

# OmniFlow: Any-to-Any Generation with Multi-Modal Rectified Flows

Shufan Li<sup>\*1</sup>, Konstantinos Kallidromitis<sup>\*2</sup>, Akash Gokul<sup>\*3</sup>, Zichun Liao<sup>1</sup>

Yusuke Kato<sup>2</sup>, Kazuki Kozuka<sup>2</sup>, Aditya Grover<sup>1</sup>

<sup>1</sup> UCLA <sup>2</sup>Panasonic AI Research <sup>3</sup>Salesforce AI Research

<sup>\*</sup>Equal Contribution

Correspondence to jacklishufan@cs.ucla.edu

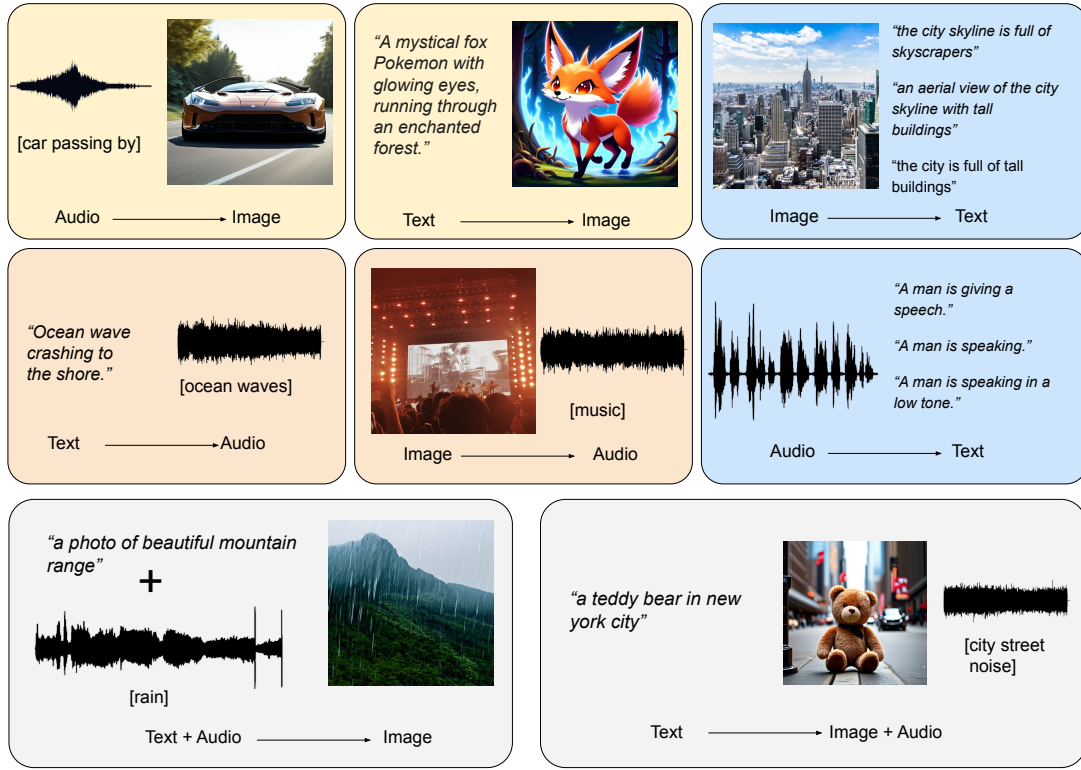


Figure 1. **OmniFlow is capable of a diverse range of any-to-any generation tasks.** OmniFlow supports generation of any output modalities given any input modality, such as text-to-image, text-to-audio, audio-to-image generations. It also supports tasks in multiple input modalities, such as text+audio-to-image.

## Abstract

We introduce *OmniFlow*, a novel generative model designed for any-to-any generation tasks such as text-to-image, text-to-audio, and audio-to-image synthesis. *OmniFlow* advances the rectified flow (RF) framework used in text-to-image models to handle the joint distribution of multiple modalities. It outperforms previous any-to-any models on a wide range of tasks, such as text-to-image and text-to-audio synthesis. Our work offers three key contributions: First, we

extend RF to a multi-modal setting and introduce a novel guidance mechanism, enabling users to flexibly control the alignment between different modalities in the generated outputs. Second, we propose a novel architecture that extends the text-to-image MMDiT architecture of *Stable Diffusion 3* and enables audio and text generation. The extended modules can be efficiently pretrained individually and merged with the vanilla text-to-image MMDiT for fine-tuning. Lastly, we conduct a comprehensive study of the design choices of rectified flow transformers for large-scale audio and text

generation, providing valuable insights into optimizing performance across various modalities. Code is available at <https://github.com/jacklishufan/OmniFlows>.

## 1. Introduction

Generative modeling has witnessed considerable advancements in recent years. Notably, diffusion models such as DALL-E-3 [40], Stable Diffusion 3 [11], AudioLDM2 [33] achieves state-of-the-art performance on text-to-image and text-to-audio tasks. However, these models can only perform a single task while requiring considerable computing resources and data for training. To achieve any-to-any generations, previous works such as CoDi [46] and UIO [36] typically combine a set of modality-specific encoders (*e.g.* ViT [1]) and decoders (*e.g.* Stable Diffusion [44]). However, this design limits these models’ ability to integrate information across modalities and generate multi-modal outputs coherently. For example, to perform audio+text-to-image ( $A+T \rightarrow I$ ) generation, CoDi simply takes a weighted average of the audio embedding and text embedding to condition an image generator. However, there is no guarantee that the averaged embedding can faithfully represent the two input modalities, as arbitrarily many modality embeddings can average to the same embedding.

An alternative approach for any-to-any generation is to use a single multi-modal model to learn the joint distribution of multiple modalities. This approach has often led to strong performance as it allows information to flow across modalities. However, existing single-model designs typically involve training from scratch, and thus require a considerable amount of data. Existing works in this area, such as UniDifuser [4] and Chameleon [47] only experiment with text and image modalities. They also require considerable compute resources. To the best of our knowledge, there has yet to be a unified open-sourced multi-modal generative model that supports text, image, and audio simultaneously.

We propose OmniFlow, a unified multi-modal generative model for any-to-any generation. Unlike previous unified multi-modal models, OmniFlow does not need to be trained from scratch with a large amount of data because of its modular design, saving considerable computing resources for its training. OmniFlow is inspired by the MMDiT architecture used in Stable Diffusion 3 [11], which performs text-to-image generation using a two-stream network that combines a text-input stream and an image-output stream through a series of joint attention blocks. OmniFlow builds on MMDiT by incorporating additional input and output streams, extending its text-to-image capability to support any-to-any generation. Crucially, since the parameters for each stream are mostly independent, we can pretrain them separately or initialize them with a pretrained single-task expert model (*e.g.* SD3).

To effectively train OmniFlow, we propose a novel multi-modal rectified flow formulation that incorporates a diverse set of tasks, such as text-to-audio and audio-to-image, into a unified learning objective. Multi-modal rectified flow is built upon a decoupled, time-differentiable interpretation between the distribution of a multi-modal data pair and i.i.d. Gaussian noise. In this formulation, each of the any-to-any generation tasks can be represented by a path connecting two noise levels. For example, given text, image, and audio modalities, the task of text+audio-to-image ( $T+A \rightarrow I$ ) can be represented by a path between the distribution of (clean text, clean audio, Gaussian noise) to (clean text, clean audio, clean image).

We conducted extensive evaluations of OmniFlow. Experiment results show that OmniFlow outperforms previous any-to-any models on a wide range of tasks, including text-to-image and text-to-audio generation. Compared to single-task specialist models, OmniFlow achieves competitive performance with state-of-the-art methods.

In summary, our contributions are three-fold:

- First, we extend rectified flow formulation to the multi-modal setting and support flexible learning of any-to-any generation in a unified framework.
- Second, we proposed OmniFlow, a novel modular multi-modal architecture for any-to-any generation tasks. It allows multiple modalities to directly interact with each other while being modular enough to allow individual components to be pretrained independently or initialized from task-specific expert models.
- Lastly, to the best of our knowledge, we are the first work that provides a systematic investigation of the different ways of combining state-of-the-art flow-matching objectives with diffusion transformers for audio and text generation. We provide meaningful insights and hope to help the community develop future multi-modal diffusion models beyond text-to-image generation tasks.

