Applications of Diffusion Models

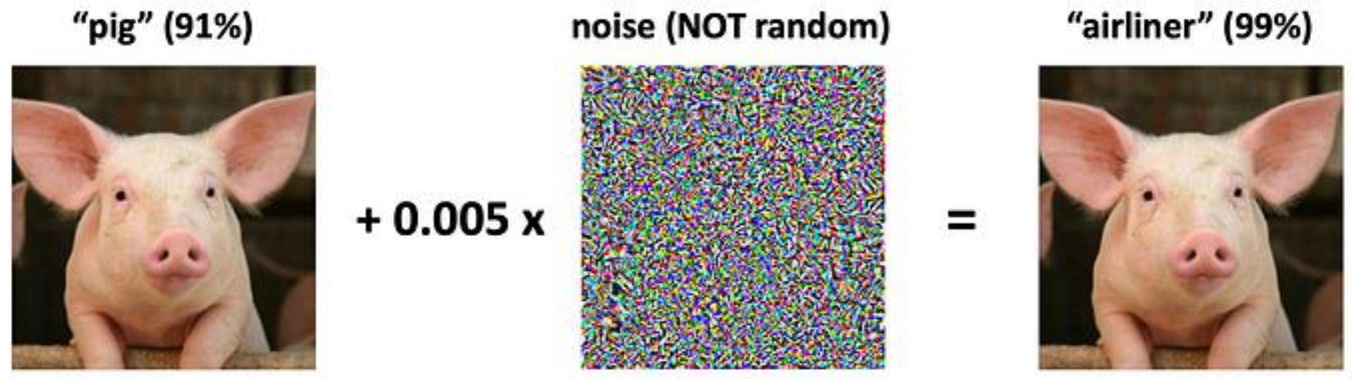
Jian Li

IIIS, Tsinghua

ATCS 2023

Adversarial Robustness
and Score-based generative model

A Brief Introduction to Adversarial Robustness



Adversarial Training

- Training adversarially robust classifiers

$$\underset{\theta}{\text{minimize}}\, \hat{R}_{\text{adv}}(h_\theta, D_{\text{train}}) \equiv \underset{\theta}{\text{minimize}}\, \frac{1}{|D_{\text{train}}|} \sum_{(x,y)\in D_{\text{train}}} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

- we would repeatedly choose a minibatch B⊆D$_{\text{train}}$, and update θ according to its gradient

$$\theta := \theta - \frac{\alpha}{|B|} \sum_{(x,y)\in B} \nabla_\theta \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

- By Daskin theorem

$$\nabla_\theta \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y) = \nabla_\theta \ell(h_\theta(x+\delta^\star)), y)$$

$$\delta^\star = \underset{\delta\in\Delta(x)}{\text{argmax}}\, \ell(h_\theta(x+\delta)), y)$$

https://adversarial-ml-tutorial.org/introduction/

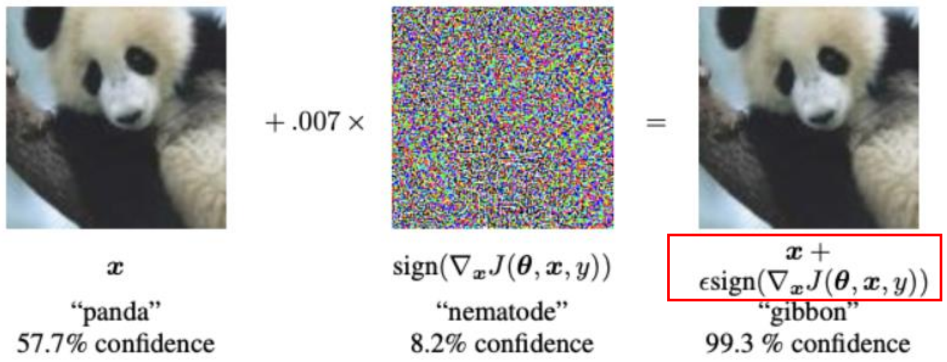
Adversarial examples

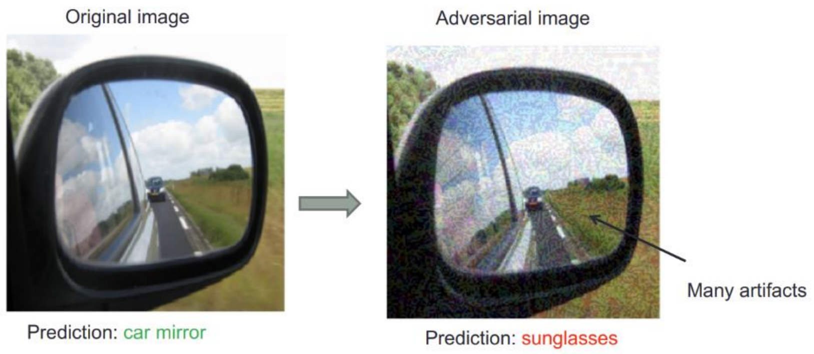
- Solving the inner maximization problem (adversarial attack):

$$\underset{\|\delta\|\leq\epsilon}{\text{maximize}}\,\ell(h_\theta(x),y)$$

- The Fast Gradient Sign Method (FGSM)
  - First compute the gradient g at x (use BP, gradient w.r.t. x)
  - In order to maximize loss, we want to adjust delta in the direction of this gradient, and project to epsilon l_infinity ball

$\delta := \operatorname{clip}(\alpha g, [-\epsilon, \epsilon]).$ or $\delta := \epsilon \cdot \operatorname{sign}(g).$


$+.007\times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}}J(\boldsymbol{\theta},\boldsymbol{x},y))$
"nematode"
8.2% confidence
$\boldsymbol{x}+\epsilon\text{sign}(\nabla_{\boldsymbol{x}}J(\boldsymbol{\theta},\boldsymbol{x},y))$
"gibbon"
99.3 % confidence



  - Sometimes, FGSM requires large eps in order to succeed (human-perceptible)

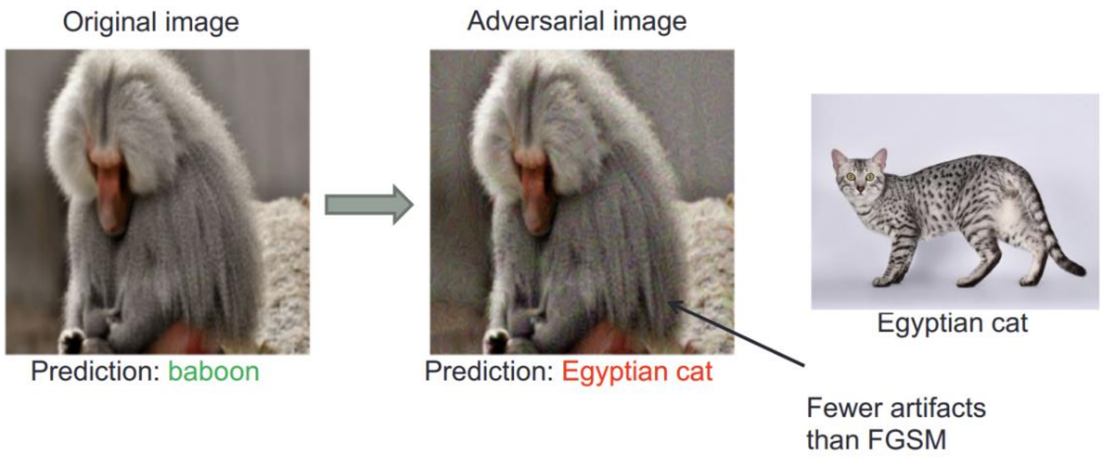
Adversarial examples

Projected gradient (steepest) descent (PGD)

(also known as I-FGSM which expands for Iterative-Fast Gradient Sign Method)

$$x^t = \text{clip}_{(-\epsilon,\epsilon)}\left(x^{t-1} + \gamma \cdot \text{sign}\left(\nabla_x \mathcal{L}(C(x^{t-1}, w), y)\right)\right)$$

- for those pixels with perturbation size larger than eps, “clip” truncates it to eps
- Another difference: PGD uses random initialization, by adding random noise to the original image from a uniform distribution in the range (-eps,eps)



Other powerful attacks: Carlini and Wagner attack (C&W Attack), One-pixel Attack

https://hongyanz.github.io/slides/cs886_s22_Lecture3.pdf

Adversarial Training

$$\nabla_\theta \max_{\delta \in \Delta(x)} \ell(h_\theta(x+\delta)), y) = \nabla_\theta \ell(h_\theta(x+\delta^\star)), y)$$

$$\delta^\star = \operatorname*{argmax}_{\delta \in \Delta(x)} \ell(h_\theta(x+\delta)), y)$$

- Danskin's theorem only technically applies to the case where we are able to compute the maximum exactly
- *The key aspects of adversarial training is incorporate a strong attack into the inner maximization procedure*
- Projected gradient descent is one of the strongest attacks

```
Repeat:
  1. Select minibatch $B$, initialize gradient vector $g := 0$
  2. For each $(x, y)$ in $B$:
     a. Find an attack perturbation $\delta^\star$ by (approximately) optimizing
            $\delta^\star = \operatorname*{argmax}_{\|\delta\| \leq \epsilon} \ell(h_\theta(x+\delta), y)$
     b. Add gradient at $\delta^\star$
            $g := g + \nabla_\theta \ell(h_\theta(x+\delta^\star), y)$
  3. Update parameters $\theta$
            $\theta := \theta - \frac{\alpha}{|B|} g$
```

In practice, use PGD to find the inner maximizer

Adversarial Purification:

Another approach for the adversarial defense is to purify attacked images before feeding them to classifiers

Adversarial Purification

- Learn a purification model whose goal is to remove any existing adversarial noise from potentially attacked images into clean images so that they could be correctly classified when fed to the classifier.
- The purification model is usually trained independently of the classifier
- The most common way is for adversarial purification to learn a generative model:

Generate clean images from attacked images.

