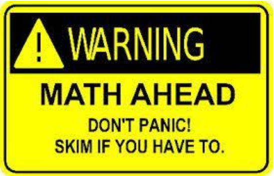


CSC413/2516 Lecture 10:

Diffusion Models & Vision Language Models

Bo Wang

Diffusion Probabilistic Models

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

ICLR 2015

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

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

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Motivation: Estimating small perturbations is more tractable than explicitly describing the full distribution.

The essential idea, inspired by non-equilibrium statistical physics, is to

- systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process.
- learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data.

https://arxiv.org/abs/1503.03585
https://www.youtube.com/watch?v=XCUlnHP1TNM

Denoising Diffusion Probabilistic Model (DDPM)

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset,

Demonstrate that diffusion models are capable of generating high quality samples.

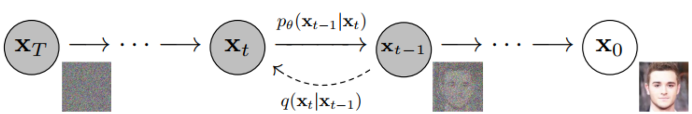


Figure 2: The directed graphical model considered in this work.

https://github.com/hojonathanho/diffusion

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020

Denoising Diffusion Probabilistic Model (DDPM)

Forward diffusion process: gradually adds noise to input image $q(x_t|x_{t-1})$

Reverse denoising process: learns to generate data by denoising $q(x_{t-1}|x_t) \approx p_\theta(x_{t-1}|x_t)$

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1,\dots,T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
5: $\quad$ Take gradient descent step on
$$\nabla_\theta \boxed{\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}, t)\right\|^2}$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
2: **for** $t = T,\dots,1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \boxed{\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}}$
5: **end for**
6: **return** $\mathbf{x}_0$

How are these formulas derived?

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020

Denoising Diffusion Probabilistic Model (DDPM)

DDPM: Forward Process $q(x_t|x_{t-1})$

Fixed, no trainable parameters

$x_0 \sim q(x)$ $\longrightarrow$ x_t $\longrightarrow$ x_T

Q: How to obtain x_t at any arbitrary time step t? $\beta_0 = 10^{-4}, \beta_T = 0.02$

Def: $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ $\quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$

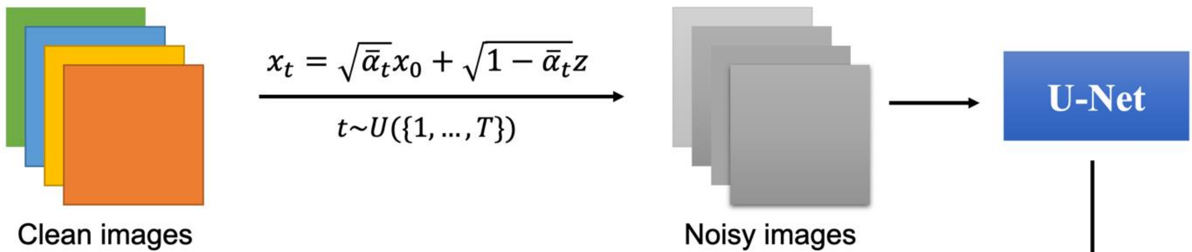
Reparameterization trick
A way to sample data x from $N(\mu, \sigma^2)$
- Sample z from $N(0,1)$
- $x = \mu + \sigma z$

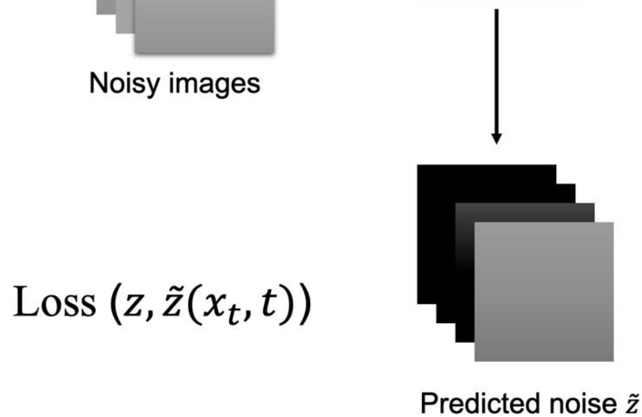
A nice property of the above process is that we can sample $\mathbf{x}_t$ at any arbitrary time step t in a closed form using reparameterization trick. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$:

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} && \text{;where } \mathbf{z}_{t-1}, \mathbf{z}_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\mathbf{z}}_{t-2} && \text{;where } \bar{\mathbf{z}}_{t-2} \text{ merges two Gaussians (*).} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z} \\
q(\mathbf{x}_t|\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})
\end{aligned}
$$

(*) Recall that when we merge two Gaussians with different variance, $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$, the new distribution is $\mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$. Here the merged standard deviation is $\sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$.

DDPM: Training Process





Algorithm 1 Training

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

DDPM: An alternative way to derive the loss

Training is performed by optimizing the usual variational bound on negative log likelihood:

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \le \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L \quad (3)$$

Efficient training is therefore possible by optimizing random terms of L with stochastic gradient descent. Further improvements come from variance reduction by rewriting L (3) as:

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big] \quad (5)$$

Furthermore, with the parameterization (11), Eq. (10) simplifies to:

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

$$\begin{aligned}
L_{\mathrm{CE}} &= -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \Big(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}\Big) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \Big(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T}\Big) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \Big(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big) \\
&\le -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] = L_{\mathrm{VLB}}
\end{aligned}$$

Jensen inequality

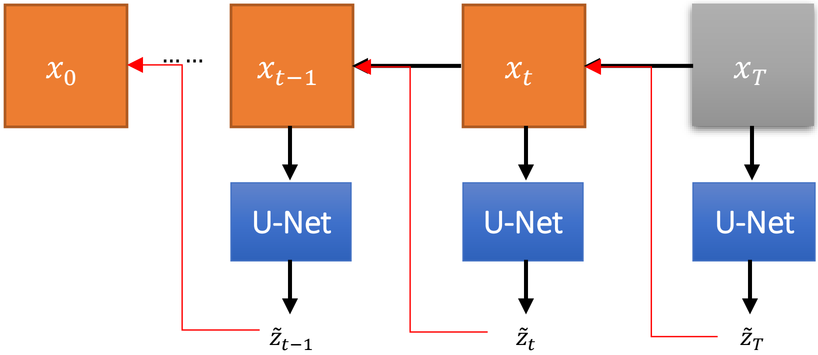
$$\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \Big(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\Big) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Big] \\
&= \mathbb{E}_q\big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\big]
\end{aligned}$$

Monster comes…

Separate case t=1

Bayes' Rule

DDPM: Reverse Process



$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} z_t$$

Reverse process: $(x_t, \tilde{z}) \rightarrow x_{t-1}$

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} (x_t - \sqrt{1 - \bar{\alpha}_t} z_t)$$

One step can recover x_0!
Why don't we use it?

Denoising Diffusion Probabilistic Model (DDPM)

DDPM: $x_t \to x_0$

Figure 7: When conditioned on the same latent, CelebA-HQ 256 $\times$ 256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

appear first and details appear last. Figure 7 shows stochastic predictions $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ with $\mathbf{x}_t$ frozen for various t. When t is small, all but fine details are preserved, and when t is large, only large scale features are preserved. Perhaps these are hints of conceptual compression [18].

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} z_t$$

Reverse process: $(x_t, \tilde{z}) \to x_{t-1}$

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t} z_t)$$

One step can recover x_0!
Why don't we use it?