## 2. Backgrounds

### 2.1. Flow-Based Generative Models

Flow-based generative models [23, 31, 34, 48], represent the coupling of data points  $x^0$  and noise distribution  $x^1$  using an ordinary differential equation (ODE):

$$dx^t = v_\theta(x^t, t)dt \quad (1)$$

where the velocity  $v$  is parameterized by a neural network. Directly solving this equation is expensive. However, we can define a forward process  $x^t = a(t)x^0 + b(t)x^1$  to directly regress a conditional vector field using the Conditional Flow Matching (CFM) objective [48] as follows:

$$\mathcal{L}_{\text{CFM}} = \left(-\frac{b(t)\lambda'(t)}{2}\right) \mathbb{E}_{t, x^1, x^t | x^1} \|\epsilon_\theta(x^t, t) - x^1\|^2 \quad (2)$$

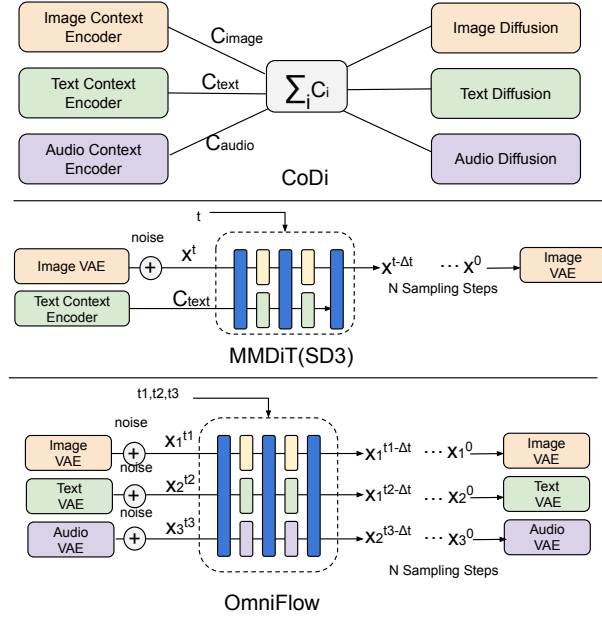


Figure 2. **Pipeline of OmniFlow.** Previous any-to-any models such as CoDi [46] (Top) concatenate multiple modality-specific encoders and decoders, and naively average the embedding of multiple modalities to achieve joint conditioning. By contrast, OmniFlow (Bottom) is a unified, modular multi-modal model, where features from different modalities directly interact with each other through joint attention layers. OmniFlow is inspired by the modular design of Stable Diffusion 3 [11] (Middle), a text-to-image model.

where  $\lambda(t) = \log \frac{\alpha(t)^2}{\beta(t)}$  is the signal-to-noise ratio (SNR),  $\epsilon_\theta(x^t, t) = -\frac{2}{\lambda'(t)b(t)}(v_\theta(x^t, t) - \frac{\alpha'(t)}{\alpha(t)}x^t)$  is parameterized by  $v_\theta$ . The optimum of this objective remains unchanged when introducing time-dependent weighting, and hence we can rewrite it following [22] as:

$$L_w(x_0) = -\frac{1}{2}\mathbb{E}_{t, x^1} w(t) \lambda'(t) \|\epsilon_\theta(z_t, t) - \epsilon\|^2 \quad (3)$$

where,  $w(t) = -\frac{1}{2}\lambda'(t)b(t)^2$  for CFM and  $x^1 \sim \mathcal{N}(0, I)$  follows noise distribution. This formulation gives a unified representation for a variety of generative modeling approaches. For example, a rectified flow’s forward process is defined as  $x^t = (1-t)x^0 + tx^1$ , which corresponds to  $w^{\text{RF}} = \frac{t}{1-t}$ . Esser et al. [11] summarized many configurations of common methods under this unified formulation, including (LDM)-Linear [44] and Cosine [39]. They also explored a logit-normal distribution of timestep  $t$  for text-to-image generation. We explore all these variants in the context of multi-modal generation, particularly for audio and text, as it is unclear if the results from the text-to-image domain can be directly generalized.

## 2.2. Any-to-Any Generation

Prior works have explored any-to-any generation. CoDi [46] achieved it first by combining multiple modality-specific encoders (e.g. ViT) and decoders (e.g. Stable Diffusion) through bridge alignment. However, its design has limited cross-modality interaction. For example, to achieve text+audio-to-image (T+A→I generation), it simply computes the weighted average of text embeddings and audio embedding. Unified-IO [36] models any-to-any generation as a sequence-to-sequence problem, and uses an autoregressive model to achieve any-to-any generation, such as text-to-image or text-to-audio. Our work is the first to use a multi-modal flow matching objective for any-to-any tasks.

Additional works focus exclusively on unifying text-to-image and image-to-text generation. Chameleon [47] uses an LLM-like large autoregressive model to handle multi-modal data. It represents images as VQGAN tokens [50]. Transfusion [52] adopted a similar design, but uses a non-autoregressive diffusion loss for image modeling, while maintaining an autoregressive loss for text generation. Despite their successes, these unified multi-modal models require considerable training resources, because they are less modular than previous works that combine multiple models. OmniFlow achieves a good balance by separating the parameters of each individual modality, while allowing the features of each modality to freely interact with each other at every layer.

## 3. Method

### 3.1. Multi-Modal Rectified Flow

We consider the joint distribution  $(x_1^0, x_2^0, \dots, x_n^0) \sim \pi_{data}$  over the space of paired multi-modal data where  $x_i \subseteq \mathbb{R}^{d_i}$  is a sample of modality  $i$  represented by a vector of  $d_i$  dimension. Let  $(x_1^1, x_2^1, \dots, x_n^1) \sim \pi^1$  be the i.i.d Gaussian distribution where  $x_i^1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is a Gaussian vector of  $d_i$  dimension. Given empirical observations  $x^0 \sim \pi_{data}$ , and  $x^1 \sim \pi^1$ , we consider the decoupled, continuous, time-differential interpolation given by:

$$\frac{\partial x_i^{t_i}}{\partial t_i} = v_i(x_1^{t_1}, x_2^{t_2}, \dots, x_i^{t_i}, t_1, \dots, t_i) \quad (4)$$

$$\frac{\partial x_i^{t_i}}{\partial t_j} = 0; i \neq j \quad (5)$$

$$x_i^{t_i} = (1-t_i)x_i^0 + t_i x_i^1 \quad (6)$$

where the independence condition of Eq (2) indicates  $x_i^{t_i}$  only moves when  $t_i$  moves. Over this interpretation space, we can use a path  $\tau : t \rightarrow (t_1, t_2, \dots, t_n); [0, 1] \rightarrow [0, 1]^n$  to model any-to-any generation tasks involving these modalities. For example, given  $(x_1, x_2, x_3) \sim p_{data}$  where  $x_1, x_2, x_3$  are image, text, and audio modalities. We can

---

**Algorithm 1** Multi-Modal Rectified Flow

---

**Input:** Dataset  $\mathcal{D}$  consists of modality 1, ...,  $N$ , where each sample  $x = (x_{i1}^0, x_{i2}^0, \dots)$  consists of a subset (or all) of modalities  $i_1, i_2, \dots \in \{1, 2, \dots, N\}$ .

**Output:**  $v_{\theta,i} : (x_1^{t_1}, x_2^{t_2}, \dots, x_n^{t_n}) \rightarrow v_i^{t_i}$  for each  $i = 1, 2, \dots, N$ , parameterized by  $\theta$

Initialize  $\theta$

```
1: while not converged do
2:   Sample  $x = (x_{i1}^0, x_{i2}^0, \dots) \sim \mathcal{D}$ 
3:    $x_j^0 \leftarrow \mathbf{0}; \forall j \in \{1, 2, \dots, N\} \setminus \{i_1, i_2, \dots\}$ 
4:   Sample path  $\tau$ .*
5:   Sample  $t \sim \text{Uniform}([0, 1])$ 
6:    $(t_1, \dots, t_N) \leftarrow \tau(t)$ 
7:    $x_i^{t_i} \leftarrow x_i^{t_i} = (1 - t_i)x_i^0 + t_i x_i^1; \forall i \in 1, 2, \dots, N$ 
8:    $\mathcal{L} = \sum_{i \in \{i_1, i_2, \dots\}} \|v_i - v_{\theta,i}(x_1^{t_1}, \dots, x_n^{t_n}, t_1, \dots, t_n)\|^2$ 
9:   Perform optimizer step using  $\nabla_{\theta} \mathcal{L}$ 
10: end while
11: Return  $\theta$ 
12:   ▷ *  $\tau$  encodes a task involving only modality  $i_1, i_2, \dots$ ,
   hence  $t_j = 1; \forall j \notin \{i_1, i_2, \dots\}$ 
```

---

model text-to-image (T→I) tasks as a path  $\tau_{t2i}$  such that  $\tau_{t2i}(0) = (0, 0, 1)$ , which represents a clean text-image pair and  $\tau_{t2i}(1) = (1, 0, 1)$ , which represents clean text. We can similarly model the joint sampling of text, image and audio set as a path from  $(0, 0, 0)$  to  $(1, 1, 1)$  and text+image-to-audio ( $T + I \rightarrow A$ ) as a path from  $(0, 0, 0)$  to  $(0, 0, 1)$ .

The flow matching objective would be solving  $n$  least squares regression problems for each modality of the form:

$$\min_{v_{\theta}} \mathbb{E}_{\tau} \int_{\tau} \mathbb{E}_{x^0, x^1} \|v_i - v_{\theta,i}(x_1^{t_1}, x_2^{t_2}, \dots, x_n^{t_n}, t_1, \dots, t_n)\|^2 ds \quad (7)$$

where  $v_i = x_i^0 - x_i^1$ , and  $v_{\theta,i}$  is a neural network parameterized by  $\theta$ . We use the same network  $\theta$  to predict outputs for all modalities 1, 2, ...,  $N$ . The outer expectation is over some prior of paths encoding generation tasks which we are interested in. The integral is calculated over a path  $\tau(t) = (t_1, \dots, t_n)$ , and  $ds = \frac{\partial t_i}{\partial t} dt$ . Concretely, we consider three modalities: image, text, audio in our experiments as modalities: 1, 2 and 3 respectively. We consider the distribution of all possible linear paths  $\tau(t) = (t_1, t_2, t_3)$  in  $[0, 1]^3$  following the rectified flow formulation. They can encode a diverse set of tasks such as text-to-image or text+image-to-audio.

During training, we do not necessarily need all modalities for each data point. For data points that only contain a subset of three modalities (e.g. text-image pairs), we can set the time step of remaining modalities (e.g. audio) to 1, which corresponds to complete Gaussian noise. The full training algorithm is given as follows:

At inference, we simply pick a path and use the network

prediction to solve for Eq. (5). Notably, for standard text-to-image generation with  $(x_1, x_2)$  pairs where  $x_1$  is image and  $x_2$  is text, and  $x_3$  is the missing audio modality, picking a linear path from  $(1, 0, 1)$  to  $(0, 0, 1)$  is equivalent to the standard single-modality rectified flow (Text→Image) formulation used by Stable Diffusion 3 [11].

