

# Dual Diffusion for Unified Image Generation and Understanding

Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, Peng Wang

Pengxi Liu

November 14, 2025

# Multimodal Generative Models



Image Captioning

**Model:**  
 $P(\text{text} \mid \text{image})$

*"A Samoyed  
puppy coding  
on a laptop."*

*"A Samoyed  
puppy coding  
on a laptop."*

Text-based Generation

**Model:**  
 $P(\text{image} \mid \text{text})$



- Image **generation** and image **understanding** are traditionally *separated* worlds.

# Multimodal Generative Models



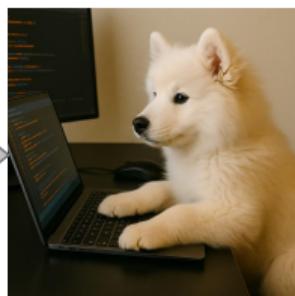
Image Captioning  
Model:  
 $P(\text{text} \mid \text{image})$

*"A Samoyed puppy coding on a laptop.*

- Image **generation** and image **understanding** are traditionally *separated* worlds.
- Can a fully **multimodal** model capable of **both generating and understanding** images?

*"A Samoyed puppy coding on a laptop.*

Text-based Generation  
Model:  
 $P(\text{image} \mid \text{text})$



# Multimodal Generative Models

- Other models tried to unify both directions, but not *pure diffusion*.

	Modality		Task		
	Image Backbone	Text Backbone	Image Gen	Image Cap.	Visual Question Answering
Versatile Diffusion [80]	<b>Diffusion</b>	Diff. + AR	✓	✓	✗
Unidiffuser [7]	<b>Diffusion</b>	Diff. + AR	✓	✓	✗
Show-O [79]	<b>Diffusion</b>	AR	✓	✓	✓
Transfusion [84]	<b>Diffusion</b>	AR	✓	✓	✓
Ours	<b>Diffusion</b>	<b>Diffusion</b>	✓	✓	✓

## Why diffusion though?

- Flexible **computational budget**: controllable generation quality.

## Why diffusion though?

- Flexible **computational budget**: controllable generation quality.
- Parallelizable **any-order generation**: not autoregressive, thus don't need a fixed left-to-right order.

## Why diffusion though?

- Flexible **computational budget**: controllable generation quality.
- Parallelizable **any-order generation**: not autoregressive, thus don't need a fixed left-to-right order.
- **Iterative refinement** of predictions: diffusion refines its prediction iteratively, thus stable and controllable.

## Background: Continuous Diffusion (Forward Process)

Consider the forward process from image  $\mathbf{x}_0$  to Gaussian noise  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$

$$\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}).$$

with  $\alpha_t, \sigma_t > 0$ ,  $\alpha_t/\sigma_t$  decreases monotonically. This process can be represented by ODE of velocity field  $\mathbf{v}(\mathbf{x}_t, t)$

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{v}(\mathbf{x}_t, t), \quad \mathbf{v}(\mathbf{x}_t, t) = \dot{\alpha}_t \mathbf{x} + \dot{\sigma}_t \epsilon.$$

$\alpha_t = 1 - t, \sigma = t$  would define a “straight line” between  $\mathbf{x}_0$  and  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ . In this case, we have  $\mathbf{v} = \epsilon - \mathbf{x}$ . The neural network for regressing the velocity field is trained by optimizing the **flow matching loss**

$$L_{\text{FM}} = \mathbb{E}_{t, q(\mathbf{x}_t | \mathbf{x})} \|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\epsilon - \mathbf{x})\|_2^2.$$

## Background: Continuous Diffusion (Backward Process)

Start from Gaussian noise  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$  and integrate the ODE *backward* in time:

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{v}_\theta(\mathbf{x}_t, t), \quad t : T \rightarrow 0.$$

Discretized form (Euler method):

$$\mathbf{x}_{t_{k-1}} = \mathbf{x}_{t_k} - \Delta t_k \mathbf{v}_\theta(\mathbf{x}_{t_k}, t_k), \quad t_K = T > \dots > t_0 = 0.$$

Optionally, higher-order solvers (e.g., Heun / Runge–Kutta) improve stability:

$$\hat{\mathbf{x}}_{t_{k-1}} = \mathbf{x}_{t_k} - \Delta t_k \mathbf{v}_\theta(\mathbf{x}_{t_k}, t_k),$$

$$\mathbf{x}_{t_{k-1}} = \mathbf{x}_{t_k} - \frac{\Delta t_k}{2} \left[ \mathbf{v}_\theta(\mathbf{x}_{t_k}, t_k) + \mathbf{v}_\theta(\hat{\mathbf{x}}_{t_{k-1}}, t_{k-1}) \right].$$

After integrating from  $t = T$  to  $t = 0$ , the final  $\mathbf{x}_0$  represents a generated image sample following  $p_{\text{data}}(\mathbf{x}_0)$ .