DDPM: Reverse Process $q(x_{t-1}|x_t)$

Reverse denoising process: learns to generate data by denoising $q(x_{t-1}|x_t) \approx p_\theta(x_{t-1}|x_t)$

$x_0 \sim q(x)$

x_T

Q2: How to estimate the true reverse process $q(x_{t-1}|x_t)$?

Remark 1. If the variance is small enough during forward process, $q(x_{t-1}|x_t)$ will be Gaussian as well.

Remark 2. The reverse conditional probability $q(x_{t-1}|x_t)$ is intractable, but $q(x_{t-1}|x_t, x_0)$ would be tractable.

Reminder: Gaussian pdf: $N(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

Denoising Diffusion Probabilistic Model (DDPM) (optional)

DDPM: Reverse Process $q(x_{t-1}|x_t)$

Q3: How to obtain the mean and variance of $q(x_{t-1}|x_t)$?

It is noteworthy that the reverse conditional probability is tractable when conditioned on $\mathbf{x}_0$:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

Using Bayes' rule, we have:

$$\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1} + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\big)\Big)
\end{aligned}$$

DDPM: Reverse Process $q(x_{t-1}|x_t)$

Q3: How to obtain the mean and variance of $q(x_t|x_{t-1})$?

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t,\mathbf{x}_0)\big)\Big)$$

Following the standard Gaussian density function, the mean and variance can be parameterized as follows

$$\tilde{\beta}_t = 1/(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}) = 1/(\frac{\alpha_t-\bar{\alpha}_t+\beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

Variance: $\tilde{\beta}_t = \frac{1}{a}$

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) &= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}) \\ &= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t \\ &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0\end{aligned}$$

Mean: $\tilde{\mu}(x_t,x_0) = -\frac{b}{2a}$

Using the fact that $ax^2+bx+C=a(x+\frac{b}{2a})^2$, rearrange the second line with regards to x_{t-1}

- x_{t-1}^2: $\frac{\alpha_t}{\beta_t}x_{t-1}^2 + \frac{1}{1-\sqrt{\bar{\alpha}_{t-1}}}x_{t-1}^2$. so $a = \frac{\alpha_t}{\beta_t}+\frac{1}{1-\sqrt{\bar{\alpha}_{t-1}}}$.
- x_{t-1}: $(-\frac{2\sqrt{\alpha_t}}{\beta_t}x_t)x_{t-1} + (-\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\sqrt{\bar{\alpha}_{t-1}}}x_0)x_{t-1}$, so $b = -(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\sqrt{\bar{\alpha}_{t-1}}}x_0)$

Absorb x_0 by substituting it with

$$x_0 = \frac{1}{\sqrt{\bar{a}_t}}(x_t - \sqrt{1-\bar{a}_t}z_t)$$

$$\longrightarrow \quad \tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}z_t\right)$$

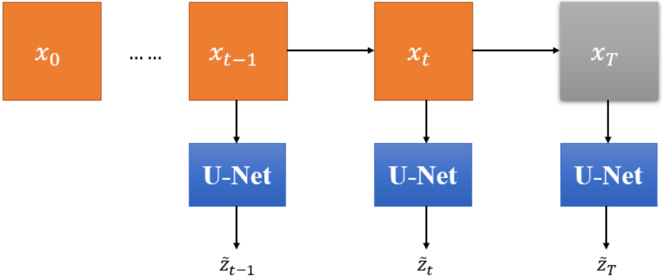
Denoising Diffusion Probabilistic Model (DDPM)

DDPM: Reverse Process $q(x_{t-1}|x_t)$

$$q(x_{t-1}|x_t) \sim N\left(\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_t\right), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I\right)$$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} z_t\right) + \sigma z$$

$$z \sim N(0, I), z_t \approx \tilde{z} = UNet(x_t, t)$$



Algorithm 2 Sampling

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

DDPM: Summary

Forward

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

$$q(x_t|x_0) = N(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + (1-\bar{\alpha}_t)z, \; z \sim N(0,1)$$

$$\tilde{z} = UNet(x_t, t)$$

$$loss(z, \tilde{z})$$

Reverse $\boldsymbol{x_{t-1} \sim N(\mu, \sigma^2)}$

$q(x_{t-1}|x_t) \rightarrow q(x_{t-1}|x_t, x_0) \rightarrow \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)}$ →derive mean and variance

$$q(x_{t-1}|x_t) = N\left(\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_t\right), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I\right)$$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_t\right) + \sigma z$$

Algorithm 1 Training

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

Algorithm 2 Sampling

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

Denoising Diffusion Probabilistic Model (DDPM)

DDPM: Sample Quality

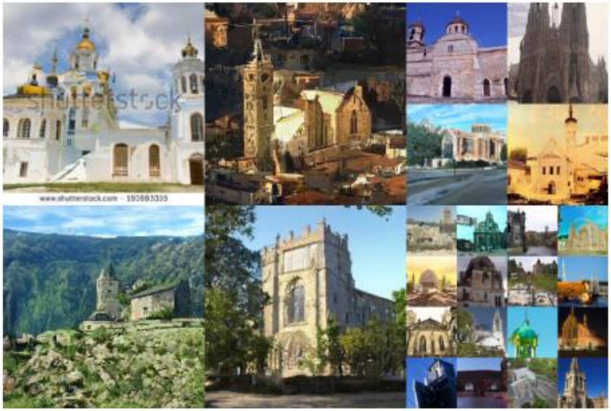


Figure 3: LSUN Church samples. FID=7.89

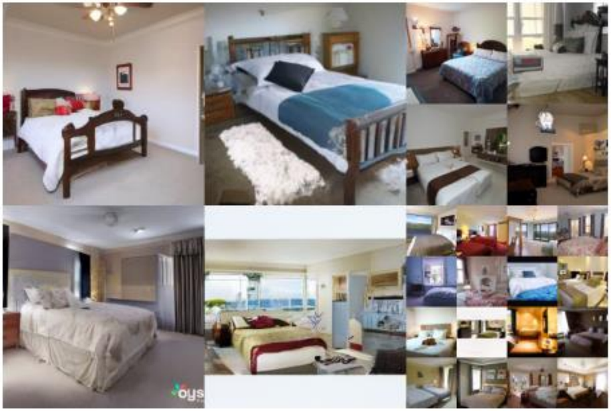


Figure 4: LSUN Bedroom samples. FID=4.90

We find that training our models on the true variational bound yields better codelengths than training on the simplified objective, as expected, but the latter yields the best sample quality.

DDPM: Progressive generation

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).

Large scale image features appear first and details appear last.

Demo Time: Stable Diffusion

https://stablediffusionweb.com/

Trade-offs of Generative Approaches

- So far, we have seen four different approaches:
  - Autoregressive models (Lectures 3, 7, and 8)
  - Generative adversarial networks (last lecture)
  - Variational Auto-encoder (this lecture)
  - Diffusion models (this lecture)
- They all have their own pro and con. We often pick a method based on our application needs.
- Some considerations for computer vision applications:
  - Do we need to evaluate log likelihood of new data?
  - Do we prefer good samples over evaluation metric?
  - How imporant is representation learning, i.e. meaningful code vectors?
  - How much computational resource can we spent?

Vision Language Models (VLMs)

Vision Language Models (VLMs)

What can VLM do?

