

# FM Copyright Infringement

Group 6

Feng Guo, Tongxuan Tian, Weifeng Yu, Yanxi Liu, Kefan Song

# Agenda

1. Foundation Models And Fair Use
2. Copyright Plug-in Market for The Text-to-Image Copyright Protection
3. Extracting Training Data from Diffusion Models
4. A Comprehensive Survey of AI-Generated Content (AIGC):A History of Generative AI from GAN to ChatGPT
5. Llama 2: Open Foundation and Fine-Tuned Chat Models

# FOUNDATION MODELS AND FAIR USE

Peter Henderson\*, Xuechen Li\*, Dan Jurafsky, Tatsunori Hashimoto, Mark A.  
Lemley, Percy Liang

Stanford University

Presented by Kefan Song(ks8vf), Yanxi Liu(kww7ur)

## Outline:

- Overview US case Law on Fair Use
- Examples on Generated Text, Code and Images
- Strategies to Mitigate the Risk

## Fair Use Defense:

### Data creator

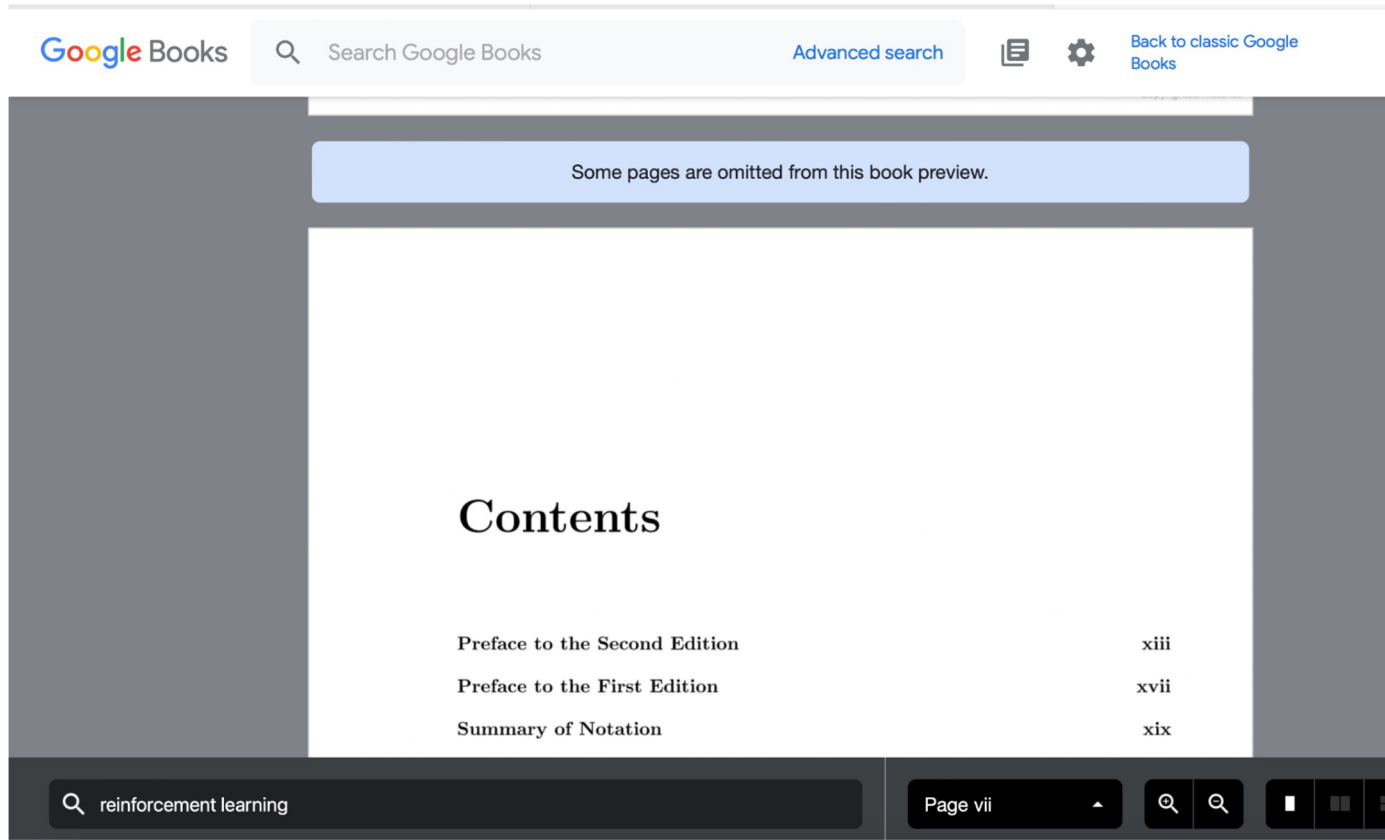
- Creates content that might be used for GenAI training.
- Whose copyright may be violated.
- May sue Tech Company who deploys GenAI

### Tech Company

- When Tech Companies who deploys GenAI is sued for copyright violation, they can use the Fair Use Defense to not get charged.

## Previous Example of Fair Use Defense not involving GenAI

### Google Books



Google Books

Search Google Books

Advanced search

Back to classic Google Books

Some pages are omitted from this book preview.

### Contents

Preface to the Second Edition	xiii
Preface to the First Edition	xvii
Summary of Notation	xix

reinforcement learning

Page vii

# Four “Arguments” Tech Company Can Use for Defense

If the use of unlicensed copyrighted materials:

1. satisfy transformativeness
2. (Nature of the work) Is factual vs creative
3. the amount of the portion used is small
4. has little effect on market of the copyrighted materials

then such use is legal.

# Examples of Fair Use Defense

	Protected by Fair Use	Unprotected by Fair Use
Transformativeness	“Different Purpose” Google Books allowing searching through the copyrighted books	A cover to cover reproduction
Nature of work	Facts, Ideas	The Expression of facts or ideas.
Amount	Long, but small portion of the material	Short but large portion of the material (Tattoo)
Effect on market		A derivative book



# Natural Language Text - Examples of Fair Use Defense

Text generation : One of the most prevalent, and earliest, use-cases of foundation models, like GPT.

Applications: Copy-editing, text-based games, and general-purpose chatbots.

Training data sources: internet, books, court documents.

Fair Use Considerations:

- (1) The role of transformation in determining fair use.
- (2) Examination of relevant cases paralleling foundation model outputs.

# Natural Language Text - Examples of Fair Use Defense

## Verbatim Copying and Hypotheticals:

- (1) Google Books case: Limited content provision as fair use.
- (2) Hypothetical scenario: Virtual assistant reading books aloud.

## Implications for Foundation Models:

- (1) The thin line between transformative use and copyright infringement.
- (2) The importance of model output transformation for fair use defense.

# Natural Language Text - Examples of Fair Use Defense

## Challenges in Determining Fair Use:

- (1) Difficulty in applying fair use to verbatim and minimally transformed outputs.
- (2) The significance of the amount and substantiality of the used portion.

## Strategies for Compliance:

- (1) Enhancing model outputs for greater transformation.
- (2) Legal and technical strategies to align with fair use doctrine.

# Code - Examples of Fair Use Defense

Natural language text and code generation models have similar training processes, in fair use assessments, they have each different case law with slightly varied assessments.

Literal vs. Non-literal Infringement:

Literal infringement (verbatim copying) unlikely to be fair use, especially for significant portions of the code.

Introduction of tests for non-literal infringement: Abstraction-Filtration-Comparison and SSO tests, focusing on copyrightable, expressive aspects of code (e.g., inter-modular relationships).

# Code - Examples of Fair Use Defense

## Challenges in Non-literal Copyright:

- (1) Judges acknowledge unclear boundaries for non-literal program structure copyright protection.
- (2) Difficulty in proving nonliteral infringement due to protection limitations on non-expressive, functional elements of programs.

## Criteria for Fair Use in Code:

- (1) Small amounts of copied code, significant transformation, or different overall product may indicate fair use.
- (2) The importance of transforming generated content to reduce infringement risk.

# Code - Examples of Fair Use Defense

## Copyright Protection Limitations:

- (1) Functional aspects of code have limited copyright protection compared to creative works.
- (2) Encouragement for transformation in generated software to minimize legal risks.

## Additional Concerns in Code Generation:

- (1) Potential right of publicity issues with verbatim output of usernames.
- (2) DMCA §1202 and right of publicity considerations for transformative works.

# Generated Images - Examples of Fair Use Defense

The third commonly produced category of generative AI is image generation.

**Complexities of fair use with images.** -> Hypothetical 2.5: Generate Me Video-Game Assets.

While fair use might offer some defense, the direct appropriation of artists' work with only slight alterations poses a significant legal risk for the company, indicating that their use might not qualify as fair use.

## Hypothetical 2.5: Generate Me Video-Game Assets.

One direction for generative art is creating video game assets. There are already mechanisms to generate 3D models from text (Poole et al., 2022). Consider a situation where a video game company builds a machine learning model into their system that generates art on the fly within the game to populate a virtual world dynamically. The game is a hit, but artists begin to notice that their artwork shows up in the game with only slight modifications, for example on tattoos for video game characters. Is this fair use? While their lawsuit is not guaranteed to succeed, there is still some risk for the video game company if the outcome follows *Alexander v. Take-Two Interactive Software, Inc.* (S.D. Ill. 2020).

# Generated Images - Examples of Fair Use Defense

The third commonly produced category of generative AI is image generation.

## **Style Transfer.**

More abstract scenarios, where art is generated in different styles.

Three components to consider:

1. The rights of the original image that is being transformed into a different style.
2. The rights of the artist whose style is being mimicked.
3. Other intellectual property considerations with images:  
the right to publicity and trademark infringement.