High level idea

- Adversarial purification can be understood as a denoising procedure
- Learning scores in diffusion models is equivalent to learning the noise (denoising)
- Recall the objective for learning the score network s

$$\mathcal{L}(\theta, \{\sigma_j\}_{j=1}^{L}) = \sum_{j=1}^{L} \sigma_j^2 \ell(\theta, \sigma_j), \qquad \ell(\theta, \sigma) = \mathbb{E}_{q(\tilde{x}|x)p_{\text{data}}(x)} \left[\frac{1}{2} \left\| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q(\tilde{x}|x) \right\|^2 \right].$$

- Purification (deterministic update):

$$x_0 = x' + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$
$$x_t = x_{t-1} + \alpha_{t-1} s_\theta(x_{t-1}).$$

- Since the norm of v is bounded due to the perceptual indistinguishability constraint, the added noise ε can “screen out” the relatively small perturbation v.
- The score network s is trained to denoise images perturbed by Gaussian noises. Adding Gaussian noises makes x_0 more similar to the data used to train the score network.

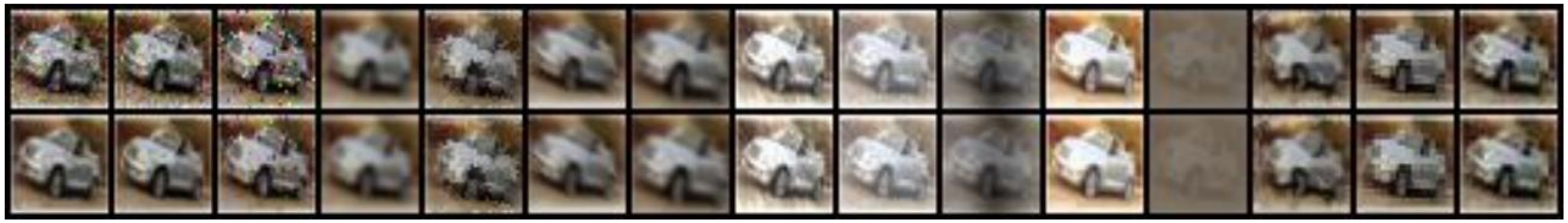


Figure 5. Examples of corrupted and purified images. From left: {Gaussian, shot, impulse} noise, {Defocus, glass, motion, zoom} blur, {snow, frost, fog, brightness} weather, {contrast, elastic, pixelate, JPEG} digital corruptions.

DiffPure

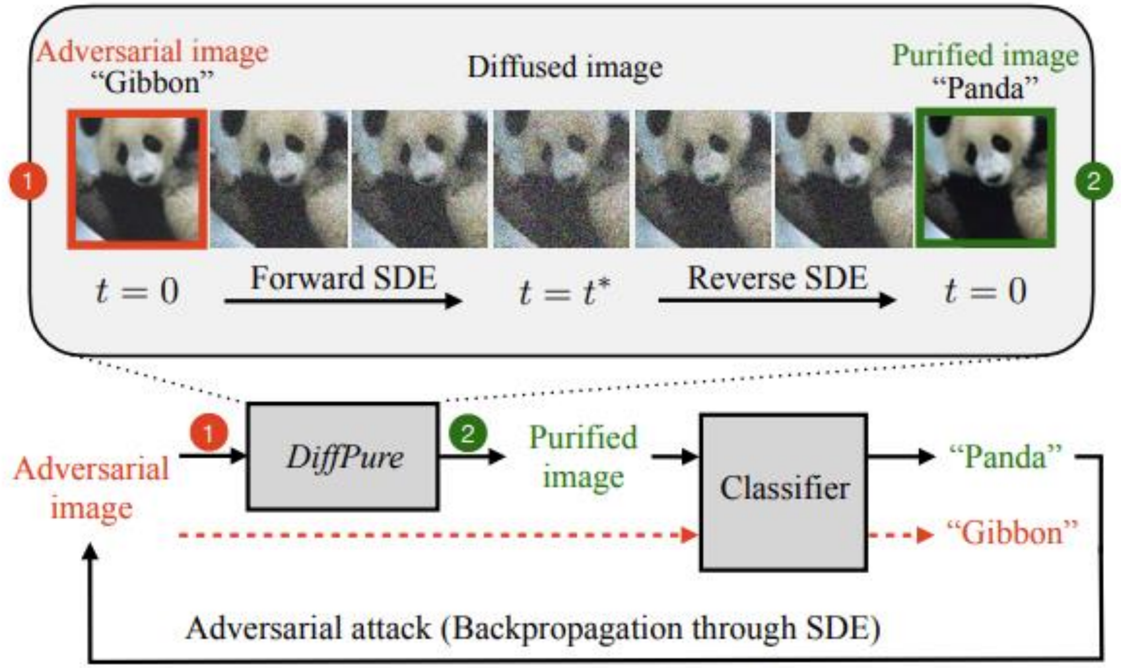
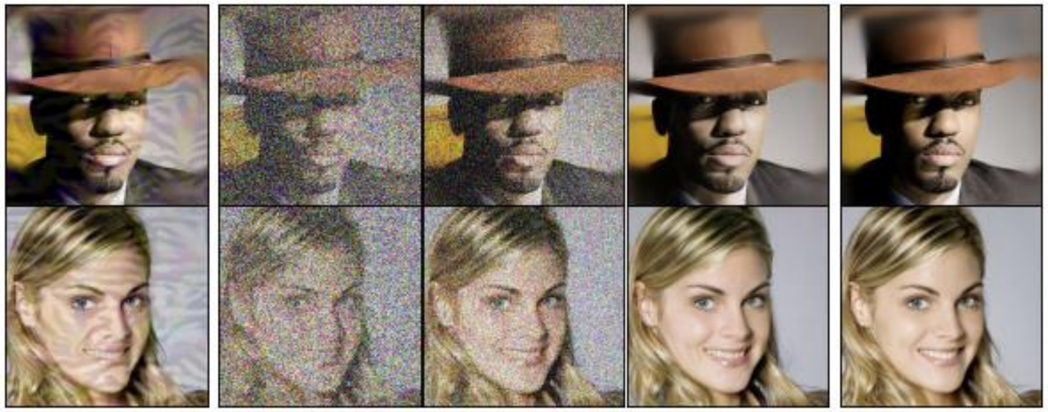
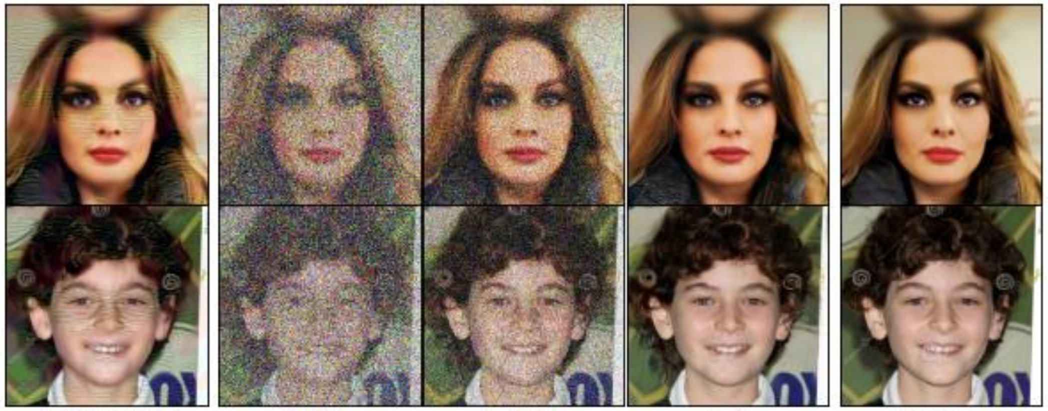


Figure 1. An illustration of *DiffPure*. Given a pre-trained diffusion model, we add noise to adversarial images following the forward diffusion process with a small diffusion timestep t^* to get diffused images, from which we recover clean images through the reverse denoising process before classification. Adaptive attacks backpropagate through the SDE to get full gradients of our defense system.