An example of recent open-source VLM: Molmo

https://molmo.allenai.org/blog

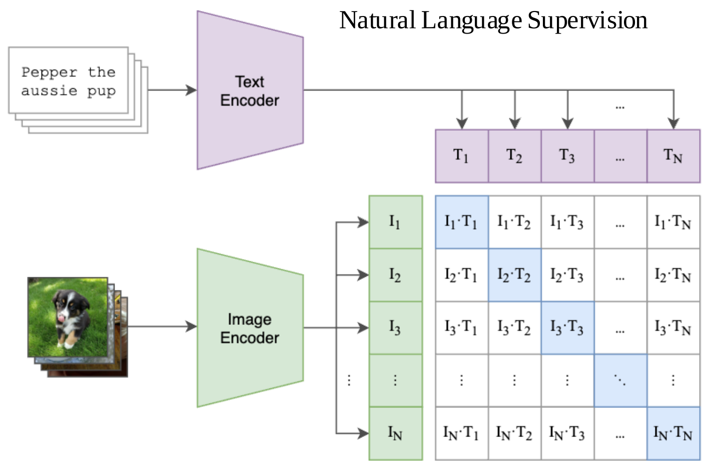
Outline

- Overview: From CLIP to GPT-4V
- Recent Advances on VLMs
- Summary and Emerging Trends

Vision Language Models (VLMs)