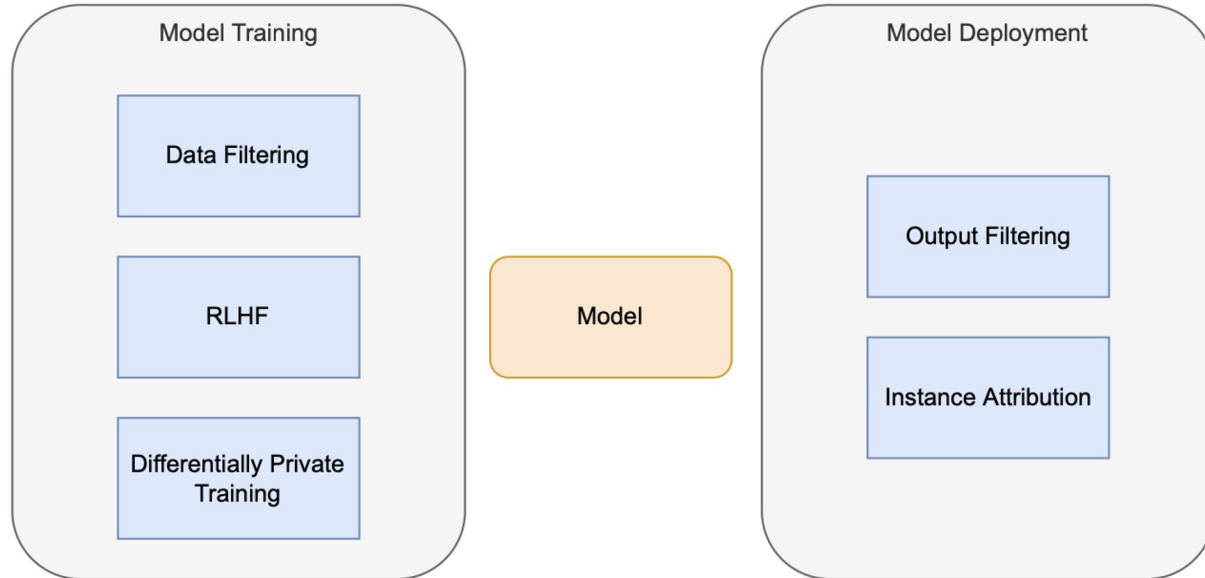


# Technical Mitigation

<b>Non-Technical Mitigation</b>	<b>Technical Mitigation</b>
Target market	Transformativeness
Commercial Use	Amount of Material
Good Faith	Identifying Parody
	Facts or Expression of Facts

# Technical Mitigation

## Training Time Mitigation vs Deployment Time Mitigation



# 1. Data Filtering

## Two Types of Data Filtering

1. Not train on dataset.
  - a. E.g. AlphaCode only trained on unlicensed Github source code
  - b. Restrict to robot.txt for webcrawled data
  
1. Deduplication to reduce memorization
  - a. Problematic: Given different images of an NBA player, a tattoo may still be memorized.

## 2. Output Filtering

Apply a filter to detect output similar to training data

E.g. Github Copilot

Disadvantages of Current Output Filters

1. Additional inference costs
2. Easily bypassed by minor style-transfer

Future direction:

An output filter that detects high-level semantic similarity?

### 3. Instance Attribution

Given training examples  $\mathcal{Z}_1, \dots, \mathcal{Z}_n$ ,

Train a parameter by Empirical Risk Minimization :

$$\hat{\theta} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta).$$

Remove one example  $\mathcal{Z}$

Retrain a parameter

$$\hat{\theta}_{-z} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \sum_{z_i \neq z} L(z_i, \theta)$$

Obtain the difference between two parameters:

$$\hat{\theta}_{-z} - \hat{\theta}$$

### 3. Instance Attribution

Application to Fair Use:

For a copyrighted datapoint  $z$

A larger difference on  $\hat{\theta}_{-z} - \hat{\theta}$

Indicates a higher risk of violating fair use.

### 3. Instance Attribution

Disadvantage :

High Computation costs (leave one out retraining or inverting Hessian)

Alternatives:

Retrieval Augmented Methods

It naturally selects the instance before inferencing

## 4. Differentially Private Training

*Definition 1.* A randomized mechanism  $\mathcal{M}: \mathcal{D} \rightarrow \mathcal{R}$  with domain  $\mathcal{D}$  and range  $\mathcal{R}$  satisfies  $(\epsilon, \delta)$ -differential privacy if for any two adjacent inputs  $d, d' \in \mathcal{D}$  and for any subset of outputs  $S \subseteq \mathcal{R}$  it holds that

$$\Pr[\mathcal{M}(d) \in S] \leq e^\epsilon \Pr[\mathcal{M}(d') \in S] + \delta.$$

For example:

In DP-SGD, noise is added to the gradient, and the output of such randomized mechanisms would be parameters like  $\theta_d$  and  $\theta_{d'}$  and is guaranteed to have DP guarantee.



## 4. Differentially Private Training

### Benefits in Fair Use:

DP trained models naturally less likely to memorize a single instance.

### Challenges in Fair Use:

1. High computation costs
2. Trade off between privacy and accuracy
3. Similar examples to the single example removed

#### **Hypothetical 4.1: Differentially Private Lyric Generation.**

Imagine that a developer intends to train a machine learning model to aid musicians to create lyrics. The developer scrapes copyrighted lyrics of songs from music websites. However, the lyrics of the same song are scraped multiple times, each of which is treated as a single example in the dataset. Additionally, the developer isn't careful about removing duplicates before training the model with DP. The final model thus ends up reproducing verbatim chunks of lyrics of certain songs. The lyricist whose lyrics were reproduced by the deployed model sues an end user who wrote a song with the help of this model.

# 5. Learning from Human Feedback

For Human Annotations,

Provide the closest copyrighted content to the LLM output  
Ask to flag outputs that are not transformative enough.

# Copyright Plug-in Market for The Text-to-Image Copyright Protection

Anonymous authors

ICLR 2024

Feng Guo (grj4jc)

# Outline

- Introduction
- @Plug-in Market
- Features
- Experiments
- Limitation



Civit AI

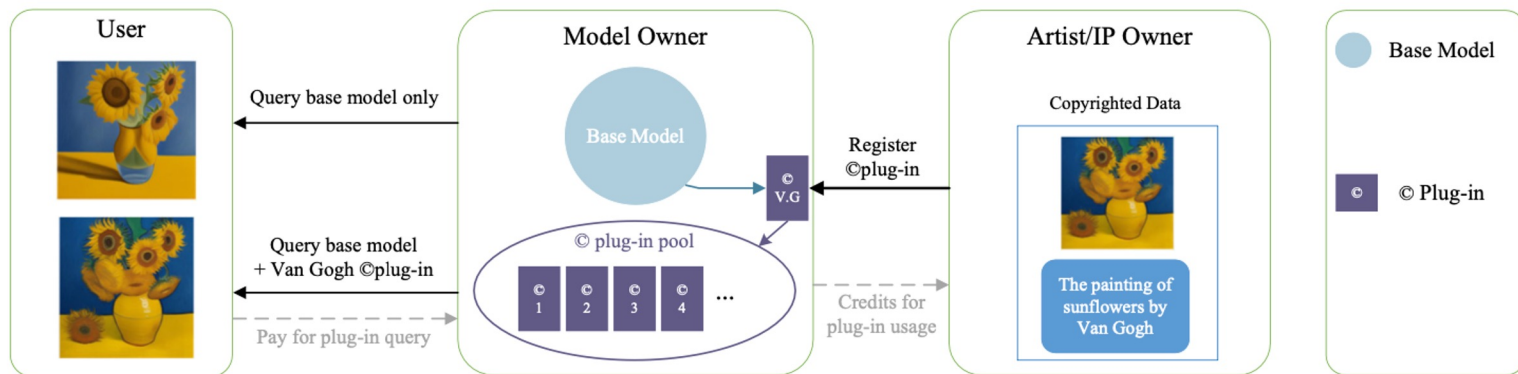
<https://civitai.com/images/6347481>

# Motivation

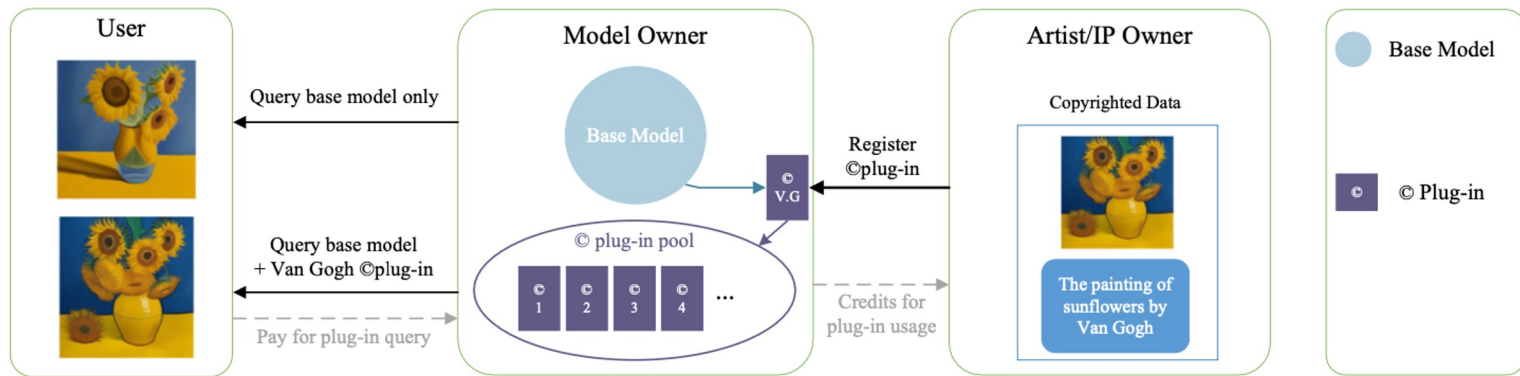
- “whether the copyright laws prohibit using copyrighted data to train machine learning models”
  - Debate between **AI developers, content creator, legislation & judicature department**
  - It’s ok to use for “fair use”, but can we say training procedure is “fair use”
- Impact
  - LLM keep improving the quality of generated images (Diffusion Model)
  - But it cannot attribute credits to the original data in the training set
  - Adding anxiety to artist community
  - Replicate character from major IP ( Disney’s Mickey Mouse, ...)

