A Model Architecture and Configuration

A.1 Overview

In this section, we provide more details on the model architecture as shown in Table 11, where each modality specific diffuser is based on UNet architecture with different variations detailed in the table. Another notable difference is the video architecture where we add temporal attention and temporal shift as discussed in Section 3.3 and we will discuss its detail in the next section.

Table 11: Hyperparameters for our diffusion models. Note the video and image generation uses the same diffuser.

Modality	**Video (Image) LDM**	**Audio LDM**	**Text LDM**
Hyperparameter			
Architecture	LDM	LDM	LDM
z-shape	$4 \times$ #frames $\times 64 \times 64$	$8 \times 256 \times 16$	$768 \times 1 \times 1$
Channels	320	320	320
Depth	4	2	2
Channel multiplier	1,2,4,4	1,2,4,4	1,2,4,4
Attention resolutions	64,32,16	64,32,16	64,32,16
Head channels	32	32	32
Number of heads	8	8	8
CA embed dim	768	768	768
CA resolutions	64,32,16	64,32,16	64,32,16
Autoencoders	AutoKL	AudioLDM	Optimus
Weight initialization	Stable Diffusion-1.4	-	Versatile Diffusion
Parameterization	ϵ	ϵ	ϵ
Learning rate	$2e-5$	$5e-6$	$5e-5$
Total batch size	256	1024	1024
Diffusion Setup			
Diffusion steps	1000	1000	1000
Noise schedule	Linear	Linear	Linear
β_0	0.00085	0.00085	0.00085
β_T	0.0120	0.0120	0.0120
Sampling Parameters			
Sampler	DDIM	DDIM	DDIM
Steps	50	50	50
η	1.0	1.0	1.0
Guidance scale	2.0	7.5	2.0

A.2 Video LDM Architecture

Except for the base image UNet architecture, we also add temporal attention and temporal shift [2] before each residual block. Following VDM [21], the temporal attention is a transformer attention module where we flatten the height and width dimension to batch size dimension and the self-attention is performed on the time dimension. The temporal shift is illustrated in Fig. 6 where we first split channels into k chunks. Then, we shift the channel dimension numbered 0 to $k-1$ by temporal dimension from 0 to $k-1$ times respectively. Eventually, we concatenate the shifted chunks by the hidden dimension. Note that we use $k=3$ in the illustration for simplicity but $k=8$ in our implementation. We then add a convolution layer before the temporal shift module. Finally, we use residual connection [18] and add the output to the input before the convolution layer. The complete video UNet layer is shown in Fig. 7.

B Model Training

Prompt Encoders Training. As discussed in Section 3.2, we use bridging alignment to perform contrastive learning between all prompt encoders. We use Adam [26] optimizer with learning rate 1e-4 and weight decay 1e-4.

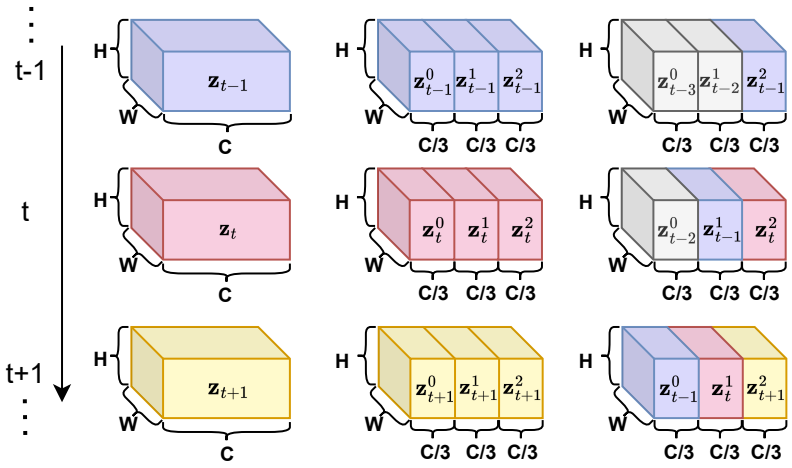

Figure 6: Temporal shift [2] illustration. C, H, W represent channel, height, width, respectively. The vertical line represents time steps from $t-1$, t, and $t+1$. The grey blocks denote "padding tensors".

