SYNCPHONY: SYNCHRONIZED AUDIO-TO-VIDEO GENERATION WITH DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-video and image-to-video generation have made rapid progress in visual quality, but they remain limited in controlling the precise timing of motion. In contrast, audio provides temporal cues aligned with video motion, making it a promising condition for temporally controlled video generation. However, existing audio-to-video (A2V) models struggle with fine-grained synchronization due to indirect conditioning mechanisms or limited temporal modeling capacity. We present *Syncphony*, which generates 380×640 resolution, 24fps videos synchronized with diverse audio inputs. Our approach builds upon a pre-trained video backbone and incorporates two key components to improve synchronization: (1) **Motion-aware Loss**, which emphasizes learning at high-motion regions; (2) **Audio Sync Guidance**, which guides the full model using a visually aligned off-sync model without audio layers to better exploit audio cues at inference while maintaining visual quality. To evaluate synchronization, we propose **CycleSync**, a video-to-audio-based metric that measures the amount of motion cues in the generated video to reconstruct the original audio. Experiments on AVSync15 and The Greatest Hits datasets demonstrate that **Syncphony** outperforms existing methods in both synchronization accuracy and visual quality.

1 INTRODUCTION

Video generation has achieved remarkable progress especially in text-to-video (T2V) and image-to-video (I2V). They synthesize visually crisp and temporally coherent videos that match the given text prompt and/or a starting frame. However, we still need additional ways to control the motions that are difficult to control by the texts or the starting frames. For example, texts inherently lack explicit timings of when and how motions would occur, although they may describe motions, e.g., “dog barking” and “striking bowling”. In what rhythm would the dog bark? When is the ball released, how fast does it roll, and when does it hit the pins? Similarly, image-based conditions also face inherent limitations. An image can convey information about the appearance, pose, background, and layout of the scene, but it represents only a static snapshot of a single moment.

In contrast, audio signals inherently carry temporal clues because audio and video share the same temporal axis. Returning to the earlier examples, the accompanying audio would provide how many times and exactly when the dog barks, when the ball is released, how quickly it travels, and when it hits the pins. Therefore, we tackle generating videos that are synchronized to audios.

Even with audio, text, or image conditions, existing audio-to-video methods (Lee et al., 2023; Jeong et al., 2023; Yariv et al., 2023; Zhang et al., 2024) struggle with fine-grained synchronization between audio and motion. These approaches rely on indirect mappings, such as magnitude-based adjustments (Lee et al., 2023) or audio-to-text projections (Jeong et al., 2023; Yariv et al., 2023), which fail to reflect the complex and detailed temporal structures in audio signals. Instead, we directly inject audio features into the visual generation process via a cross-attention mechanism, enabling audio-motion alignment. In parallel, compared to T2V models (Jin et al., 2024a; HaCohen et al., 2024; Blattmann et al., 2023; Wan et al., 2025) which generate high-resolution, high-frame-rate, temporally coherent videos, Zhang et al. (2024) adds temporal layers to an image backbone, training them from scratch with limited data (e.g., 6 fps at 256×256 resolution) leading to broken temporal coherence, such as flickering and saturation artifacts. We address this by leveraging a pretrained video backbone with strong temporal modeling capabilities, resulting in more stable and consistent motion.

Despite these advancements, audio-to-video generation still faces a fundamental challenge: MSE-based objectives alone are insufficient for modeling accurate motion timing and appropriate motion magnitude. In diffusion or flow models, MSE is effective for reconstructing overall visual appearance, but it provides weak supervision for localized, precisely timed motion changes. Therefore, stronger and more targeted supervision is needed in regions that exhibit significant motion.

To this end, we propose Syncphony, which generates high-quality videos at 380×640 resolution, 24fps, and up to 5 seconds in length, and most of all, synchronized to audio. We design Syncphony to have joint self-attention of text-video and audio cross-attention with RoPE on top of a DiT architecture.

For training, we introduce a motion-focused loss that places greater emphasis on regions exhibiting significant visual movement. This encourages the model to better detect motion that is causally linked to the audio signal, even when such motion is highly localized. Furthermore, for sampling, we introduce a novel synchronization guidance strategy that enhances audio-driven motion without compromising visual fidelity. Motivated by the observation that applying traditional classifier-free guidance to audio conditions makes it difficult to train scenes without audio, we instead propose a guidance method that skips the audio layer itself, rather than dropping the audio condition.

Also, we provide a comprehensive set of experiments that evaluate synchronization, visual quality, and semantic alignment across real-world scenarios. In particular, we propose a novel synchronization metric, CycleSync, designed for high-frame-rate video generation, overcoming the limitations of existing metrics that assume unrealistic one-to-one audio-video mappings or operate only at low temporal resolution. Using CycleSync, we demonstrate that our approach successfully models varied audio-motion dynamics. Overall, our method, Syncphony, outperforms existing approaches across all aspects, and we will release our code, models, and evaluation tools to support future research in this direction.

2 Related works

2.1 Text&Image-to-Video generation

Models. Based on the autoregressive models (Yan et al., 2021; Hong et al., 2022; Jin et al., 2024b) and diffusion models (Ho et al., 2022; Brooks et al., 2024), video generative models have been advanced dramatically. Notably, adapting DiT allows huge improvements in high-quality video generation with scalability (Peebles & Xie, 2023; Chen et al., 2023; Wang et al., 2023; HaCohen et al., 2024). Chen et al. (2024), and Valevski et al. (2024) proposed a hybrid approach that combines autoregressive and diffusion models. Upon them, Jin et al. (2024a) proposes both a spatial and temporal feature compression enabling the generation of long videos with high fidelity at a lower training cost. Notably, they only allow text or an image as conditions. On the other hand, our method takes audio as condition.