# Background: Discrete Diffusion (Mask)

**Problem:** Tokens are discrete; we cannot add Gaussian noise.

**Idea:** Corrupt information by *masking* tokens over time. This is the discrete analogue of “increasing noise variance” in continuous diffusion.

**Modeling primitives:**

- Vocabulary  $\mathcal{X} = \{1, \dots, N\}$ ; clean token is one-hot  $\mathbf{x} \in \{0, 1\}^N$ .
- A special one-hot  $\mathbf{m}$  denotes a *mask absorbing state*.
- A schedule  $\alpha_t \in (0, 1]$ , monotonically decreasing in  $t$ .

**Interpretation:** As  $t$  grows, information decreases: with probability  $\alpha_t$  keep  $\mathbf{x}$ , otherwise replace by  $\mathbf{m}$ .

## Background: Discrete Diffusion (Forward Marginal)

For each token position (positions are treated independently), define the forward marginal:

$$q(\mathbf{x}_t \mid \mathbf{x}) = \text{Cat}[\mathbf{x}_t \mid \alpha_t \mathbf{x} + (1 - \alpha_t) \mathbf{m}]$$

where  $\text{Cat}(\cdot \mid \pi)$  is a categorical distribution with probabilities  $\pi$ .

### Interpretation:

- $\alpha_t$ : probability the token remains unchanged at time  $t$ . This work uses  $\alpha_t = 1 - t$ .
- $(1 - \alpha_t)$ : probability it is masked.
- $q(\mathbf{x}_0 \mid \mathbf{x}) = \text{Cat}[\mathbf{x}]$  (clean),  $q(\mathbf{x}_1 \mid \mathbf{x}) = \text{Cat}[\mathbf{m}]$  (fully masked).

## Background: Discrete Diffusion (Posterior)

**Case 1:**  $\mathbf{x}_t \neq \mathbf{m}$ . The token is not masked at  $t$ . By absorption, it was also not masked at any  $s < t$ . By Postulate B,  $\mathbf{x}_s = \mathbf{x}_t$ . Hence the posterior is a degenerate categorical

$$q(\mathbf{x}_s | \mathbf{x}_t, \mathbf{x}) = \text{Cat}(\mathbf{x}_s | \mathbf{x}_t) \equiv \delta(\mathbf{x}_s - \mathbf{x}_t).$$

**Case 2:**  $\mathbf{x}_t = \mathbf{m}$ . At time  $t$  the token is masked. For  $s < t$  there are two possibilities

$$\mathbb{P}(\mathbf{x}_s = \mathbf{m} | \mathbf{x}_t = \mathbf{m}) = \frac{\mathbb{P}(\mathbf{x}_s = \mathbf{m}, \mathbf{x}_t = \mathbf{m})}{\mathbb{P}(\mathbf{x}_t = \mathbf{m})} = \frac{\mathbb{P}(\mathbf{x}_s = \mathbf{m})}{\mathbb{P}(\mathbf{x}_t = \mathbf{m})} = \frac{1 - \alpha_s}{1 - \alpha_t}, \quad \mathbb{P}(\mathbf{x}_s = \mathbf{x} | \mathbf{x}_t = \mathbf{m}) = \frac{\alpha_s - \alpha_t}{1 - \alpha_t}.$$

Thus

$$q(\mathbf{x}_s | \mathbf{x}_t=\mathbf{m}, \mathbf{x}) = \text{Cat}\left(\mathbf{x}_s \middle| \frac{(1 - \alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}}{1 - \alpha_t}\right).$$

## Background: Discrete Diffusion (Infinitesimal Posterior)

Let  $s = t - \Delta t$  with  $\Delta t \rightarrow 0$ .