### 3.2. Multi-Modal Guidance

To flexibly control the multi-modal generation process, we extend the classifier free guidance (CFG)[16] to multi-modal rectified flow setting. Recall that CFG of single modalities are formulated as follows:

$$\hat{v}_{\theta}(x^t, c) = v_{\theta}(x^t, c) + (\alpha - 1)(v_{\theta}(x^t, c) - v_{\theta}(x^t)) \quad (8)$$

where  $c$  is a condition and  $x^t$  is the noised latent at timestep  $t$  of the single-modal output. We extend this formulation to multi-modal setting by defining  $\delta_{ij} = v_{\theta}(x_i^t, x_j^0) - v_{\theta}(x_i^t)$ , which represents the influence of input modality  $j$  to output modality  $i$ . In particular, we obtain  $v_{\theta}(x_i^t, x_j^0)$  and  $v_{\theta}(x_i^t)$  by setting inputs of modalities not present in the formula to Gaussian noise. For example, given three modalities  $x_1, x_2, x_3$ , we can obtain  $v_{\theta}(x_1^t, x_2^0)$  by computing  $v_{\theta}(x_1^t, x_2^0, x_3^1)$  and obtain  $v_{\theta}(x_1^t)$  by computing  $v_{\theta}(x_1^t, x_2^1, x_3^1)$ . Note that  $x_2^1, x_3^1$  is just Gaussian noise.

Given the set of  $\delta_{ij}$ , we can guide the output generation of modality  $i$  by the following formula:

$$\hat{v}_{\theta}(x_1^{t_1}, \dots, x_n^{t_n}) = v_{\theta}(x_1^{t_1}, \dots, x_n^{t_n}) + \sum_{j \neq i} (\alpha_{ij} - 1) \delta_{ij} \quad (9)$$

where  $\alpha_{ij}$  is the equivalent of  $\alpha$  in a multi-modal setting. This scheme allows the user to precisely control the interaction between each of the input and output modalities. When there are only two modalities, our multi-modal guidance Eq. (9) is equivalent to the standard single-modal classifier-free guidance Eq. (8).

### 3.3. Model Architecture

We propose OmniFlow, a modular, effective extension to the MMDiT architecture used in Stable Diffusion 3. Concretely, given multi-modal inputs that consist of text, image, and audio, we first convert them to latents  $x_1, x_2, x_3$  using modality-specific VAEs. We then add random Gaussian noise to the latents following the forward process defined in Eq. (6). We use the three sinusoidal embeddings to encode,  $t_1, t_2, t_3$  which correlate to the noise scale for each modality. These three timestep embeddings are passed to an MLP to obtain  $y$ , a single embedding representing all modality-specific time steps. The final input to OmniFlow are the unified timestep embedding  $y$ , and noised latents  $(x_1, x_2, x_3)$ . These four input vectors are passed to  $N$  consecutive Omni-Transformer



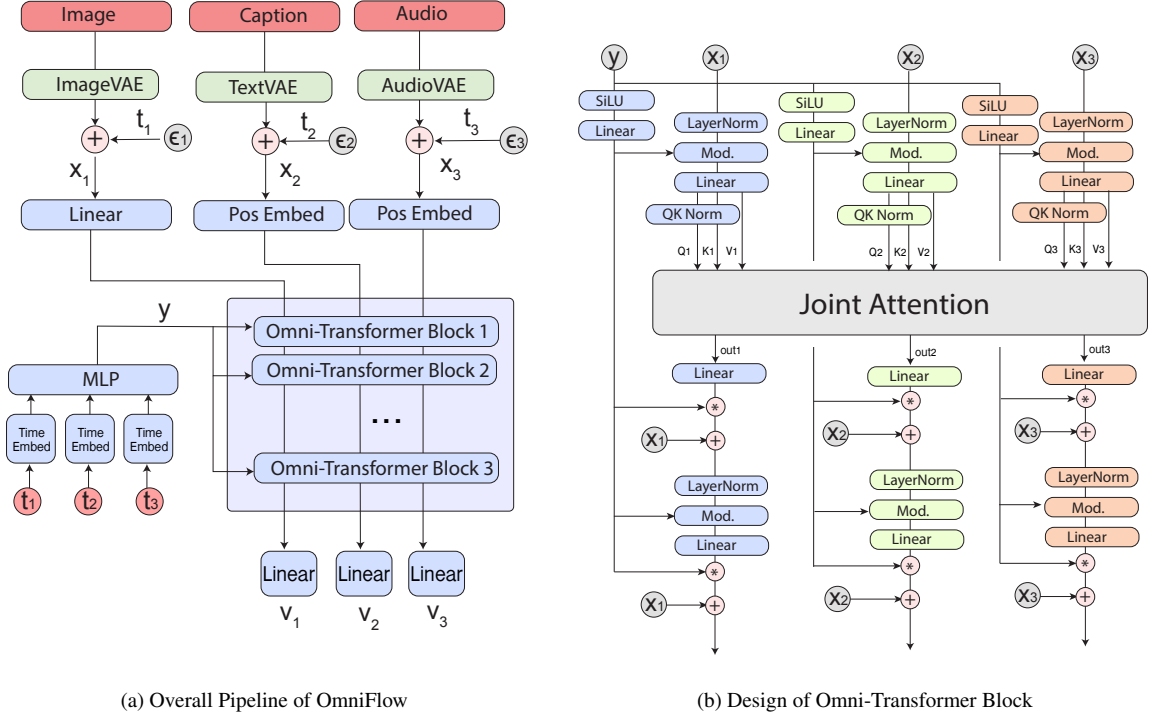


Figure 3. **Architecture of OmniFlow.** Left: We highlight the architecture of OmniFlow. Right: We show the design of an individual Omni-Transformer Block.

blocks. The final hidden states of each modality, are then processed by the linear output layer to obtain predictions of  $v$ .

Within each Omni-Transformer block, the inputs  $x_1, x_2, x_3$  are processed by modality-specific projections to obtain  $q_1, k_1, v_1, q_2, k_2, v_2, q_3, k_3, v_3$ . We then concatenate the queries, keys, and values to obtain  $Q = \text{Concat}(q_1, q_2, q_3), K = \text{Concat}(k_1, k_2, k_3), V = \text{Concat}(v_1, v_2, v_3)$ . The joint attention output for  $i^{\text{th}}$  modality  $\text{out}_i$  is given by:

$$\text{out}_i = \text{SoftMax}\left(\frac{q_i^T K}{\sqrt{d}}\right)V \quad (10)$$

where  $d$  is the dimension of each attention head. The output is passed to a feed forward network (FFN) to get the final output of the Omni-Transformer block. Following the design of DiT [41], we use the unified time embedding to modulate the qkv projection and FFN. We add skip connections after the joint attention operation and after the FFN.

We illustrate the model architecture in Fig. 3. Notably, different modalities are handled by different projection and feed-forward layers with independent parameters. The only multi-modal operation is the joint attention, with no trainable parameters of its own. This allows us to pretrain layers of different modalities individually and combine them for fine-

tuning, which significantly improves the training efficiency.

## 4. Setup

### 4.1. Training Dataset

We use text-image pairs, text-audio pairs, and audio-image pairs during training. We also make use of a small amount of text-image-audio triplets. The text-image pairs include 5M images sampled from COYO-700M dataset [5], 2M images sampled from LAION-Aesthetic-3M subset [25], 7M images from LAION-COCO subset [26], the full CC12M dataset [6], and 2M high-quality image dataset generated by flux-dev and DALLE-3 [14]. We put high weights on images from LAION-Aesthetic-3M and the 2M high-quality images to maintain good aesthetic quality in the output. The text-audio pairs include the full training set of AudioSet [12], Audiotaps [21] and WavCaps [37]. The audio-image pairs include the training data of VGGSound [7] and SoundNet [2]. While SoundNet contains 2M images and is larger than VGGSound, we set the sample weight of VGGSound and SoundNet to 2:1 since SoundNet contains many improperly resized images with bad aspect ratios.

To generate text-image-audio triplets, we use BLIP [28] to generate synthetic captions for videos in VGGSound and SoundNet. We provide further details of the dataset construction in the Appendix.

## 4.2. Training Recipe

At a high level, we initialize OmniFlow with the text and image modules of Stable-Diffusion 3 (Model 1). We first train a separate text-to-audio model with text-audio pairs (Model 2). Then, we merge Model 1 and Model 2 to obtain a combined model with text, image, and audio modules (Model 3). Since Model 1 and Model 2 have separate text modules, we average their weights during the merge process. Finally, we fine-tune Model 3 on a diverse set of any-to-any tasks using the methods described in Sec. 3.1.

Due to our modular design, we can initialize and pretrain each module individually. This saves immense computational cost when compared to previous unified multi-modal models (e.g. UniDiffuser [4]) which are trained from scratch. We use a global batch size of 64 and train Model 2 and Model 3 for 100k, and 150k steps each. We provide further training and implementation details in the Appendix.

## 5. Main Results

### 5.1. Evaluation Metrics

We perform extensive experiments on paired generation (text-to-image, text-to-audio) and generic any-to-any generation such as text-to-audio+image (T→I+A), audio-to-text+image (A→T+I). For text-to-image generation, we report FID [15] and CLIP [43] scores on MSCOCO-30K benchmark [30]. Following the official implementation, the cosine similarities between CLIP embeddings are multiplied by 100. We also report results on the GenEval benchmark [13]. For audio generation, we report FAD [20] and CLAP [10] score on AudioCaps. Results are reported with a 16kHz sampling rate. We also use CLAP scores for caption evaluations.

