

A ADDITIONAL GENERATION EXAMPLES

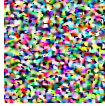
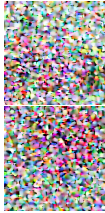



Generated Text	Generated Images
<p>PROMPT: I want to paint flowers but I need a photo for inspiration. Can you show me two pictures of flowers and describe them?</p> <p>$t = 0.2$: first a pink of out in appealing roses and overall the. seems be the, appreciation for. a pink a flower pink with possibly vase is base the flower appeal simple backdrop the point of the image</p> <p>$t = 0.4$: In first, a arrangement pink flowers is, of six flowers out in appealing manner.< image > roses are delicate and pink the overall of the. The bouquet seems be the captured,oking and appreciation for the nature. This image a view pink sitting a table flower be ger with a pink with prominent center possibly a. vase is base the flower visual appeal table a simple backdrop, allowing flower be the point of the image.</p> <p>$t = 0.6$: In first, a arrangement of pink flowers is, of six flowers out in a appealing manner.< image > roses are delicate and have a, pink, the overall of the. The bouquet seems be focus the image captured, showcasing their The display and atmosphere,oking of and appreciation for the nature. This image a view of a pink sitting a table.< image > flower appears be ger with a pink with prominent center possibly a bud. vase is the base the flower, enhancing visual appeal of the. The table vase a simple backdrop, allowing flower be the point of the image.</p> <p>$t = 0.8$: In this first image, a beautiful arrangement of pink flowers is on display, with total of six flowers spread out in a appealing manner.< image >The roses are delicate and have a soft, pink hue, which adds the overall charm of the scene. The bouquet seems be the main focus of the image flowers captured a close shot, showcasing their intricate details. The display and atmosphere, evoking a sense of romance and appreciation for the beauty nature. This image a view of a pink flower sitting a glass table.< image >The flower appears be ger with a pink color with a a prominent center, possibly a bud. vase is the base of the flower, enhancing visual appeal of the. The table the vase provides a simple backdrop, allowing flower to be the focal point of the image.</p> <p>$t = 1.0$: In this first image, a beautiful arrangement of pink flowers is on display, with a total of six flowers spread out in a visually appealing manner.< image >The roses are delicate and have a soft, pink hue, which adds to the overall charm of the scene. The bouquet seems to be the main focus of the image, with the flowers captured in a close-up shot, showcasing their intricate details. The display creates an elegant and serene atmosphere, evoking a sense of romance and appreciation for the beauty of nature. This image features a close-up view of a beautiful pink flower sitting in a glass vase on a table.< image >The flower appears to be a gerbera, with a vibrant pink color, with a vibrant pink color and a prominent center, possibly a bud. The glass vase is positioned at the base of the flower, enhancing the visual appeal of the scene. The table beneath the vase provides a simple yet elegant backdrop, allowing the flower to be the focal point of the image.</p> <p>$t = 2.0$: In this first image, a beautiful arrangement of pink flowers is on display, with a total of six flowers spread out in a visually appealing manner.< image >The roses are delicate and have a soft, pink hue, which adds to the overall charm of the scene. The bouquet seems to be the main focus of the image, with the flowers captured in a close-up shot, showcasing their intricate details. The display creates an elegant and serene atmosphere, evoking a sense of romance and appreciation for the beauty of nature. This image features a close-up view of a beautiful pink flower sitting in a glass vase on a table.< image >The flower appears to be a gerbera, with a vibrant pink color, with a vibrant pink color and a prominent center, possibly a bud. The glass vase is positioned at the base of the flower, enhancing the visual appeal of the scene. The table beneath the vase provides a simple yet elegant backdrop, allowing the flower to be the focal point of the image.</p>	<p>No image</p>     

Figure 8: Example interleaved generation process showing flowers. An animated version is included in the supplementary material.

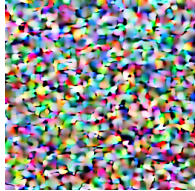
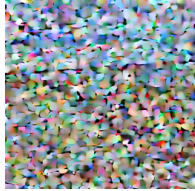

Generated Text	Generated Images
PROMPT: I'm looking for a new hobby that's relaxing and will allow me to be outdoors to get some fresh air. Offer me a suggestion and show me an image of the type of place I can do it	
$t = 0.2$: ? It's that. senseness tranqu promotes.	No image
$t = 0.4$: Why try? It's a that you a wide of, lake trail. in can create sense-ness tranqu promotes stress and.	No image
$t = 0.6$: Why not try nature? It's a that you a wide variety of natural, like this lake trail,. in can create sense ofness and tranqu promotes stress and.	No image
$t = 0.8$: Why not try nature? It's a relaxing that allows you to in a wide variety of natural environments, like this lake trail, for instance.< image >Walking in can create sense of calmness and tranqu that promotes stress and relaxation.	
$t = 1.0$: Why not try nature walking? It's a relaxing hobby that allows you to immerse yourself in a wide variety of natural environments, like this lake trail, for instance.< image >Walking in nature can create a sense of calmness and tranquility that promotes stress reduction and relaxation.	
$t = 2.0$: Why not try nature walking? It's a relaxing hobby that allows you to immerse yourself in a wide variety of natural environments, like this lake trail, for instance.< image >Walking in nature can create a sense of calmness and tranquility that promotes stress reduction and relaxation.	

Figure 9: Example interleaved generation process showing outdoor exercise. In this example, the image is inserted late in the generation process when the text is almost fully denoised. An animated version is included in the supplementary material.