Guidance. Guidance mechanisms play a crucial role in improving sample quality across generative models. Classifier-Free Guidance (Ho & Salimans, 2022) interpolates between conditional and null-conditional predictions to enhance visual fidelity, but requires models to be explicitly trained with null conditions. Spatiotemporal Skip Guidance (Hyung et al., 2025) constructs a weak model by skipping visually sensitive layers, and interpolates its predictions with those of the full model to improve quality without additional training. However, in T2I and T2V architectures, visual and semantic representations are often deeply entangled, making such selective skipping difficult.

2.2 Audio-to-Video generation

Recent works on Audio-to-Video (A2V) generation have explored how to synthesize temporally aligned videos conditioned on audio inputs. Lee et al. (2023) modulates cross-attention weights based on audio amplitude to control video. Although this approach is simple, amplitude alone does not transfer the semantic and temporal structure of audio, resulting in weak fine-grained synchronization. On the other hand, Jeong et al. (2023); Yariv et al. (2023) project audio embeddings into a text embedding space and generate frames using pre-trained text-to-video (T2V) models. This indirect audio-to-motion mapping is a bottleneck in delivering temporal expressiveness and hinders precise alignment between audio cues and motion transitions. AVSyncD (Zhang et al., 2024) injects audio layers into a Stable Diffusion-based text-to-image (T2I) model, but it is limited to the T2I backbone's spatial resolution and suffers from relatively shallow temporal modeling capacity. Although Zhang et al. (2024) further introduces synchronization guidance, this requires additional training and often causes flickering, degrading visual smoothness. While talking head models have shown strong lip-sync performance for speech (Wang et al., 2025), they are limited to facial motion and human voice. We instead focus on non-speech sounds and general visual motion, which could complement lip-sync systems in real-world audio scenarios.

In contrast to these prior approaches, our method builds on the strengths of diffusion transformer-based T2V models to directly incorporate fine-grained temporal audio cues. By leveraging a high-capacity backbone capable of high-resolution, high-frame-rate generation and introducing targeted synchronization guidance and loss-level modifications, our model achieves accurate audio-motion synchronization across diverse domains while preserving visual fidelity.

Synchronization metrics. Existing synchronization metrics, such as RelSync (Zhang et al., 2024) and AlignSync (Zhang et al., 2024), require downsampling to 6 fps, which reduces temporal resolution and undermines the evaluation of fine-grained motion. AV-Align (Yariv et al., 2023) assumes a one-to-one correspondence between motion and audio peaks, which fails to generalize to real-world scenarios involving preparatory or residual motion. For example, a hammer moves before the impact sound and stops at the sound. To address these limitations, we propose a new synchronization metric that supports high frame rates and generalizes to real-world audio–motion scenarios.

3 SYNCPHONY

3.1 OVERVIEW

Our goal is to generate high-quality videos that have motions aligned with audio inputs. We build upon a pretrained autoregressive diffusion transformer (Jin et al., 2024a), which sequentially synthesizes consecutive video chunks by denoising each chunk for given a previous chunk and a text prompt. As shown in Figure 1, our model takes an initial frame, a text prompt, and an audio waveform as input. The initial frame is encoded into a latent $\boldsymbol{z}_0$ using a VAE, which serves as the starting point for generating video latents $\{\boldsymbol{z}_l\}_{l=1}^{L}$. Text features are extracted from pretrained encoders (Raffel et al., 2020; Radford et al., 2021), and audio features $\{\boldsymbol{a}_i\}_{i=0}^{L_{\text{audio}}}$ are obtained from DenseAV (Hamilton et al., 2024) encoder. Each transformer block includes a joint self-attention layer, which attends over the concatenated sequence of text tokens and video latents. To incorporate audio, we insert a cross-attention layer before the joint self-attention layer in the later blocks, allowing each video latent to attend to its aligned audio segment for fine-grained synchronization.

In the following subsections, we propose a motion-aware loss that puts more weight on the regions with large motions for training (3.2), introduce a sampling strategy designed to sample the videos toward better audio-conditional outputs (3.3), and describe additional architectural details (3.4).

3.2 MOTION-AWARE LOSS

Conventional video generation models typically use Mean Squared Error (MSE) loss, which measures the pixel or latent-level discrepancy between predicted and ground-truth frames. While MSE is effective for general reconstruction, it treats all spatial and temporal regions equally, without distinguishing between static and dynamic areas. As a result, even when the model produces inaccurate motion timing or insufficient movement—e.g., a delayed or insufficient gun firing motion—the error remains low if the overall appearance is visually close to the ground truth. This may lead the model to interpret poorly synchronized predictions as successful outputs, weakening its ability to learn precise audio-visual alignment.

This limitation is particularly critical in real-world scenarios where audio cues correspond to distinct, temporally localized motion, such as a drum hit or bowling pin collision. In such cases, accurate timing and appropriate motion magnitude are essential for maintaining natural synchronization.