Overview: From CLIP to GPT-4V

(1) Contrastive pre-training

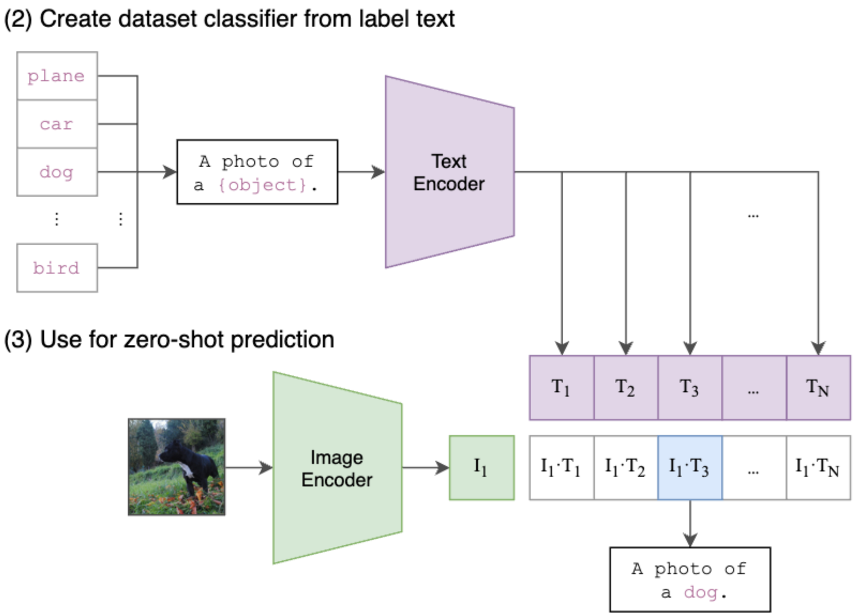



```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

https://openai.com/index/clip/

Vision Language Models (VLMs)

Overview: From CLIP to GPT-4V



https://openai.com/index/clip/

Vision Language Models (VLMs)

Overview: From CLIP to GPT-4V

Food101

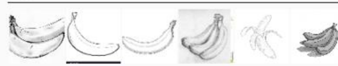
guacamole (90.1%) Ranked 1 out of 101 labels

- ✓ a photo of **guacamole**, a type of food.
- × a photo of **ceviche**, a type of food.
- × a photo of **edamame**, a type of food.
- × a photo of **tuna tartare**, a type of food.
- × a photo of **hummus**, a type of food.

SUN397

television studio (90.2%) Ranked 1 out of 397 labels

- ✓ a photo of a **television studio**.
- × a photo of a **podium indoor**.
- × a photo of a **conference room**.
- × a photo of a **lecture room**.
- × a photo of a **control room**.

Dataset	ImageNet ResNet101	CLIP ViT-L
 ImageNet	76.2%	76.2%
 ImageNet V2	64.3%	70.1%
 ImageNet Rendition	37.7%	88.9%
 ObjectNet	32.6%	72.3%
 ImageNet Sketch	25.2%	60.2%
 ImageNet Adversarial	2.7%	77.1%

Vision Language Models (VLMs)

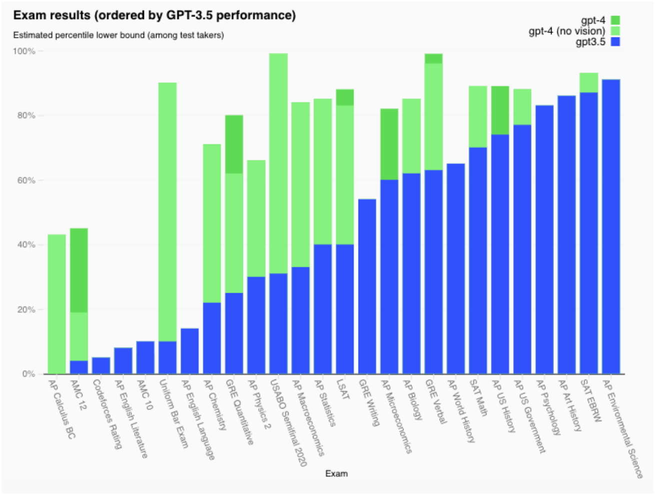
Overview: From CLIP to GPT-4V

GPT-4 Technical Report

OpenAI*

Abstract

We report the development of GPT-4, a large-scale, multimodal model which can accept image and text inputs and produce text outputs. While less capable than humans in many real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the top 10% of test takers. GPT-4 is a Transformer-based model pre-trained to predict the next token in a document. The post-training alignment process results in improved performance on measures of factuality and adherence to desired behavior. A core component of this project was developing infrastructure and optimization methods that behave predictably across a wide range of scales. This allowed us to accurately predict some aspects of GPT-4's performance based on models trained with no more than 1/1,000th the compute of GPT-4.



https://openai.com/index/gpt-4-research/

Vision Language Models (VLMs)

Overview: From CLIP to GPT-4V

User What is funny about this image? Describe it panel by panel.

Source: hmmm (Reddit)

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

User What is unusual about this image?

Source: Barnorama

GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

Vision Language Models (VLMs)

Open-source VLM

GPT-4V shows impressive capability for image and test understanding, but

- Model architecture and training protocols (e.g., data, optimizer…) are not clear

From 2023: How can we build GPT-4V like models?

- Year 2023: a large gap between open-source models and GPT-4V
- Year 2024: ~90% performance on public benchmarks

VLM Leaderboard: Chatbot Arena

Arena (battle) | Arena (side-by-side) | Direct Chat | Leaderboard | About Us

Chatbot Arena LLM Leaderboard: Community-driven Evaluation for Best LLM and AI chatbots

Blog | GitHub | Paper | Dataset | Twitter | Discord | Kaggle Competition

Chatbot Arena (lmarena.ai) is an open-source platform for evaluating AI through human preference, developed by researchers at UC Berkeley SkyLab and LMSYS. With over 1,000,000 user votes, the platform ranks best LLM and AI chatbots using the Bradley-Terry model to generate live leaderboards. For technical details, check out our paper.

Category: English

Apply filter: Style Control | Show Deprecated

English Prompts

#models: 37 (100%) #votes: 78,383 (62%)

Rank* (UB)	Delta	Model	Arena Score	95% CI	Votes	Organization	License	Knowledge Cutoff
1	0	ChatGPT-4o-latest (2024-09-03)	1258	+17/-13	2283	OpenAI	Proprietary	2023/10