(a) Smiling



(b) Eyeglasses

Figure 2. Our method purifies adversarial examples (first column) produced by attacking attribute classifiers using PGD ℓ_∞ ($\epsilon = 16/255$), where $t^* = 0.3$. The middle three columns show the results of the SDE in Eq. (4) at different timesteps, and we observe the purified images at t=0 match the clean images (last column). Better zoom in to see how we remove adversarial perturbations.

Table 4. Standard accuracy and robust accuracies against unseen threat models on ResNet-50 for CIFAR-10. We keep the same evaluation settings with (Laidlaw et al., 2021), where the attack bounds are $\epsilon = 8/255$ for AutoAttack ℓ_∞, $\epsilon = 1$ for AutoAttack ℓ_2, and $\epsilon = 0.05$ for StAdv. The baseline results are reported from the respective papers. For our method, the diffusion timestep is $t^* = 0.125$.

Method	Standard Acc	Robust Acc		
		ℓ_∞	ℓ_2	StAdv
Adv. Training with ℓ_∞ (Laidlaw et al., 2021)	86.8	49.0	19.2	4.8
Adv. Training with ℓ_2 (Laidlaw et al., 2021)	85.0	39.5	47.8	7.8
Adv. Training with StAdv (Laidlaw et al., 2021)	86.2	0.1	0.2	53.9
PAT-self (Laidlaw et al., 2021)	82.4	30.2	34.9	46.4
Adv. Craig (Dolatabadi et al., 2021)	83.2	40.0	33.9	49.6
Adv. GradMatch (Dolatabadi et al., 2021)	83.1	39.2	34.1	48.9
Ours	**88.2±0.8**	**70.0±1.2**	**70.9±0.6**	**55.0±0.7**

StAdv

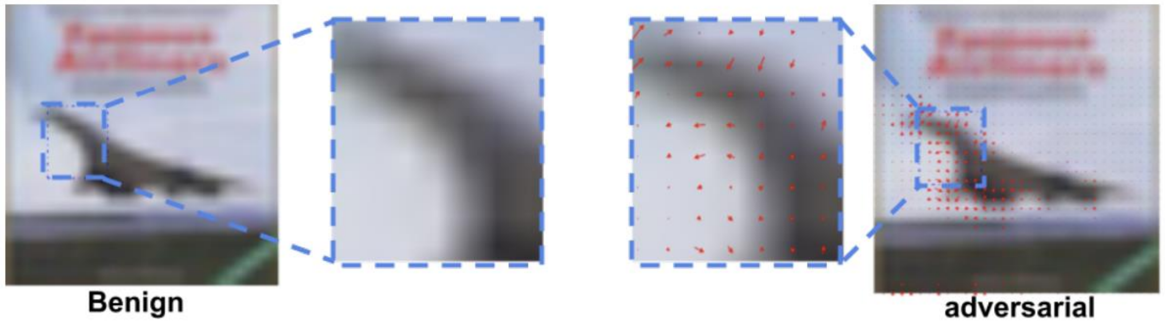


Figure 6: Flow visualization on CIFAR-10. The example is misclassified as bird.

High-resolution image reconstruction with latent diffusion models from human brain activity

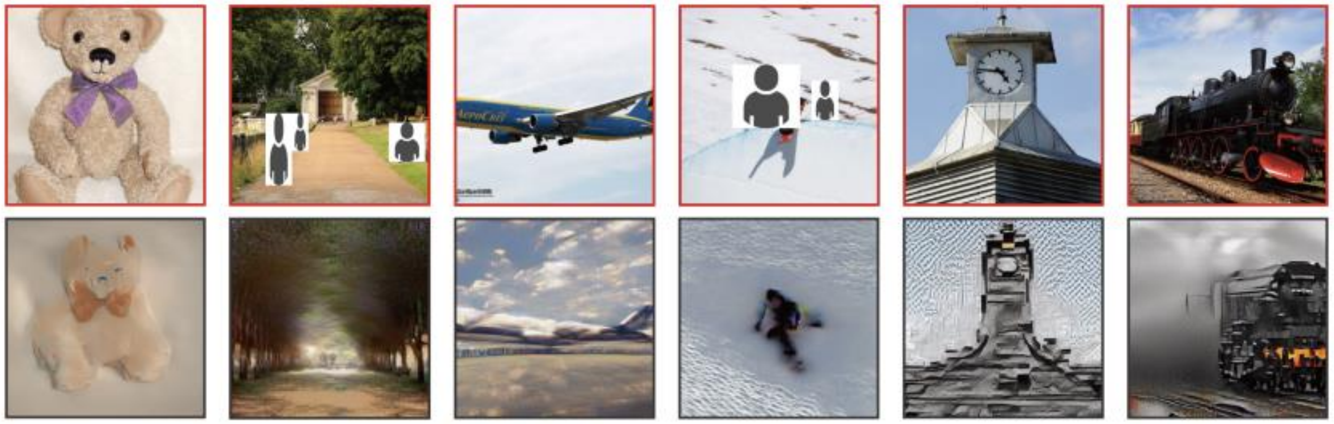


Figure 1. Presented images (red box, top row) and images reconstructed from fMRI signals (gray box, bottom row) for one subject (subj01).

https://sites.google.com/view/stablediffusion-with-brain/home

Motivations

- A new method based on a diffusion model (DM) to reconstruct images from human brain activity obtained via functional magnetic resonance imaging (fMRI)

- Reconstructing visual images from fMRI is challenging:
  - The underlying representations in the brain are largely unknown
  - The sample size typically associated with brain data is relatively small
  - Low signal-to-noise ratio with fMRI data

- Overarching goal:
  - use DMs for high resolution visual reconstruction
  - use brain encoding framework to better understand the underlying mechanisms of DMs and its correspondence to the brain.

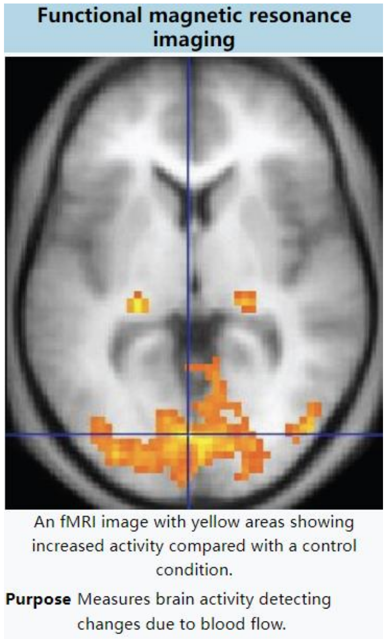
fMRI

fMRI measures brain activity by detecting changes associated with blood flow.

When an area of the brain is in use, blood flow to that region also increases

Noises in fMRI:

1. thermal noise
2. system noise
3. physiological noise
4. random neural activity
5. differences across people and across tasks within a person

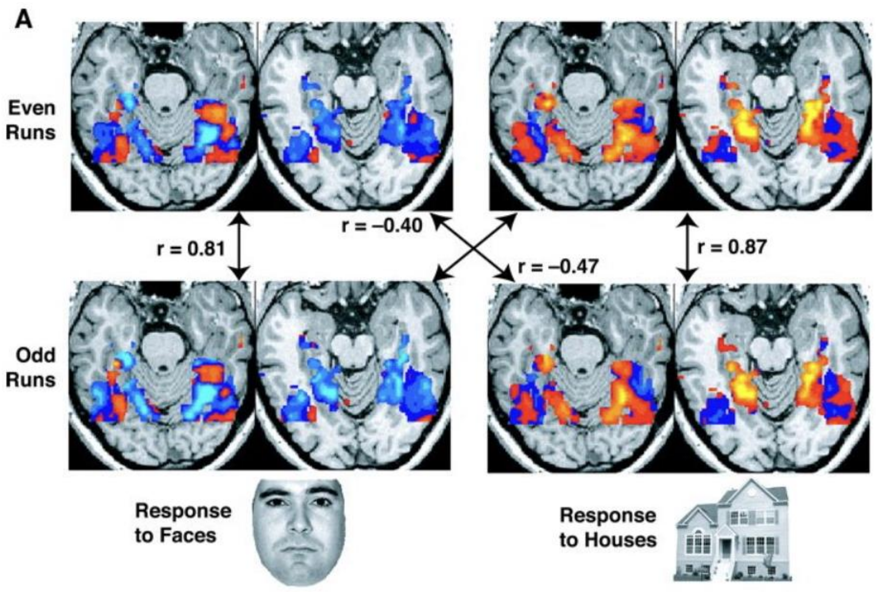


An fMRI image with yellow areas showing increased activity compared with a control condition.

Purpose Measures brain activity detecting changes due to blood flow.