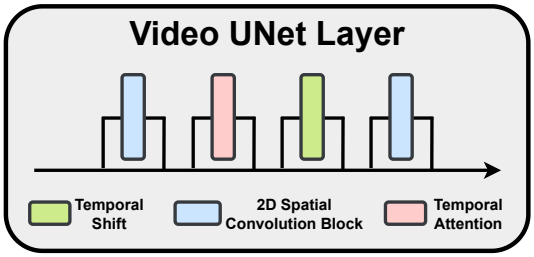

Figure 7: Video UNet layer architecture details including normalization & activation, 2D temporal attention, followed by temporal shift and 1D spatial convolution.

Diffusion Model Training. We train diffusion model with training objectives and hyperparameters detailed in Table 1 and Table 11. For video LDM, we adopt a more specific training curriculum. We adopt curriculum learning on frame resolution and frames-per-second (FPS). First, the diffuser is trained on the WebVid dataset of a 256-frame resolution, with the training objective being text-conditioned video generation. The training clips are sampled from 2-second video chunks with 4 FPS. Second, the model is further trained on HDVILLA and ACAV datasets, with a 512-frame resolution and 8 FPS, and the training objective is image-conditioned video generation (the image is a randomly sampled frame of the clip). Each training clip contains 16 frames sampled from a 2-second video chunk with 8 FPS.

Joint Generation Training. As discussed in Section 3.2, we train joint generation by aligning environment encoders and optimize cross-attention layers only in the diffusion models. We use Adam optimizer with learning rate 1e-5 and weight decay 1e-4.

C Training Datasets

In this section, we introduce more details about the video and audiovisual training datasets.

Video. WebVid [4] is a large-scale dataset of web videos with diverse content, spanning over 40 categories such as sports, cooking, and travel. It contains over 1.2 million video clips (all without sound) that are all at least 30 seconds in duration with video descriptions. We perform **text→video** and video-text contrastive learning task with this dataset. HD-Villa-100M [54] is a large-scale video dataset with over 100 million video clips sourced from YouTube. The dataset covers a wide range of video categories and includes high-quality videos with a resolution of at least 720P. Since it lacks

curated video description and we use the middle frame as image input to perform **image→video** generation.

Audiovisual. SoundNet originally contains over two million sounds and spans a wide range of categories including music, animal sounds, natural sounds, and environmental sounds. We collected all currently accessible 1M videos.

D Limitations & Broader Impacts

While the paper primarily focuses on the technical advancements and potential applications of CoDi, we also consider potential negative social impacts that could arise from the development and deployment of such technology. These impacts can include:

Deepfakes and Misinformation. As part of a common issue for generative AI models, the ability of CoDi to generate realistic and synchronized multimodal outputs also raises concerns about the creation and dissemination of deepfakes. Malicious actors could exploit this technology to create highly convincing fake content, such as fabricated videos or audio clips, which can be used for misinformation, fraud, or other harmful purposes.

Bias and Stereotyping. If the training data used for CoDi is biased or contains stereotypes, the generated multimodal outputs may also reflect these.

E License

We will publicly release our code and checkpoints. We cite licenses from the individual dataset or package we use from the community and provide the following links for references.

LAION-400M: Creative Common CC-BY 4.0

AudioSet: Creative Common CC-BY 4.0

AudioCaps: MIT

Freesound: Creative Commons

BBC Sound Effect: The BBC's Content Licence

SoundNet: MIT

Webvid10M: Webvid

HD-Villa-100M: Research Use of Data Agreement v1.0

PyTorch: BSD-style

Huggingface Transformers: Apache

Torchvision: BSD 3-Clause

Torchaudio: BSD 2-Clause