# @Plug-in Market

- Motivated by the copyright law: reward creators for their work
  - Crediting and share revenue with creator
  - Decode generated image into similar example, so that can credit its original creditors
  - propose a conceptual framework named @Plug-in Market

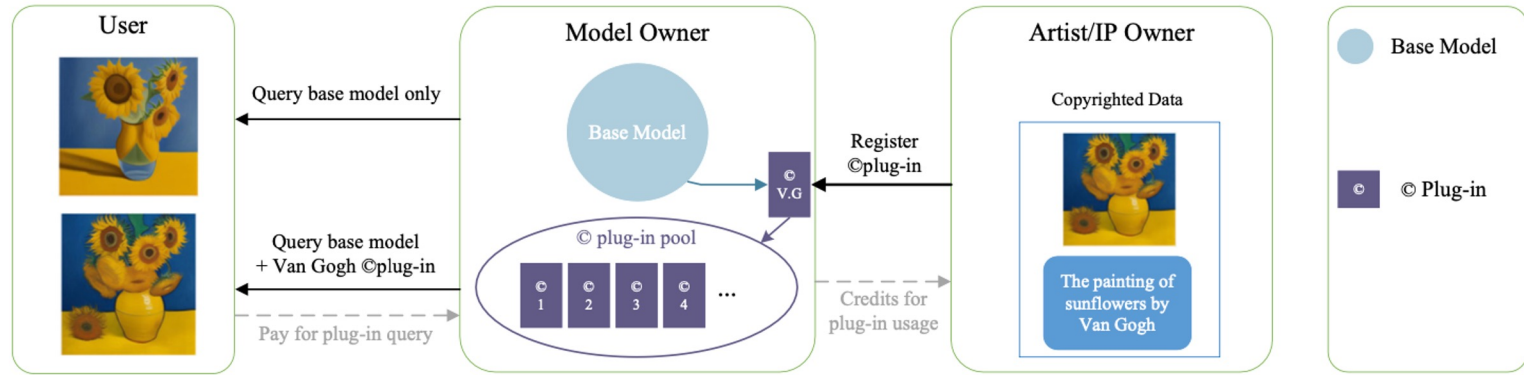


# @Plug-in Market



- Model owner (OpenAI) acts as a platform
- Artist/IP owner: register copyright data as “Plug-in”
- Query base model: not affiliate with the creator
- Query base model with “Plug-in”: credit to creator, user pay for query

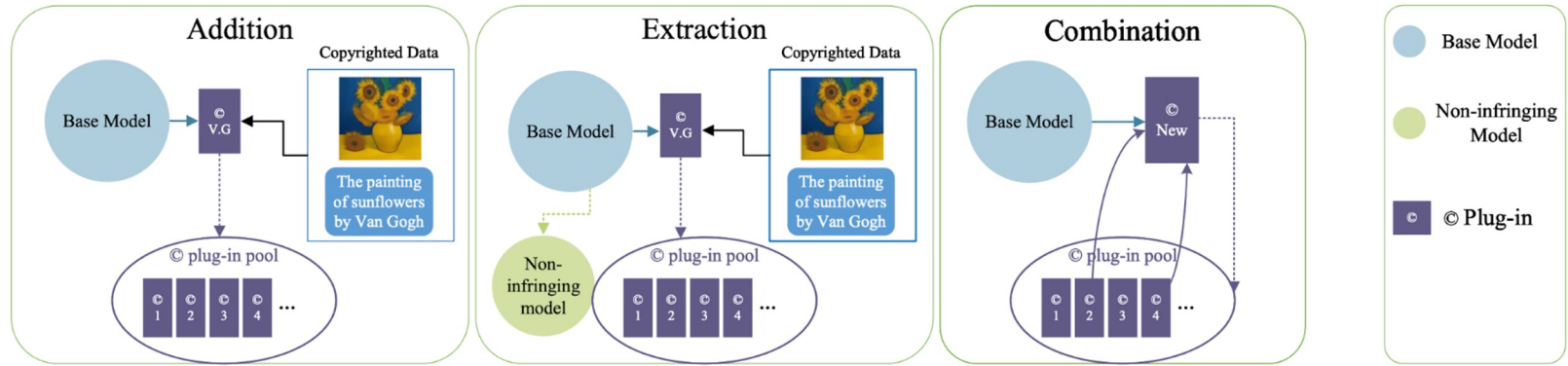
# @Plug-in Market: Benefit to Everyone



- Creator are well compensated for creating new works
- User pay for using copyrighted plug-ins and avoid being accused of copyright infringement in their own creations
- Model owner makes profits for the plug-in registration and usage.
- Market can track the usage of the copyrighted works in an explicit way



# @Plug-in Market Operations



- Addition: creator can easily add work as plugin In a good performance manner
- Extraction: model owner can remove works that are infringed from base model
- Combination
  - Creator can combine their work together
  - User can use different creators' work to create new images

# Background

## Diffusion Model

- Probabilistic models that aim to learn a data distribution
- After training, one can use model to generate new images, which can be based on input (e.g. a prompt text)
- This work based on Stable Diffusion Model

## LoRA(**L**ower **R**ank **A**dapter)

- It locks the pre-trained model weights in place
- It adds trainable rank decomposition matrices to each layer of the Transformer architecture
- It can be shared and used to build many small LoRA modules for different tasks

# Addition

- Can be implemented straightforwardly under LoRA
  - LoRA can server as a plug-in for SDM and learn them with copyright work
  - Track the usage and fairly attribute the reward
- Examples
  - Available in model sharing platforms

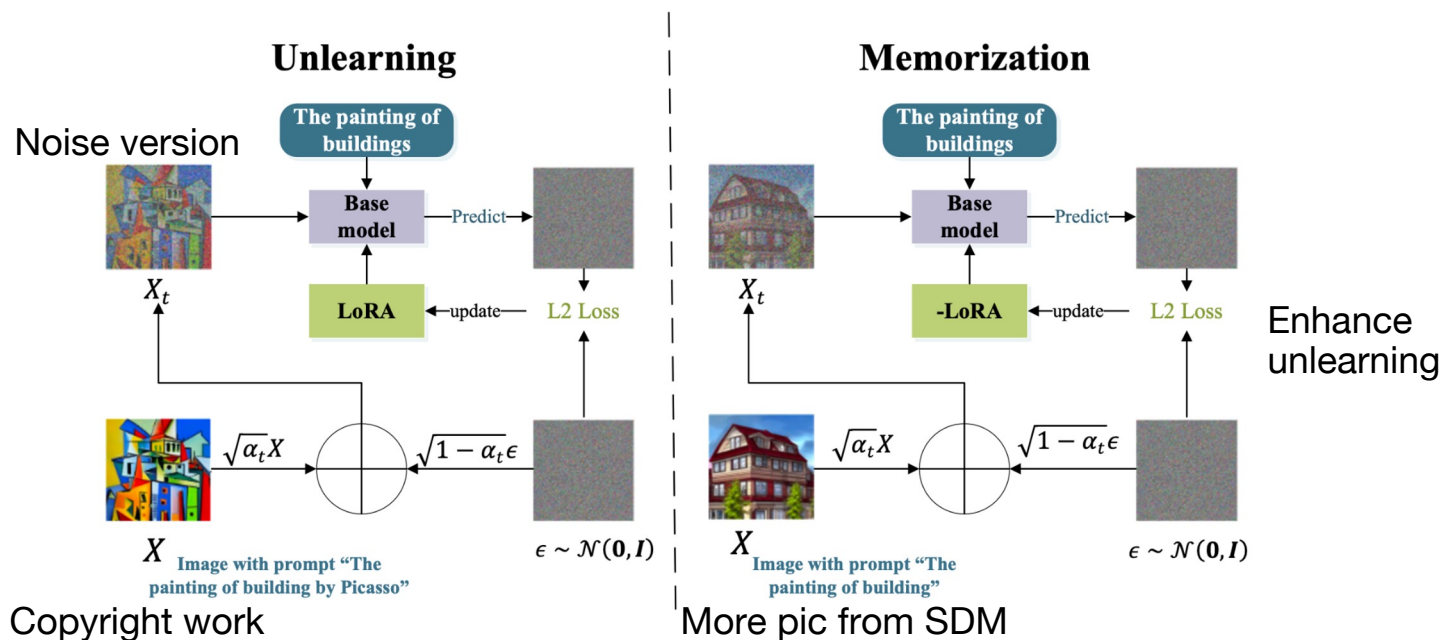


Civit Ai

# Extraction

- Traditional Solution
  - Retrain model from scratch only use non-infringing data
  - High cost, complex data clearing, hard to implement
- Instead, “ Inverse LoRA”
  - Unlearn the target concept
  - Tunes the inversed LoRA to memorize surrounding concepts
  - Inverse LoRA to obtain the non-infringing model