Nuffield Department of Clini…

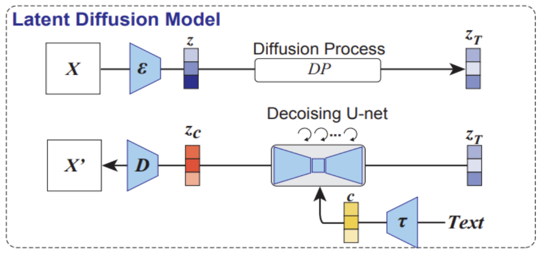
Introduction to FMRI —…



Data

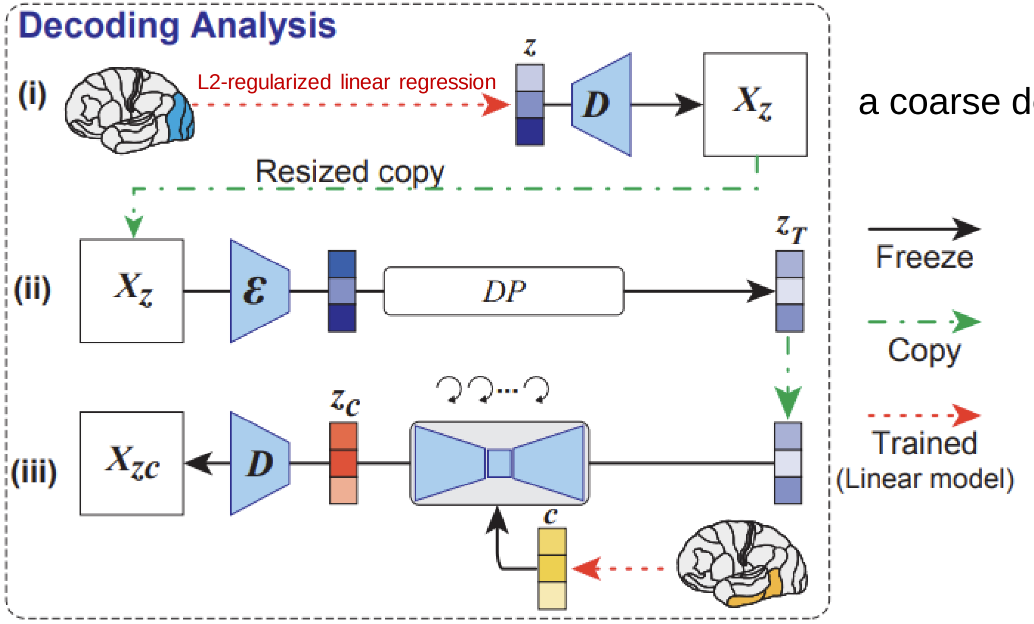
- Natural Scenes Dataset (NSD)
- NSD provides data acquired from a 7-Tesla fMRI scanner over 30–40 sessions during which each subject viewed three repetitions of 10,000 images.
- The images used in the NSD experiments were retrieved from MS COCO and cropped to 425*425

Framework



The only training required in our method is to construct linear models that map fMRI signals to each LDM component

z as the latent representation of the original image compressed by the autoencoder



a coarse decoded image Xz

To construct models from fMRI to the components of LDM, we used L2-regularized linear regression, and all models were built on a per subject basis

Text input is projected to a fixed latent representation by a pretrained text encoder (CLIP)

We decoded latent representations of the presented image (z) and associated text c from fMRI signals within early (blue) and higher (yellow) visual cortices, respectively.

These latent representations were used as input to produce a reconstructed image Xzc.

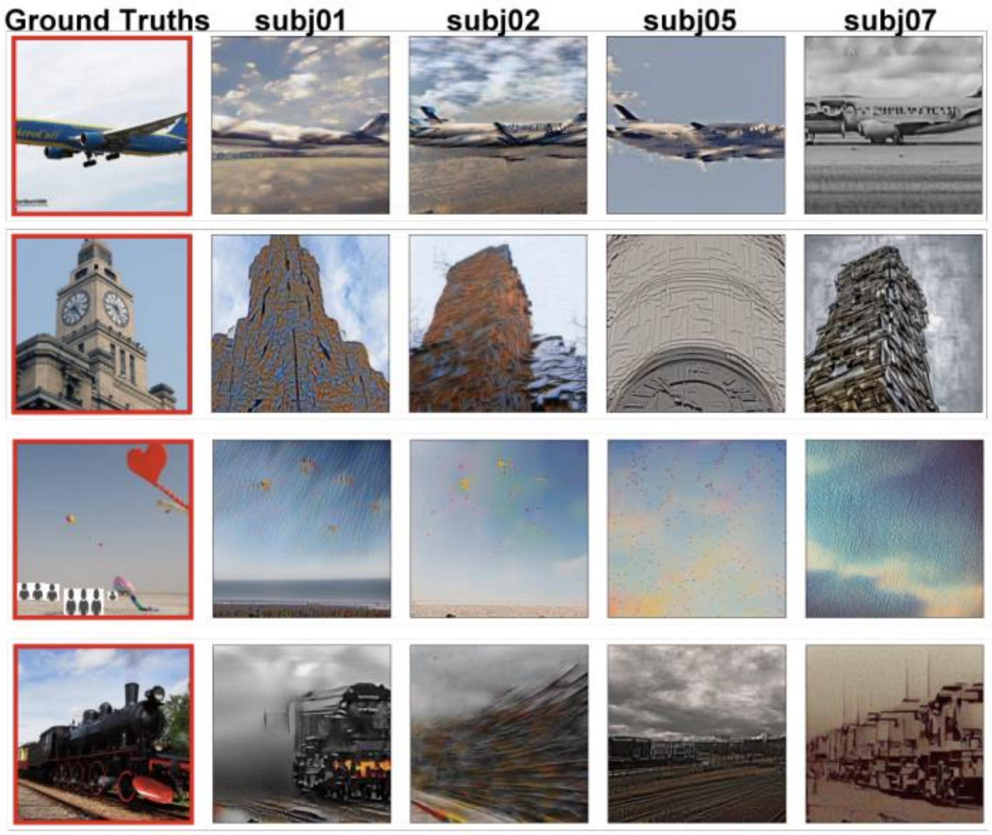
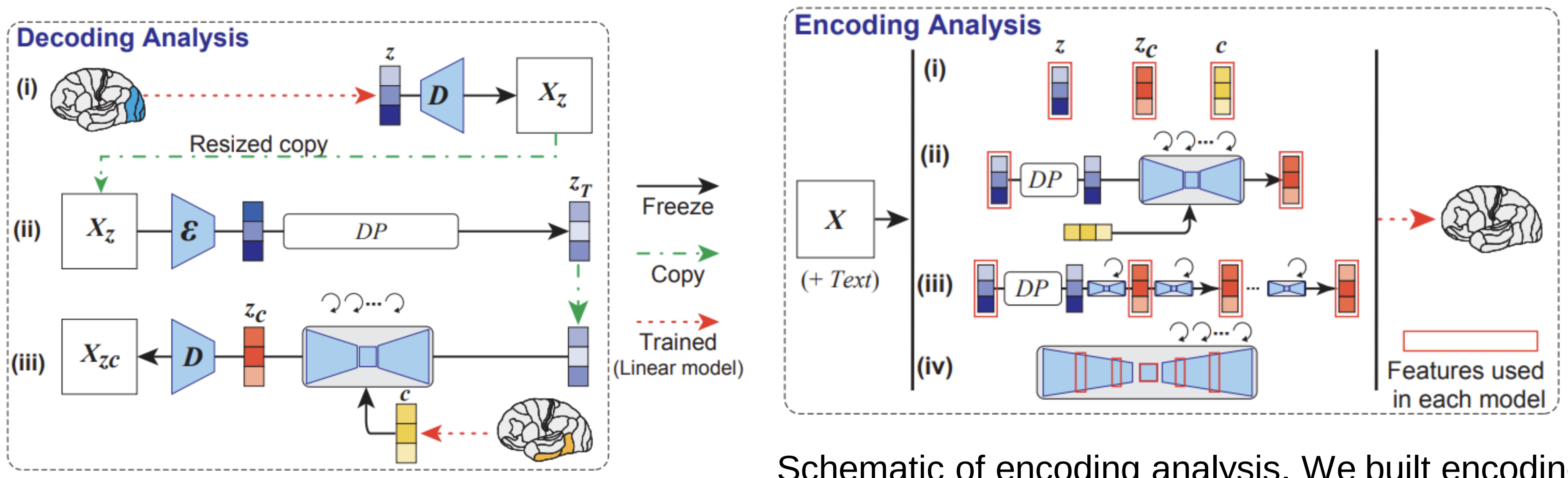


Figure 4. Example results for all four subjects.

Encoding: Whole-brain Voxel-wise Modeling

Encoding: Whole-brain Voxel-wise Modeling
map representations in LDM to brain activity



Schematic of encoding analysis. We built encoding models to predict fMRI signals from different components of LDM, including z, c, and zc.