**Case 1:**  $\mathbf{x}_t \neq \mathbf{m}$ . The token cannot become masked in the past (absorption). Therefore

$$q(\mathbf{x}_{t-\Delta t} | \mathbf{x}_t, \mathbf{x}) = \delta(\mathbf{x}_{t-\Delta t} - \mathbf{x}_t),$$

and contributes zero KL.

**Case 2:**  $\mathbf{x}_t = \mathbf{m}$ . Use the expansion  $\alpha_{t-\Delta t} = \alpha_t - \alpha'_t \Delta t + o(\Delta t)$

$$\mathbb{P}(\mathbf{x}_{t-\Delta t} = \mathbf{x}) = \frac{\alpha_{t-\Delta t} - \alpha_t}{1 - \alpha_t} = \frac{-\alpha'_t}{1 - \alpha_t} \Delta t + o(\Delta t),$$

$$\mathbb{P}(\mathbf{x}_{t-\Delta t} = \mathbf{m}) = 1 - \frac{-\alpha'_t}{1 - \alpha_t} \Delta t + o(\Delta t).$$

Define the **unmasking rate**

$$r_t^* := \frac{-\alpha'_t}{1 - \alpha_t} \quad (\geq 0).$$

Thus

$$q(\mathbf{x}_{t-\Delta t} | \mathbf{x}_t=\mathbf{m}, \mathbf{x}) = \begin{cases} r_t^* \Delta t, & \mathbf{x}, \\ 1 - r_t^* \Delta t, & \mathbf{m}. \end{cases}$$

# Background: Discrete Diffusion (Backward Parameterization)

Given the infinitesimal forward posterior, we now construct a model for the **reverse (generative) process**. When  $\mathbf{x}_t = \mathbf{m}$ , the forward process has an “unmasking” event with rate  $r_t^*$ . The reverse process should *choose which token to unmask to*.

**True infinitesimal posterior:**

$$q(\mathbf{x}_{t-\Delta t} | \mathbf{x}_t = \mathbf{m}, \mathbf{x}) = \begin{cases} r_t^* \Delta t, & \mathbf{x}, \\ 1 - r_t^* \Delta t, & \mathbf{m}. \end{cases}$$

**Model reverse transition:** We let the *same unmasking rate*  $r_t^*$  be used, but the model predicts **which token to unmask into**:

$$p_\theta(\mathbf{x}_{t-\Delta t} | \mathbf{x}_t = \mathbf{m}) = \begin{cases} r_t^* \Delta t (\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t) \cdot \mathbf{x}), & \mathbf{x}, \\ 1 - r_t^* \Delta t, & \mathbf{m}. \end{cases}$$

**Interpretation:**

- $r_t^*$ : when an “unmask” event happens.
- $\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t)$ : model’s distribution over tokens.
- $\mathbf{x}$  is one-hot, so  $\mathbf{x}_\theta \cdot \mathbf{x}$  is model’s probability for the true token.

## Background: Discrete Diffusion (Instantaneous KL Term)

To derive the training objective, compute the instantaneous **KL divergence** between true and model reverse transitions.

Since Case 1 ( $\mathbf{x}_t \neq \mathbf{m}$ ) gives a delta posterior, its KL is zero. Thus only the masked case contributes

$$\begin{aligned}\text{KL}(q\|p_\theta) &= \sum_{z \in \{x, m\}} q(z) \log \frac{q(z)}{p_\theta(z)} \\ &= r_t^* \Delta t \log \frac{r_t^* \Delta t}{r_t^* \Delta t (\mathbf{x}_\theta \cdot \mathbf{x})} + (1 - r_t^* \Delta t) \log \frac{1 - r_t^* \Delta t}{1 - r_t^* \Delta t} \\ &= r_t^* \Delta t \log \frac{1}{\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t) \cdot \mathbf{x}} + \mathcal{O}(\Delta t^2)\end{aligned}$$

Thus the **instantaneous KL density** (divide by  $\Delta t$ ) is

$$\boxed{\frac{\text{KL}}{\Delta t} = r_t^* [-\log(\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t) \cdot \mathbf{x})].}$$

## Background: Discrete Diffusion (NELBO Integral)

The total negative ELBO is the time integral of instantaneous KL

$$L_{\text{NELBO}} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[ \int_0^1 r_t^* \left[ -\log(\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t) \cdot \mathbf{x}) \right] dt \right].$$

