Zhao


Justin Lovelace


Chao Wan


Eliot Shekhtman


Jin Zhou


Yoav Artzi


Vivian Chen


Sofian Zalouk


Christian Belardi


Adhitya Polavaram


Varsha Kishore