(i) linear models to predict voxel activity from latent representations independently: z, c, and zc
(ii) we incorporated them into a single model
(iii) examine how zc changes through the denoising process.
(iv) we extracted features from different layers of U-Net

z produced high prediction performance in the posterior part of visual cortex, namely early visual cortex

c produced the highest prediction performance in higher visual cortex.

zc carries a representation that is very similar to z, showing high prediction performance for early visual cortex

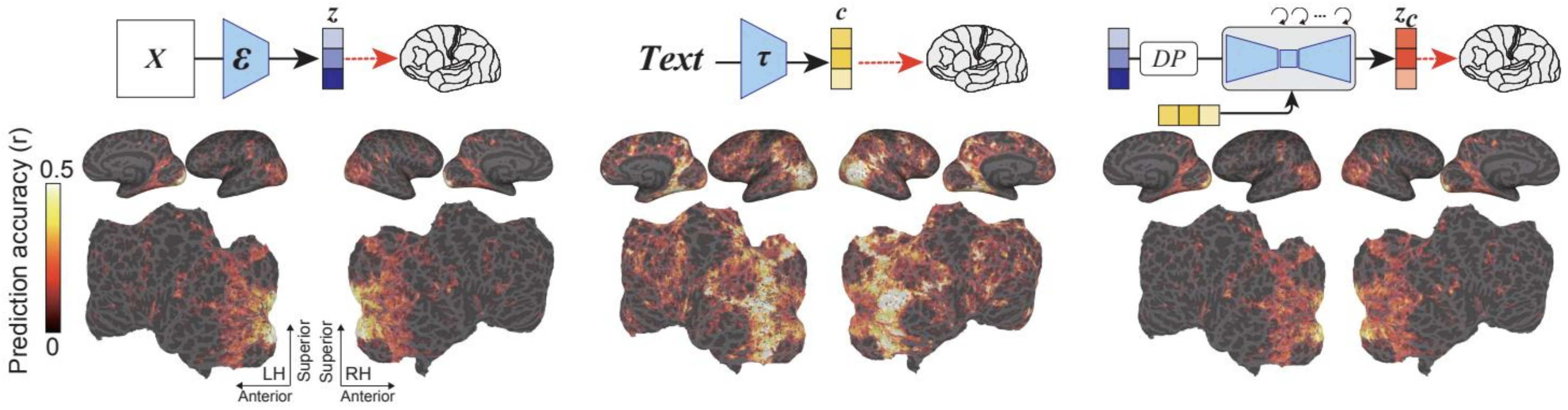


Figure 6. Prediction performance (measured using Pearson's correlation coefficients) for the voxel-wise encoding model applied to held-out test images in a single subject (subj01), projected onto the inflated (top, lateral and medial views) and flattened cortical surface (bottom, occipital areas are at the center), for both left and right hemispheres. Brain regions with significant accuracy are colored (all colored voxels $P < 0.05$, FDR corrected).

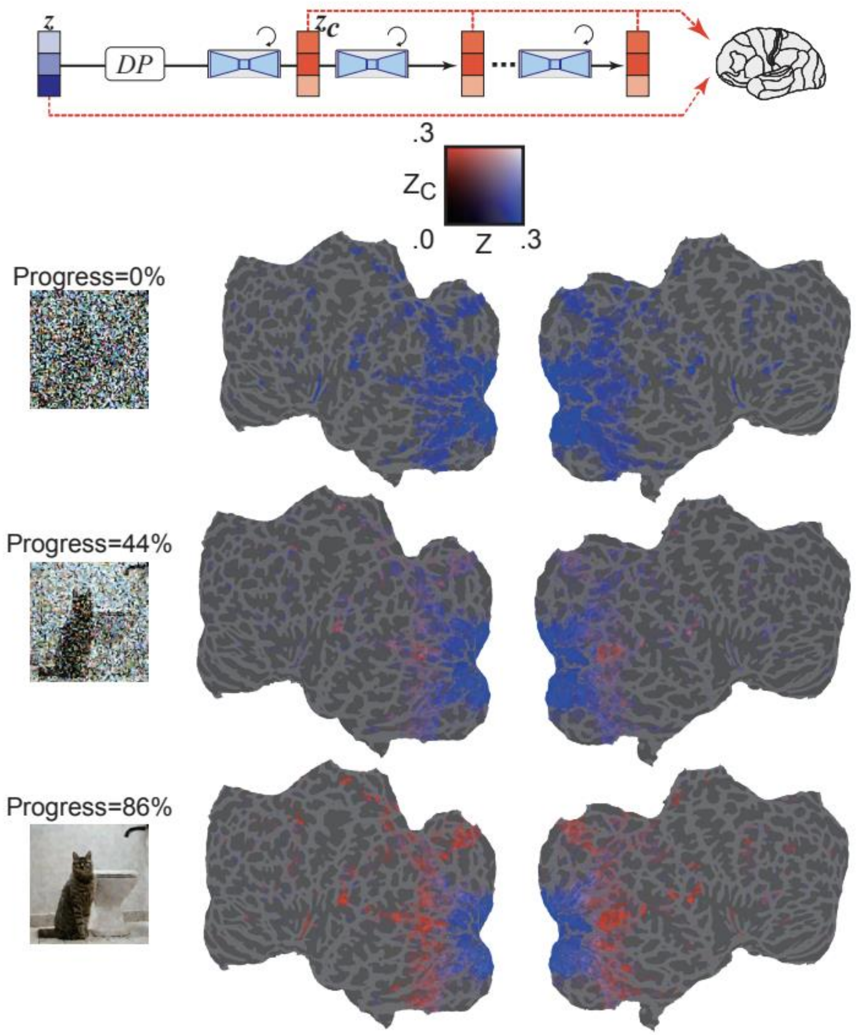


Figure 8. Unique variance accounted for by $\mathbf{z_c}$ compared with $\mathbf{z}$ in one subject (subj01), obtained by splitting accuracy values from the combined model. While fixing $\mathbf{z}$, we used $\mathbf{z_c}$ with different denoising stages from early (top) to late (bottom) steps. All colored voxels $P < 0.05$, FDR corrected.

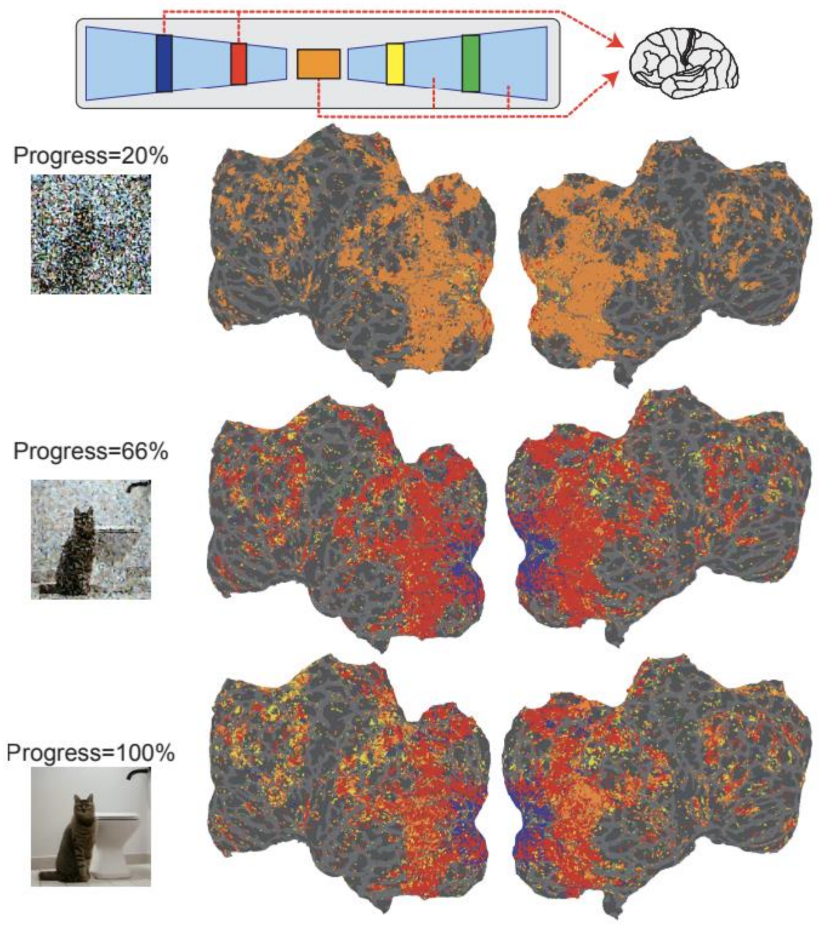


Figure 9. Selective engagement of different U-Net layers for different voxels across the brain. Colors represent the most predictive U-Net layer for early (top) to late (bottom) denoising steps. All colored voxels $P < 0.05$, FDR corrected.