### 5.2. Text-to-Image Generation

Model	Param	FID↓	CLIP↑
UniDiffuser	0.9B	<b>9.71</b>	30.93
CoDi	4.3B	11.26	30.69
UIO-2XXL	6.8B	13.39	-
SDv1.5	0.9B	11.12	30.63
SDXL*	2.6B	16.49	31.36
SD3-Medium*	2B	20.94	30.65
OmniFlow*	3.4B	13.40	<b>31.54</b>

Table 1. **Text-to-Image Generation on MSCOCO-30K Benchmark.** \*Indicates models pretraining data consists of high quality images and captions that do not follow the distribution of COCO dataset, which can negatively affect FID scores.

We report results on MSCOCO-30k in Tab. 1, and results on GenEval in table Tab. 2. On MSCOCO-30k, we achieve a lower FID than state-of-the-art models such as SDXL and SD3-Medium. While our FID number is higher than some

Model	Param	Images	Gen.↑
<i>Text-to-Image Specialist</i>			
SD1.5	0.9B	4.0B	.43
SDv2.1	0.9B	2.3B	.50
SDXL	2.6B	1.6B	.55
DALL-E 2	4.2B	2.6B	.52
SD3-Medium	2B	1B	.62
SD3-Large	8B	2.0B	.68
<i>Generalist</i>			
CoDi	4.3B	400M*	.38
UniDiff.	0.9B	2B	.43
OmniFlow	3.4B	30M*	<b>.62</b>
Chameleon	7B	3.5B	.39
Transfusion	7B	3.5B	.63

Table 2. **Text-to-Image Generation on GenEval Benchmark.** We compare the model size, number of training images and GenEval benchmark Score. \* Indicates fine-tuning dataset. CoDi and MMDiT-O are both initialized with pretrained text-to-image diffusion models (SD and SD3).

previous models such as SDv1.5, it should be noted that more recent models such as SDXL and SD3 tend to have higher FID numbers because they are trained on high-quality text-image pairs that do not match the distribution of COCO images [42]. Notably, SD3 has a FID of 20.94 while SDv1.5 has 11.12, even though SD3 is considered a better model according to human evaluations. SDXL, which is widely recognized as the state-of-the-art open-source model before the release of SD3, also has a higher FID than SDv1.5.

In terms of CLIP scores, OmniFlow significantly outperforms previous models. In particular, when contrasted with generalist models UniDiffuser and CoDi, we achieve a gain of +0.61 and +0.85 respectively, showing superior text-to-image alignment. On GenEval Benchmark, which better measures the text-to-image capabilities, OmniFlow achieves a score of 0.62, a competitive score even when compared to the state-of-the-art specialist SD3-Medium. In addition, OmniFlow significantly outperforms previous any-to-any baselines at the same scale, such as CoDi (+.24) and UniDiffuser (+.19). Compared with larger models trained on a lot more images, OmniFlow outperforms Chameleon-7B and achieves competitive performance as Transfusion-7B.

Notably, unlike Chameleon, Transfusion, and UniDiffuser which need to be trained from scratch, OmniFlow achieves high performance with only 30M training images, highlighting the effectiveness of our modular design. While the design of CoDi also allows it to make use of pretrained text-to-image model as its initialization, it is trained with considerably more images than OmniFlow while performing worse.

Model	Param	FAD↓	CLAP↑
<i>Text-to-Audio Specialist</i>			
AudioGen-L[24]	1B	1.82	-
Make-an-Audio[19]	0.4B	2.66	-
AudioLDM-L[32]	0.7B	1.96	.141
Make-an-Audio 2[18]	0.9B	2.05	.173
AudioLDM 2-Full-L[33]	0.7B	1.86	.182
<i>Generalist</i>			
CoDi	3.4B	1.80	.053*
OmniFlow	3.4B	<b>1.75</b>	<b>.183</b>
UIO-2XXL	6.7B	2.64	-

Table 3. **Text-to-Audio Generation on AudioCaps Evaluation Set.** Comparison of FAD and CLAP scores for various audio generators. \*Reproduced from official checkpoint, see Appendix for details.

### 5.3. Text-to-Audio Generation

We report text-to-audio generation results on AudioCaps in Tab. 3. Compared with previous state-of-the-art, OmniFlow achieves strong performance on FAD and CLAP scores. It outperforms AudioLDM2 on FAD (-0.11) and achieves equivalent performance on CLAP (+0.001). When compared with generalist models, OmniFlow significantly outperforms CoDi on both FAD (-0.05) and CLAP (+.13) metrics.

### 5.4. Recipes for Audio and Text Diffusions

	Audio Gen. FAD↓	Text Gen. CLAP↑
<i>Continuous Flow Matching</i>		
eps/linear	2.08	.141
v/cos	2.01	.203
v/linear	1.86	.126
rf/uniform	1.82	.227
rf/lognorm	<b>1.79</b>	<b>.254</b>
<i>Discrete Text Diffusion</i>		
SEDD[35]	-	.180
MDLM[45]	-	.163

Table 4. **Various Formulations for Audio and Text Generation.** We report FAD for audio generation and CLAP for text generation on AudioCaps dataset.

We explore various recipes for training audio and text diffusion transformers for multi-modal generation, which is a relatively under-explored area. Concretely, we explored five formulations mentioned in the section Sec. 2.1. For these experiments, we used a model with only audio and text modules (Model 2 in Sec. 4.2) and trained for 50k steps. We report FAD score for text-to-audio generation and CLAP score for audio-to-text generation. Amongst all five formulations, rf/lognorm performs the best with the lowest FAD (1.79) and highest CLAP score (.254). We also explored two

discrete space diffusion models, SEDD [35] and MDLM [45] which showed advantages over continuous-space diffusion models in recent literature. Specifically, we use the absorbing state version of SEDD. For these experiments, the text-vae encoder is replaced with a token-embedding layer, and, text-vae decoder is replaced with a simple linear output layer to predict token logits. We also replace the flow-matching loss on the text-embedding with the loss function of SEDD and MDLM respectively, which operates on token logits instead of continuous embeddings. We report the CLAP score on audio-to-text generation. We do not see considerable advantages over continuous alternatives.

## 6. Sampling

On the sampling side, we explored the effect of guidance and timestep shift. The timestep shift was originally introduced by SD3 to balance the sampling process of images at different resolutions. Concretely, it augments the inference schedule as:

$$\hat{t} = \frac{\gamma t}{1 + (1 - \gamma)t} \quad (11)$$

where  $\gamma = \sqrt{\frac{m}{n}}$ , with  $m$  being the target sample resolution and  $n$  being a reference resolution. For audio and text generation, there is no concept of varying resolution, as the input audio spectrogram and text embedding have fixed resolutions. However, we empirically observe applying a shift can improve the generation quality. Concretely, incorporating the shift term  $\gamma > 1$  will lead to a concave schedule, where the denoising process progresses slowly at the beginning and accelerates towards the end. We find that this improves sample quality for text-to-audio and audio-to-text generation tasks.

We employ the multi-modal guidance mentioned in Sec. 3.2. For simple audio-to-text and text-to-audio generation, our formulation is reduced to standard classifier-free guidance. We show the effect of guidance and timestep shift in Fig. 4. Generally, we find that shift=3.0 works well for both tasks. For audio generation, a guidance scale of 8 achieves the highest performance. For text generation, a guidance scale of 4 achieves the best result.

To explore the effect of multi-modal guidance in Sec. 3.2, we provide qualitative results for audio+image-to-text (A+I→T) task. Recall that we use  $x_1, x_2, x_3$  to denote image, text, and audio modalities. The multi-modal guidance for this task can be controlled by  $\alpha_{21}$  and  $\alpha_{23}$  where  $\alpha_{21}$  controls text-image alignment and  $\alpha_{23}$  controls text-audio alignment. For simplicity, we denote  $\alpha_{21}$  as  $\alpha_{im}$  and  $\alpha_{23}$  as  $\alpha_{au}$ . We vary  $\alpha_{im}, \alpha_{au}$  between the interval  $[1.0, 2.0]$  such that  $\alpha_{im} + \alpha_{au} = 3.0$ . We show the results in Fig. 5. Qualitatively, higher  $\alpha_{au}$  will make the model’s output resemble more the audio captions, and  $\alpha_{im}$  will make the model’s output resembles more the image captions. Interestingly, we observe that it also reflects the subtle differences in the

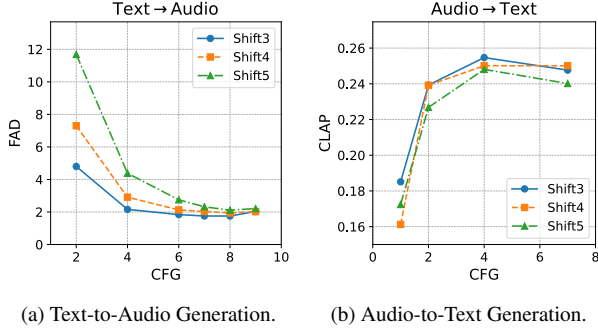


Figure 4. **Effect of CFG and Shift for audio and text generation.** We evaluate the impact of guidance and timestep shift on text-to-audio and audio-to-text tasks.

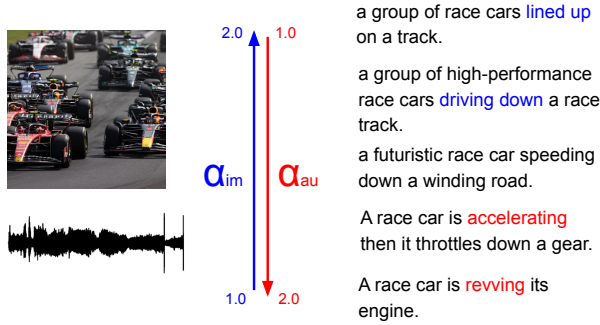


Figure 5. **Effect of Multi-Modal Guidance.** In this example, the user can flexibly control the alignment between output text and input image, audio independently by varying  $\alpha_{au}$  and  $\alpha_{im}$ . Higher  $\alpha_{im}$  will make the output texts resemble image captions, with visual descriptions such as *lined up*, *driving down*. Higher  $\alpha_{au}$  will make the output texts resemble audio captions, with descriptions such as *accelerating*, *revving*.

style of audio and image captions in the training data (*e.g.* whether the first letter is capitalized). By varying these two parameters, users can achieve flexible control of generation.