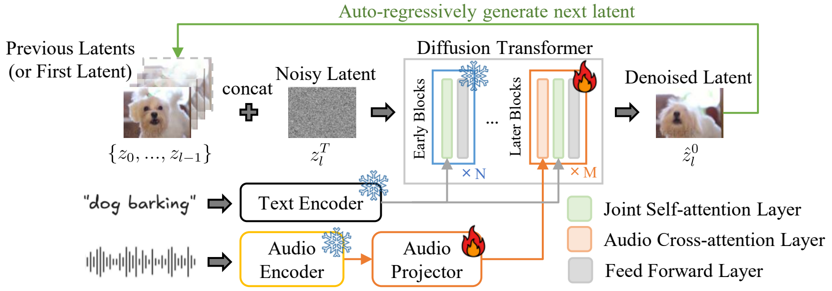
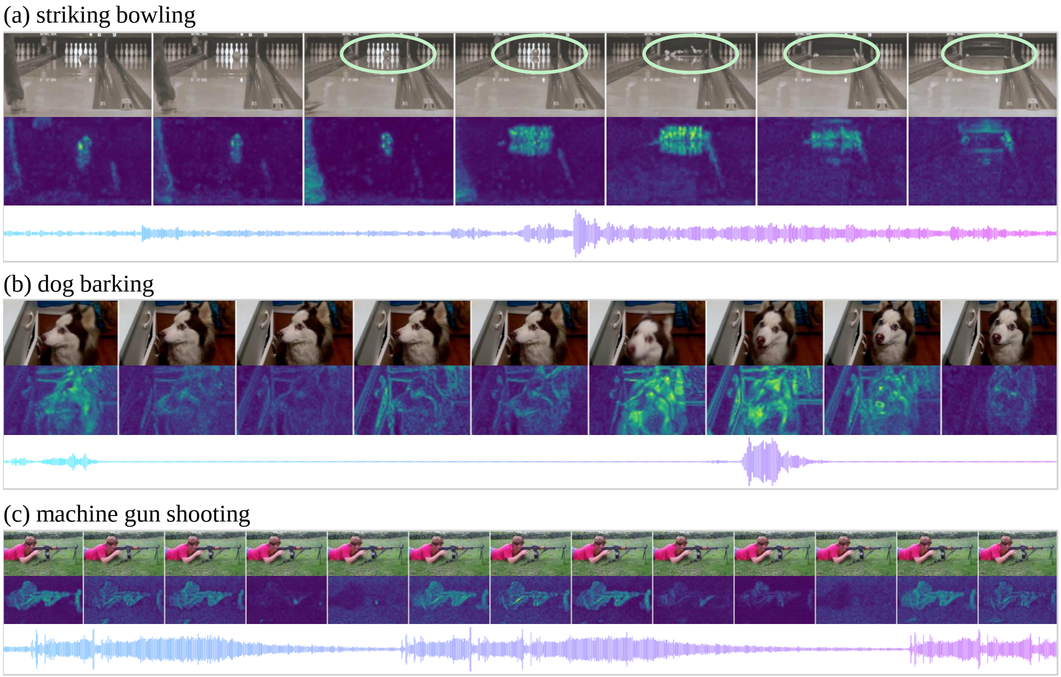


Figure 1: **Overview of our video generation framework.** Given an initial frame, a text prompt, and an audio waveform, the model autoregressively predicts each video latent through iterative denoising. The Diffusion Transformer is divided into two groups of transformer blocks: the early blocks (frozen, blue) and the later blocks (trainable, orange). Text features are injected into all blocks via joint self-attention, whereas audio cross-attention layers are inserted *only into the later (trainable) blocks*. For brevity, latents are visualized as RGB frames, but they are spatiotemporal features extracted by VAE.



(a) striking bowling

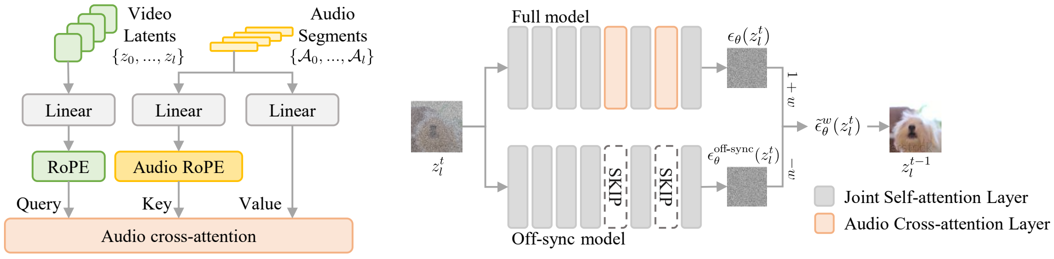
(b) dog barking

(c) machine gun shooting

Figure 2: Visualization of video frames (top), latent difference maps (middle), and audio waveforms (bottom) over time. In (a) and (b), the latent differences correspond to key audio events such as pin collisions and barking. In (c), although motion is not clearly visible in raw frames, latent differences still reveal temporal alignment with machine gun audio signals.

Therefore, it is necessary to provide stronger and more focused supervision to areas involved in high-motion events.

In Figure 2, we observe that latent differences between adjacent frames tend to correlate with audio events, even when the corresponding motion is not clearly visible in the video frames, as in (c). Based on this observation, we propose a **Motion-aware Loss** that amplifies the learning signal according to the intensity of ground-truth motion. This amplifies supervision at moments of significant movement, encouraging the model to better capture and align motion with audio cues.



(a) Audio Cross-attention with Audio RoPE

(b) Audio Sync Guidance

Figure 3: (a) Audio cross-attention with Audio RoPE. Each video latent attends to a local audio segment using cross-attention. RoPE is applied to both video queries and audio keys, using a shared positional embedder to align modalities in relative position space. (b) Audio Sync Guidance. An off-sync model that skips the audio cross-attention layers guides the full model to better utilize audio cues during sampling.

The proposed loss function is defined as:

$$\mathcal{L} = \|\hat{\boldsymbol{z}}_l - \boldsymbol{z}_l^{GT}\|^2 + \lambda \cdot \|(\hat{\boldsymbol{z}}_l - \boldsymbol{z}_l^{GT}) \odot \underbrace{(\boldsymbol{z}_l^{GT} - \boldsymbol{z}_{t-1}^{GT})}_{\text{motion}}\|^2, \tag{1}$$

where $\hat{\boldsymbol{z}}_l$ and $\boldsymbol{z}_l^{GT}$ denote the predicted and ground-truth latents at the t-th position in the video latent sequence, respectively, and $\odot$ denotes element-wise multiplication. The second term weights prediction errors according to the magnitude of ground-truth motion between consecutive frames, with λ as a hyperparameter (we set $\lambda = 1$).