# Extraction Example: Picasso Building



Unlearning: tune LoRA to match copyrighted image with “The painting of the building”

Memorization: guide the generation far away from the target concept “Picasso”

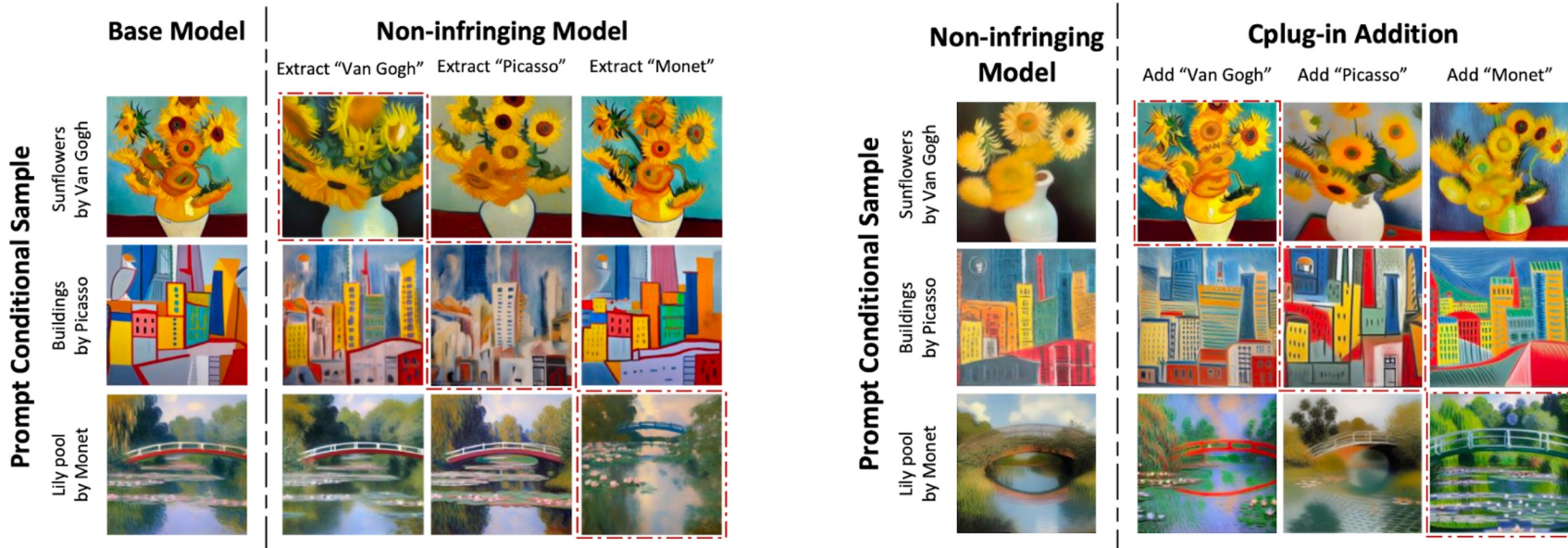
# Combination

- Simply adding two plug-ins will yield unpredictable outcomes (“Snoopy” and “Mikey”)
- EasyMerge: a data-free layer-wise distillation method
  - Data-free: only requiring plug-ins and corresponding text prompts
  - With layer-wise distillation: accomplish the combination in a few iterations

# Experiment

- Example
  - Style transfer: Extraction and Combination
  - Cartoon IP recreation: Extraction and Combination

# Experiments: Style Transfer



(a) Results of *extraction* in style transfer.

(b) Results of *combination* in style transfer

1) Vincent van Gogh 2) Pablo Ruiz Picasso 3) Oscar-Claude Monet



# Experiments: Cartoon IP recreation

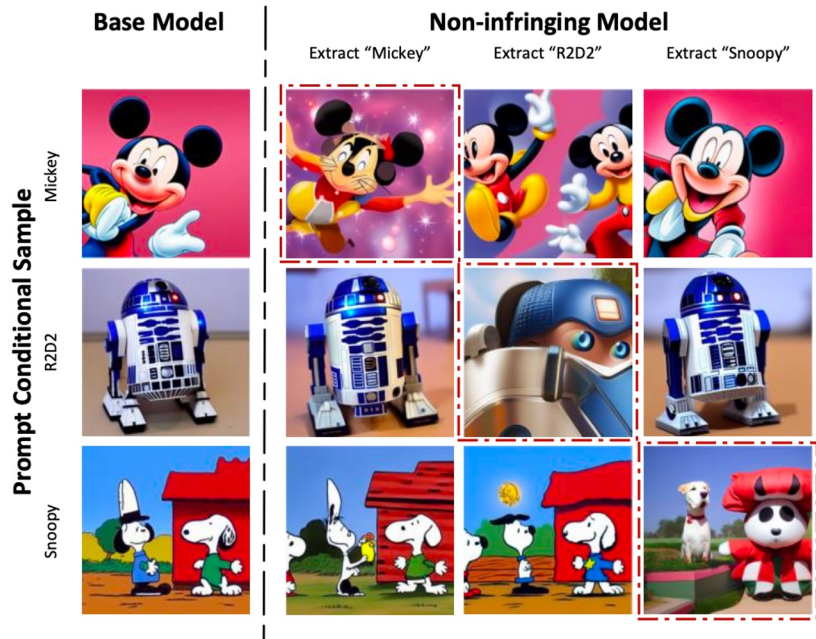


Figure 6: *IP addition within a single image.* We can add ©Plug-in to generate Mickey or Vader in a single image or add combined ©Plug-in to generate both.

- 1) Mickey Mouse
- 2) R2D2
- 3) Snoopy
- 4) Vader

# Limitation

- Search
  - How to manage plug-ins with its growth?
  - How user can find the right plug-in effectively?
- Backward compatibility
  - When the base model is upgraded, the pool of plug-ins need to be retrained, which adds huge cost.
- Performance
  - Non-infringing model may degrade if conducting too many extraction operations, and the influence is not thoroughly evaluated.

# Extracting Training Data from Diffusion Models

*Nicholas Carlini*<sup>\*1</sup>    *Jamie Hayes*<sup>\*2</sup>    *Milad Nasr*<sup>\*1</sup>

*Matthew Jagielski*<sup>+1</sup>    *Vikash Sehwal*<sup>+4</sup>    *Florian Tramèr*<sup>+3</sup>

*Borja Balle*<sup>†2</sup>    *Daphne Ippolito*<sup>†1</sup>    *Eric Wallace*<sup>†5</sup>

<sup>1</sup>Google    <sup>2</sup>DeepMind    <sup>3</sup>ETHZ    <sup>4</sup>Princeton    <sup>5</sup>UC Berkeley

**Tongxuan Tian**  
nua3jz@virginia.edu

## Motivation

- Whether do generative models memorize and regenerate training example

Yes, state-of-the-art diffusion models **do** memorize training samples!



Figure 1: Diffusion models memorize individual training examples and generate them at test time. **Left:** an image from Stable Diffusion’s training set (licensed CC BY-SA 3.0, see [49]). **Right:** a Stable Diffusion generation when prompted with “Ann Graham Lotz”. The reconstruction is nearly identical ( $\ell_2$  distance = 0.031).

## Motivation

- Whether do generative models memorize and regenerate training examples ?  
Yes, state-of-the-art diffusion models **do** memorize training samples!
- ➔ How and why do memorization occur?
- Understanding privacy risks
  - Understanding generalization

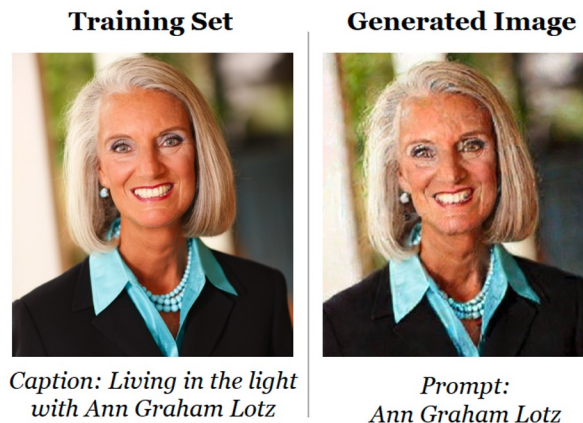
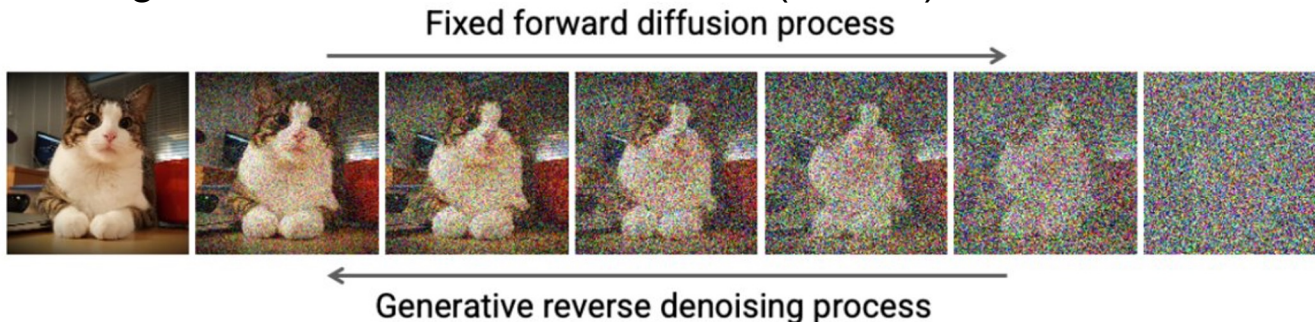


Figure 1: Diffusion models memorize individual training examples and generate them at test time. **Left:** an image from Stable Diffusion’s training set (licensed CC BY-SA 3.0, see [49]). **Right:** a Stable Diffusion generation when prompted with “Ann Graham Lotz”. The reconstruction is nearly identical ( $\ell_2$  distance = 0.031).