Stable Diffusion: Text-to-Image

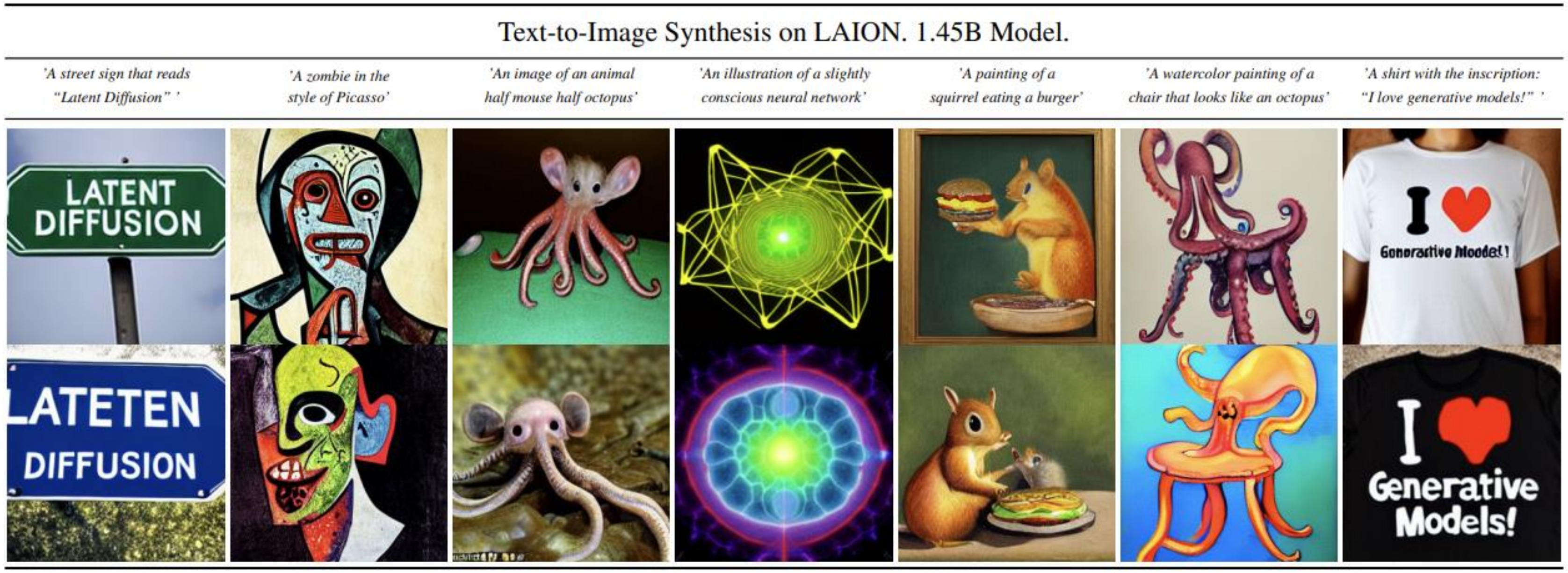

Text-to-Image Synthesis on LAION. 1.45B Model.
'A street sign that reads "Latent Diffusion" '
'A zombie in the style of Picasso'
'An image of an animal half mouse half octopus'
'An illustration of a slightly conscious neural network'
'A painting of a squirrel eating a burger'
'A watercolor painting of a chair that looks like an octopus'
'A shirt with the inscription: "I love generative models!" '
LATENT DIFFUSION
LATETEN DIFFUSION
Generative Models!

Figure 5. Samples for user-defined text prompts from our model for text-to-image synthesis, *LDM-8 (KL)*, which was trained on the LAION [78] database. Samples generated with 200 DDIM steps and $\eta = 1.0$. We use unconditional guidance [32] with $s = 10.0$.

Stable Diffusion: Layout-to-Image

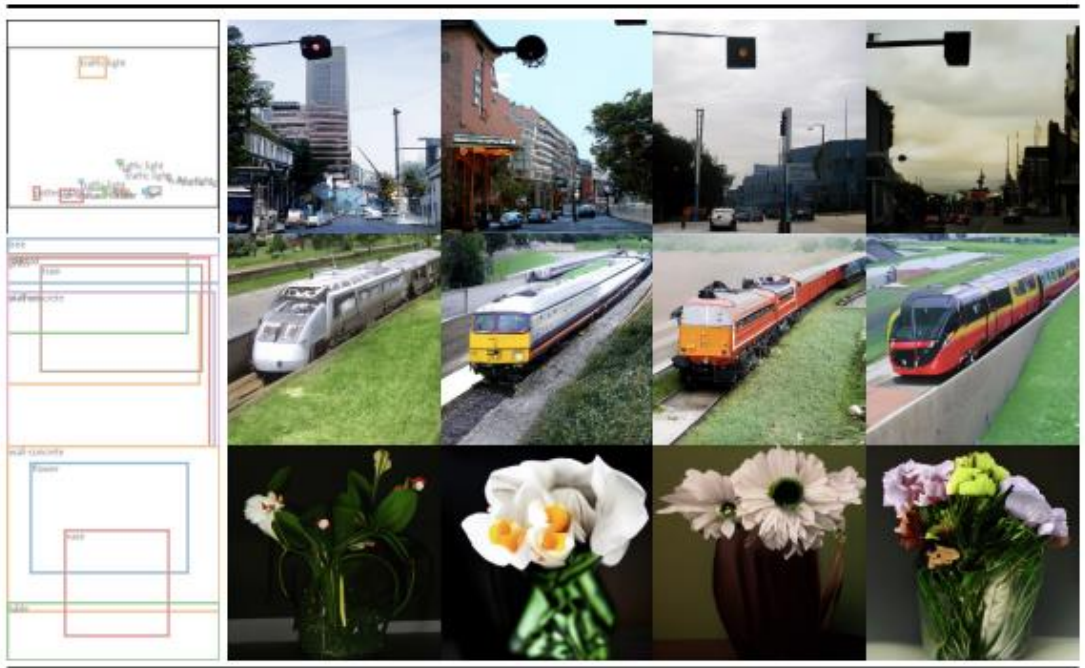
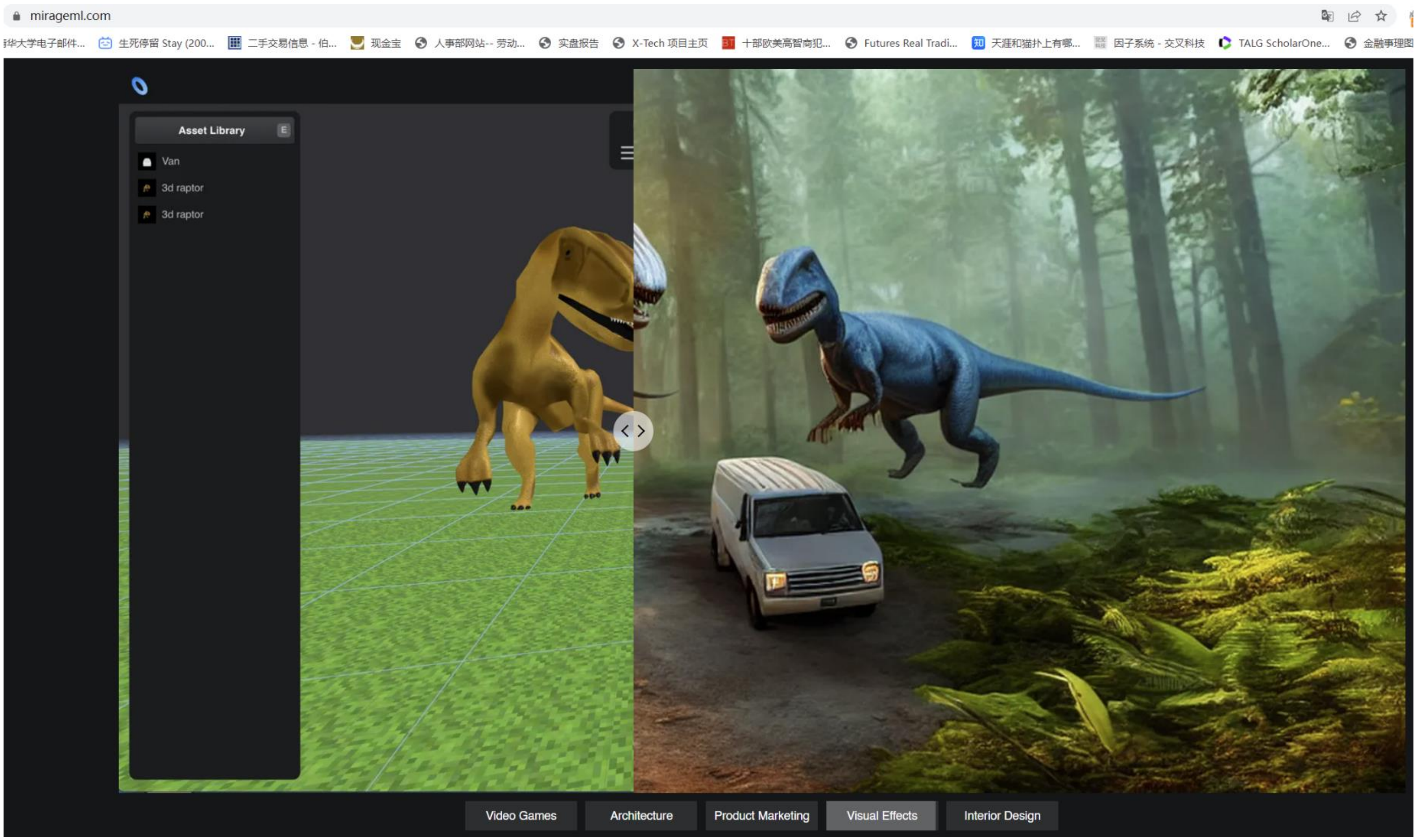
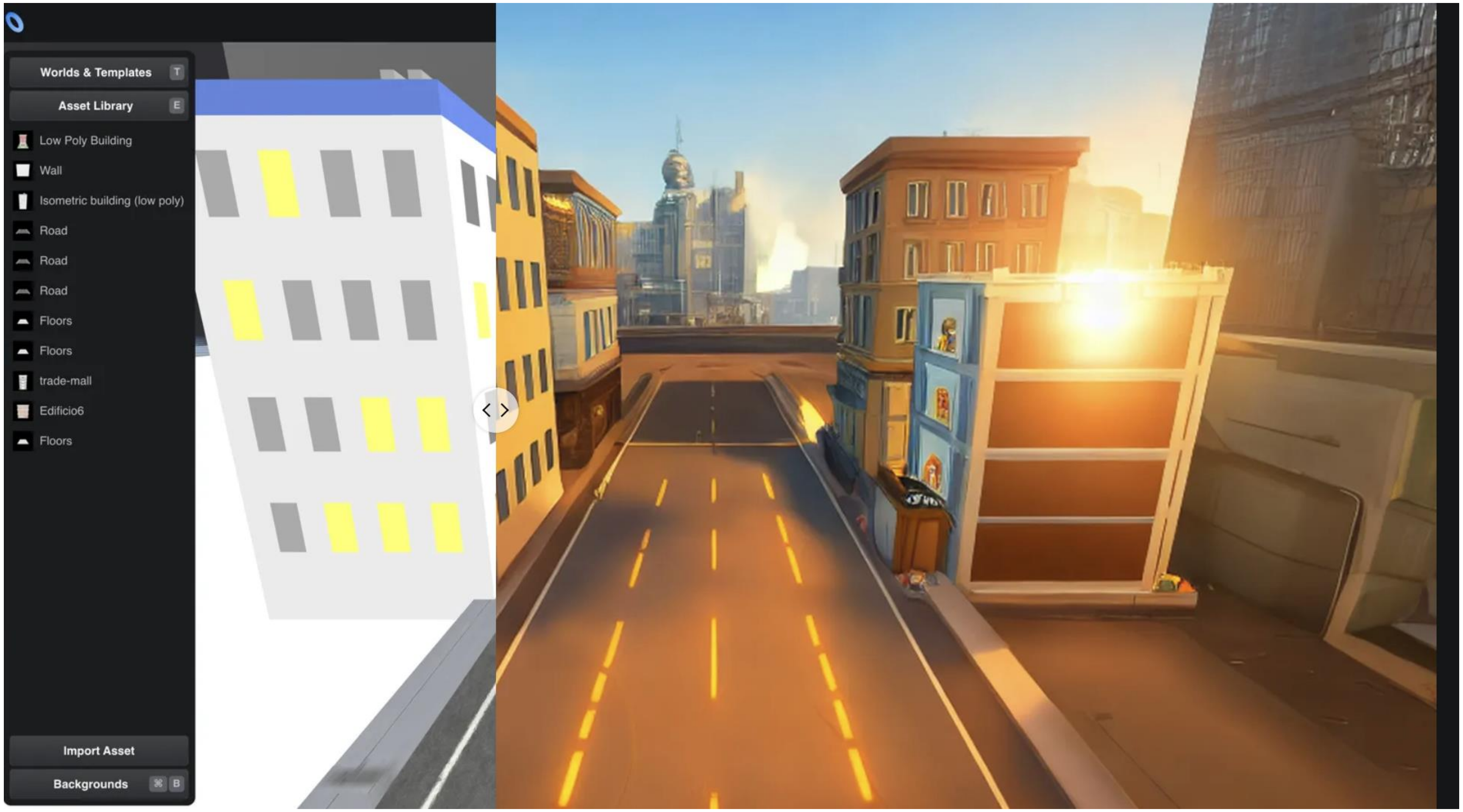