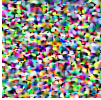







Generated Text	Generated Images
<p>PROMPT: I am teaching an ESL lesson on home types. Please show me images of a bungalow, a two-story home and a townhouse that I can show my students, and briefly explain about each of them.</p> <p>t = 0.2: good. **B is typically for with a-p simple including suburban towns. is of a.Two** type residential building levels with the typically bedrooms on the. example-story. **Town town unit town. They vary typically than homes image example.< image ></p> <p>t = 0.4: which good show to them about different types1. **Bungalow** ungalow is of is typically cosy designed for. Bungal are single-story, with a-p simple can including suburban neighbourhood towns and areas. Below is of a.Two-** A type residential building levels, with the typically on bedrooms on the. below example of-story. **Town A town multi residential unit that town walls. They can vary size are typically smaller than single homes image is example of.< image ></p> <p>t = 0.6: images which would good to show students to them about different types: 1. **Bungalow** ungalow is a type of is typically small, cosy and designed for comfortable. Bungal are often single-story, with a-p simple can found various locations including suburban neighbourhood towns and areas. Below is an image of a.Two-** A two a type residential building has levels, with the typically on the first floor and bedrooms and additional on the. image below an example of-story. **Town A town multi-level residential unit that attached town shared walls. They can vary size, are typically smaller than single homes. The image is example of of.< image ></p> <p>t = 0.8: are images which would be good to show your ESL students to them about different types: 1. **Bungalow** A bungalow is a type of that is typically small, cosy and designed for comfortable living. Bungalows are often single-story, with a-pitched and a simple floor can be found various locations, including suburban neighbourhoods, beach towns and areas. Below is an image of a b2.Two-Story home** A two home a type residential building has levels or floors, with the typically on the first floor and bedrooms and additional spaces on the second. image below is an example of a two-story home.3. **Townhouse** A town a multi-level residential unit that attached one more other town shared walls. They can vary size, but are typically smaller than single-family homes. The image below is example of a of-houses.< image ></p> <p>t = 1.0: Here are three images which would be good to show your ESL students to teach them about different home types: 1. **Bungalow** A bungalow is a type of house that is typically small, cosy, and designed for comfortable living. Bungalows are often single-story homes, with a low-pitched roof and a simple, open floor plan. They can be found in various locations, including suburban neighbourhoods, beach towns, and rural areas. Below is an image of a bungalow:< image >2. **Two-Story home** A two-story home is a type of residential building that has two levels or floors, with the main living areas typically on the first floor and bedrooms and additional living spaces on the second floor. The image below is an example of a two-story home.< image >3. **Townhouse** A townhouse is a multi-level residential unit that is attached to one or more other townhouses via shared walls. They can vary in size, but are typically smaller than single-family homes. The image below is an example of a row of townhouses.< image ></p> <p>t = 2.0: Here are three images which would be good to show your ESL students to teach them about different home types: 1. **Bungalow** A bungalow is a type of house that is typically small, cosy, and designed for comfortable living. Bungalows are often single-story homes, with a low-pitched roof and a simple, open floor plan. They can be found in various locations, including suburban neighbourhoods, beach towns, and rural areas. Below is an image of a bungalow:< image >2. **Two-Story home** A two-story home is a type of residential building that has two levels or floors, with the main living areas typically on the first floor and bedrooms and additional living spaces on the second floor. The image below is an example of a two-story home.< image >3. **Townhouse** A townhouse is a multi-level residential unit that is attached to one or more other townhouses via shared walls. They can vary in size, but are typically smaller than single-family homes. The image below is an example of a row of townhouses.< image ></p>	       

Figure 10: Example interleaved generation process showing home types. An animated version is included in the supplementary material.

Input image	Captions with varying classifier-free guidance weights
	<p>[CFG 0.0:] A llama and a horse standing in a field.</p> <p>[CFG 2.0:] A white llama and a white horse standing in a field.</p>
	<p>[CFG 0.0:] A group of toy animals sitting on a table.</p> <p>[CFG 1.0:] A table topped with a variety of toy animals, including a giraffe , a cow , and a bird , as well as a toy duck . In the background, there is a frame attached to the wall.</p> <p>[CFG 1.5:] A green table topped with toy animals, including a giraffe, a cow, a yellow bird, and a toy duck, next to a red box . In the background, there is a photo frame attached to the wall.</p>
	<p>[CFG 0.0:] A bathroom with a large tub and a sink.</p> <p>[CFG 1.0:] A bathroom with a claw foot tub and three windows .</p> <p>[CFG 2.5:] A brown bathroom or master bathroom with a classic claw foot tub and three windows.</p>
	<p>[CFG 0.0:] A cat sitting on a wooden deck looking up.</p> <p>[CFG 1.0:] A cat sitting on a wooden deck looking at its reflection in a window.</p> <p>[CFG 2.5:] Two fluffy ginger and white cats sit and gaze at their reflection in a glass window on a green wooden deck in Japan .</p>
	<p>[CFG 0.0:] A white plate topped with a cake and a spoon.</p> <p>[CFG 1.0:] A plate with a dessert and two spoons on it.</p> <p>[CFG 2.0:] A white plate topped with ice cream , accompanied by two spoons, a bottle , a glass , and a tissue paper on the table .</p> <p>Through the glass window in the background, we can see the water and the sky.</p>
	<p>[CFG 0.0:] A glass bowl filled with colorful paper cranes.</p> <p>[CFG 1.0:] Colorful origami cranes in a glass bowl shaped like a heart .</p> <p>[CFG 2.0:] A table with a heart-shaped bowl filled with colorful origami cranes in various colors . The background is slightly blurred , giving the focus to the vibrant colors of the cranes.</p>

Figure 11: Text generation examples from OneFlow, which allows the use of classifier-free guidance (CFG). We observe that CFG produces longer and more **detailed** captions and also increased chance of **hallucinations** . Highlighted text show increased levels of detail when using higher CFG weights.

B FULL DERIVATIONS

We provide the derivations of the model here. We briefly summarize the Edit Flow (Havasi et al., 2025) formulation and derivation, and then derive the interleaved time schedule when insertions and image denoising are performed simultaneously.