## Background

- Diffusion models
  - Denoising Diffusion Probabilistic Models (DDPM)



- Training data privacy attacks
  - **Membership inference attacks:** “Was this example in the training set?”
  - **Inversion attacks:** extract representative examples from a target class
  - **Attribute inference attacks:** reconstruct subsets of attributes of training samples
  - **Extraction attacks:** completely recover training examples

*This paper explores **3 attacks** on **diffusion models**.*

## Threat Model System Overview

- Image-generation systems
  - $x_{gen} \leftarrow Gen(r)$   $r$  is fresh noise
  - $x_{gen} \leftarrow Gen(p; r)$ ,  $p$  is prompt,  $r$  is noise
- Adversary capabilities
  - Black-box adversary on *Stable Diffusion* and *Imagen*
  - White-box adversary on 16 diffusion models trained on CIFAR-10
- Adversary goals
  - Data extraction (Inversion attacks): successfully extract identical image
  - Data reconstruction (Attribute inference attacks): given partial knowledge to recover full image
  - Membership inference (Membership inference attacks): given image  $x$ , infer whether  $x$  is in the training set

## Data Extraction Attacks

- Extracting training data from state-of-the-art diffusion model: **Stable Diffusion** and **Imagen**

### Measurement for Extraction and Memorization

**Definition 1 (( $\ell, \delta$ )-Diffusion Extraction)** [adapted from [11]]. We say that an example  $x$  is extractable from a diffusion model  $f_\theta$  if there exists an efficient algorithm  $\mathcal{A}$  (that does not receive  $x$  as input) such that  $\hat{x} = \mathcal{A}(f_\theta)$  has the property that  $\ell(x, \hat{x}) \leq \delta$ .

$$\ell_2(a, b) = \sqrt{\sum_i (a_i - b_i)^2 / d}$$

$d$  is the dimension of input for normalization

**Definition 2 (( $k, \ell, \delta$ )-Eidetic Memorization)** [adapted from [11]]. We say that an example  $x$  is ( $k, \ell, \delta$ )-Eidetic memorized<sup>2</sup> by a diffusion model if  $x$  is extractable from the diffusion model, and there are at most  $k$  training examples  $\hat{x} \in X$  where  $\ell(x, \hat{x}) \leq \delta$ .



## Data Extraction from Stable Diffusion (Black-box attacks)

- **Preprocessing: Identifying duplicates in the training data to reduce computational cost**
  - **Embedding:** Embed each images to 512 dimension vector using CLIP
  - **Near-duplication:** Search for any training samples that are nearly duplicated with a pixel-level L2 distance below some threshold
  - **Attack:** For each of these near-duplicate images, they use corresponding prompts as input to extraction attack

## Data Extraction from Stable Diffusion (Black-box attacks)

- **Preprocessing: Identifying duplicates in the training data to reduce computational cost**
  - **Embedding:** Embed each images to 512 dimension vector using CLIP
  - **Near-duplication:** Search for any training samples that are nearly duplicated with a pixel-level L2 distance below some threshold
  - **Attack:** For each of these near-duplicate images, they use corresponding prompts as input to extraction attack
- **Extraction**
  - Generating images using selected prompts
  - 500 images for each prompt with different seeds
  - Performing membership inference to get images that appear to be memorized

## Extraction Result for Stable Diffusion

- Compare with training images using definition 1, 94 images are successfully extracted under the threshold 0.15 for L2 distance
- Still 13 images are memorized after human annotation

Original:



Generated:



Figure 3: Examples of the images that we extract from Stable Diffusion v1.4 using random sampling and our membership inference procedure. The top row shows the original images and the bottom row shows our extracted images.

- Imagen is less private than stable diffusion

## Extraction Result for Stable Diffusion

- For 175 million generated images, they will sort them by the mean distance between images in the clique

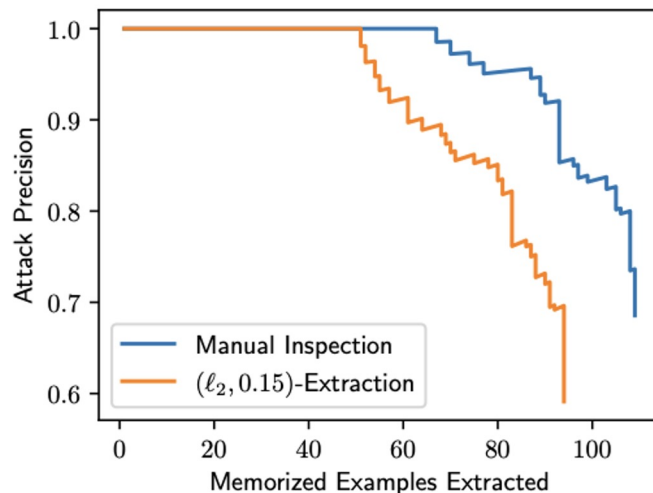


Figure 4: Our attack reliably separates novel generations from memorized training examples, under two definitions of memorization—either  $(\ell_2, 0.15)$ -extraction or manual human inspection of generated images.

# Investigating Memorization

## Experiment Setup

- CIFAR-10 dataset
- 16 diffusion models
- Privacy attacks:
  - Membership inference attacks (class-conditional models)
  - Data reconstruction attacks (inpainting models)

## Membership Inference Attacks

### *White-box attacks*

- The loss threshold attack

Training examples are expected to have **lower loss** than non-training ones.

$l = \mathcal{L}(x; f)$  , reports “member” if  $l < \tau$

- The likelihood Ratio Attack (LiRA)

- First train a collection of *shadow models*
- Compute loss of  $\mathcal{L}(x; f_i)$  under each shadow models
- Losses are split into 2 sets:  $IN = l^{in_i}$  and  $OUT = l^{out_i}$
- In initialization, fitting Gaussians  $N_{IN}$  to  $IN$  and  $N_{out}$  to  $OUT$  set of losses
- For a new model  $f^*$ , compute  $l^* = \mathcal{L}(x; f^*)$  and measure whether  $Pr[l^*|N_{IN}] > Pr[l^*|N_{OUT}]$

## Membership Inference Attack Results

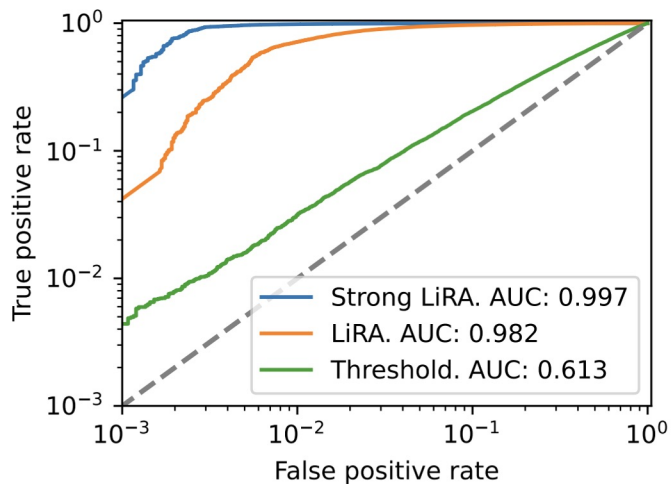


Figure 10: Membership inference ROC curve for a diffusion model trained on CIFAR-10 using the loss threshold attack, baseline LiRA, and “Strong LiRA” with repeated queries and augmentation (§5.2.2).

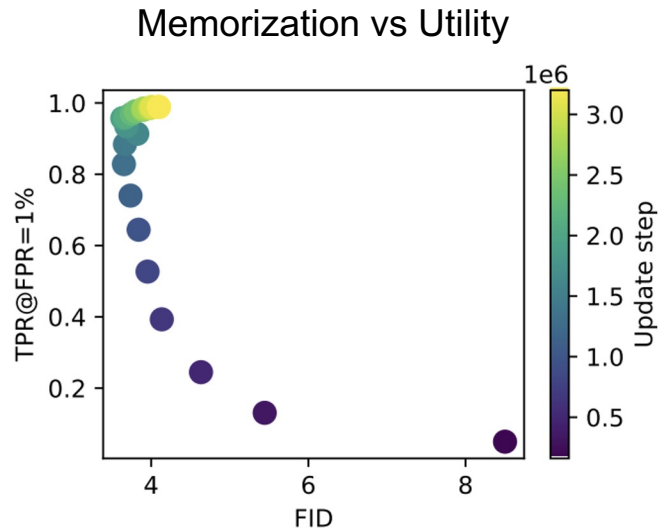


Figure 11: Better diffusion models are more vulnerable to membership inference attacks; evaluating with TPR at an FPR of 1%. As the FID decreases (corresponding to a quality increase) the membership inference attack success rate grows from 7% to nearly 100%.