Recall the unmasking rate

$$r_t^* = \frac{-\alpha'_t}{1 - \alpha_t}.$$

Plug into the integral

$$L_{\text{NELBO}} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[ \int_0^1 \frac{-\alpha'_t}{1 - \alpha_t} \left[ -\log(\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t) \cdot \mathbf{x}) \right] dt \right].$$

Removing the double negatives

$$L_{\text{NELBO}} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[ \int_0^1 \frac{\alpha'_t}{1 - \alpha_t} \log(\mathbf{x}_\theta(\mathbf{x}_t, \alpha_t) \cdot \mathbf{x}) dt \right].$$

## Background: Discrete Diffusion (Monte Carlo Training)

To implement the continuous objective, discretize time and sample timesteps during training.

Sampling  $t_i$  uniformly

$$t_1, \dots, t_K \in (\delta, 1].$$

Monte-Carlo approximation of NELBO

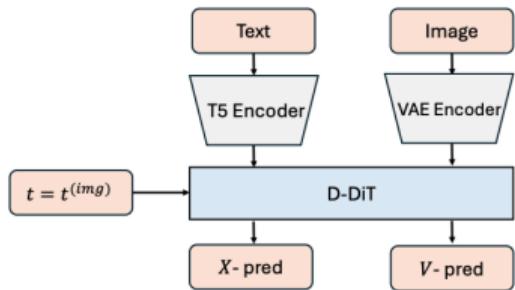
$$L_{\text{text}} \approx -\frac{1}{K} \sum_{i=1}^K \frac{\alpha'_{t_i}}{1 - \alpha_{t_i}} \log \left( \mathbf{x}_\theta(\mathbf{x}_{t_i}, \alpha_{t_i}) \cdot \mathbf{x} \right).$$

### Interpretation:

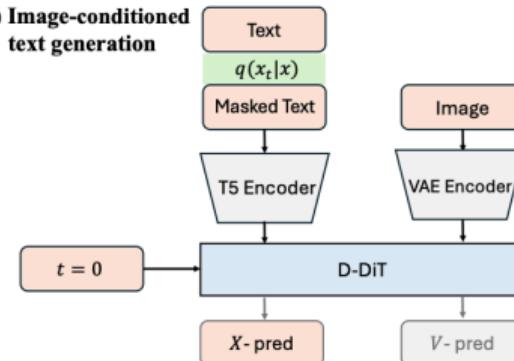
- At timestep  $t_i$ , we draw a masked-corrupted input  $\mathbf{x}_{t_i}$ .
- The model predicts the distribution over original tokens.
- The weighting  $\frac{\alpha'_{t_i}}{1 - \alpha_{t_i}}$  acts like a “time-dependent learning rate”.

# Method: Architecture

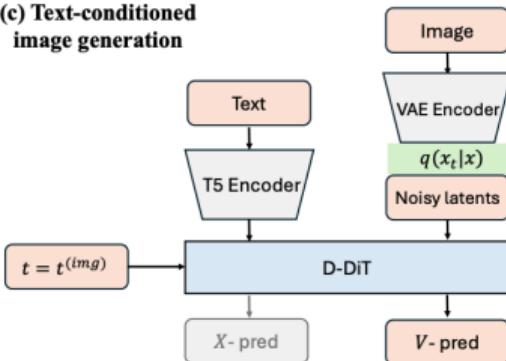
(a) Overview



(b) Image-conditioned  
text generation



(c) Text-conditioned  
image generation



## Image Branch:

- Inputs latents  $x_t^{(img)}$  from VAE encoder (lower resolution).
- Predicts **velocity**  $v_\theta$  (continuous diffusion target).

## Text Branch:

- Inputs masked text tokens  $x_t^{(txt)}$ .
- Predicts **token categorical distribution**  $x_\theta$  (discrete diffusion target).
- A bi-directional **text encoder** is stacked on top (no causal mask).