2	0	Gemini-1.5-Pro-002	1225	+15/-14	1848	Google	Proprietary	Unknown
2	0	Gemini-1.5-Flash-002	1215	+16/-21	1542	Google	Proprietary	Unknown
2	0	GPT-4o-2024-05-13	1212	+9/-7	12516	OpenAI	Proprietary	2023/10
4	0	Claude 3.5 Sonnet (20241022)	1188	+13/-16	1573	Anthropic	Proprietary	2024/4
4	0	Claude 3.5 Sonnet (20240620)	1187	+7/-6	13820	Anthropic	Proprietary	2024/4
7	0	Gemini-1.5-Pro-001	1159	+7/-7	10951	Google	Proprietary	2023/11
7	0	GPT-4-Turbo-2024-04-09	1157	+7/-8	8705	OpenAI	Proprietary	2023/12
7	2	Gemini-1.5-Flash-8B-Exp-0827	1137	+15/-13	2126	Google	Proprietary	2023/11
8	1	GPT-4o-2024-08-06	1131	+21/-21	815	OpenAI	Proprietary	2023/10
9	0	Gemini-1.5-Flash-8B-001	1132	+14/-15	1605	Google	Proprietary	Unknown
9	0	GPT-4o-mini-2024-07-18	1124	+8/-8	8803	OpenAI	Proprietary	2023/10
9	4	Molmo-72B-0924	1119	+20/-20	1120	AI2	Apache 2.0	Unknown
9	0	Qwen2-VL-72b-Instruct	1115	+15/-14	1625	Alibaba	Qwen	2024/9

78K+ human votes

https://lmarena.ai/

Vision Language Models (VLMs)

VLM Leaderboard: Vision Arena

Arena | Video Arena | Direct Chat | Leaderboard | About Us | Failure Case Examples

WildVision Arena Leaderboard

Image-Arena Leaderboard | Video-Arena Leaderboard

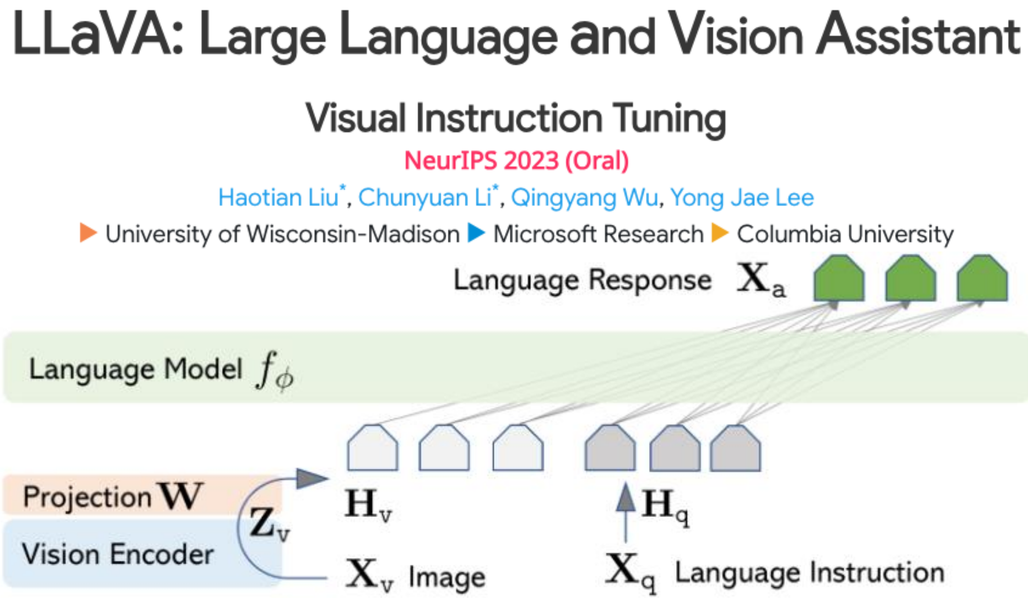
Total #models: **38**. Total #votes: **15626**. Last updated: 2024-11-07 01:39:00 PST.

15K+ human votes

Rank	V-L Model	WV-Arena Elo	95% CI	Battles	MMMU
1	gpt-4o	1195	+17/-13	2149	OpenAI
2	claude-3-5-sonnet-20240620	1160	+16/-20	1156	Anthropic
3	llava-onevision-qwen2-72b-ov-chat	1151	+137/-78	35	Bytedance
4	Aria	1148	+129/-119	26	Rhymes
5	gpt-4o-mini	1135	+53/-42	117	OpenAI
6	gemini-1.5-pro-latest	1110	+33/-39	283	Google
7	Pixtral-12B-2409	1101	+67/-62	106	Mistral
8	gpt-4-turbo	1099	+50/-44	128	OpenAI
9	gpt-4-vision-preview	1089	+12/-12	3384	OpenAI
10	gemini-1.5-flash-latest	1085	+21/-17	1295	Google
11	Qwen2-VL-7B-Instruct	1061	+64/-48	116	Alibaba

https://huggingface.co/WildVision

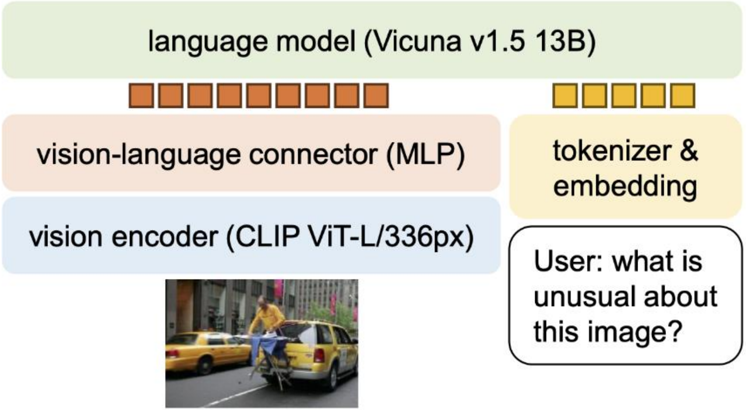
VLM Architectures: LLaVA



https://llava-vl.github.io/

Vision Language Models (VLMs)

VLM Architectures: LLaVA 1.5



Main architecture modifications

- ➢ Better image encoder: CLIP ViT-L 336
- ➢ Replace linear connector with MLP

Liu, Haotian, et al. "Improved baselines with visual instruction tuning." *CVPR* 2024. https://arxiv.org/abs/2310.03744

Vision Language Models (VLMs)

VLM Architectures: LLaVA Next

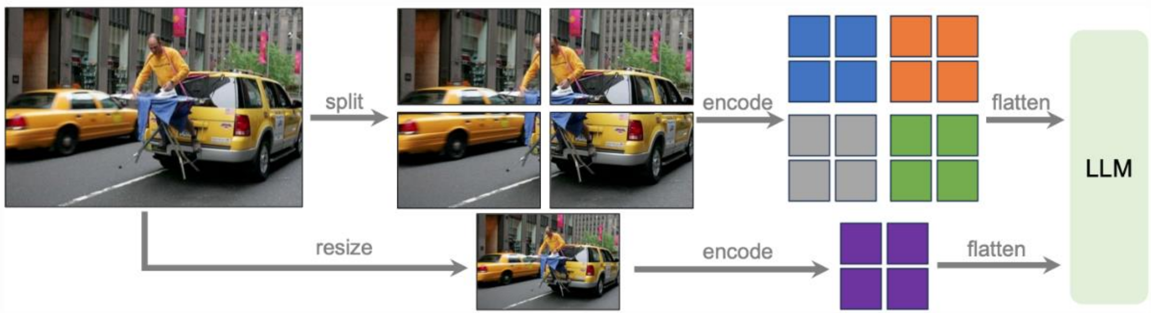


Illustration of dynamic high resolution scheme: a grid configuration of 2 × 2

- Dynamic high resolution
- Improve data quality with better diversity and instructions
- Scale LLM backbone from 7B to 34B

https://llava-vl.github.io/blog/2024-01-30-llava-next/

Vision Language Models (VLMs)

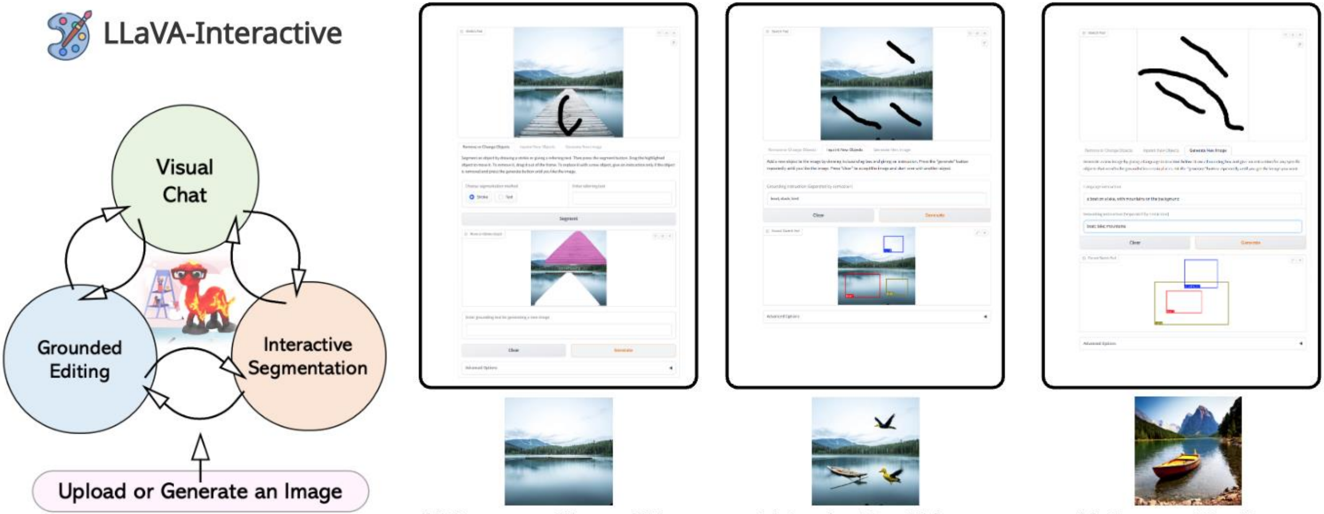
VLM Architectures: LLaVA Next

High-quality data

Datasets	LLaVA-ReCap (LLaVA-NeXT-34B)	Raw Captions
 COCO118K	The image shows a meal served in a blue tray with compartments. In the top left compartment, there is a slice of bread with a spread that appears to be butter, accompanied by a few almonds and a slice of what looks like a baked potato or sweet potato. The top right compartment contains a variety of fruits, including what seems to be pineapple, orange slices, and possibly a piece of melon.\n\nIn the bottom left compartment, there is a piece of bread with a spread that could be butter or margarine, and a small portion of what might be a meatball or a similar type of meat covered in a sauce. The bottom right compartment contains a serving of broccoli, which appears to be steamed or lightly cooked.\n\nThe meal is presented in a way that suggests it is a balanced meal, with a variety of food groups represented carbohydrates (bread), protein (meatball), healthy fats (almonds and butter), and fruits and vegetables (broccoli and the fruit assortment). The image is a close-up photograph with a focus on the food, and the colors are vibrant, indicating freshness.	• a meal is presented in brightly colored plastic trays. • closeup of bins of food that include broccoli and bread. • colorful dishes holding meat, vegetables, fruit, and bread. • a bunch of trays that have different food. • there are containers filled with different kinds of foods