This design ensures that prediction errors during dynamic motion are penalized more heavily than those in static periods, encouraging the model to better capture the timing and intensity of important motions.

Importantly, we do not directly use audio signal strength as a supervision signal. This is because audio and motion do not always exhibit a one-to-one temporal alignment: motion may precede or follow audio events, or span multiple frames. For instance, a lion may move before roaring, or a bowling ball may roll before impact. By focusing on ground-truth motion magnitude rather than audio signal strength, our loss design allows the model to learn natural synchronization patterns without rigidly assuming direct temporal alignment.

One might worry that the motion-aware loss could be negatively affected by camera motion or background movement that is unrelated to audio events. However, because our formulation weights the loss based on ground-truth motion intensity itself, the model naturally learns to differentiate between motion that is causally linked to audio cues and motion that is not.

Overall, Motion-aware Loss strengthens the model's attention to motion-relevant regions, encouraging the model to learn diverse audio-motion relationships and generate natural, well-aligned motion sequences. Additional notes on motion-aware loss are provided in Appendix A.

3.3 Audio Sync Guidance

In audio-conditioned video generation, audio-driven layers are responsible for injecting timing cues into the visual dynamics. However, these cues from audio are not always strong or clear, so it's hard for the model to determine whether to reflect them in the generated motion. For example, when a drumstick hits a plastic surface, a subtle crinkling sound helps specify the exact target. Relying only on the coarse impact sound can misplace the strike.

To address this, we propose **Audio Sync Guidance** (ASG) that reinforces the influence of audio signals so the model better captures and reflects them in motion. As illustrated in Figure 3(b), we run two branches that share the same visual backbone: a *full* model with audio cross-attention layers enabled, and an *off-sync* model where only those layers are disabled. We found that the off-sync model produces outputs that are visually similar to the full model's, yet desynchronized (Please see the

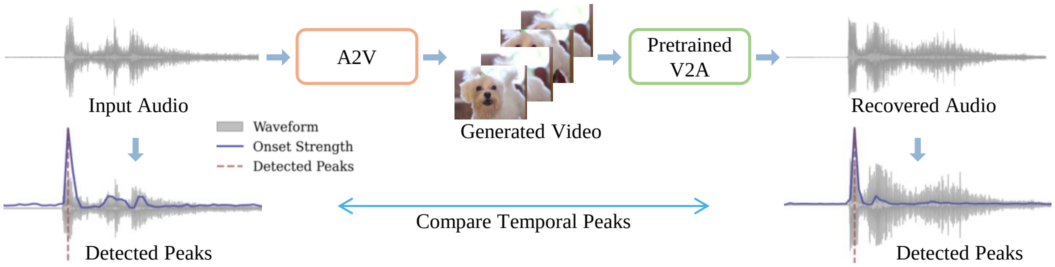


Figure 4: **CycleSync** metric pipeline. The generated video is fed into a pretrained Video-to-Audio (V2A) model to reconstruct audio. We compare temporal peaks between the reconstructed and original audio signals. High peak correspondence indicates that the generated video accurately preserves the timing structure of the original audio.

supporting experiments in Appendix B.2). Thus, the difference between the two predictions isolates the synchronization component and could serve as guidance for synchronization. By adding this difference back into the full model's output, ASG amplifies the influence of audio and encourages more synchronized motion generation.

Formally, given a latent z_l^t at denoising timestep t, the guided prediction is

$$\tilde{\epsilon}_\theta^w(\boldsymbol{z}_l^t) = \epsilon_\theta(\boldsymbol{z}_l^t) + w\left(\epsilon_\theta(\boldsymbol{z}_l^t) - \epsilon_\theta^{\text{off-sync}}(\boldsymbol{z}_l^t)\right), \tag{2}$$

where $\epsilon_\theta(\boldsymbol{z}_l^t)$ is the denoising output of the full model, $\epsilon_\theta^{\text{off-sync}}(\boldsymbol{z}_l^t)$ is the output with audio layers skipped, and w is the guidance-strength hyperparameter controlling the degree of audio emphasis. For clarity, we omit the integration with Classifier-Free Guidance; please see an Appendix B.3 for the connection to CFG.

In summary, ASG highlights audio cues by disabling only the audio cross-attention layers in the off-sync model, improving audio–motion alignment while preserving visual fidelity without additional training.

3.4 Architectural Details

Training layer selection. To leverage the pretrained video backbone effectively, we identify which transformer blocks to fine-tune through a layer-wise sensitivity analysis. We find that earlier layers primarily control spatial structure and semantic fidelity, whereas later layers govern temporal dynamics and motion refinement. Based on this, we insert audio-driven cross-attention only into the later blocks and fine-tune them jointly. This strategy allows the model to focus on synchronizing motion with audio signals while maintaining high visual fidelity and leveraging the strong generalization capability of the pretrained I2V backbone. Details are provided in Appendix D.1.2.

Audio conditioning. To synchronize video motion precisely with audio cues, we apply Rotary Positional Embedding to inject relative temporal information into the audio features during cross-attention (Audio RoPE), as illustrated in Figure 3(a). We confirm that Audio RoPE leads to tighter temporal alignment between motion and sound events. Implementation details and an ablation study are provided in Appendix D.2.

4 Evaluating Audio–Motion Synchronization

Although prior synchronization metrics (Zhang et al., 2024; Yariv et al., 2023) are useful, they require a low fixed frame-rate or introduce wrong assumption that the peak magnitudes of audio and video should match. It makes them less reliable for high-frame-rate videos or real-world audio-motion scenarios.