**Cross Attention:** Image tokens and text tokens attend to each other in every layer.

## Method: Training

Jointly train the model using two diffusion objectives:

- (1) Image loss: continuous flow-matching diffusion.

$$L_{\text{image}} = \mathbb{E}_{t, q^{(\text{img})}} \left\| \mathbf{v}_\theta(\mathbf{x}_t^{(\text{img})}, t, \mathbf{x}^{(\text{txt})}) - (\epsilon - \mathbf{x}^{(\text{img})}) \right\|_2^2$$

- (2) Text loss: discrete masked diffusion.

$$L_{\text{text}} = \mathbb{E}_{q^{(\text{txt})}} \left[ -\frac{1}{K} \sum_{i=1}^K \frac{\log (\mathbf{x}_\theta(\mathbf{x}_{t_i}^{(\text{txt})}, \mathbf{x}^{(\text{img})}) \cdot \mathbf{x})}{t_i} \right]$$

**Final joint loss:**  $L_{\text{dual}} = L_{\text{image}} + \lambda_{\text{text}} L_{\text{text}}$ .

# Method: Inference (Text-to-Image Generation)

**Goal:** Sample an image conditioned on text

$$\mathbf{x}^{(\text{img})} \sim p(\mathbf{x}^{(\text{img})} \mid \mathbf{x}^{(\text{txt})}).$$

**Approach:** Classifier-Free Guidance (CFG). We solve the ODE of image diffusion using a re-weighted velocity:

$$\tilde{\mathbf{v}}_t = s \mathbf{v}_\theta(\mathbf{x}_t^{(\text{img})}, t, \mathbf{x}^{(\text{txt})}) + (1 - s) \mathbf{v}_\theta(\mathbf{x}_t^{(\text{img})}, t, \emptyset)$$

**Interpretation:**

- $s$ : guidance scale (larger  $\rightarrow$  stronger text alignment).
- $\emptyset$ : null text embedding (unconditional branch).
- The model performs continuous image denoising, while text is kept fixed.

**Procedure:**

- ① Encode text into tokens  $\mathbf{x}^{(\text{txt})}$ .
- ② Sample noisy latent  $\mathbf{x}_T^{(\text{img})} \sim \mathcal{N}(0, I)$ .
- ③ Integrate ODE backward from  $T \rightarrow 0$  using  $\tilde{\mathbf{v}}_t$ .

# Method: Inference (Image-to-Text Generation)

**Goal:** Generate text conditioned on an image

$$\mathbf{x}^{(\text{txt})} \sim p(\mathbf{x}^{(\text{txt})} \mid \mathbf{x}^{(\text{img})}).$$

**Approach:** Use the trained text diffusion model to iteratively “unmask” tokens conditioned on image features. We sample from the posterior

$$q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x})$$

using the model’s prediction

$$\mathbf{x} \approx \mathbf{x}_\theta \left( \mathbf{x}_t^{(\text{txt})}, \mathbf{x}^{(\text{img})}; t = 0 \right).$$

**Procedure:**

- ① Start from masked sequence:

$$\mathbf{x}_1^{(\text{txt})} = \mathbf{m}.$$

- ② For  $t = 1 \rightarrow 0$ :

$$\mathbf{x}_{t-\Delta t}^{(\text{txt})} \sim q(\mathbf{x}_{t-\Delta t} \mid \mathbf{x}_t, \mathbf{x}_\theta(\mathbf{x}_t, \mathbf{x}^{(\text{img})})).$$

- ③ Output  $\mathbf{x}_0^{(\text{txt})}$ .

# Method: Inference (Image-to-Text In-filling)

**Goal:** Generate text *answers* conditioned on an image and a *given question*:

$$\mathbf{x}^{(\text{ans})} \sim p(\mathbf{x}^{(\text{ans})} \mid \mathbf{x}^{(\text{img})}, \mathbf{x}^{(\text{question})}).$$

**Approach:** Masked text diffusion with fixed question tokens.

**Setup:**

- Question tokens are fixed (never updated).
- Answer tokens start fully masked.
- Image features condition the denoising.

**Procedure:**

① Initialize:  $\mathbf{x}_1^{(\text{txt})} = [\text{question tokens}, \mathbf{m}, \mathbf{m}, \dots]$

② For  $t = 1 \rightarrow 0$ , sample:

$$\mathbf{x}_{t-\Delta t}^{(\text{txt})} \sim q(\mathbf{x}_{t-\Delta t} \mid \mathbf{x}_t, \mathbf{x}_\theta(\mathbf{x}_t, \mathbf{x}^{(\text{img})}))$$

but *keep question tokens fixed*.

③ Output  $\mathbf{x}_0^{(\text{txt})}$ .

# Experiments: Datasets

D-DiT is trained in three sequential stages using  $\sim 40M$  image–text pairs.