Setup. We make use of a blank token ε to denote empty spaces within a sequence. This token is only used for tracking token deletions during training and is not part of the vocabulary. Let $\mathcal{Z} = \bigcup_{n=0}^N (\mathcal{T} \cup \{\varepsilon\})^n$ be an extended space of aligned sequences. Furthermore, define $f_{\text{rm-blanks}} : \mathcal{Z} \rightarrow \mathcal{X}$ as the function that removes all blank tokens from the sequence. Lastly, we define the delta function over sequences $\delta_{z_1}(z_2) = \prod_i \delta_{z_1^i}(z_2^i)$ which is one if all tokens are the same otherwise zero (i.e. Kronecker’s delta function).

Continuous-time Markov chain (CTMC). A CTMC is a continuous-time discrete-space process which iteratively jumps between discrete values, with transitions

$$\mathbb{P}(X_{t+h}|X_t) = \delta_{X_t}(X_{t+h}) + hu_t(x|X_t) + o(h), \quad (13)$$

where u_t can be interpreted as a first-order characterization of the transition kernel. Since with insertions, the sequence lengths of X_t can change over time. To simplify notation, Havasi et al. (2025) used an augmented space of (X_t, Z_t) , where it is basically always enforced that $X_t = f_{\text{rm-blanks}}(Z_t)$. The role of Z_t is only for training, to keep track of which tokens are deleted and to compute the loss, and it is neither seen by the model nor used during sampling.

To briefly summarize the construction below, the Flow Matching recipe makes use of a prescribed conditional CTMC that generates single data sequences, which is then marginalized over the data distribution. The resulting marginal CTMC will then sample from the data distribution.

Conditional probability path. Given a data sequence $X_1 \sim p_{\text{data}}$, we prescribe a conditional probability path over Z_t of the same sequence length which interpolates between the empty sequence and this data sequence. We then obtain X_t by applying the $f_{\text{rm-blanks}}$ function. Concretely, we can express the conditional probability path as

$$p_t(X_t, Z_t|X_1) = p_t(X_t|Z_t, X_1) \cdot p_t(Z_t|X_1) \quad (14)$$

$$= p_t(X_t|Z_t) \cdot p_t(Z_t|X_1) \quad (15)$$

$$= \delta_{f_{\text{rm-blanks}}(Z_t)}(X_t) \cdot \left(\prod_{i=1}^n (1 - \kappa_t) \delta_{\varepsilon}(Z_t^i) + \kappa_t \delta_{X_1^i}(Z_t^i) \right), \quad (16)$$

where κ_t is a scheduler where $\kappa_0 = 0, \kappa_1 = 1$, and n is the sequence length of X_1 . In English, (16) is a mixture distribution where each token Z_t^i can either be equal to ε with probability $1 - \kappa_t$ or equal to data value X_1^i with probability κ_t .

Conditional CTMC rate. As discussed in Havasi et al. (2025), a conditional CTMC that samples from this conditional probability path can be constructed as

$$u_t(x, z|X_t, Z_t, X_1) = \left(\sum_{i=1}^n \frac{\dot{\kappa}_t}{1 - \kappa_t} (\delta_{X_1^i}(z^i) - \delta_{Z_t^i}(z^i)) \right) \delta_{f_{\text{rm-blanks}}(z)}(x), \quad (17)$$

where $x = \text{ins}(X_t, i, a)$ for some $i \in [n]$ and $a \in [M]$

which denotes the infinitesimal change in probability of going from the state $(X_t, Z_t) \rightarrow (x, z)$, constrained to next sequences x that are one token insertion difference from X_t . In English, (17) assigns a rate of $\frac{\dot{\kappa}_t}{1 - \kappa_t}$ if Z_t^i is not yet equal to X_1^i ; otherwise, it is zero. This ensures that a sample starting with all blanks $Z_0 = [\varepsilon, \dots, \varepsilon]$ at $t = 0$ will eventually turn into X_1 at $t = 1$. This ratio $\frac{\dot{\kappa}_t}{1 - \kappa_t}$ is the infinitesimal rate that each token changes its value, matching the distribution imposed by the scheduler κ_t , and conditioned on that it is still the ε token at time t .

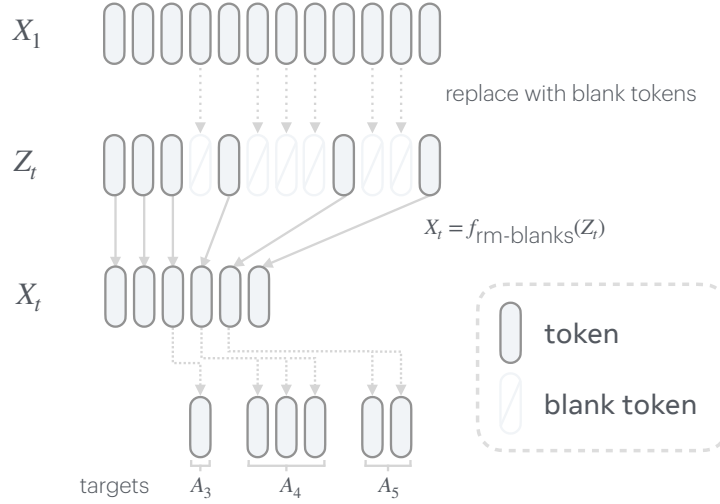


Figure 12: During training we construct Z_t by replacing tokens with the blank token (ε), with the original tokens used to construct the target bag-of-tokens \mathcal{A}_i .