To address these limitations, we propose **CycleSync**, a synchronization metric based on a video-to-audio (V2A) reconstruction process. Instead of directly comparing motion and audio peaks, CycleSync evaluates whether the motion in a video provides enough signal to reconstruct the temporal structure of the original audio. As illustrated in Figure 4, we feed the generated video into a state-of-the-art V2A model (Viertola et al., 2025), and compare the resulting audio to the original input audio by aligning their temporal peaks. By aligning audio peaks between the original and recovered audio, we can assess whether the generated video contains sufficient timing and motion cues to reproduce the original audio structure.

Formally, given an original audio signal $\boldsymbol{a}$ and a generated video $\hat{\boldsymbol{v}}$, we reconstruct the audio $\hat{\boldsymbol{a}}$ using a pretrained video-to-audio model f_{v2a}:

$$\hat{\boldsymbol{a}} = f_{\text{v2a}}(\hat{\boldsymbol{v}}). \tag{3}$$

Let $\mathbb{A} = P(\boldsymbol{a})$ and $\hat{\mathbb{A}} = P(\hat{\boldsymbol{a}})$ be the sets of onset peaks extracted from $\boldsymbol{a}$ and $\hat{\boldsymbol{a}}$, respectively. The CycleSync score is computed via symmetric temporal matching with tolerance δ:

$$\text{CycleSync} = \frac{1}{2|\mathbb{A} \cup \hat{\mathbb{A}}|} \left(\sum_{\boldsymbol{a} \in \mathbb{A}} \mathbf{1} \left[\exists\, \hat{\boldsymbol{a}} \in \hat{\mathbb{A}},\ |\boldsymbol{a} - \hat{\boldsymbol{a}}| \leq \delta \right] + \sum_{\hat{\boldsymbol{a}} \in \hat{\mathbb{A}}} \mathbf{1} \left[\exists\, \boldsymbol{a} \in \mathbb{A},\ |\boldsymbol{a} - \hat{\boldsymbol{a}}| \leq \delta \right] \right), \tag{4}$$

where δ is a temporal tolerance and $\mathbf{1}[\cdot]$ is the indicator function.

A higher CycleSync score indicates that the generated video preserves the timing structure of the original audio.

5 Experiment

5.1 Experimental setup

Dataset. We evaluate our model using AVSync15[1] (Zhang et al., 2024) and TheGreatestHits[2] (Owens et al., 2016), whose samples have synchronized audio and video.

Baselines. We compare our method against the following baseline models: We employ the Pyramid Flow Video model (**I+T2V**) (Jin et al., 2024a), which conditions on text and image inputs, TempoTokens (**T+A2V**), which conditions on audio and text inputs, and AVSyncD (**I+T+A2V**), which conditions on audio and image inputs. For a closer comparison between I2V and A2V, we also employ a fine-tuned version of our model without audio layers, denoted as Pyramid Flow (fine-tuned).

Evaluation metrics. To assess visual quality, we report **FID** (Heusel et al., 2017) (Fréchet Inception Distance) and **FVD** (Unterthiner et al., 2019) (Fréchet Video Distance). FID measures the fidelity of individual frames, while FVD evaluates the spatiotemporal coherence of the entire video. To assess semantic alignment with conditioning modalities, we use **Image-Text Similarity (IT)** and **Image-Audio Similarity (IA)**. IT evaluates how well the generated frames correspond to the input text prompt using CLIP (Radford et al., 2021), while IA measures semantic alignment between audio signals and visual content using ImageBind (Girdhar et al., 2023). To assess audio-motion synchronization, we report **CycleSync**, which evaluates whether the generated videos contain sufficient motion cues synchronized with audio signals. We also conduct a user study on 150 videos from the AVSync15 dataset. Participants compare video pairs across three criteria—**synchronization (Sync)**, **image quality (IQ)**, and **frame consistency (FC)**. Implementation details of the user study are provided in Appendix E.

Implementation details. We use the pretrained Pyramid Flow Video model (Jin et al., 2024a) as the backbone. Generated videos are up to 5 seconds long at 24 fps and 380 × 640 resolution. Audio is sampled at 16kHz. During training, we randomly sample training clips from different temporal segments of each video to improve generalization to various audio-motion alignments. During evaluation, we extract three 2-second clips at distinct time points per video. The AVSync15 dataset

[1] AVSync15 is a curated subset of the VGGSound dataset consisting of 1,500 videos from 15 action-related classes.

[2] TheGreatestHits is a dataset where a person strikes various objects with drumsticks, producing distinct impact sounds closely tied to visual motion.

Table 1: Quantitative results on the AVSync15 dataset.

Input	Model	FID ↓	FVD ↓	IA ↑	IT ↑	CycleSync ↑	User Study IQ ↑	User Study FC ↑	User Study Sync ↑
T+A	TempoTokens (Yariv et al., 2023)	8.9	4187.2	27.24	27.88	$13.10_{\pm 1.16}$	-	-	-
I+T	Pyramid Flow (Jin et al., 2024a)	8.9	550.7	34.99	29.34	$14.25_{\pm 1.39}$	-	-	-
	Pyramid Flow (fine-tuned)	**8.5**	294.6	36.89	30.02	$12.34_{\pm 1.14}$	-	-	-
I+T+A	AVSyncD (Zhang et al., 2024)	9.2	491.5	35.23	30.18	$16.38_{\pm 1.38}$	30	18	78
	Ours	**8.5**	**293.1**	**37.02**	**30.23**	$\mathbf{16.48}_{\pm \mathbf{1.28}}$	**270**	**282**	**222**
	Groundtruth	-	-	37.06	30.18	$22.15_{\pm 1.8}$			

Table 2: Quantitative results on the TheGreatestHits dataset.