### 6.1. Qualitative Comparison

We directly compare OmniFlow with two recent any-to-any generation methods: CoDi [46] and UniDiffuser [4]. In addition to the quantitative results, we present qualitative text-to-image comparisons in Fig. 6. These examples demonstrate that OmniFlow achieves a significant improvement in generation quality compared to previous any-to-any models. Specifically, in the first example (top), our model successfully follows the prompt while maintaining high aesthetic quality, accurately capturing both the cat’s features and its mirrored reflection. In contrast, CoDi is unable to change the cat’s eyes, and UniDiffuser fails to depict the cat looking at the mirror. A similar trend is evident in the third example: OmniFlow correctly positions lanterns tied to a rope,

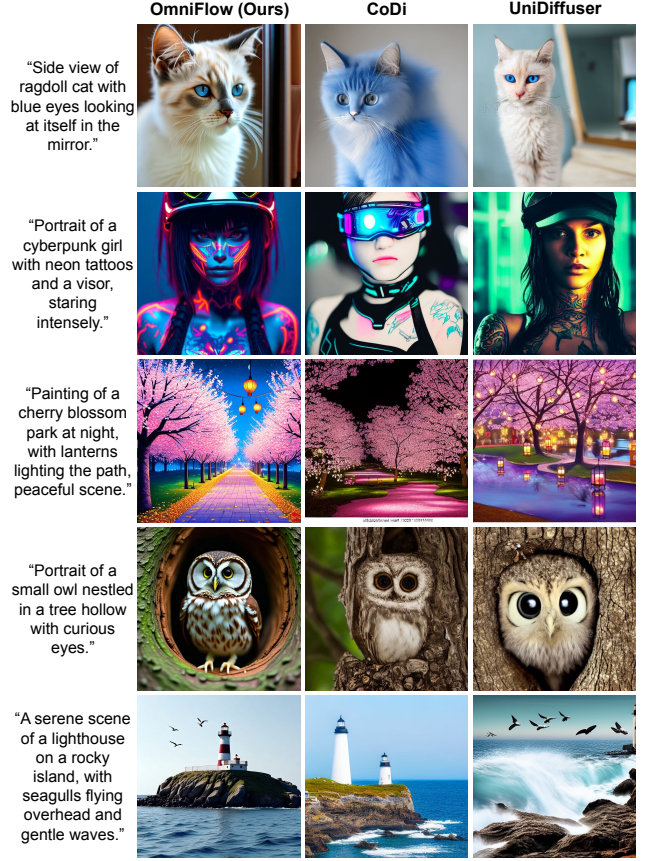


Figure 6. **Qualitative Comparison with baselines on text-to-image generation.** OmniFlow achieves better image quality and prompt alignment when compared to previous generalist models.

while UniDiffuser places them on the river. Finally, in the lighthouse example, CoDi fails to incorporate seagulls, and UniDiffuser ignores the adjective “gentle,” instead producing an image with rough waves and an out-of-focus lighthouse.

Our results show that OmniFlow achieves a much higher generation quality compared with previous any-to-any models, both in terms of image-text alignment and image fidelity.

## 7. Conclusion

We present OmniFlow, a unified early-fusion multi-modal generative model for any-to-any generation tasks. OmniFlow adapts a modular design that enables individual components to be pretrained separately, while allowing features from different modalities to directly interact with each other, through a joint attention mechanism. We conduct extensive experiments to show that OmniFlow outperforms previous any-to-any models on a wide range of challenging generation tasks, including text-to-image and text-to-audio generation. We provide further analysis on the limitation of OmniFlow in the Appendix.



## 8. Acknowledgments

This research was supported by NSF Career #2341040 and a Schmidt Science Fellowship.

## References

- [1] Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2
- [2] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29, 2016. 5, 12
- [3] JISHENG BAI, Haohe Liu, Mou Wang, Dongyuan Shi, Wenwu Wang, Mark D Plumbley, Woon-Seng Gan, and Jianfeng Chen. Audiosetcaps: Enriched audio captioning dataset generation using large audio language models. In *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*, 2024. 12
- [4] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, pages 1692–1717. PMLR, 2023. 2, 6, 8, 15
- [5] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. <https://github.com/kakaobrain/coyo-dataset>, 2022. 5
- [6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 5
- [7] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 5, 12
- [8] Wenxi Chen, Ziyang Ma, Xiquan Li, Xuenan Xu, Yuzhe Liang, Zhisheng Zheng, Kai Yu, and Xie Chen. Slam-aac: Enhancing audio captioning with paraphrasing augmentation and clap-refine through llms. *arXiv preprint arXiv:2410.09503*, 2024. 15
- [9] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. 13
- [10] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 6
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2, 3, 4, 13
- [12] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017. 5
- [13] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36, 2024. 6
- [14] Jacky Hate. Text-to-image-2m dataset, 2024. Accessed: 2024-11-14. 5
- [15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6
- [16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4
- [17] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 14
- [18] Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023. 7
- [19] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR, 2023. 7
- [20] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\`echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018. 6
- [21] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, 2019. 5
- [22] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [23] Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [24] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022. 7

- [25] LAION. Aesthetics for open source, 2023. Accessed: 2024-11-14. [5](#)
- [26] LAION. Laion coco: 600m synthetic captions from laion2b-en, 2023. Accessed: 2024-11-14. [5](#)
- [27] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*, 2020. [13](#)
- [28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. [5](#)
- [29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. [13](#), [15](#)
- [30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. [6](#)
- [31] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. [2](#)
- [32] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. [7](#), [13](#)
- [33] Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024. [2](#), [7](#)
- [34] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. [2](#)
- [35] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. [7](#), [14](#)
- [36] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024. [2](#), [3](#)
- [37] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024. [5](#)
- [38] MidJourney AI. Image generated using midjourney ai, 2024. Accessed on November 21, 2024. URL: <https://www.midjourney.com/>. [15](#), [17](#)
- [39] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. [3](#)
- [40] OpenAI. Dall-e 3, 2023. [2](#)
- [41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. [5](#)
- [42] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. [6](#)
- [43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. [6](#)
- [44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. [2](#), [3](#), [13](#)
- [45] Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [7](#), [14](#)
- [46] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems*, 36, 2024. [2](#), [3](#), [8](#), [14](#), [15](#), [18](#)
- [47] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. [2](#), [3](#)
- [48] Alexander Tong, Nikolay Malkin, Guillaume Hugué, Yanlei Zhang, Jarrod Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3), 2023. [2](#)
- [49] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. [15](#)
- [50] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. [3](#)
- [51] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024. [13](#)
- [52] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamir, Jacob Kahn, Xuezhe Ma,

Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. [3](#)

- [53] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, WANG HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *The Twelfth International Conference on Learning Representations*, 2023. [13](#)

# OmniFlow: Any-to-Any Generation with Multi-Modal Rectified Flows

## Supplementary Material

	Size	Modality
LAION-Aesthetics-3M	2M*	T,I
CC12M	12M	T,I
COYO-700M(Subset)	5M	T,I
LAION-COCO	7M	T,I
SoundNet	2M	T,A,I†
VGGSound	0.2M	T,A,I†
T2I-2M	2M	T,I
AudioSet	2M	T,A
AudioCaps	46K	T,A
WavCaps	0.4M	T,A

Table 5. List of all datasets used in training. \*Some image URLs are no longer accessible. † We generate synthetic captions using BLIP.

## A. Implementation Details

### A.1. Dataset

In Tab. 5, we list the size of all datasets used in training. We filter out all images whose shortest side is less than 256 pixels. To obtain data with all modalities (image, audio, text), we use BLIP to generate synthetic captions for images in the SoundNet[2] and VGGSound[7] dataset, which are extracted from videos. Since AudioSet only comes with class labels, we use synthetic captions generated by audio-language models provided by AudioSetCaps[3].

### A.2. Schedules

Recall from Section 3 that we can represent different tasks with different paths in  $[0, 1]^3$ . We visualize this in Fig. 7. We adopted simple linear tasks for any-to-any generation tasks so that for simple cases like text-to-image and text-to-audio, our formulation matches the standard rectified flow.

### A.3. Training Pipeline

We initialize our model with SD3 (Model 1 in Fig. 8). We first train the model on text-audio pairs to obtain Model 2. The text branch of Model 2 is initialized with weights of SD3, while the audio branch is randomly initialized. After the training, we merge Model 1, which contains a text branch and an image branch, and Model 2, which contains a text branch and an audio branch, to Model 3, which contains text, image, and audio branches. The text branch of Model 3 is obtained by averaging the weights of the text branches from Model 1 and 2. Finally, we train the Model 3 on all datasets mentioned in Suppl. A.1. This training pipeline is illustrated in Fig. 8.

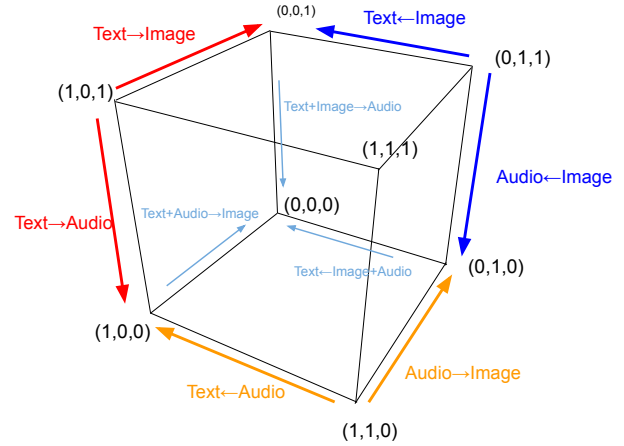


Figure 7. Paths encoding different any-to-any generation tasks.  $(t_1, t_2, t_3)$  represents the “noise level” of image, text and audio modalities.  $(0, 0, 0)$  represents clean (image, text, audio) triplets, and  $(1, 1, 1)$  represents pure Gaussian noise.