## Stage 1: Dual diffusion pretraining.

- Dataset: recaptioned DataComp-1B (30M images used).
- Text length: 64; image resolution: 256.
- Iterations: 60K, batch size 512.

## Stage 2: Continued pretraining on high-quality data.

- Datasets: ShareGPT4V pretrain set (1.3M), Recaptioned OpenImages subset (1.9M), re-captioned LAION-1024 (9M), Midjourney (1M)
- Text length: 256; image resolution: 256 or 512.
- Iterations: 200K (or +80K for 512 model).

## Stage 3: Visual instruction tuning to promote joint text-image conditioned text generation.

- LLaVA-Pretrain (558K) + LLaVA-v1.5-mix (665K).
- Add TextVQA and VizWiz training splits.
- Iterations: 50K.
- Use task-specific instruction prompts.

# Experiment: Results (Text-to-image Generation)

	Model	params (B)	Overall	Objects		Counting	Colors	Position	Color attribution
				Single	Two				
T2I	PixArt- $\alpha$ [10]	0.6	0.48	0.98	0.50	0.44	0.80	0.08	0.07
	SD V2.1	0.9	0.50	0.98	0.51	0.44	0.85	0.07	0.17
	DALL-E 2 [65]	6.5	0.52	0.94	0.66	0.49	0.77	0.10	0.19
	SDXL [60]	0.9	0.55	0.98	0.74	0.39	0.85	0.15	0.23
	DALL-E 3	-	0.67	0.96	0.87	0.47	0.83	0.43	0.45
Multimodal models	CoDI [74]	-	0.31	0.89	0.16	0.16	0.65	0.02	0.01
	LWM [51]	7	0.47	0.93	0.41	0.46	0.79	0.09	0.15
	SEED-X [25]	17	0.49	0.97	0.58	0.26	0.80	0.19	0.14
	Chameleon [75]	7	0.39	-	-	-	-	-	-
	Show-O [83]	1.3	0.68	0.98	0.80	0.66	0.84	0.31	0.50
	Transfusion [88]	8	0.67	-	-	-	-	-	-
	SD3 [20]	2	0.62	0.98	0.74	0.63	0.67	0.34	0.36
	D-DiT (ours)	2	0.65	0.97	0.80	0.54	0.76	0.32	0.50

# Experiment: Results (Text-to-image Generation)



A wooden lightbulb on a countertop.



A teapot made from tree bark set on the forest floor.



An extreme monochrome long exposure shot of Shibuya at rush hour.



A picturesque lakeside view from a balcony in Hallstatt at sunrise.



A floating castle surrounded by clouds at sunrise, with waterfalls cascading from its base.



A cozy, dimly-lit library with towering shelves, dusty books, and orbs of light floating gently above.



A peaceful forest clearing bathed in soft moonlight, with glowing mushrooms nestled in the trees.



A neon-lit cyberpunk street bustling with futuristic characters, rain-soaked pavement reflecting vivid signs and holograms.

# Experiment: Results (Multi-modal Understanding)

	<b>Model</b>	<b>Params</b> # trainable	<b>Text</b> Backbone	<b>Image</b> Backbone	<b>MS-COCO</b> CIDEr ↑	<b>VQAv2</b> Acc. ↑	<b>VizWiz</b> Acc. ↑	<b>OKVQA</b> Acc. ↑	<b>MME</b> Acc. ↑	<b>GQA</b> Acc. ↑	<b>POPE</b> Acc. ↑
Unified model	InternVL-2.0 [13]	8B	AR	-	-	-	62.9	62.9	1648.1	61.0	86.9
	LLaVA-Next [49]	13B	AR	-	-	82.8	60.5	-	1575.0	65.4	86.2
	BLIP-2 [41]	13B	AR	-	-	65.0	19.6	-	1293.8	41.0	85.5
	IDEFICS [38]	9B	AR	-	-	50.9	-	-	-	-	-
	QWEN-VL [6]	7B	AR	-	-	78.2	38.9	-	1487.5	57.5	-
	OpenFlamingo [5]	9B	AR	-	65.5	43.5	-	-	-	-	-
	Flamingo [2]	9B	AR	-	79.4	51.8	28.8	44.7	-	-	-
Bi-directional model	CM3Leon [86]	7B	AR	AR	61.6	47.6	37.6	23.8	-	-	-
	Chameleon [75]	7B	AR	AR	18.0	-	-	-	-	-	-
	LWM [52]	7B	AR	AR	-	55.8	11.6	-	-	44.8	75.2
	Show-O (256×256) [83]	1.3B	AR	Diffusion	-	64.7	-	-	1014.9	54.2	76.2
	Show-O (512×512) [83]	1.3B	AR	Diffusion	-	69.4	-	-	1097.2	58.0	80.0
	Transfusion [88]	7B	AR	Diffusion	29.0	-	-	-	-	-	-
D-DiT (Ours, 256×256)		2B	Diffusion	Diffusion	-	59.5	19.4	28.5	897.5	55.1	79.2
D-DiT (Ours, 512×512)		2B	Diffusion	Diffusion	56.2	60.1	29.9	25.3	1124.7	59.2	84.0