Input	Model	FID ↓	FVD ↓	IA ↑	IT ↑	CycleSync ↑
I+T	Pyramid Flow (Jin et al., 2024a)	**6.5**	350.5	13.95	18.42	$7.41_{\pm 0.83}$
	Pyramid Flow (fine-tuned)	6.9	195.6	**14.13**	20.86	$9.23_{\pm 0.92}$
I+T+A	AVSyncD (Zhang et al., 2024)	6.8	327.8	12.35	**21.77**	$9.89_{\pm 0.84}$
	Ours	6.7	**166.2**	13.83	19.64	$\mathbf{16.18}_{\pm \mathbf{1.26}}$
	Groundtruth	-	-	14.68	19.47	$15.99_{\pm 1.5}$

provides 450 clips, and TheGreatestHits provides 732 clips for evaluation. We use CLIP (Radford et al., 2021) and DenseAV (Hamilton et al., 2024) audio backbone as our text encoder and audio encoder, respectively. We train our model on 4 NVIDIA RTX 3090 GPUs (24GB).

5.2 Main results

5.2.1 Model comparison

Quantitative results. Tables 1 and 2 show results on AVSync15 and TheGreatestHits. Across both datasets, our model consistently outperforms baselines in synchronization accuracy while maintaining competitive visual and semantic quality. Compared to AVSyncD, our model achieves higher CycleSync scores and lower FID/FVD, indicating improved temporal coherence. User study further confirms these gains, with clear preference for our model in synchronization, image quality, and frame consistency.

On TheGreatestHits, our model even surpasses the ground-truth CycleSync score. We attribute this to the generated videos exhibiting strong and clear motion aligned with audio events, whereas ground-truth videos often contain off-event movements or sounds, such as hovering or background noise. These results suggest that our model demonstrates greater sensitivity to audio cues under synchronization metrics. Additional results using existing metrics (AV-Align, RelSync, AlignSync) are reported in Appendix C.4.

Qualitative results. Figure 5 presents qualitative comparisons among ours, Pyramid Flow (fine-tuned), and AVSyncD. Our method produces clearer motion dynamics and stable appearances, whereas AVSyncD often suffers from saturation artifacts and weakened motion. We recommend watching the supplemental videos to see additional qualitative results (Appendix F.1).

5.2.2 Metric comparison

Controlled metric comparison. We analyze synchronization robustness under controlled temporal shifts. Details are provided in Appendix C.2. As shown in Figure 7, CycleSync is markedly more sensitive to temporal misalignment than other metrics, clearly differentiating synchronized from desynchronized cases.

Human alignment validation for CycleSync. We conduct a user study to assess how well CycleSync aligns with human perception. Details are provided in Appendix C.3. As shown in Table 7 and Table 8, CycleSync achieves the highest positive correlation with human preference, while prior metrics show weak or negative trends.

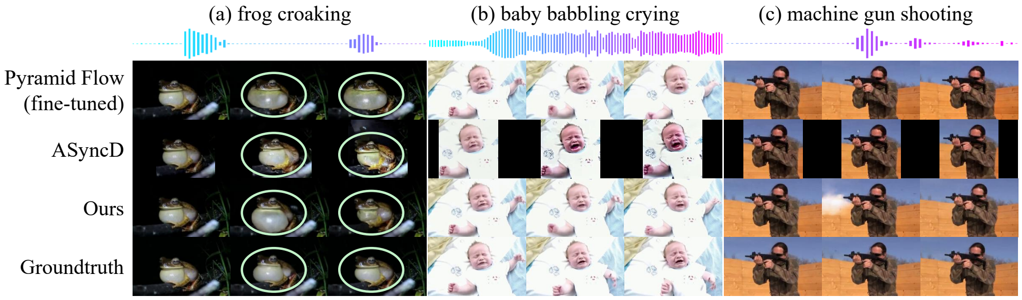


Figure 5: **Qualitative comparison of generated videos.** Our method produces more explicit and temporally consistent motion compared to both baselines.

Table 3: Ablation results on AVSync15.

Model Variant	**FID** ↓	**FVD** ↓	**CycleSync** ↑
w/o Motion-aware Loss	**8.4**	305.9	15.18±1.48
Full model w/o ASG	8.5	299.1	15.31±1.49
Full model w/ ASG ($w = 1$)	8.5	294.2	15.94±1.56
Full model w/ ASG ($w = 4$)	8.7	298.3	16.26±1.4
Full model w/ ASG ($w = 2$)	8.5	**293.1**	**16.48**±1.28

5.3 Ablation Study

Effect of Motion-aware Loss. When trained without Motion-aware Loss, the model tends to produce weaker and less clearly timed motions. As shown in Figure 6, it often fails to initiate or terminate motion in sync with the corresponding audio events. Incorporating Motion-aware Loss improves both the magnitude and temporal precision of motion, particularly at the onset and offset of dynamic actions. This is because Motion-aware Loss selectively amplifies learning signals at points of high motion intensity, guiding the model to learn more precisely on the timing structure of audio-driven actions.

Effect of Audio Sync Guidance. As shown in Tables 3 , applying Audio Sync Guidance (ASG) with scale $w = 2$ improves synchronization metrics while preserving visual fidelity. Increasing the scale to $w = 4$ yields marginal gains in synchronization, but introduces over-exaggerated motion (e.g., frog inflation or recoil motion), which slightly degrades visual realism reflected in higher FVD, while FID remains stable.

6 Conclusion

We introduced *Syncphony*, a high-quality *audio-synchronized video* generation framework. By conditioning on text, image, and audio inputs, our model captures both the semantic context and the fine-grained temporal dynamics of motion. To improve audio-motion alignment, we incorporated two key techniques: **Motion-aware Loss** encourages accurate timing by emphasizing high-motion regions, and **Audio Sync Guidance** enhances sensitivity to audio signals during inference while maintaining visual quality. To better evaluate synchronization accuracy, we proposed **CycleSync**, a video-to-audio-based metric that measures whether the generated video retains sufficient motion cues to reconstruct the original audio. This enables a more reliable assessment than the existing metrics in real-world scenarios.