Training loss. In order to train a model that transport sequences via insertions,

$$u_t^\theta(x|X_t), \quad \text{where } x = \text{ins}(X_t, i, a) \text{ for some } i \text{ and } a \quad (18)$$

we would need to marginalize out the auxiliary process Z_t and the data X_1 . Havasi et al. (2025) showed this can be done by using a loss based on any Bregman divergence while summing up over all possible sequences z such that $x = f_{\text{rm-blanks}}(z)$. Concretely, given a convex function ϕ that defines a Bregman divergence $D_\phi(a, b) = \phi(a) - \phi(b) - \langle a - b, \frac{d}{db}\phi(b) \rangle$, we can use the loss

$$\mathbb{E}_{X_t, Z_t \sim p_t(X_t, Z_t | X_1), X_1 \sim p_{\text{data}}} D_\phi\left(\sum_z u_t(\cdot, z | X_t, Z_t, X_1), u_t^\theta(\cdot | X_t)\right). \quad (19)$$

Plugging in the entropy $\phi(u) = \langle u, \log u \rangle$, this results in the Edit Flow loss

$$\mathbb{E}_{t, p_t(X_t, Z_t | X_1), X_1 \sim p_{\text{data}}} \left[\sum_{x \neq X_t} u_t^\theta(x | X_t) - \sum_{i=1}^n \mathbf{1}_{[Z_t^i = \varepsilon]} \frac{\dot{\kappa}_t}{1 - \kappa_t} \log u_t^\theta(\text{ins}(X_t, j, X_1^i) | X_t) \right], \quad (20)$$

where j is the position in X_t that corresponds to the first non- ε token on the left of Z_t^i . This ensures that inserting at the i -th position corresponds to changing the value of Z_t^i from ε to X_1^i .

Loss simplification. We deviate from Havasi et al. (2025) and use a t -independent parameterization. In particular, for sequences x that are one token insertion of X_t , i.e., $x = \text{ins}(X_t, i, a)$, we use

$$u_t^\theta(\text{ins}(X_t, i, a) | X_t) = \frac{\dot{\kappa}_t}{1 - \kappa_t} \lambda^i(X_t) Q^i(a | X_t), \quad (21)$$

where the neural network parameterizes λ and Q . Using this parameterization, letting \mathcal{A}_j be the set of missing tokens to the right of position j of X_t , the training loss (20) can be decomposed into

$$\mathbb{E}_{(\dots)} \left(\frac{\dot{\kappa}_t}{1 - \kappa_t} \right) \left(\sum_{j=1}^{n(X_t)} \lambda^j(X_t) - \sum_{j=1}^{n(X_t)} \sum_{a \in \mathcal{A}_j} \log(\lambda^j(X_t) Q^j(a | X_t)) \right) \quad (22)$$

$$= \mathbb{E}_{(\dots)} \left(\frac{\dot{\kappa}_t}{1 - \kappa_t} \right) \sum_{j=1}^{n(X_t)} \underbrace{\left(\lambda^j(X_t) - |\mathcal{A}_j| \log \lambda^j(X_t) \right)}_{(4)} + \underbrace{\sum_{a \in \mathcal{A}_j} \log Q^j(a | X_t)}_{(6)} + \text{const.} \quad (23)$$

which recovers the losses for λ and Q in (4) and (6) respectively, after removing the coefficient $\frac{\dot{\kappa}_t}{1 - \kappa_t}$. While keeping this coefficient relates the loss to an evidence lower bound (Havasi et al., 2025), we found that removing this coefficient in the loss gave better results in practice.

B.1 INTERLEAVED TIME SCHEDULE

In order to model image insertions, we would make a choice. We can either (i) fully denoise images at the time of insertion, or (ii) insert only noise and denoise later. We choose the latter approach, as this allows simultaneous generation across images and text, and provides the best parallelism as only a single model forward at each step is needed for both modalities. Without loss of generality, assume there is only a single image.

Generation starts by advancing the sequence time, denoted $t_{\text{text}} = 0$. When the image is inserted, we associate the image with its own time t_{img} .

The main difficulty is that we can not simply set t_{img} and t_{text} independently during training, as evidently we always have $t_{\text{text}} \geq t_{\text{img}}$. In fact, an independent scheduler induces the wrong distribution for our insertion prediction, and it will not insert the correct distribution at generation time. Instead, we need to ensure that training and generation see the same distribution of time values. To achieve this, we first note that the image exists in the sequence according to the scheduler κ_t , which means that the *insertion times* are distributed according to

$$p(t_{\text{insert}}) = \dot{\kappa}_t, \quad (24)$$

where t_{insert} is the time at which an image is inserted, *i.e.*, κ_t is the cumulative distribution function (CDF) of the insertion times. Equivalently, to sample the insertion time, we can apply the inverse CDF sampling,

$$t_{\text{insert}} = \kappa^{-1}(u), \quad u \sim \text{Unif}(0, 1). \quad (25)$$

If we set $t_{\text{img}} = 0$ when an image is inserted, then the difference between t_{text} and t_{img} is distributed according to the insertion time. This gives us the relation

$$t_{\text{text}} - t_{\text{img}} = t_{\text{insert}} \quad (26)$$

when $0 \leq t_{\text{text}}, t_{\text{img}}, t_{\text{insert}} \leq 1$. Since t_{text} will reach 1 before t_{img} , and we want to train for the entire process until $t_{\text{img}} = 1$, we can construct an extended time interval

$$\tau_{\text{text}} \in [0, 2], \quad t_{\text{text}} = \text{clip}(\tau_{\text{text}}), \quad (27)$$

where $\text{clip}(\tau) = \min\{1, \max\{0, \tau\}\}$ clips the time values back into the interval $[0, 1]$.

During training, we first sample τ_{text} , then sample

$$\tau_{\text{img}} = \tau_{\text{text}} - \kappa^{-1}(u), \quad u \sim \text{Unif}(0, 1). \quad (28)$$

This will sample an extended time for the image in the interval $[-1, 2]$. If $\tau_{\text{img}} < 0$, then it has not yet been inserted, hence it is deleted from the sequence. Otherwise, it is clip,

$$t_{\text{img}} = \text{clip}(\tau_{\text{img}}), \quad (29)$$

and we proceed to use the Flow Matching loss (9) to train the image denoising.

C FRAMEWORK COMPARISON

<i>Method</i>					
Text	AR	AR	Masked Diff	Discrete FM	Edit Flow
Image	AR	Diff / FM	Masked Diff	Discrete FM	FM
<i>Training</i>					
Attention Mask	Casual	Block Casual	Bidirectional	Bidirectional	Bidirectional
# Tokens per iter.	Seq Len	up to 2x seq Len	Seq Len	Seq Len	≈50% Seq Len
<i>Capabilities</i>					
Image understanding	✓	✓	✓	✓	✓
Single image generation	✓	✓	✓	✓	✓
Variable length generation	✓	✓	×	×	✓
Interleaved generation	×	✓	×	×	✓
Models	Chameleon, JanusPro	Transfusion, Bagel	MMaDA	FUDOKI	OneFlow

Table 3: High-level comparison of different frameworks for combining text and image generation.