## Membership Inference Attack Qualitative Results

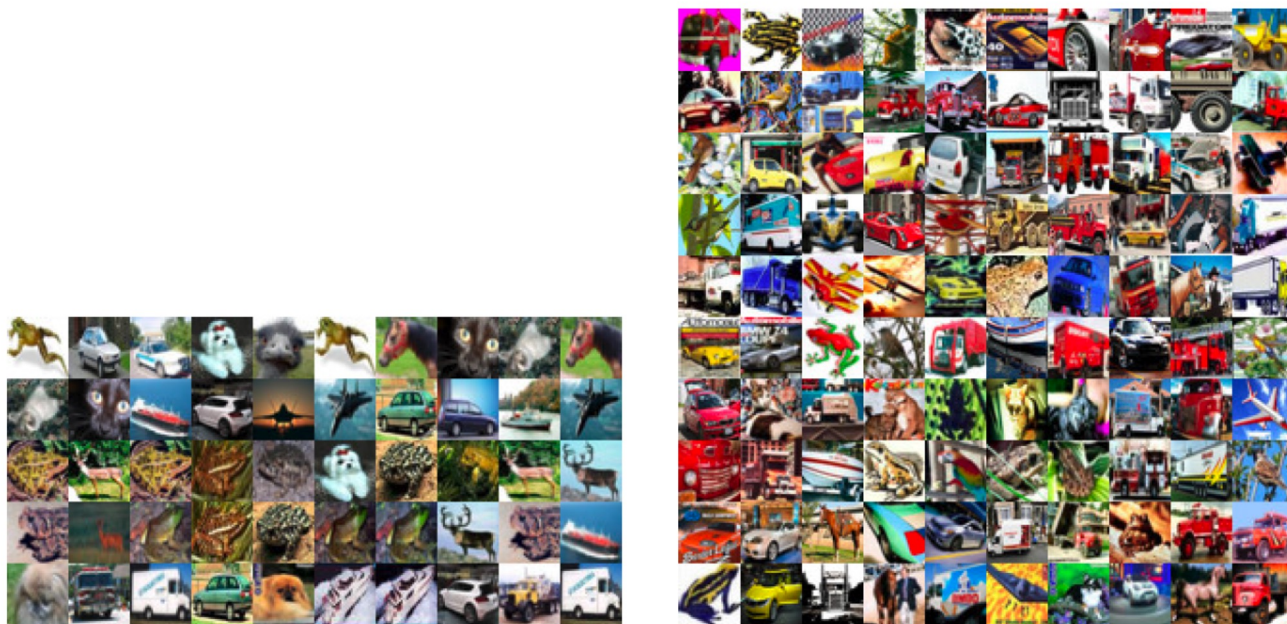
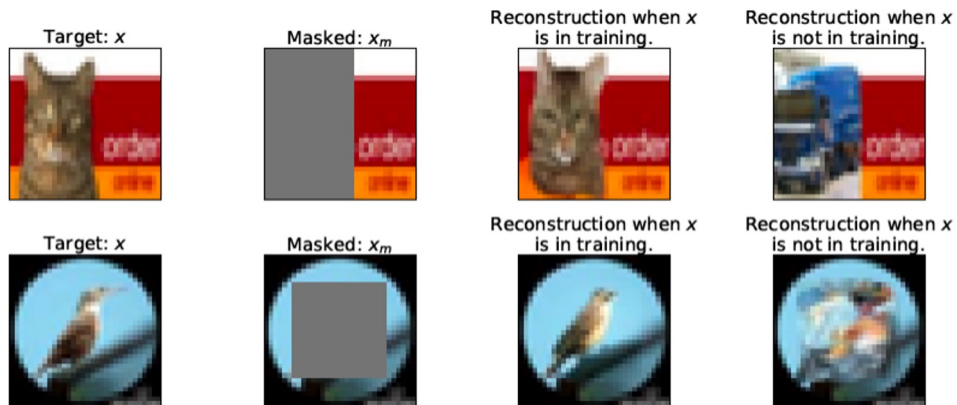


Figure 20: When performing our membership inference attack, the hardest-to-attack examples (left) are all duplicates in the CIFAR-10 training set, and the easiest-to-attack examples (right) are visually outliers from CIFAR-10 images.



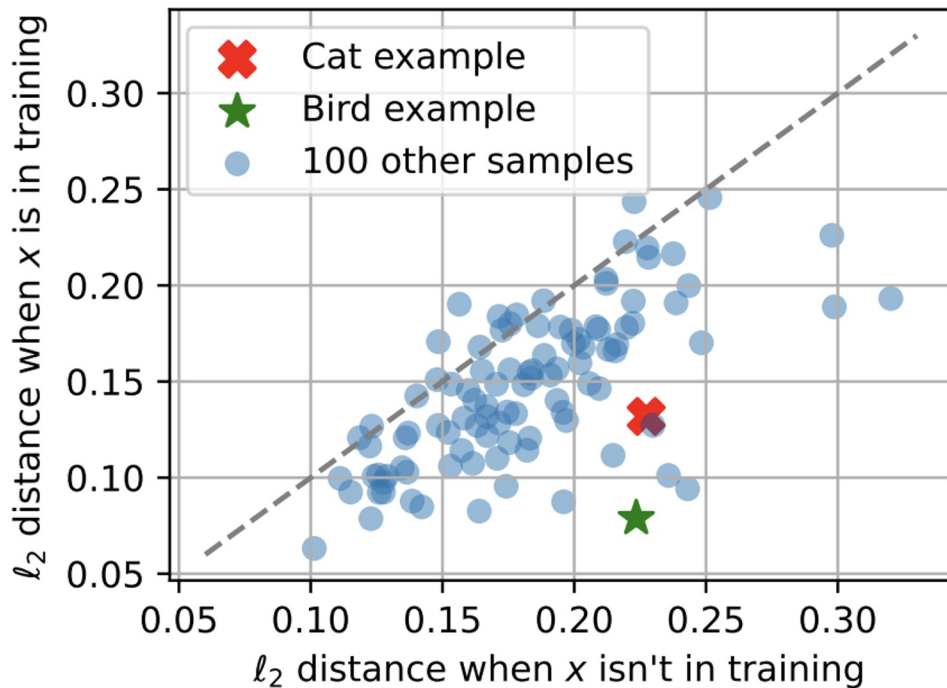
## Inpainting Attacks

- Recover masked region of a image



- Take top-10 scoring reconstruction results for each image

## Inpainting Attacks Result



## Diffusion Models vs GANs

- Data extraction attacks

Architecture	Images Extracted	FID
GANs	StyleGAN-ADA [43]	150 2.9
	DiffBigGAN [82]	57 4.6
	E2GAN [69]	95 11.3
	NDA [63]	70 12.6
	WGAN-ALP [68]	49 13.0
DDPMs	OpenAI-DDPM [52]	301 2.9
	DDPM [33]	232 3.2

Table 1: The number of training images that we extract from different off-the-shelf pretrained generative models out of 1 million unconditional generations. We show GAN models sorted by FID (lower is better) on the top and diffusion models on the bottom. Overall, we find that diffusion models memorize more than GAN models. Moreover, better generative models (lower FID) tend to memorize more data.

## Diffusion Models vs GANs

- Data extraction attacks



(a) StyleGAN



(b) MHGAN

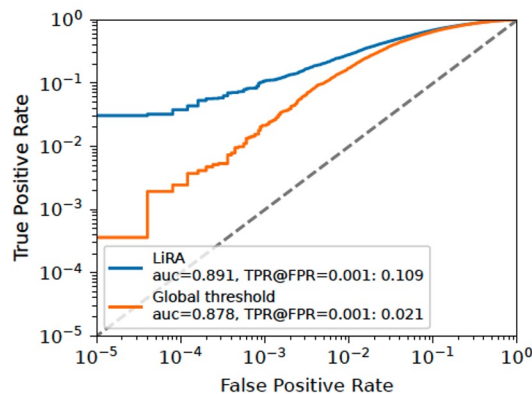


(c) BigGAN

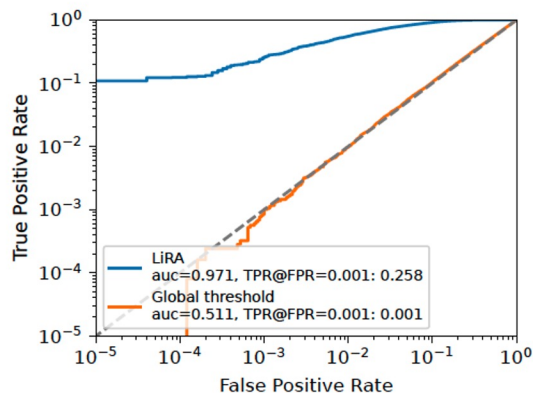
Figure 15: Selected training examples we extract from three GANs trained on CIFAR-10 for different architectures. **Top** row: generated output from a diffusion model. **Bottom** row: nearest ( $l_2$ ) example from the training dataset. Figure 25 in the Appendix contains all unique extracted images.

## Diffusion Models vs GANs

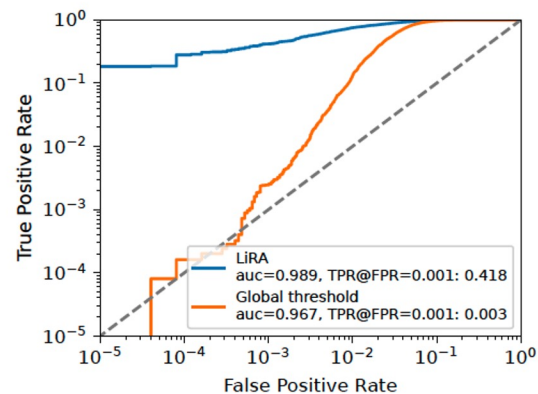
- Membership inference attacks



(a) StyleGAN FID avg = 3.7



(b) MHGAN FID avg = 7.9



(c) BigGAN FID avg = 7.7

Figure 14: Membership inference results on GAN models using the loss threshold and LiRA attacks on the discriminator. Overall, GANs are significantly more private than diffusion models under default training configurations.