Limitation. While our Motion-aware Loss improves synchronization in audio-to-video generation by emphasizing motion intensity, it assumes that the motion intensity corresponds to the audio signal, and it does not distinguish semantically meaningful movements from noisy movements, which may induce wrong supervision. Addressing this limitation by incorporating semantic understanding of motion or refining the noisy movements could not only improve synchronization but also enable

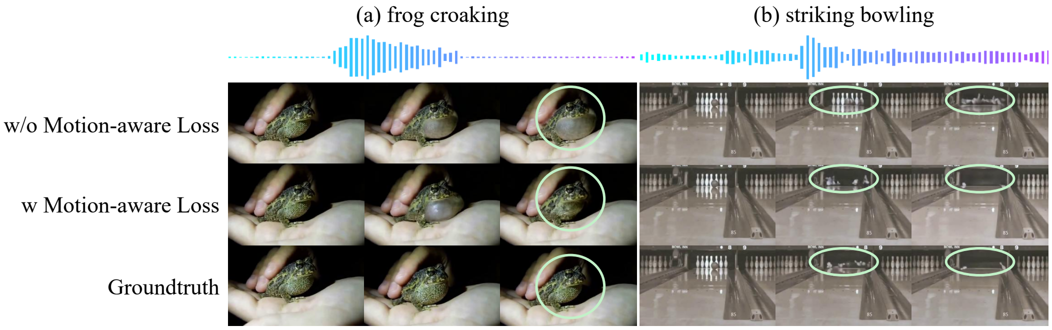


Figure 6: **Ablation of Motion-aware Loss.** (a) Without Motion-aware Loss, the model fails to terminate the motion correctly at the end of the audio. (b) It also fails to trigger motion at the correct audio onset. In contrast, with Motion-aware Loss, the model generates motion that more accurately aligns with the beginning and end of the audio event.

broader applicability to general video generation tasks without audio, where dynamic and expressive motion is important. We also note that the limitations of CycleSync as a synchronization metric are discussed in Appendix C.5.

7 Ethics statement

As a generative model, our method could be used to facilitate deceptive interactions that would cause harm, such as fraud. It could be used to impersonate public figures and influence political processes, or as a tool to promote hate speech or abuse. To address this, we will include explicit license terms and usage guidelines to promote ethical and lawful use, referencing best practices such as the Adobe Generative AI User Guidelines. If the model is released, implement safeguards such as prompt or image filtering to restrict high-risk applications, including impersonation or politically manipulative content.

8 Reproducibility statement

Key components of our implementation are provided in the supplementary materials, and detailed descriptions of our method, training, inference, and evaluation are included in the appendix. We will release our code, trained models, and evaluation tools to ensure reproducibility.

References

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023.

Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis.

In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28901–28911, 2025.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Mark Hamilton, Andrew Zisserman, John R Hershey, and William T Freeman. Separating the" chirp" from the" chat": Self-supervised visual grounding of sound and language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13117–13127, 2024.

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. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2022.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11006–11015, 2025.

Yujin Jeong, Wonjeong Ryoo, Seunghyun Lee, Dabin Seo, Wonmin Byeon, Sangpil Kim, and Jinkyu Kim. The power of sound (tpos): Audio reactive video generation with stable diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7822–7832, 2023.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024a.

Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. *arXiv preprint arXiv:2402.03161*, 2024b.

Seungwoo Lee, Chaerin Kong, Donghyeon Jeon, and Nojun Kwak. Aadiff: Audio-aligned video synthesis with text-to-image diffusion. In *CVPR Workshop on Content Generation*, 2023.

Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. Visually indicated sounds. In *CVPR*, 2016.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. In *arXiv*, 2019.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. Temporally aligned audio for video with autoregression. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report, 2023.

Mengchao Wang, Qiang Wang, Fan Jiang, Yaqi Fan, Yunpeng Zhang, Yonggang Qi, Kun Zhao, and Mu Xu. Fantasytalking: Realistic talking portrait generation via coherent motion synthesis. *arXiv preprint arXiv:2504.04842*, 2025.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Guy Yariv, Itai Gat, Sagie Benaim, Lior Wolf, Idan Schwartz, and Yossi Adi. Diverse and aligned audio-to-video generation via text-to-video model adaptation, 2023.

Lin Zhang, Shentong Mo, Yijing Zhang, and Pedro Morgado. Audio-synchronized visual animation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024.

A Additional notes on Motion-aware Loss

Motion-aware Loss is designed based on ground-truth motion magnitude, not audio amplitude. This reflects the fact that motion peak and audio peak do not always exhibit one-to-one temporal alignment. Instead, audio events are often accompanied by diverse motion patterns that vary by context.

For example, some events, such as a gunshot or a dog's bark, occur with motion and sound nearly perfectly aligned. However, many others do not. A lion may start moving its mouth and body before emitting a roar. A person winds up before throwing a ball. A hammering motion may start before the sound and end just as the impact occurs. A trombone player moves the instrument before the sound begins. A bowling ball rolls with a low rumble before producing a sharp impact sound when it hits the pins.

Therefore, synchronizing motion to audio does not mean matching peak amplitudes. Rather, it involves capturing the causal and contextual patterns of motion that correspond to different types of audio events. Our loss focuses on motion regions, encouraging the model to learn this alignment without relying on rigid audio-based timing. This design encourages the model to learn various audio-motion relationships, leading to natural audio-visual aligned video generation.

B Audio Sync Guidance

B.1 Classifier Free Guidance for Audio Guidance

Conventional classifier-free guidance (CFG) is trained by randomly dropping conditioning inputs, a strategy that has proven highly effective in models such as text-to-image and class-conditional image generation. However, we observe that directly applying this random-drop strategy to audio conditioning leads to degraded performance.

Table 4: **Analysis of Audio Sync Guidance.** The full model includes audio layers, whereas the off-sync model skips them.