 BLIP558K	The image shows a scene of a significant vehicle accident. A white car is severely damaged, with its body crumpled and parts of the vehicle scattered around. The car appears to have been involved in a high-speed impact, as suggested by the extensive damage and the disarray of the surrounding area.\n\nIn the foreground, there is a person wearing a high-visibility vest, which is typically worn by emergency personnel or workers to ensure they are easily seen in such situations. This individual is standing near the wreckage, possibly assessing the situation or preparing to provide assistance.\n\nThe setting appears to be an outdoor area with grass, and there are various items scattered around, including what looks like a suitcase and other debris, indicating that the accident may have occurred in a public space or near a parking area.\n\nIn the background, there is a sign with the word \"Reliant\" on it, which could be part of a larger sign or structure, possibly indicating the name of a nearby establishment or facility. The sign is partially obscured by the debris from the accident.\n\nThe overall scene suggests a serious incident that has likely attracted the attention of emergency services and possibly led to disruption in the area. The image captures the immediate aftermath of the accident, with no visible people other than the individual in the high-visibility vest.	a crashed white honda car in a field at a parking lot
 CC3M	The image shows an interior view of a modern kitchen. The kitchen features a clean and minimalist design with a color scheme that includes white cabinetry and black countertops. The cabinets have a glossy finish and are complemented by stainless steel appliances, including a refrigerator, oven, and microwave. The countertops appear to be made of a dark stone or composite material.\n\nIn the foreground, there is a glass-topped kitchen island with a stainless steel faucet and a red kettle on the countertop. The island also has a built-in oven and a cooktop. The backsplash is made of a light-colored stone or tile, and there are several small items on the countertops, such as a few books, a bottle, and a small plant.\n\nThe ceiling has recessed lighting and a large, rectangular skylight that allows natural light to flood the space. The floor is not fully visible, but it appears to be a light-colored hardwood or laminate. The overall impression is of a well-lit, spacious, and contemporary kitchen.	a city for sale property .

VLM Architectures: LLaVA Interactive

An All-in-One Demo for Image Chat, Segmentation and Generation/Editing

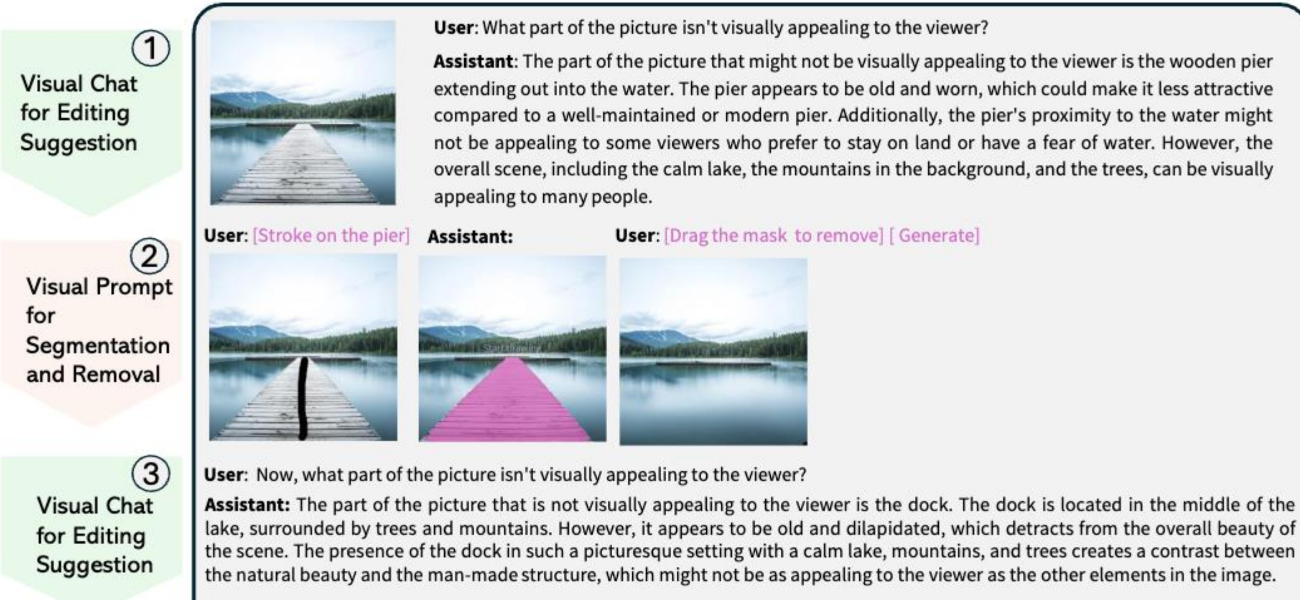


(b) Remove or Change Objects (c) Inpaint New Objects (d) Generate New Image

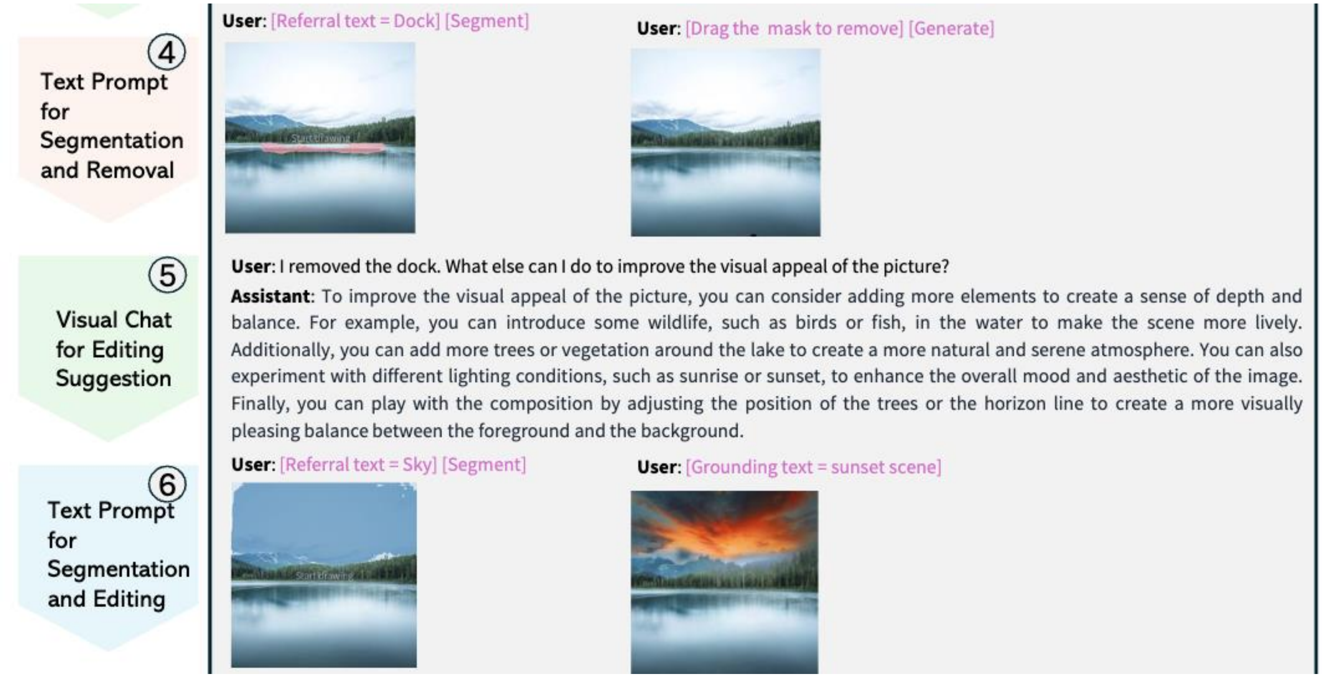
https://llava-vl.github.io/llava-interactive/

Vision Language Models (VLMs)

VLM Architectures: LLaVA Interactive



VLM Architectures: LLaVA Interactive



Vision Language Models (VLMs)

VLM Architectures: LLaVA Interactive

⑦ Visual Chat for Editing Suggestion

User: I really like the idea of a sunset and changed the sky to a sunset scene. Does it look good to you?