## Defenses and Recommendations

- Deduplicating training data
- Differentially-Private Training
  - Differentially-private stochastic gradient descent (DP-SGD)

## Summary

- State-of-the-art diffusion models memorize training images
- Define memorization in diffusion models
- Stronger diffusion models are less private than weaker diffusion models
- Propose attack techniques to help estimate privacy risks of trained models

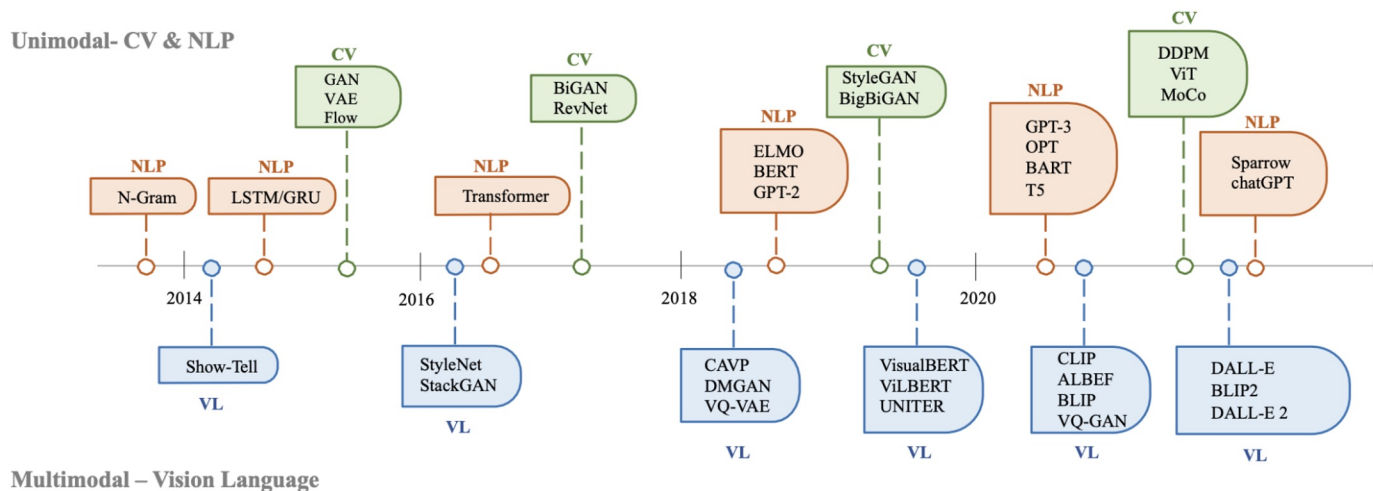
# A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT

Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun

Present by: Ellery (Weifeng) Yu

## Emergence from technical approach:

The transformer architecture, introduced in 2017, has revolutionized AI by becoming the backbone of major generative models in NLP and CV.



Innovations like the Vision Transformer and SwinTransformer have furthered this by adding visual components

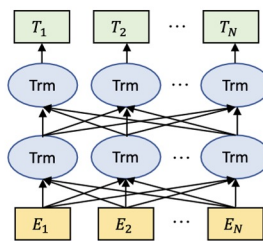


## Foundation Model

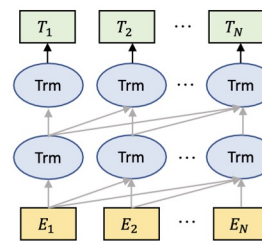
**Transformer:** self-attention mechanism allow the model to attend different parts in a input sentence. For each layer, encoder and decoder consists a multihead attention and a feed forward NN.

### Pretrained model:

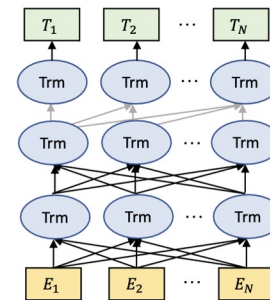
- Encoder Models (Masked Language Models)
- Decoder Models (autoregressive models)



Encoder (BERT)



Decoder (GPT)



Encoder-Decoder (T5/BART)

## Reinforcement Learning from Human Feedback

**Purpose:** To better align AIGC output with human preferences

- Pre-training
- Reward learning
- Fine-tuning with reinforcement learning.

# Computing

## Hardware

### Distributed Training

The training workload is split among multiple processors or machines, allowing the model to be trained much faster.

### Cloud Computing

Service providers let researchers access to powerful computing resources to boost their model training. eg. AWS (Amazon) & Azure (Microsoft)

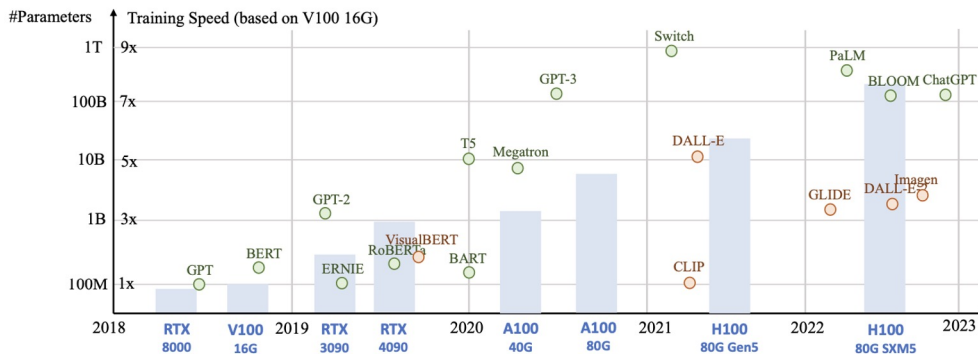
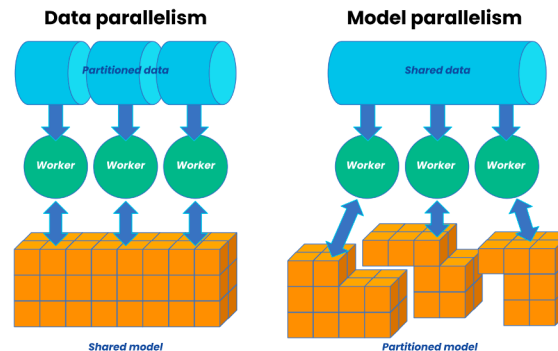


Fig. 5. Statistics of model size [52] and training speed <sup>1</sup> across different models and computing devices.



# Generative AI

## Unimodal Model

### Generative Language Models:

Decoder Models (Autoregressive Models):

Predicting the probability of a masked token given context information

Eg. GPT3, OPT

Encoder Models (Masked Language Models)

Model the probability of the next token given previous tokens

Eg. BERT RoBERTa

Encoder- Decoder Models

Combines transformer-based encoders and decoders together for pre-training.

Eg. T5, BART

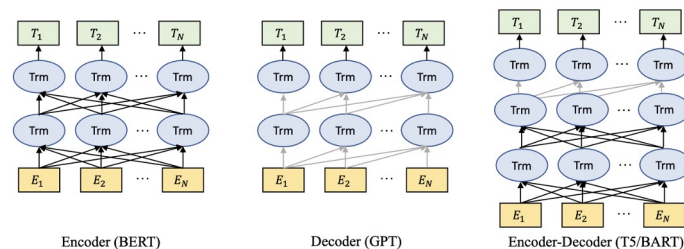
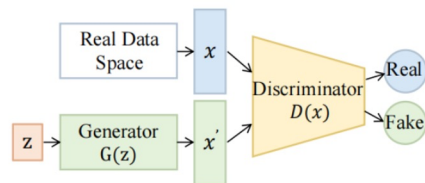


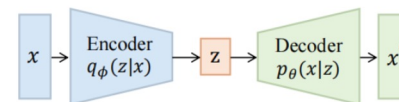
Fig. 4. Categories of pre-trained LLMs. Black line represents information flow in bidirectional models, while gray line represents left-to-right information flow. Encoder models, e.g. BERT, are trained with context-aware objectives. Decoder models, e.g. GPT, are trained with autoregressive objectives. Encoder-decoder models, e.g. T5 and BART, combines the two, which use context-aware structures as encoders and left-to-right structures as decoders.

## Vision Generative Models

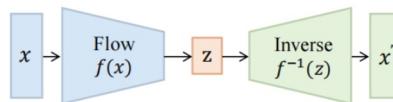
- GANs
- VAEs
- Flow
- Diffusion



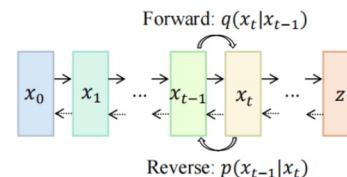
(1) Generative adversarial networks



(2) Variational autoencoders



(3) Normalizing flows



(4) Diffusion models

Fig. 7. Categories of vision generative models.

# GANs

## LAPGAN (Laplacian Pyramid GAN):

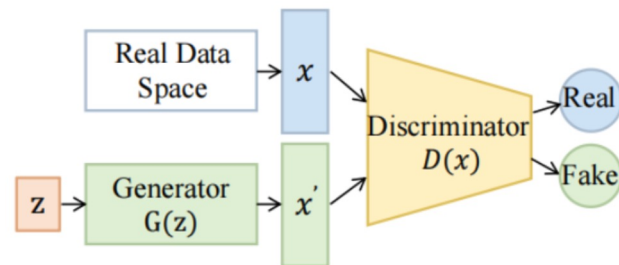
- Utilizes a cascade of convolutional networks.
- Generates high-quality images through a coarse-to-fine approach.
- Enhances detail at each level of the image pyramid.

## DCGAN (Deep Convolutional GAN):

- Employs architectural constraints for more stable training.
- Simplifies and stabilizes the structure of convolutional networks.
- Pioneered features like strided convolutions and batch normalization in GANs.

## BigGAN:

- Known for high-resolution and diverse image synthesis.
- Implements large scale models and improved training dynamics.
- Uses class-conditional generation to produce highly detailed images.