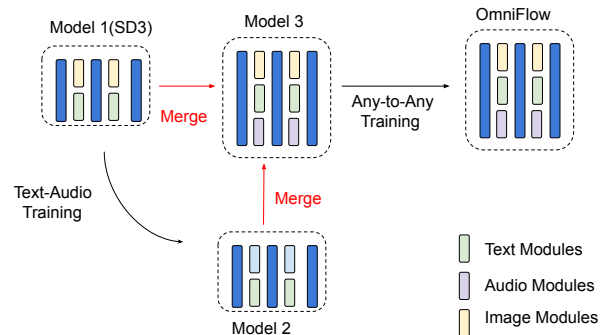


Figure 8. Training Pipeline of OmniFlow. We initialize our model with SD3 (Model 1). We then train the model on text-audio pairs to obtain Model 2. We merge Model 1 and Model 2 to obtain Model 3. The final model is obtained by further training Model 3 on any-to-any generation tasks.

We train Model 2 for 100k steps and Model 3 for 150k steps. We use 8 A6000 GPUs with a per GPU batch size of 8. We use AdamW optimizer with a learning rate of  $1e-5$  for Model 2 and  $5e-6$  for Model 3. The learning rate undergoes a linear warmup in the first 1000 steps and a cosine decay throughout the rest of the training. We adopt exponential moving average (EMA), which are updated every 100 training steps with a decay factor of 0.999.



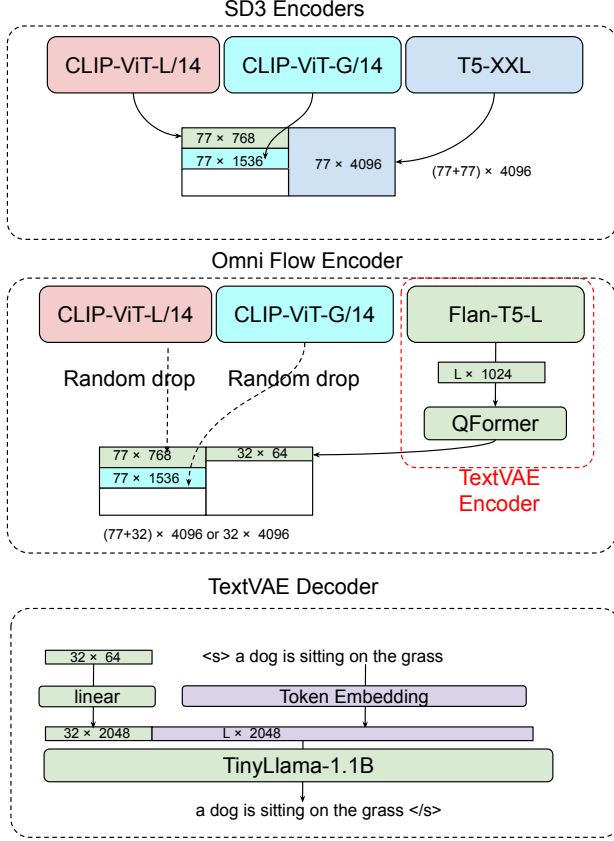


Figure 9. **Architecture of Text VAE and Text Encoders in OmniFlow.** SD3 (Top) uses three text encoders: CLIP-L, CLIP-G, and T5-XXL. OmniFlow (Middle) replaces the 4.7B T5-XXL with a VAE encoder based on Flan-T5-L. CLIP encoders become optional and are not used for tasks without clean text inputs. The decoder of VAE (Bottom) is based on TinyLlama-1.1B. The VAE embedding is used as the prefix for decoding.

#### A.4. Text VAE

We train a text VAE on caption data using Flan-T5-L [9]. Recall that SD3[11] makes use of three text encoders: CLIP-L, CLIP-G and T5-XXL. We replace the 4.7B T5-XXL with Flan-T5-L [27] to save computation cost and use it as part of a text VAE. Specifically, given an input caption of length  $L$ , it is first encoded by Flan-T5-L to obtain a vector of size  $L \times 1024$ . We then pass it to a QFormer[29] and obtain an output vector of size  $32 \times 64$ . This vector is used as the VAE embedding. In the decoding process, the VAE embedding is first processed by a linear projection layer to obtain a vector of size  $32 \times 2048$ . This is used as the prefix embedding for a TinyLlama-1.1B decoder [51]. These architecture designs are shown in Fig. 9. Note that while we introduced a 1.1B text-decoder, the overall system actually has fewer parameters since we replaced the 4.7B T5-XXL with a 783M Flan-T5-L.

We employ the auto-encoding training objective of OPTIMUS [27]. We freeze the Flan-T5-L encoder and fine-tune the QFormer and TinyLlama decoder end-to-end. We train the text VAE on all caption data mentioned in Suppl. A.1 for 2 epochs, with a learning rate of  $1e-5$ , a global batch size of 256 using AdamW optimizer.

When using the VAE encoder as the text encoder of OmniFlow, we pad the embedding to 4096 with zeros to maintain the input dimension of SD3. Additionally, we also incorporate the CLIP-L and CLIP-G encoders of SD3 as auxiliary text encoders to stabilize the training. We apply random dropout to these encoders during the training. During the inference, the CLIP encoders are not used if the input does not contain clean texts (e.g. Image-to-Text task).

#### A.5. Audio VAE

We directly adapt the audio VAE used by AudioLDM [32]. In particular, we adopt the same vocoder and preprocessing pipeline as AudioLDM2. We use HiFiGen as VAE, which is used in AudioLDM and AudioLDM2. We use AudioLDM2’s checkpoint. We also explored AudioMAE, but found it to perform significantly worse as measured by FAD (2.03 vs 1.79).

#### A.6. Omni-Transformer

We followed the architectural design of SD3 for image and text modules and initialize them with SD3 weights. The audio modules are initialized with identical setup to the image modules. Specifically, it has 24 layers and a hidden size of 1536. The positional embedding layer has a patch size of 2. Since the audio VAE outputs a feature map of dimension  $256 \times 16$ , the positional embedding layer will convert each audio to a sequence of length  $128 \times 8 = 1024$ .

#### A.7. Pooled Conditional Embeddings

SD3 makes use of additional pooled embeddings from CLIP-ViT-L/14 and CLIP-ViT-G/14 in addition to the sequence embeddings. We maintain them as is, with additional dropout during the training. We additionally incorporate an Audio Encoder to create pooled embeddings for audio inputs [53]. These embeddings are not used when clean data of respective modality is not available.

#### A.8. Baselines

In this section, we describe the specific variants studied in Tab. 4. Except for the discrete text diffusions (SEDD and MDLM), these variants fit into the unified formulation of Eq. (3) by varying its parameters.

**linear** is a variant of DDPM used in LDM [44]. It discretizes the timesteps to  $0, 1, \dots, T-1$  and uses the formulation  $b_t = \sqrt{1 - \alpha_t^2}$ , where  $a_t = \sqrt{\prod_{i=0}^t (1 - \beta_i)}$ , and  $\beta_t = (\sqrt{\beta_0} + \frac{t}{T-1}(\sqrt{\beta_{T-1}} - \sqrt{\beta_0}))^2$ . We explored  $\epsilon$ -prediction and  $v$ -prediction objectives for this variant.

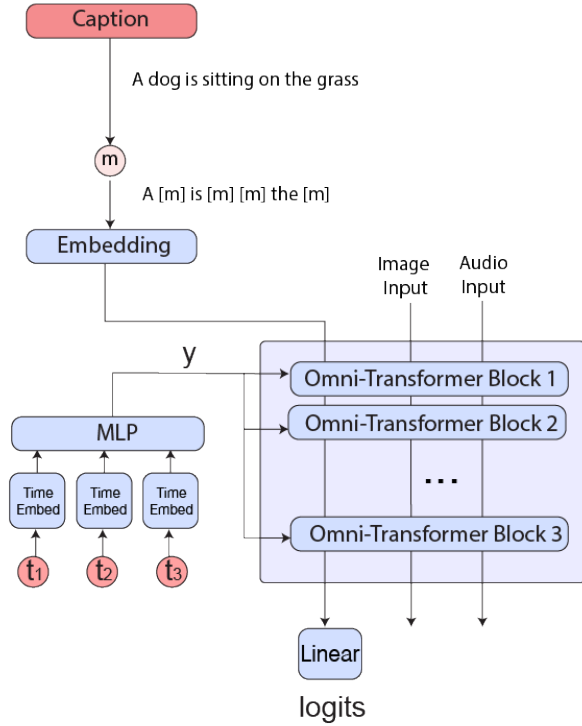


Figure 10. **Discrete Diffusion Variant of OmniFlow.** In this setup, we remove the text VAE and directly pass token embedding to the Omni-Transformer layers. “[m]” indicates a mask token.

**cosine** is defined by the forward process

$$x^t = \cos\left(\frac{\pi}{2}t\right)x^0 + \sin\left(\frac{\pi}{2}t\right)x^1 \quad (12)$$

The weighting function is  $w_t = e^{-\lambda_t/2}$  for  $v$ -prediction objectives[17].