Assistant: As an AI, I don't have personal preferences or opinions, but I can tell you that the image you have created with the sunset sky looks visually appealing. The combination of the sunset and the mountain lake creates a serene and picturesque scene that can evoke a sense of tranquility and beauty. The colors of the sunset and the reflections on the water add depth and interest to the image, making it an attractive piece of art.

User: What do you mean by the reflections on the water?

Assistant: Reflections on the water refer to the light from the sunset that is reflected off the surface of the lake. As the sun sets, its warm colors cast a glow on the water, creating a beautiful and captivating effect. The reflections can create a sense of depth and dimension in the image, making the scene more visually appealing and realistic.

⑧ Text Prompt for Segmentation and Editing

User: [Referral text = lake surface] [Segment]

User: [Grounding text = reflection of the sunset scene in the sky] [Generate]

⑨ Visual Chat for Editing Suggestion

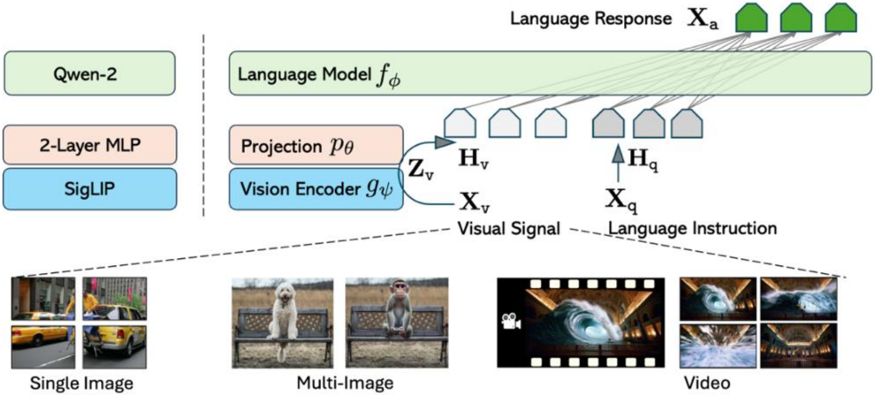
User: Thank you for mentioning the reflection. I added the reflection of the sunset. What do you think of my final product?

Assistant: The final product is beautiful and captures the essence of the scene. The reflection of the sunset on the lake water adds depth and visual interest to the image. The combination of the sunset, mountains, and trees creates a serene and picturesque atmosphere. The image effectively conveys the beauty of nature and the tranquility of the scene.

Vision Language Models (VLMs)

VLM Architectures: LLaVA OneVision

The first single open-source model that can simultaneously handle three important computer vision scenarios: **single-image, multi-image and video scenarios**.



LLaVA-OneVision Network Architecture.
Left: Current model instantiation; Right: The general form of LLaVA extended to more visual signals.

- LLM: Qwen-2
- Vision Encoder: SigLIP
- Projector: 2-layer MLP

Li, Bo, et al. "Llava-onevision: Easy visual task transfer." *arXiv preprint arXiv:2408.03326* (2024).

Vision Language Models (VLMs)

VLM Architectures: LLaVA OneVision

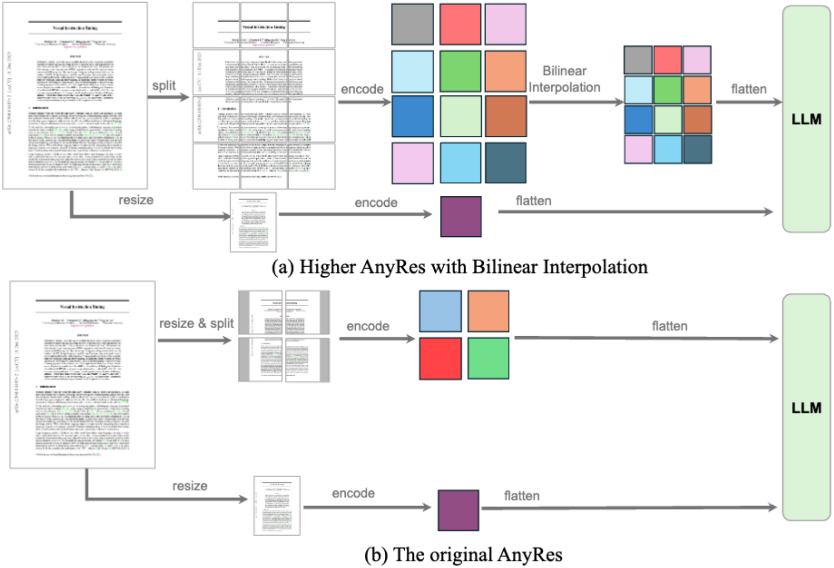


Figure 2: The visual representations. Top: The new Higher AnyRes scheme with Bilinear Interpolation to deal with images of higher resolution; Bottom: the original AnyRes in [82].

Li, Bo, et al. "Llava-onevision: Easy visual task transfer." *arXiv preprint arXiv:2408.03326* (2024).

VLM Architectures: LLaVA OneVision

Visual Representation Strategy in LLaVA-OneVision.
The maximum number of visual tokens across different scenarios is designed to be similar, ensuring balanced representations to accommodate cross-scenario capability transfer.

Li, Bo, et al. "Llava-onevision: Easy visual task transfer." *arXiv preprint arXiv:2408.03326* (2024).

Vision Language Models (VLMs)

VLM Architectures: LLaVA OneVision

User: Here's an image with objects labeled in numbers. Please describe the objects that are marked as 4, 5 and 7.

LLaVA-OV: The image shows a room with various items marked with numbers. Here's the description for the items marked as 4, 5, and 7:
The item marked as 4 is a framed picture hanging on the wall. It features a black and white illustration of a dandelion with seeds blown away, creating a sense of movement. The item marked as 5 is a white bookshelf with multiple shelves. It appears to be a storage unit that is part of the room's furniture. It is located between the two desks. The item marked as 7 is a black chair with a modern design. It has a metal frame and a woven seat, suggesting a contemporary style. This chair is positioned in front of one of the desks.