Figure 8. Layout-to-image synthesis with an *LDM* on COCO [4], see Sec. 4.3.1. Quantitative evaluation in the supplement D.3.

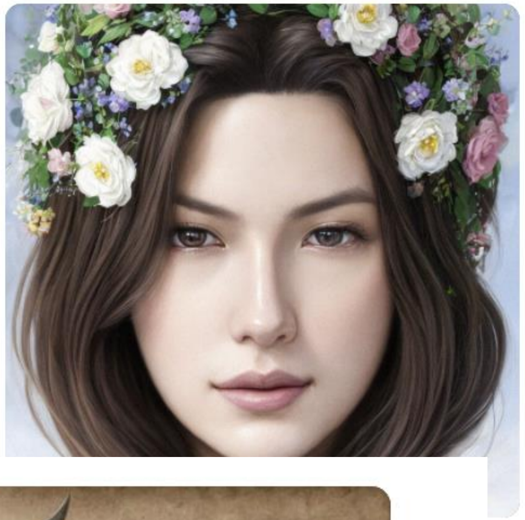
3D design


mirageml.com
Asset Library
Van
3d raptor
3d raptor
Video Games
Architecture
Product Marketing
Visual Effects
Interior Design

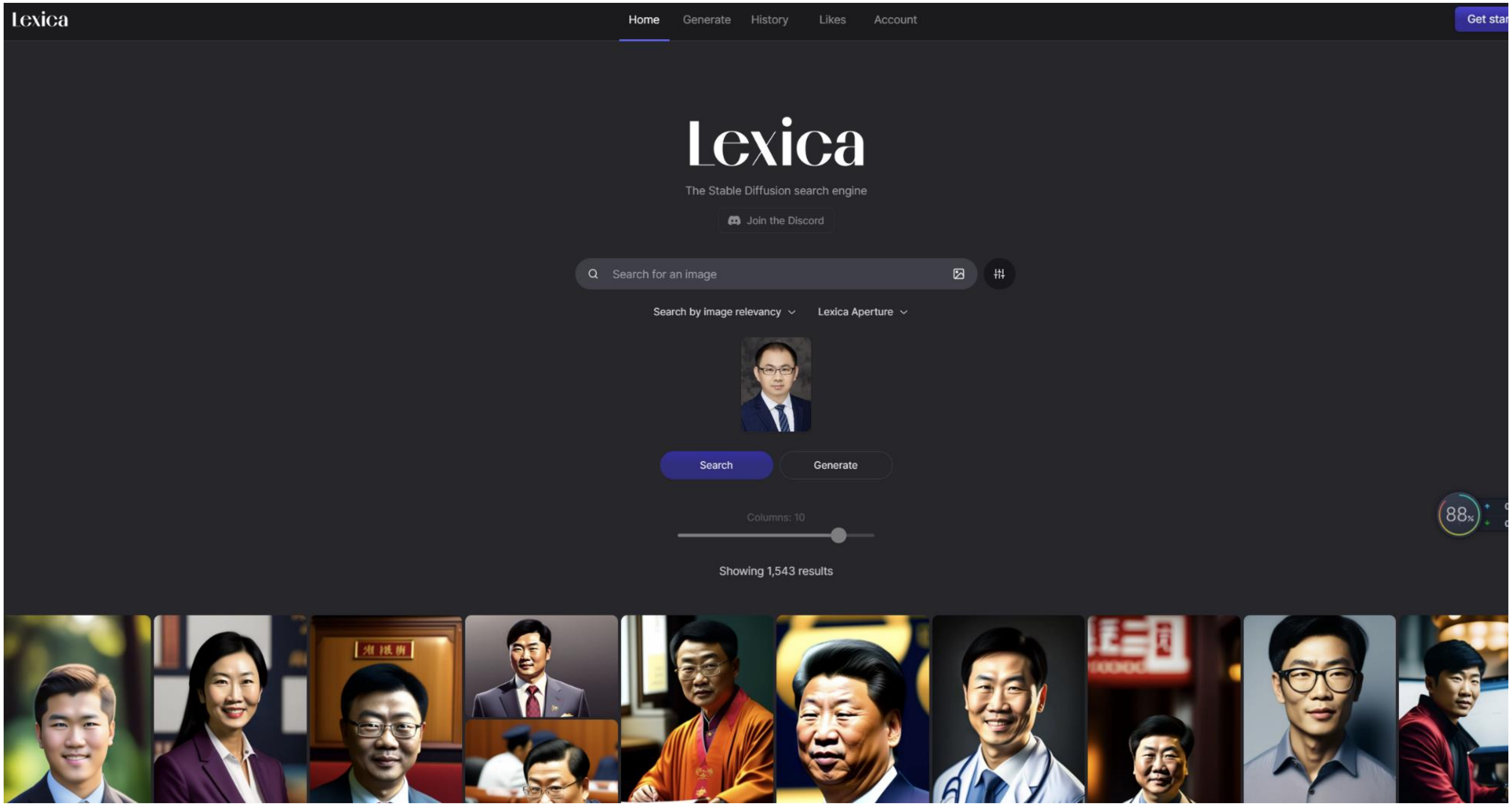
3D design


Worlds & Templates
T
Asset Library
E
Low Poly Building
Wall
Isometric building (low poly)
Road
Road
Road
Floors
Floors
trade-mall
Edificio6
Floors
Import Asset
Backgrounds
⌘ B