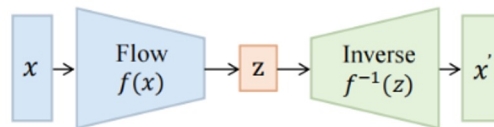


(1) Generative adversarial networks

## Flows

A Normalizing Flow is a distribution transformation from simple to complex by a sequence of invertible and differentiable mappings.

- Coupling and autoregressive flows
  - Multi-scale flows
- Convolutional and Residual Flows.
  - ConvFlow
  - RevNets
  - iRevNets

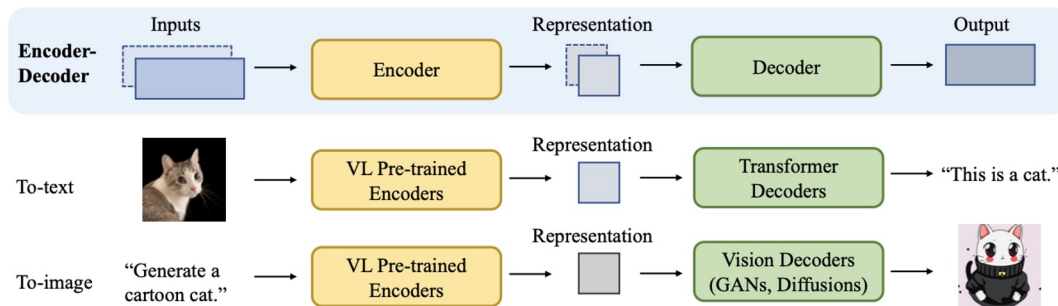


(3) Normalizing flows

# Multimodal Models

## Vision Language Generation

**Core** :Encoder-decoder architecture.



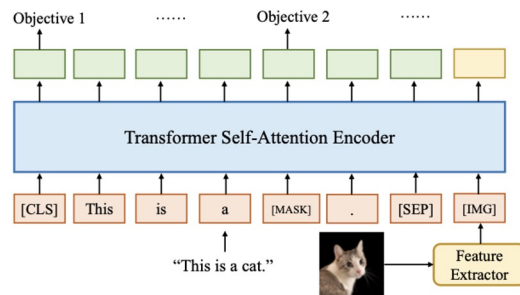
**Encoder** is responsible for learning a contextualized representation of the input data.

**Decoder** is used to generate raw modalities that reflect cross-modal interactions, structure, and coherence in the representation

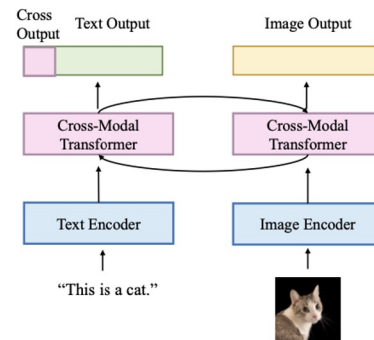


## Vision Language Encoders

- Concatenated encoders: concatenating the embeddings from single encoders



(a) Concatenated Encoder



(b) Cross-aligned Encoder

- Cross-aligned encoders: learn contextualized representations by looking at pairwise interactions between modalities.

## Vision Language Decoders

- To text decoders: Jointly- trained decoders, frozen decoders.

- To image decoders:
  - GAN-based,
  - Diffusion-based: GLIDE, Imagen
  - VAE-based: DALL-E

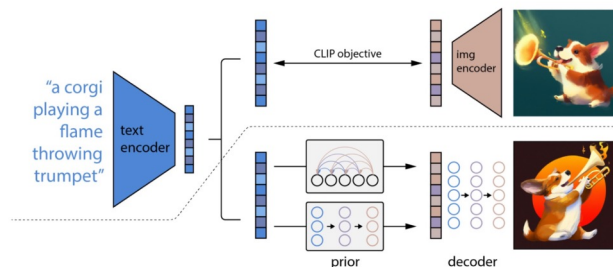


Fig. 11. The model structure of DALL-E-2. Above the dotted line is the CLIP pre-training process, which aims to align the vision and language modalities. And below the dotted line is the image generation process. The text encoder accepts an instruction and encodes it into a representation, then the prior network and diffusion model decodes this representation to generate the final output.

## Other modalities generation

- Text-audio
- Text-Graph
- Text-Code

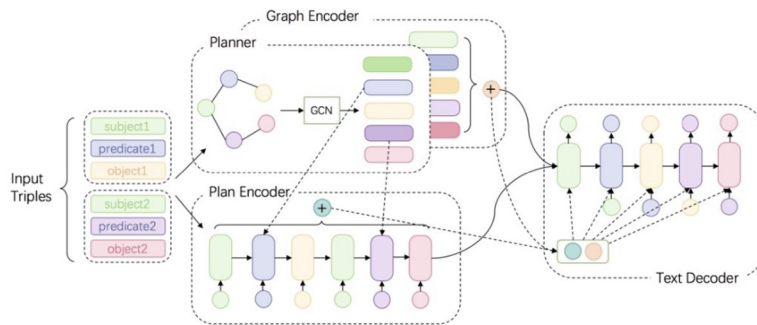


Fig. 12. DUALENC [175]: a KG-to-text generation model that bridges the structural gap between KG and graph via dual-encoding.

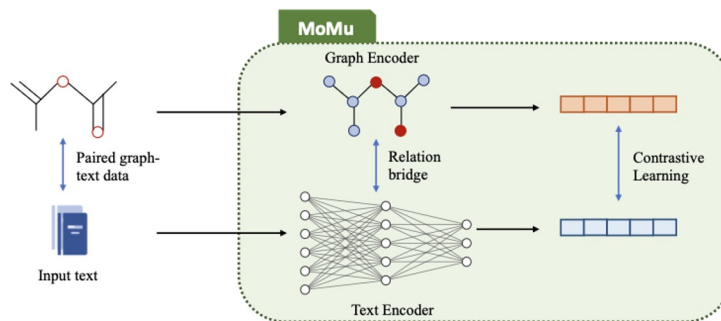
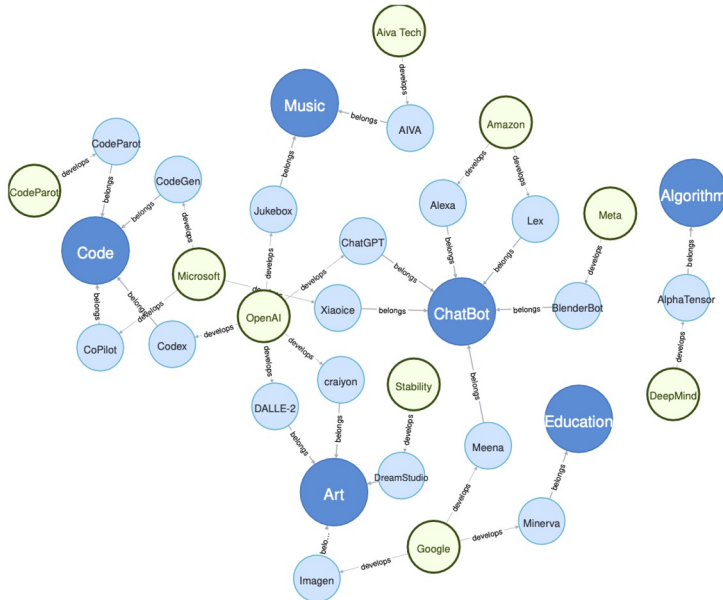


Fig. 13. MoMu [188]: A cross-modal text-molecule generation model.

# Application



Application	Platform/Software	Company	Year	Papaer	Link
ChatBot	Xiaoice	Microsoft	2018	[200]	Xiaoice
ChatBot	Meena	Google	2020	[201]	Meena Blog
ChatBot	BlenderBot	Meta	2022	[202]	Blenderbot
ChatBot	ChatGPT	OpenAI	2022	[10]	ChatGPT
ChatBot	Alexa	Amazon	2014	-	Amazon Alexa
ChatBot	Lex	Amazon	2017	-	Amazon Lex
Music	AIVA	Aiva Tech	2016	-	AIVA
Music	Jukebox	OpenAI	2020	[203]	Jukebox
Code	CodeGPT	Microsoft	2021	[204]	CodeGPT
Code	CodeParrot	CodeParrot	2022	[205]	CodeParrot
Code	Codex	OpenAI	2021	[206]	Codex blog
Code	CoPilot	Microsoft	2021	[206]	CoPilot
Art	DALL-E-2	OpenAI	2022	[5]	DALL-E-2 Blog
Art	DreamStudio	Stability	2022	[13]	Dreamstudio
Art	craiyon	OpenAI	2021	[1]	Craiyon
Art	Imagen	Google	2022	[152]	Imagen
Education	Minerva	Google	2022	[207]	Minerva Blog
Algorithm	AlphaTensor	DeepMind	2022	[208]	AlphaTensor

Table 1. Applications of Generative AI models.

Fig. 14. A relation graph of a current research areas, applications and related companies, where dark blue circles represent research areas, light blue circles represent applications and green circles represents companies.

## Efficiency

- **Inference efficiency:** This is concerned with the practical considerations of deploying a model for inference, i.e., computing the model's outputs for a given input. Inference efficiency is mostly related to the model's size, speed, and resource consumption (e.g., disk and RAM usage) during inference.
- **Training efficiency:** This covers factors that affect the speed and resource requirements of training a model, such as training time, memory footprint, and scalability across multiple

## Future Directions

- High-stakes Applications
- Specialization and Generalization
- Continual Learning and Retraining
- Reasoning
- Scaling up
- Social issue

Thanks!