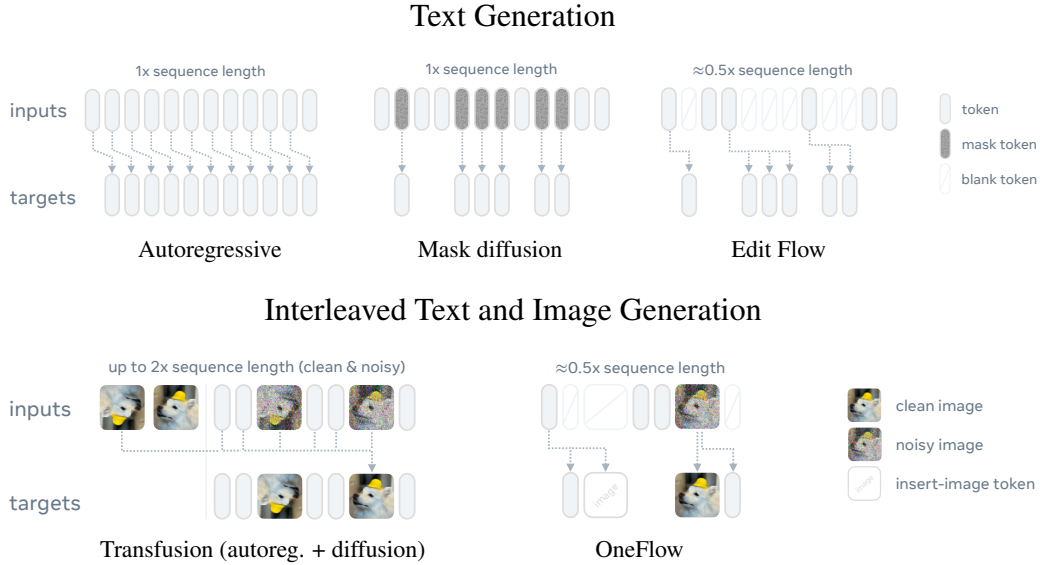


Figure 13: Illustration of the model input and targets during training for (*top*) text generation and (*bottom*) interleaved generation. To train autoregressive with diffusion denoising, the images are typically duplicated so that both the clean and the noisy images are in the sequence. On the other hand, OneFlow deletes tokens and images during training which reduces the sequence length.

D ONEFLOW ARCHITECTURE

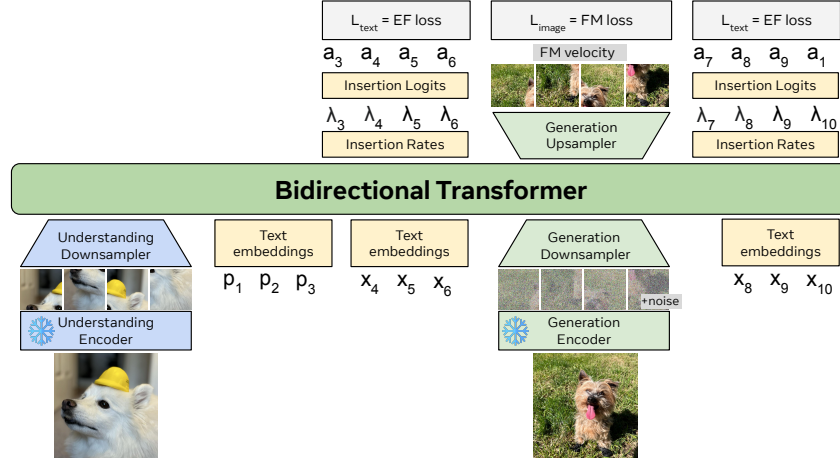


Figure 14: **Architecture.** With a multimodal prompt, OneFlow can produce variable length generations with interleaved text & images in a unified non-autoregressive sequence model, simultaneously generating all modalities with an interleaved time schedule for each generated image and text.

E ALGORITHMS

Algorithm 1 OneFlow interleaved text–image generation.

```

1: function ONEFLOWGENERATION(step size  $\Delta t$ , schedule  $\kappa$ )
2:    $X \leftarrow$  empty sequence  $\triangleright$  Text tokens (initially empty set)
3:    $\mathcal{I} \leftarrow \emptyset$   $\triangleright$  Set of image latents with per-image times
4:    $t_{\text{text}} \leftarrow 0$ 
5:   while  $t_{\text{text}} < 1$  or  $\exists Y \in \mathcal{I} : t_{\text{img}}(Y) < 1$  do
6:      $X, \mathcal{I}, t_{\text{text}}, t_{\text{img}} \leftarrow \text{ONEFLOWSTEP}(X, \mathcal{I}, t_{\text{text}}, t_{\text{img}}, \Delta t, \kappa)$ 
7:   end while
8:   return  $X$  and  $\{\text{VAEDec}(Y) : Y \in \mathcal{I}\}$   $\triangleright$  Decode VAE latents into image space
9: end function

```

Algorithm 2 OneFlow step function.