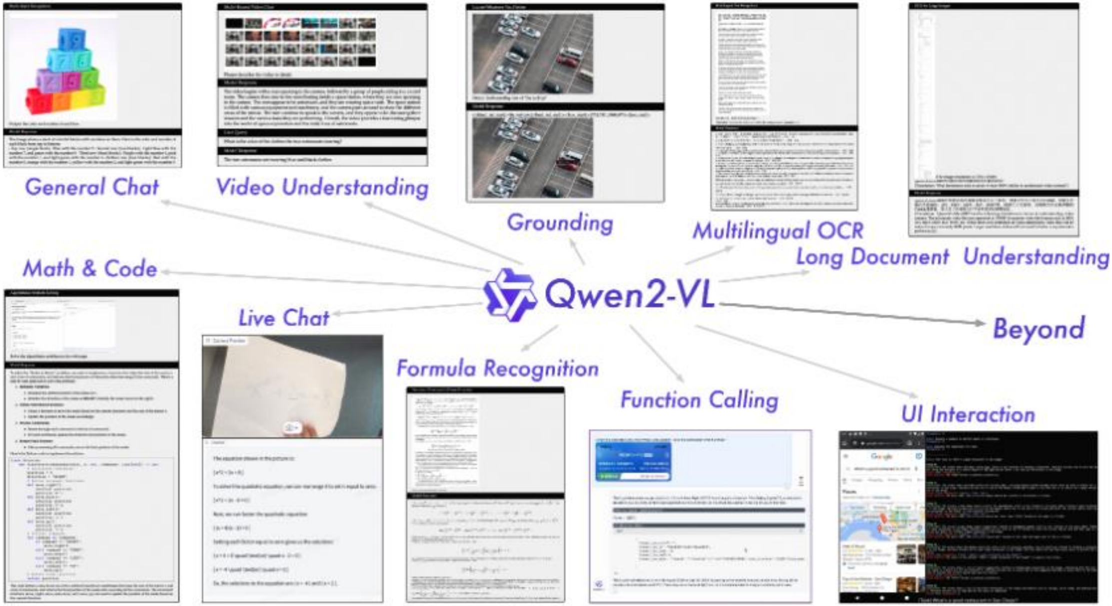
User: Here are two videos, please provide the analysis on their differences.

LLaVA-OV: The first video shows a sequence of images where a man in a blue plaid shirt approaches a woman in a red dress, engages in a conversation, and eventually gives her a kiss on the cheek. The woman in the red dress is looking at the man with a slight smile, and another woman in a light blue top is observing the scene. The setting appears to be a street with buildings in the background.
The second video shows the same man in the blue plaid shirt walking away from the woman in the red dress, who is now looking in the opposite direction. The woman in the light blue top is no longer visible in this sequence. The man is walking down the street, passing by the buildings and other pedestrians. The overall mood of the video changes from the intimate and personal interaction in the first video to a more solitary and neutral scene in the second video.

Li, Bo, et al. "Llava-onevision: Easy visual task transfer." *arXiv preprint arXiv:2408.03326* (2024).

Vision Language Models (VLMs)

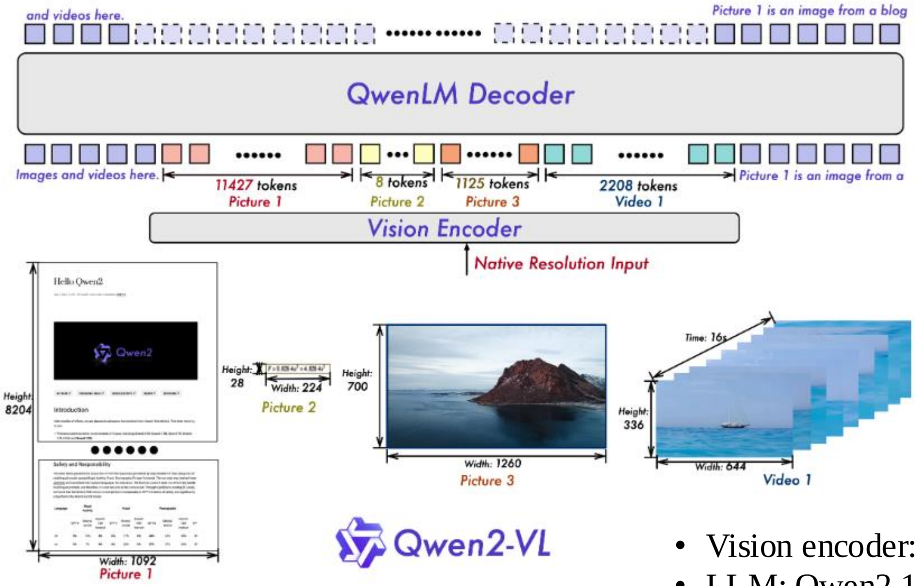
VLM Architectures: Qwen2-VL



https://qwenlm.github.io/blog/qwen2-vl/

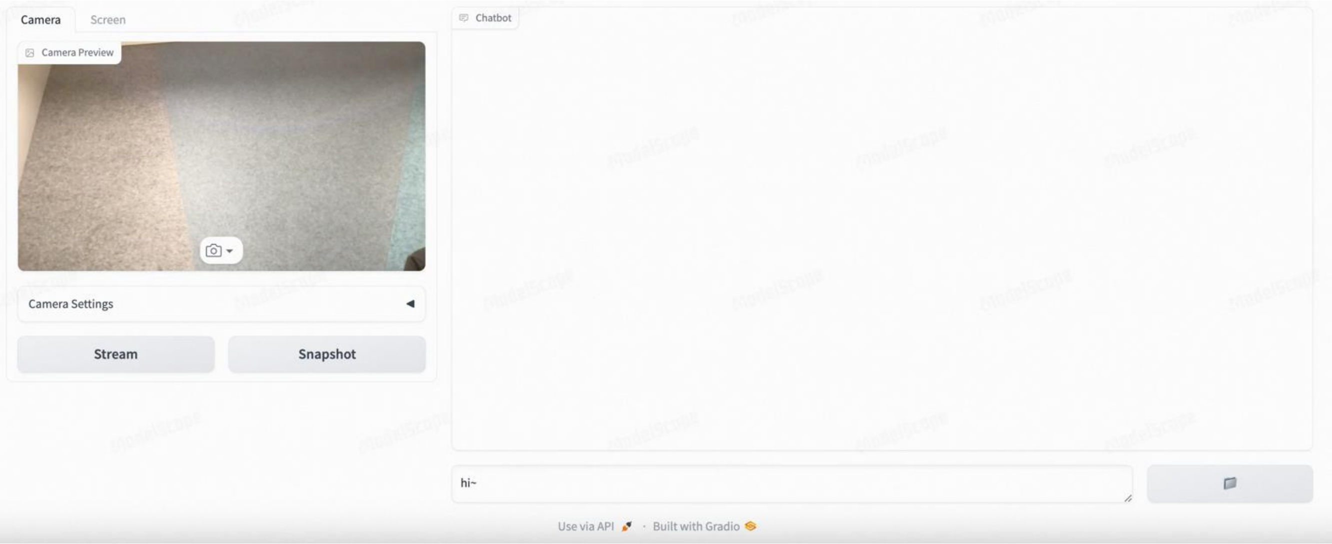
Vision Language Models (VLMs)

VLM Architectures: Qwen2-VL



- Vision encoder: ViT-14 675M
- LLM: Qwen2 1.5B-72B
- Naïve dynamic resolution

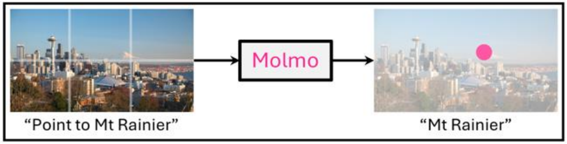
VLM Architectures: Qwen2-VL

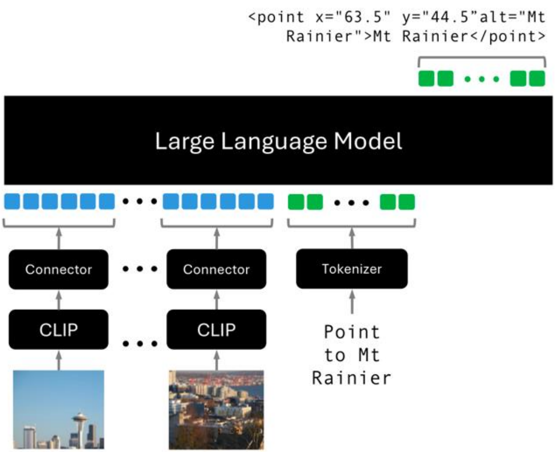


https://qwenlm.github.io/blog/qwen2-vl/

Vision Language Models (VLMs)

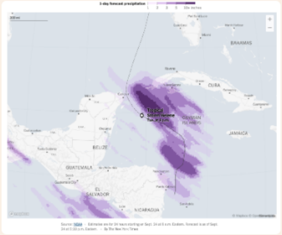
VLM Architectures: Molmo





https://molmo.allenai.org/blog

- Vision encoder: ViT-L/14 336x336
- Connector: MLP
- LLM: Qwen2 7B-72B



Describe this image.

This image is a detailed weather map depicting a one-day forecast of precipitation. The map features a color gradient scale at the top, ranging from light gray to dark purple, indicating precipitation levels from less than an inch to over 10 inches. The landmasses are shown in white, while the surrounding water is shaded in light gray.

Prominent locations such as Cuba, the Bahamas, Jamaica, Belize, Guatemala, El Salvador, and Nicaragua are clearly labeled. The map also highlights the Cayman Islands and various cities within these countries.

At the center of the map, there is a black circle with a white dot, marking the position of Tropical Storm Helene. The forecast timestamp is Tuesday at 8 p.m. The storm's precipitation is represented by a dark purple area, indicating heavy rainfall. The purple extends from the center of the map, spreading towards the left and right, with lighter shades of purple indicating less intense precipitation.

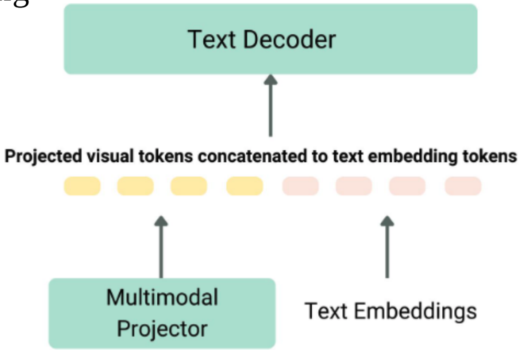
Demo Time

https://huggingface.co/spaces/garibida/ReNoise-Inversion

Vision Language Models (VLMs)

Summary

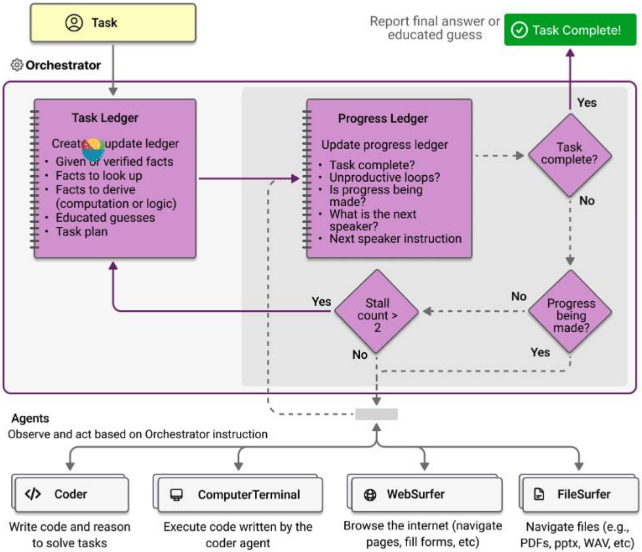
- ➢ The gap between open-source VLMs and GPT-4 are closing
- ➢ Model architectures and training recipes are converging
  - Image encoder, projector, and LLM decoder
  - Stage-wise training
    - Uni-modal pretraining
    - Projector training
    - Multimodal instruction fine-tuning



- ➢ Recommend paper:
- Liang et al. A Comprehensive Survey and Guide to Multimodal Large Language Models in Vision-Language Tasks, https://arxiv.org/abs/2411.06284
- Bordes et al. An Introduction to Vision-Language Modeling, https://arxiv.org/abs/2405.17247

Emerging Trends: Multi-Agent System

Microsoft: Magentic-One (Nov. 4)



https://github.com/microsoft/autogen/tree/main/python/packages/autogen-magentic-one