**SEDD and MDLM** are recently proposed discrete text-diffusion models. We consider MDLM[45] and the absorbing state variants of SEDD[35] in our experiments.<sup>1</sup> These models directly define a forward process in the discrete token space, where clean text tokens are progressively replaced with a special “[MASK]” token. We adapt our implementation for these methods by removing the text VAE and introducing a token embedding layer. This design is shown in Fig. 10.

<sup>1</sup>SEDD also has a uniform variant, where the tokens are not replaced with a “[MASK]” token, but a randomly token sampled from the vocabulary.

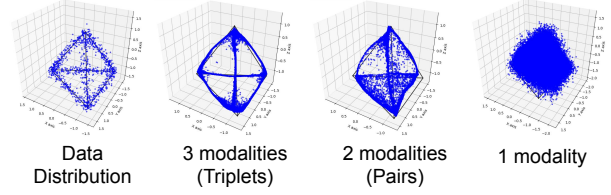


Figure 11. **Synthetic Experiments on three 1D-modalities.** We consider the joint distribution of three toy modalities  $(x_1, x_2, x_3)$ , each represented by a vector of dimension 1. Hence, a triplet consisting of three modalities be represented by a point in  $\mathbb{R}^3$ . We assume the joint distribution is a uniform distribution in the neighborhood of tetrahedron (Left). We experiment with training OmniFlow using triplets, pairs, and only individual modalities. Models trained with triplets of three modalities best represent the original distribution.

## B. Additional Discussions

### B.1. Sampling

OmniFlow does not directly model the marginals of two modalities. For example, given three modalities  $(x_1, x_2, x_3)$ , it does not directly model  $p(x_1^0|x_2^0) = \int_{x_3^1 \in \mathbb{R}^{d_3}} p(x_1^0, x_3^1|x_2^0) dA$ , where  $d_3$  is the dimension of  $x_3^1$ . Integrating over  $x_3^1$  is infeasible. Instead, we sample  $p(x_1^0, x_3^1|x_2^0)$  by first sample  $x_3^1 \sim q(x_3^1|x_2^0) = \mathcal{N}(0, I)$  and sample  $p(x_1^0|x_3^1, x_2^0)$  using path from  $(1,0,1)$  to  $(0,0,1)$ .

### B.2. Necessity of text, image, audio triplets.

Compared with previous works such as CoDi[46] which uses weighted average of embeddings to mix multiple input modalities, OmniFlow requires directly training on triplets consisting of all modalities (image, text, audio). To study the necessity of this requirement, we conduct synthetic toy examples on three modalities  $(x_1, x_2, x_3)$ , each represented by a one-dimensional vector. A triplet of three modalities can then be represented by a point  $(X, Y, Z)$  in 3D space. We show this experiments in Fig. 11. We assume the ground truth data distribution follows a uniform distribution in a small neighborhood adjacent to a tetrahedron (Leftmost Figure). We experiment with training an 8-layer MLP with triplets  $(x_1, x_2, x_3)$  (Second-Left Figure), pairs of  $(x_1, x_2), (x_1, x_3), (x_2, x_3)$  (Second-Right Figure), and only individual modalities  $(x_1), (x_2), (x_3)$  (Rightmost Figure). For each model, we plot 50k samples generated by the model. Qualitatively, models trained on triplets best represent the data distribution. This makes sense as pairs are essentially projections on XY, XZ, YZ planes and individual modalities are projections on X, Y, Z axis. These projections are not sufficient to recover the original distribution represented in this 3D space.

	Images	Parms.	AudioCaps		COCO-Karpathy	
			CLAP↑	CIDEr↑	CLIP↑	CIDEr↑
Specialist						
BLIP-2[29]	129M	2.7B	-	-	-	145.8 ‡
SLAM-AAC[8]	-	7B	-	84.1 ‡	-	-
Generalist						
OmniFlow	30M	3.4B	<b>0.254</b>	<b>48.0</b>	26.8	<b>47.3</b>
CoDi †	400M	4.3B	0.206	7.9	25.9	17.2
Unidiffuser †	2B	0.9B	-	-	<b>29.3</b>	20.5
UIO2-XXL	1B*	6.8B	-	48.9	-	125.4*
Transfusion	3.5B	7B	-	-	-	35.2

Table 6. **X-to-Text Performance comparison on AudioCaps and COCO Captions.** \* UIO2’s training data includes COCO. The fine-tuning dataset also includes 53M image understanding data, including 14 image captioning datasets.  $\dagger$  evaluated with official checkpoints.  $\ddagger$  fine-tuned on respective datasets (COCO and Audiocaps).

## C. Quantative Text Evaluation

We report quantitative results of image captioning on COCO-Karpathy-Test dataset and audio captioning on Audiocaps dataset. We report CLIP score, CLAP score, and CIDEr[49] on these two benchmarks. We compare against generalist models such as CoDi and Uni-Diffuser. Uni-Diffuser, released two checkpoints v0 and v1, where v1 is fine-tuned on internal data. We compare against v0 for a fairness. OmniFlow outperforms CoDi on both tasks, and outperforms UniDiffuser in CIDEr score (+26.8). It has a lower CLIP score (-2.5). We consider the performance of OmniFlow as competitive, considering OmniFlow is trained on significantly less data than UniDiffuser and can also perform audio captioning task. We note that the performance of generalist models significantly lags behind specialist models that are fine-tuned of respective datasets, suggesting rooms for further improvements. We provide further discussion in the limitation section.

## D. Benefits of Multi-Task Training

In this section, we discuss if joint training on multiple modalities benefit single-task performance. We provide additional results in Tab. 7 by comparing a model trained on all tasks with a model only trained on a subset of tasks. We show that Omniflow was able to leverage the training data of related tasks (e.g. T2A, I2A) and boost individual performance. In addition to results presented in Tab.R1, we also observe improvements in image generation, where OmniFlow generate high fidelity A2I outputs even though A2I datasets consist of low-res videos (+1.22 Aesthetic score), thanks to high-fidelity T2I data.

Training Data	FAD $\downarrow$
OmniFlow	<b>1.83</b>
I2A,A2I,T2A,A2T	1.89
I2A,A2I	2.03
I2A-Only	2.05

Table 7. **Performance of Various Training Data Compositions.** We compare FAD scores for Image to Audio (I2A) under different setup on VGGSound.

## E. Additional Qualitative Results

### E.1. Text-to-Image

Fig. 14 demonstrates a range of qualitative text-to-image examples for OmniFlow. We depict a wide variety of people, scenes and objects to demonstrate the robustness of our approach.

### E.2. Image-to-Text

We provide a side-by-side image-to-text comparison between OmniFlow, CoDi [46] and UniDiffuser [4] using synthetic high quality images from the Midjourney Explore page [38] in Fig. 12.

### E.3. Audio-to-Text

In Tab. 8, we show qualitative results on Audiocaps audio-to-text task. OmniFlow can generation captions that match the ground truth. While CoDi can correctly grasp the main objects in the audio such as “car”, “bird”, “sheep”, “computer”, it struggles with generating captions that accurately reflect the scene.

### E.4. Text-VAE AutoEncoding

In Tab. 9, we show reconstruction examples of Text VAE. The reconstruction mostly adheres to the semantics of the ground truth, with minor differences. For example, it may change “well-furnished” to “well-decorated”.

ID	CoDi	OmniFlow	GT
yVjivgsU2aA	Four car driver trying forcoming for a speeding car.	A race car engine revs and tires squeal.	An engine running followed by the engine revving and tires screeching.
8F-ndyrEWJ8	Fire police cars stop and red traf- fic on different highway.	A fire siren goes off loudly as a man shouts and a low hum of an engine is running throughout the whole time.	A distant police siren, then racing car engine noise, and a man calling in police code over his radio.
350OCezayrk	Four motor car driving for completing an automobile service.	A vehicle engine is revving and idling.	A motor vehicle engine starter grinds, and a mid-size engine starts up and idles smoothly.
LCwSUVuTyvg	Door, a blue hat and winter jacket.	A door is being slammed.	Glass doors slamming and sliding shut.
7XUt6sQS7nM	The sheep of the woman are the sheep of the sheep.	Multiple sheep bleat nearby.	A sheep is bleating and a crowd is murmuring.
PVvi2SDOjVc	Car going for a car coming home. Three cars coming for a blue car coming down a road after the highway.	A car horn beeps.	A car engine idles and then the horn blows.
Z.smJ66Tb3c	Men in the bird while the man in the boat.	Two men talk over blowing wind and bird chirps.	A man is speaking with bird sounds in the background followed by a whistling sound.
CMNIW6Lkwc	Two men in the fire and two men are coming towards the other man in the game.	A man speaks, followed by a loud bang and people laughing.	A man talking as a camera muffles followed by a loud explosion then a group of people laughing and talking.
JQz40TkjymY	Writing computers for people in writing.	Typing on a computer keyboard.	Typing on a computer keyboard.
U90e2P9jy30	A man shouts the word to the person on the sidewalk to walk to get him to the door the hand to fall down on the sidewalk in.	Basketballs being dribbled and people talking.	Several basketballs bouncing and shoes squeaking on a hardwood surface as a man yells in the distance.
5I8lmN8rwDM	Stationary fire drill technician drilling down a hose pipe while wearing safety gear. Railroad safety drill for motorcycle with hose or oil checking equipment.	A drill runs continuously.	Drilling noise loud and continues.
NIKIRKz8OKI	Birds on blue birds.	A woman talks and then an animal chewing.	A woman speaks with flapping wings and chirping birds.

Table 8. **Qualitative comparisons of CoDi and OmniFlow on Audiocaps audio captioning task.** Audios are randomly sampled. Audiocaps provide five ground truth captions per audio. For better presentation, we only list one in this table.