X is the token sequence, \mathcal{I} is the set of image latents each with time $t_{\text{img}}(Y)$.

```

1: function ONEFLOWSTEP( $X, \mathcal{I}, t_{\text{text}}, t_{\text{img}}, \Delta t, \kappa$ )
2:    $(\{\pi, \lambda_{\text{nonzero}}, Q\}, \{v(Y, \cdot)\}_{Y \in \mathcal{I}}) \leftarrow \text{OneFlowModel}(X, \mathcal{I}, t_{\text{img}})$ 

3:   for all  $Y \in \mathcal{I}$  with  $t_{\text{img}}(Y) < 1$  do  $\triangleright$  Image: Flow matching step on images
4:      $\Delta t_{\text{img}} \leftarrow \min\{1 - t_{\text{img}}(Y), \Delta t\}$ 
5:      $Y \leftarrow Y + \Delta t_{\text{img}} \cdot v(Y, t_{\text{img}}(Y))$ 
6:      $t_{\text{img}}(Y) \leftarrow t_{\text{img}}(Y) + \Delta t_{\text{img}}$ 
7:   end for

8:    $\Delta t_{\text{text}} \leftarrow \min\{1 - t_{\text{text}}, \Delta t\}$ 
9:   if  $\Delta t_{\text{text}} > 0$  then
10:    for all positions  $i \in \{1, \dots, n(X)\}$  do  $\triangleright$  Text: parallel insertions
11:       $p_i^\pi \leftarrow 1 - \pi^i$   $\triangleright$  If using (4) without  $\pi$ , then skip this step
12:       $p_i^\lambda \leftarrow \Delta t_{\text{text}} \cdot \frac{\dot{\kappa}(t_{\text{text}})}{1 - \kappa(t_{\text{text}})} \cdot \lambda_{\text{nonzero}}^i$ 
13:      do-insert  $\leftarrow \text{Bernoulli}(p_i^\pi)$  and  $\text{Bernoulli}(p_i^\lambda)$ 
14:      if do-insert then
15:         $a \sim Q^i(\cdot | X)$ 
16:         $X \leftarrow \text{ins}(X, i, a)$ 
17:        if  $a = \langle \text{image} \rangle$  then
18:           $Y \sim \mathcal{N}(0, I), t_{\text{img}}(Y) \leftarrow 0, \mathcal{I} \leftarrow \mathcal{I} \cup \{Y\}$ 
19:        end if
20:      end if
21:    end for
22:  end if

23:   $t_{\text{text}} \leftarrow t_{\text{text}} + \Delta t_{\text{text}}$ 
24:  return  $X, \mathcal{I}, t_{\text{text}}, t_{\text{img}}$ 
25: end function

```

Algorithm 3 OneFlow training loss with interleaved schedule

```

1: function ONEFLOWTRAININGSTEP(data sequence  $X$ , image latents  $\mathcal{I}$ , schedule  $\kappa$ )
2:    $\tau_{\text{text}} \sim \text{Unif}[0, 2]$ 
3:    $t_{\text{text}} \leftarrow \min\{1, \tau_{\text{text}}\}$ 
4:    $j \leftarrow 0$ 
5:    $X_t \leftarrow []$ 
6:   for all  $X^i \in X$  do  $\triangleright$  Keep each ground-truth token with prob  $\kappa(t_{\text{text}})$  to get noisy  $X_t$ 
7:     if  $r < \kappa(t_{\text{text}})$  where  $r \sim \text{Unif}(0, 1)$  then
8:        $X_t \leftarrow X_t + [X^i]$ 
9:        $j \leftarrow j + 1$ 
10:       $\mathcal{A}_j \leftarrow \{\}$ 
11:    else
12:       $\mathcal{A}_j \leftarrow \mathcal{A}_j \cup \{X^i\}$   $\triangleright$  Record the deleted tokens at each position in  $\mathcal{A}_j$ 
13:    end if
14:  end for
15:   $\mathcal{I}_t \leftarrow \{\}$ 
16:  for all images  $Y \in \mathcal{I}$  do
17:     $Y_1 \leftarrow \text{VAEEnc}(\text{img})$ 
18:     $u \sim \text{Unif}(0, 1)$ 
19:     $\tau_{\text{img}}(Y) \leftarrow \tau_{\text{text}} - \kappa^{-1}(u)$ 
20:    if  $\tau_{\text{img}} < 0$  then
21:      insert  $\langle \text{image} \rangle$  in the appropriate  $\mathcal{A}_i$   $\triangleright$  Image is “deleted” at this snapshot
22:    else
23:       $t_{\text{img}}(Y) \leftarrow \min\{1, \tau_{\text{img}}(Y)\}$ 
24:       $Y_0 \sim \mathcal{N}(0, I)$ 
25:       $Y_t \leftarrow t_{\text{img}}(Y)Y_1 + (1 - t_{\text{img}}(Y))Y_0$ 
26:       $\mathcal{I}_t \leftarrow \mathcal{I}_t \cup \{Y_t\}$ 
27:    end if
28:  end for  $\triangleright$  Forward pass
29:   $\{\pi, \lambda_{\text{nz}}, Q\} \leftarrow \text{OneFlowModel}(X_t, \mathcal{I}_t)$   $\triangleright$  Compute OneFlow losses
30:   $\mathcal{L}_{\text{tokens}} \leftarrow \frac{1}{n(X_t)} \sum_i \left[ - \sum_{a \in \mathcal{A}_i} \log Q^i(a \mid X_t) \right]$   $\triangleright n(X_t)$  is the length of the sequence
31:   $\mathcal{L}_{\text{count}} \leftarrow \frac{1}{n(X_t)} \sum_i \left( \lambda^i(X_t) - |\mathcal{A}_i| \log \lambda^i(X_t) \right)$ 
32:   $\mathcal{L}_{\text{img}} \leftarrow \sum_{Y \in \mathcal{I}_t} \mathbf{1}[\tau_{\text{img}}(Y) \geq 0] \cdot \|v(Y_t, t_{\text{img}}(Y)) - (Y_1 - Y_0)\|_2^2$ 
33:   $\mathcal{L} \leftarrow \mathcal{L}_{\text{tokens}} + \mathcal{L}_{\text{count}} + \mathcal{L}_{\text{img}}$ 
34:   $\Theta \leftarrow \text{optimizer\_step}(\nabla \mathcal{L}; \Theta)$   $\triangleright$  Compute gradients and update model
35: end function