https://neural.love/

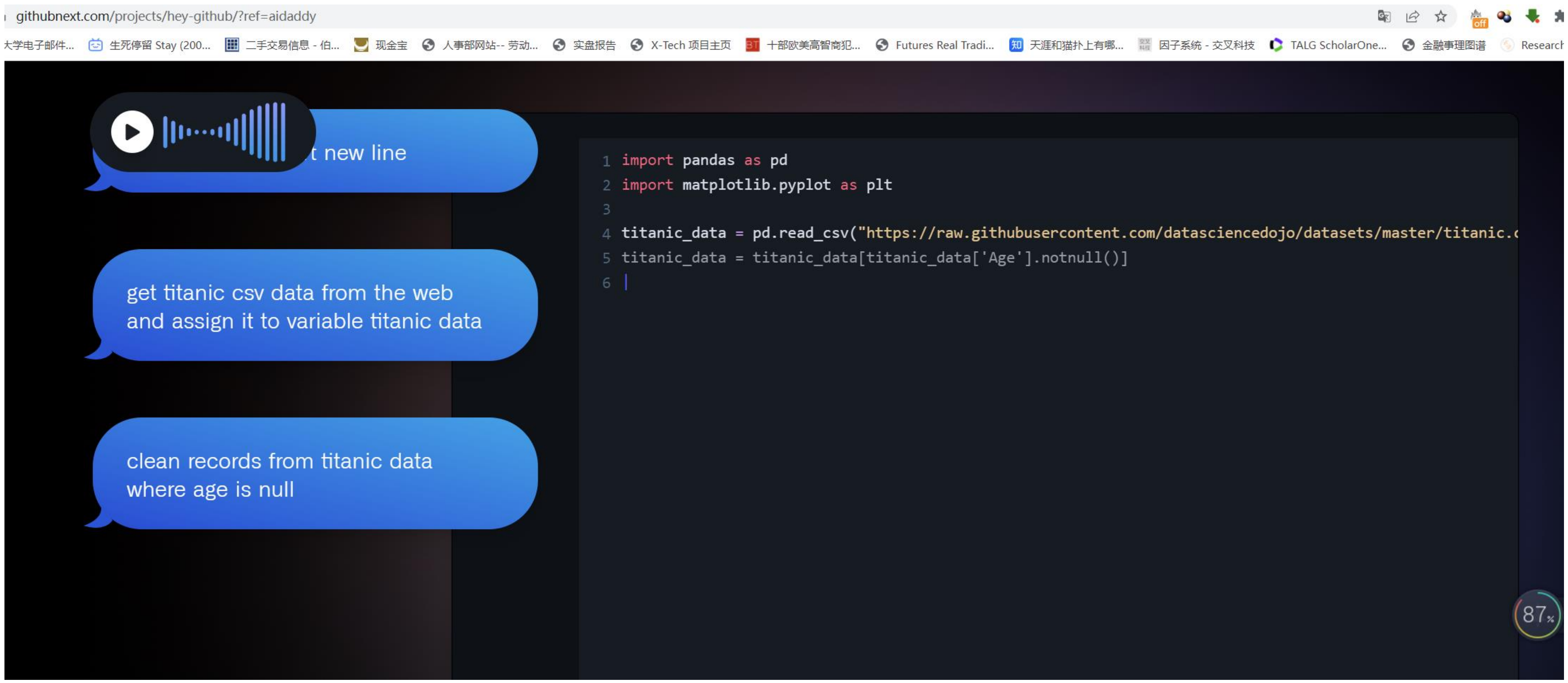


Stable diffusion search engine



AIGC: Low-code/No-code

- Write python without touching the keyboard
- https://githubnext.com/projects/hey-github/?ref=aidaddy



t new line

get titanic csv data from the web and assign it to variable titanic data

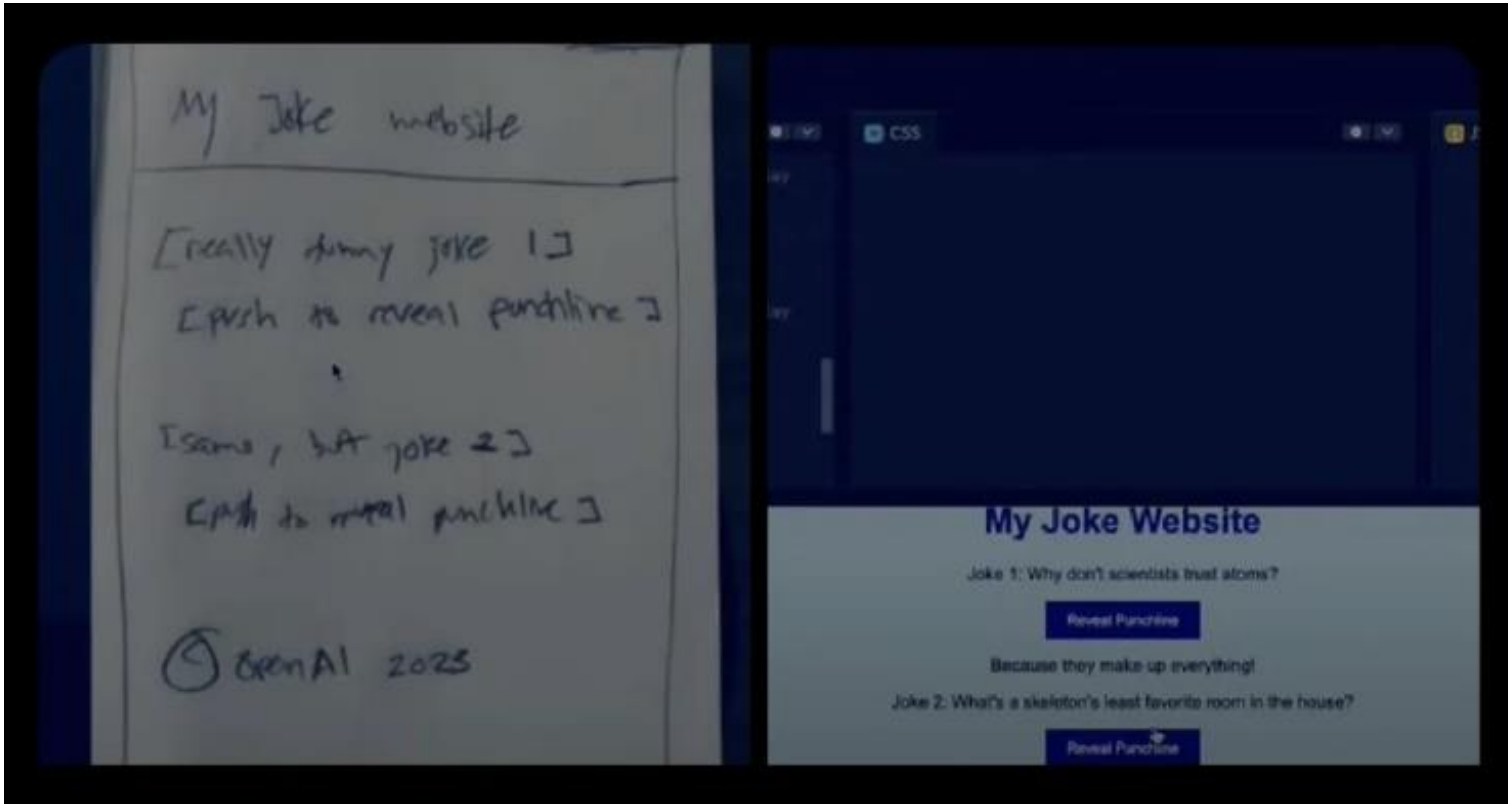
clean records from titanic data where age is null

```python
import pandas as pd
import matplotlib.pyplot as plt

titanic_data = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.
titanic_data = titanic_data[titanic_data['Age'].notnull()]

```

GPT-4: Build a webpage from a hand-drawn picture



My Joke website

[really funny joke 1]

[push to reveal punchline]

[same, but joke 2]

[push to reveal punchline]

© OpenAI 2023

My Joke Website

Joke 1: Why don't scientists trust atoms?

Because they make up everything!

Joke 2: What's a skeleton's least favorite room in the house?