## F. Limitations

On text generation tasks, our model’s performance is not state-of-the-art and has considerable room for improvements. We believe this is the side effect of incorporating large-scale data with many noisy texts of different styles (e.g. alt texts, human written prompts) that differs from the distribution of standard benchmark datasets such as MSCOCO. Additionally, for image-to-text task specifically, OmniFlow is exposed to considerably less image-text pairs (30M) during the

training compared with previous generalist models such as CoDi(400M) and UniDiffuser(2B). There is also the question of balancing datasets of different caption qualities. For example, WavCaps is a weakly-labeled dataset, but is 10x larger than higher quality AudioCaps. Additional consideration is required in order to generate captions that can achieve high scores on audiocaps benchmark. Despite these limitations, we show that OmniFlow can generate reasonable image and audio captions through quantitative and qualitative experi-



Reconstruction	GT
Crispy chicken tenders alongside a portion of a bbq sauce.	Crispy chicken tenders alongside a portion of bbq sauce.
A <b>well-furnished</b> living room with a patterned curtain <b>rod</b> , a small white side table holding a vase of flowers, and a tufted gray sofa.	A <b>well-decorated</b> living room with a patterned curtain <b>panel hanging from the window</b> , a small white side table holding a vase of flowers, and a tufted gray sofa.
A young man wearing a black shirt and holding an American flag.	A young man wearing a black shirt and holding an American flag.
An artistic painting of a futuristic city by the water.	An artistic painting of a futuristic city by the water.
Cozy and <b>well-designed</b> living room with a green velvet sofa, glass coffee table displaying potted plants, and a large skylight overhead.	Cozy and <b>stylish</b> living room with a green velvet sofa, glass coffee table displaying potted plants, and a large skylight overhead.
A silver <b>Audi Rs4</b> sedan driving on the passenger side near a mountainous coastline.	A silver <b>Acura RLX</b> sedan driving on the passenger side near a mountainous coastline.

Table 9. **Text VAE reconstruction results. We show reconstruction results (Left) and the ground truth text (Right).** The reconstruction mostly adheres to the semantics of the ground truth, with minor differences.

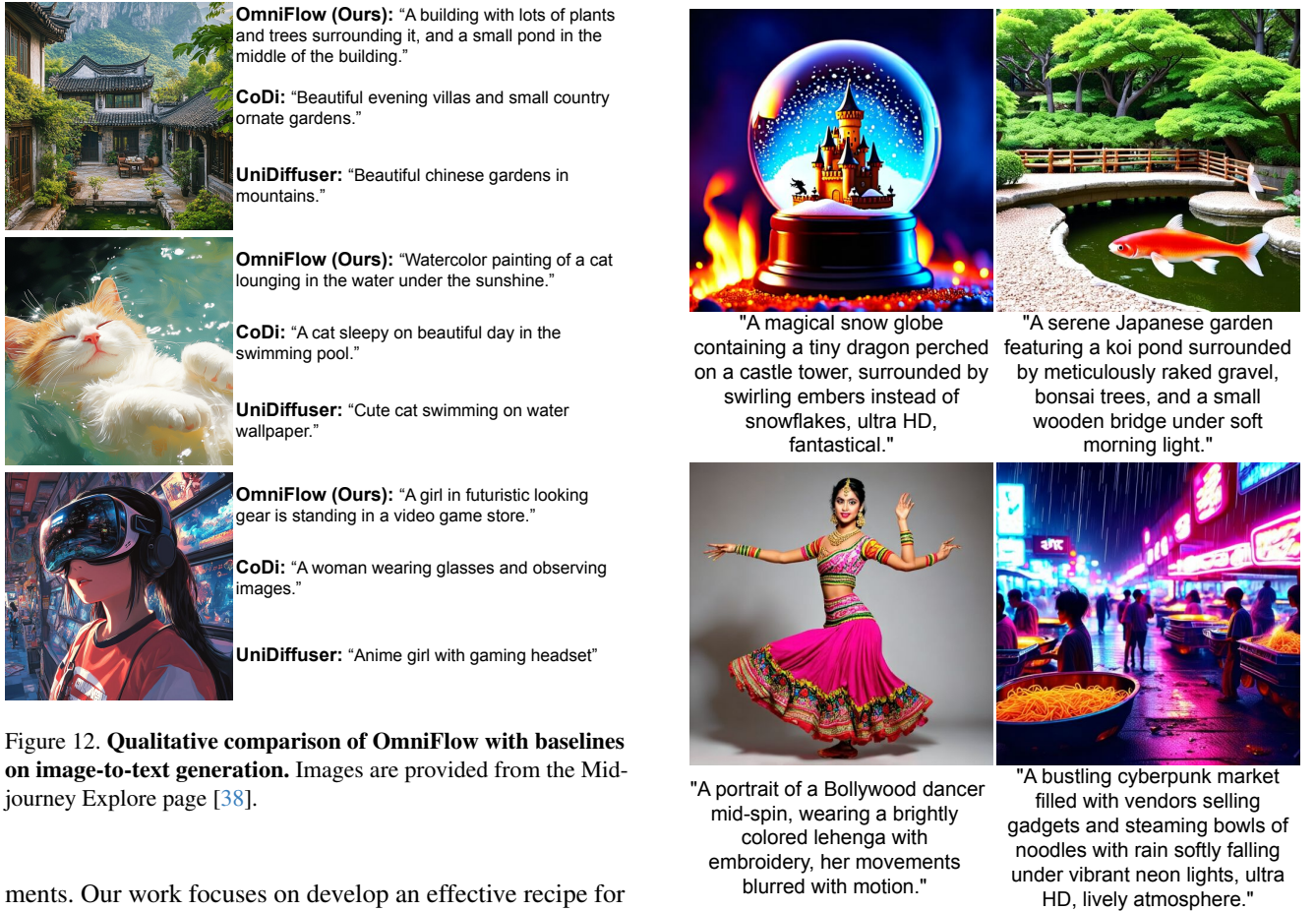


Figure 12. **Qualitative comparison of OmniFlow with baselines on image-to-text generation.** Images are provided from the Mid-journey Explore page [38].

ments. Our work focuses on develop an effective recipe for any-to-any generalist models. We leave optimizing for text generations to future works.

On Image generation tasks, while OmniFlow can generate high quality images, it has the same limitations as any text-to-image models. For example, it may inherit unintended biases from the training dataset. It may also struggle in prompts that the vanilla SD3 model also struggles with.



Figure 13. **Examples of failure cases encountered during the text-to-image generation process of OmniFlow.**

## G. Miscellaneous

### G.1. Reproducibility of CoDi

To accurately reproduce the results of CoDi [46], we follow the weights and instructions as indicated in the i-Code-V3 GitHub repository <sup>2</sup>. However, we encounter reproducibility issues, similar to open issues reported by others, which have remained unresolved <sup>3</sup>.

## H. Reproducibility Statement

All dataset used in this work are public and accessible from the Internet, except for synthetic captions of SoundNet and VGGSound we generated. We have release the code, check-points, and generated captions for these two dataset.

## I. Failure Cases

In Fig. 13 we present several failure cases of Omni-Flow when performing text-to-image generation. In the snow globe example, the model fails to interpret the prompt specifying "swirling embers instead of snowflakes," mistakenly generating snow instead. Another issue arises with the dancer, where the prompt "movements blurred with motion" is inaccurately represented as an additional arm. Lastly, the Koi pond and ramen examples highlight unnatural outputs, with the former resembling a poorly edited image of a fish in a pond and the latter depicting oversized bowls of noodles placed unnaturally on the street.

---

<sup>2</sup><https://github.com/microsoft/i-Code/tree/main/i-Code-V3>

<sup>3</sup><https://github.com/microsoft/i-Code/issues/134>





"A portrait of a young woman with striking green eyes and freckles, wearing a flowing green scarf in a windy meadow."



"Close-up of a kitten with playful eyes, wicker basket in background, ultra HD."



"A bustling Tokyo street at night, with neon signs glowing in Japanese characters and people with umbrellas walking under the soft drizzle."



"A vibrant autumn forest where sunlight filters through the red and orange leaves, casting warm shadows on a winding path, photorealistic detail."



"An astronaut standing at the base of a towering ice cliff on an alien world, with the aurora reflecting off their helmet visor."



"A peaceful countryside inn with timber framing and blooming flower beds, nestled in a small village surrounded by hills, inviting and nostalgic."



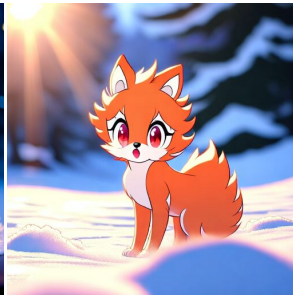
"A close-up of Christmas cookies shaped like stars, resting on a plate beside a steaming mug of cocoa."



"A rugged canyon landscape with red rock formations glowing under the setting sun, and a winding river cutting through the valley."



"A portrait of a snow globe resting on a wooden table, featuring a miniature winter village with glowing lights."



"A serene scene of a Vulpix playing in freshly fallen snow, its fur shimmering under the bright sunlight."



"A lighthouse perched on a rocky coastline, with waves crashing against the cliffs and the beacon casting its light across the sea."



"A traditional pagoda standing tall against a backdrop of a golden sunset, surrounded by lush greenery and sakura blossoms."



"Portrait of a robotic lion with metallic fur and fierce red eyes, roaring fiercely."



"A close-up portrait of an elderly African man with a wise expression, wearing a traditional Kente cloth, ultra HD, photorealistic."



"A scene of Mount Fuji reflected in the still waters of Lake Kawaguchi, surrounded by cherry blossoms under a clear blue sky."



"A vibrant close-up of a dreamcatcher hanging by a window, glowing softly in the golden afternoon light."

Figure 14. Qualitative examples of the text-to-image capability of OmniFlow.