# Experiment: Results (Multi-modal Understanding)



Human: Describe the furnishings in this room.

Assistant: In the bedroom, there is a blue and white large bed, a rug, and a chair. The room is also decorated with yellow curtains.



Human: What is unusual or funny about this image?

Assistant: The unusual and funny element of it is that it features an orange traffic cone with small white eyes on it. This cute appearance adds a playful and amusing touch to the otherwise plain and ordinary.



Human: Can you describe the subjects of this photograph?

Assistant: The photograph features two elephants, a mother elephant and a baby elephant, standing next to each other and taking advantage of a sunny day.



Human: Can you identify this landmark? What are some features of this area?

Assistant: The landmark in the area is the Golden Gate Bridge, which is a long red suspension bridge, standing tall against the backdrop of a beautiful bay in the water. The bridge is adjacent to buildings on either side, making it a focal point in the landscape. The area also includes an active, bustling bay where people can enjoy boats, sightseeing, and enjoy the view of the waterway, indicating that the location might be in the popular coastal spot of San Francisco.



Human: What is the dish in the photo?

Assistant: The dish in the photo is a square dish filled with roasted fish, with orange wedges used as garnish.

# Experiment: Ablation Studies

**Motivation.** Diffusion models are trained on massive text–image pairs. *But do their internal image features transfer to multimodal understanding?*

## Study 1: Use diffusion image features + GPT2 to caption images.

- Compare SD3 features (frozen / trainable), CLIP features, and D-DiT.
- Pipeline: **Vision Encoder → GPT2 Decoder → Caption → GPT4 VQA scoring.**

### Findings:

- SD3 diffusion features give **poor** VQA accuracy (42–50).
- Even when SD3 is trainable, performance remains far worse than CLIP (50.6).
- **D-DiT achieves 55.0 / 60.3 without any GPT2 decoder.**

**Conclusion 1:** Raw diffusion features do not align with text embedding spaces. They are excellent for generation, but not for reasoning or language decoding.

## Study 2: Vary the text diffusion sampling steps.

- VQA (short answers): accuracy plateaus with few steps.
- Captioning (long text): performance improves with more steps.

**Conclusion 2:** Text masked diffusion truly models the text likelihood. More sampling steps ⇒ lower perplexity ⇒ better captions.

# Experiment: Ablation Studies

Vision Encoder	Language Decoder	VQAv2 (val)	
		0-shot	32-shot
SD3 feature (frozen)	GPT 2	42.3	46.9
SD3 feature (trainable)	GPT 2	45.1	50.2
CLIP ViT L/14 (frozen)	GPT 2	50.6	54.8
UniDiffuser [7]	GPT 2*	46.7	49.4
D-DiT (ours)	-	<b>55.0</b>	<b>60.3</b>

Task	$T = 4$	8	16	32	64	128
VQAV2 (acc.)	58.8	58.0	59.3	60.5	60.0	59.6
MS-COCO (CIDEr)	20.2	35.3	46.5	51.3	56.2	54.5

# Accommodation

## Is it worth reading? Yes!

- Provides a clean and unified view of continuous diffusion and discrete masked diffusion.
- The training protocol is clearly written: forward corruption, dual loss design, joint DiT architecture.
- Helpful for understanding how bidirectional multimodal diffusion works (T2I, I2T, in-filling).

## Is it worth implementing? Maybe.

- The pretrained checkpoints in their Github repository make it easy to experiment.
- Discrete diffusion is still weaker than autoregressive LMs.
- For my use case: need to think carefully about how to generate two modalities in the mean time (e.g., image + clinical text).