```

F ADDITIONAL EXPERIMENT RESULTS

F.1 PERFORMANCE BETWEEN AR AND ONEFLOW DURING PRETRAINING

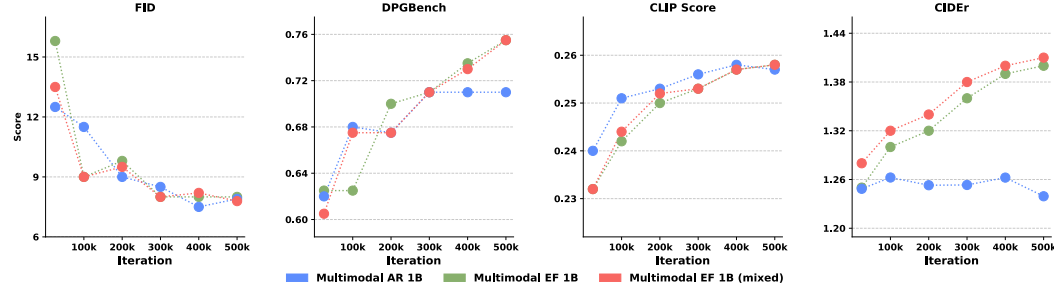


Figure 15: **Training curve for OneFlow vs. AR for multimodal pretraining.** OneFlow initially starts out lower than AR however it quickly catches up and exceeds AR, most notably on DPG and CIDEr.

F.2 PRETRAINING FROM SCRATCH VS LLAMA INIT

Model	Initialization	Image Generation				VQA
		DPG \uparrow	FID \downarrow	CLIP \uparrow	CIDEr \uparrow	Avg VQA \uparrow
OneFlow	Random	73.17	7.96	25.7	139.4	51.2
OneFlow	LLaMA	75.41 (+2.24)	7.79 (-0.17)	26.0 (+0.3)	138.2 (-1.2)	52.2 (+1.0)
OneFlow Mixed	Random	74.86	7.69	25.8	140.0	51.6
OneFlow Mixed	LLaMA	75.08 (+0.22)	7.44 (-0.25)	25.8 (+0.0)	139.1 (-0.9)	52.8 (+1.2)
AR + FM	Random	71.9	7.83	25.8	122.9	46.6
AR + FM	LLaMA	73.4 (+1.50)	7.91 (-0.08)	25.7 (-0.1)	123.9 (+1.0)	49.0 (+2.4)

Table 4: **Ablation study comparing LLaMA initialization vs. random initialization.** Except for CIDEr, using LLaMA as initialization generally offers benefits, especially for dense prompt image generation (DPG) and for VQA performance. Image generation metrics use CFG=3, and VQA results are averaged across benchmarks.

F.3 SAMPLING STEPS ON CAPTION PERFORMANCE

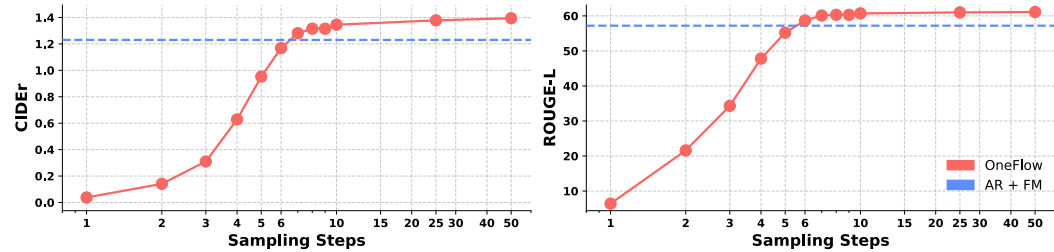


Figure 16: **Performance vs. sampling steps compared to AR.** OneFlow achieves parity with the AR model using only 6 sampling steps.

F.4 QUALITATIVE COMPARISON ON IMAGE GENERATION

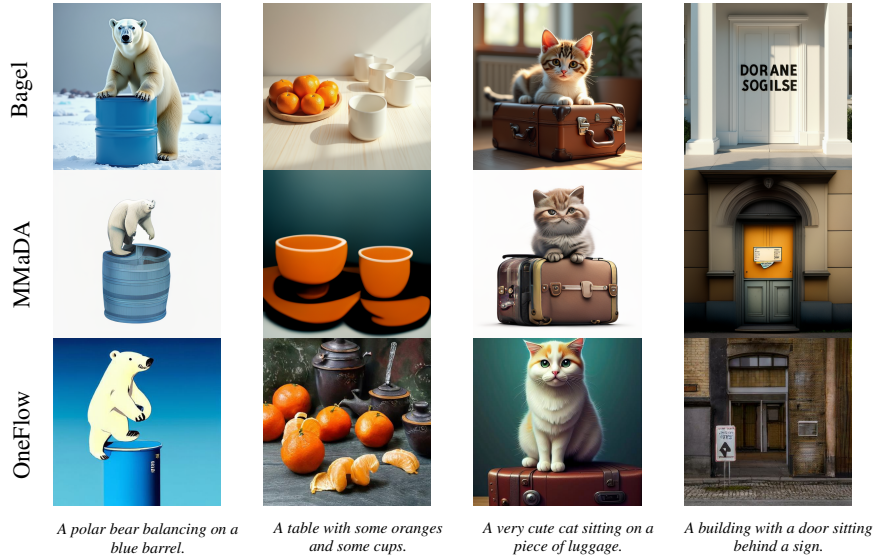


Figure 17: **Qualitative comparison of OneFlow and SOTA models.** We notice that OneFlow gets the details of the prompt correctly, for instance the polar bear is ‘balancing on a blue barrel’. The visual details of our generation are also better compared to MMaDA possibly due to using continuous image tokens rather than discrete. In the last column, the figure shows OneFlow handles common semantic challenges more effectively, as it was able to generate a building with ‘a door sitting behind a sign’.

Prompt	AR+FM	OneFlow	OneFlow Mixed
An expansive field, blanketed by the <i>soft light of morning</i> , cradles a collection of eight cabbages, their green heads round and plump. These vegetables are nestled among rows of rich soil, dotted with glistening droplets of dew that cling to their crinkled leaves. As wisps of mist begin to lift, the cabbages lie poised, ready for the day’s impending harvest.			
An elegant and modern bathroom featuring a sleek, white rectangular bathtub filled with a froth of soap bubbles. The bathtub rests upon a floor of gray, matte tiles that complement the room’s minimalist design. Against the room’s far wall stands a <i>large window that frames the warm, amber hues of a sunset</i> , casting a tranquil glow throughout the space.			
The image captures a whimsical scene with a brown tabby cat, its fur <i>patterned in shades of dark brown, black, and light taupe</i> . The cat, situated as if in the throes of space, is portrayed with a transparent, gleaming bubble encasing its head like an astronaut’s helmet. Around it, an assortment of smaller bubbles float serenely in the imagined cosmos, with a creatively interpreted Saturn adorned with rings in the backdrop, providing an aura of interstellar exploration.			

Figure 18: **OneFlow vs. AR on DPG Bench.** OneFlow outperforms AR significantly on DPG Bench. Through our inspection, we notice that OneFlow gets the subtle details correctly. We highlight the subtle details that the OneFlow model gets correctly in red whereas the AR misses. For the first row, the OneFlow model generates the ‘field of cabbages in the soft morning light’. The second row, OneFlow model generates ‘a window with the amber hue’ whereas AR did not. Lastly, the ‘brown cat has shades of brown, black and light taupe’ in the OneFlow generations.

F.5 QUALITATIVE COMPARISON ON IMAGE UNDERSTANDING

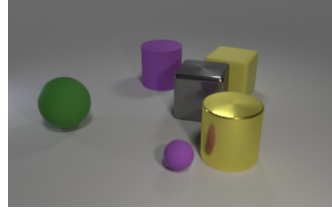
Counting

Question: Subtract all large rubber spheres. Subtract all big shiny cylinders. How many objects are left? Explain your answer:

Ground Truth: 4

There are **five objects** left in the image. To find their count, we look at the positions of the spheres relative to the cylinders and the cube. From left to right, the first sphere is the green, the second is the purple, the third is the blue, and the fourth is the brown. The fifth object, which is the yellow cylinder, is not counted since it is not listed in the options. Therefore, the total count of objects that are left with the categories (green, purple, blue, brown) is **three**.

AR



In the image, there is **one object** that matches the description of the large sphere: the green sphere, and the only large shiny cylinder, which is the yellow cylinder. Therefore, the total number of objects left in the image is **four**.

OneFlow

(a) **OneFlow compared to AR model on counting task.** OneFlow first locates the object in the prompt before performing the calculation.

RealWorld Perception

Question: In which direction is the front wheel of the car on the right side facing? You must explain your answer. Explain your answer:

Ground Truth: Right



In the image provided, the front wheel of the car on the **right side** is **aimed towards the left**. The wheel is partially visible and appears to be in the process of turning, as indicated by the position of the car and the blurred scenery. Given the context of the street and the position of the car, it is likely that the front wheel is facing towards the direction the car is turning. However, due to the tire's orientation and the tire's angle, it's not possible to determine the steering direction with absolute certainty. The most probable action, based on the visible evidence, is that the front wheel is turned to **the left**.

AR

The front wheel of the car on the right side is facing towards the camera's perspective. This is determined by the direction of the road, the orientation of the vehicles, and the position of the wheels and headlights. If facing the right side, the direction of the camera is towards the front of the image, then the front wheel of the car is forward, facing the right side of the image. Conversely, if we consider the front of the car to the left side of the image, which is the perspective of the camera, the front of the car is facing to the left. Therefore, from the perspective of the camera, and considering the orientation of the front of the car, the front wheel of the car is facing towards the **right side of the image**.

OneFlow

(b) **OneFlow compared to AR on hard perception task.** AR commits to the answer at the start of the sequence.

Visual Search

Question: In which direction is the front wheel of the car on the right side facing? You must explain your answer. Explain your answer:

Ground Truth: Right



There is **no plastic stool** visible in the image. The scene includes a canal, boats, trees, and traditional buildings, but none of these items appear to be present within the frame. The stool in question may be part of the rustic outdoor setting or an unrelated object not in the frame.

AR

The plastic stool in the image is located on the right side of the canal, near the side of the canal. It is **blue** and stands out against the natural colors of the scene, contrasting with the red colors of the buildings and the canal.

OneFlow

(c) **OneFlow compared to AR on visual search task.** OneFlow first locates the target object (stool), then analyzes its color contrast against the background to determine the answer. Red arrow added for visual clarity (not present in original image).

Figure 19: Comparison of VQA generation between AR and OneFlow.

1458 G LLM DISCLOSURE
1459

1460 We used LLMs to create latex code for the figures, generate code, and to improve writing.
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511

Template prompt of VLM-as-a-Judge

Please act as an impartial judge and evaluate the quality of the following caption provided by an AI assistant for a given image.

Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, please rate the response on a scale of 1 to 10 by strictly following this format: "[rating]", for example: "Rating: [[5]]".

Below are some guidelines for your reference:

1. **Precision:** The caption should accurately correspond to the content of the image, providing precise information about it. Common examples of imprecision include errors in color, quantity, spatial relationships, or the posture of people.
2. **Informativeness:** Salient information in the image should be reflected in the caption. Since it is impossible to include every detail, you will need to subjectively judge which aspects of the image are important. For instance, describing an otter as "a small animal" is precise, but it is less informative than specifying "an otter".
3. **Hallucination:** Captions that include descriptions of objects or elements that are clearly absent from the image should be significantly penalized.
4. **Attention to detail:** Annotators should pay close attention to the details in the image to distinguish the quality of the descriptions.
5. **Assistive description:** Imagine a visually impaired person asking you to describe the image for them. How would you convey the image to them?
6. **Reverse thinking:** What image does the caption lead us to imagine? Does the caption effectively lead you to imagine the intended image?

Image: <image> Reference Caption: <reference caption> Caption: <caption text>

Figure 20: Template prompt of VLM-as-a-Judge, taken from CapArena (Cheng et al., 2025)