Model Variant	**FID** ↓	**FVD** ↓	**CycleSync** ↑
Off-sync model	8.5	294.6	12.34$\pm$1.14
Full model	8.5	299.1	15.31$\pm$1.49
Full model w/ ASG	8.5	**293.1**	**16.48**$\pm$**1.28**

Unlike text, audio carries a meaningful semantic interpretation even when its value is *zero*. In other words, *silence* is itself an informative condition. When audio inputs are randomly dropped during training, the model becomes unable to properly model silence and fails to understand its distinct role compared to the absence of conditioning. Our experiments show that this mismatch significantly harms synchronization quality.

To address this issue, instead of dropping audio conditions, we propose an Audio Sync Guidance mechanism that selectively skips audio-conditioning layers during inference. This approach preserves the semantic meaning of silence while preventing the model from being exposed to ambiguous training signals. In the following section, we describe this strategy in more detail.

B.2 Differences between full and off-sync model in Audio Sync Guidance

To better understand how Audio Sync Guidance contributes to synchronization, we evaluate whether an off-sync model—formed by skipping the audio layers—can still retain appearance and overall motion. As shown in the last row of Figure 5 and "Off-sync model" of Table 4, the model remains out of synchronization but still preserves appearance (FID) and motion quality (FVD). Since the visual quality remains similar between the full and off-sync models, their difference primarily captures audio-related motion cues. By adding this difference back into the full model's output, Audio Sync Guidance amplifies the influence of audio and encourages more synchronized motion generation.

B.3 Integration of CFG and Audio Sync Guidance

Classifier-Free Guidance (Ho & Salimans, 2022) interpolates between conditional (full) and null-conditional predictions to enhance visual fidelity:

$$\tilde{\epsilon}_\theta(z_l^t) = \epsilon_\theta(z_l^t, c) + w_t \left(\epsilon_\theta(z_l^t, c) - \epsilon_\theta(z_l^t, c_\varnothing)\right) \tag{5}$$

At inference time, the Audio Sync Guidance and CFG are combined additively:

$$\tilde{\epsilon}_\theta(z_l^t) = \epsilon_\theta(z_l^t, c) + w_a \left(\epsilon_\theta(z_l^t, c) - \epsilon_\theta^{\text{off-sync}}(z_l^t, c)\right) + w_t \left(\epsilon_\theta(z_l^t, c) - \epsilon_\theta(z_l^t, c_\varnothing)\right) \tag{6}$$

In our implementation, we use $w_a = 2$ and $w_t = 4$.

B.4 Audio Sync Guidance compared to prior skip-based method

Audio Sync Guidance (ASG) is inspired by Hyung et al. (2025) but differs in both purpose and design.

Hyung et al. (2025) improves visual fidelity by constructing a weak model that skips visually sensitive layers and using it to guide the full model. In T2I/T2V settings, however, semantic and visual features are heavily entangled, making such selective skipping difficult, model-dependent, and prone to unintended degradations.

ASG instead targets synchronization. We skip only the audio-injection layers, creating an off-sync model that preserves appearance but ignores audio cues. The difference between this off-sync and the full model isolates synchronization as the guidance signal (see Appendix B.2).

This design is suited to A2V architectures, where audio and visual pathways are explicitly separated. Skipping only the audio pathway perturbs synchronization without affecting visual fidelity, enabling precise and stable guidance for improved audio–motion alignment.

C Evaluation metrics for synchronization

C.1 Implementation details of CycleSync

We use V-AURA (Viertola et al., 2025) as the pretrained video-to-audio model f_{v2a}, selected for its demonstrated effectiveness in generating general-class, temporally aligned audio from videos. For peak detection, we use `librosa.onset.onset_detect`, and δ is fixed at 5 milliseconds.

C.2 Controlled metric comparison

To evaluate the effectiveness of CycleSync, we compare it with three existing synchronization metrics: AV-Align (Yariv et al., 2023), AlignSync, and RelSync (Zhang et al., 2024). We apply six levels of synchronization shift to video clips from the AVSync15 (Zhang et al., 2024) and TheGreatestHits (Owens et al., 2016) datasets.

implementation detail. We extract three 2-second clips per video with linear intervals. To ensure valid comparison under delay shifts, clips are sampled starting 0.5 seconds into the video, allowing up to 0.5 seconds of temporal shift. Videos shorter than 2.5 seconds are excluded. It results in 438 clips from 150 videos in AVSync15, and 732 clips from 244 videos in TheGreatestHits.

AlignSync and RelSync are evaluated on videos downsampled to 6 fps. AV-Align is measured at 6 fps unless otherwise noted as 24 fps. CycleSync (Ours) is evaluated at 24 fps videos.

Synchronization configurations. A sample type with *"Perfect Sync"* represents that Ground-truth audio pairs with its original video. The other sample types with *"Delay 0.1s–0.5s"* represent that the video is temporally shifted by the indicated delay relative to its audio.

C.2.1 Results and analysis

Figure 7 shows how each metric responds to increasing audio-video misalignment. We observe that existing metrics often struggle to clearly separate perfectly synchronized samples from delayed ones, whereas CycleSync scores drop sharply with the misalignments. For absolute metric values, please refer to Table 5 and Table 6.

AV-Align. The performance of AV-Align varies significantly depending on the frame rate. At 24 fps, we would expect the highest score for perfectly synchronized samples, but in both AVSync15 and TheGreatestHits, delayed samples receive higher scores than the ground-truth alignment. At 6 fps, AV-Align becomes more stable, but the separation between perfect and delayed cases remains limited. This suggests that the metric may not reliably reflect fine-grained temporal misalignment at higher frame rates.

Moreover, as shown in Appendix C.4, there are cases where models without audio conditioning obtain higher AV-Align scores than models explicitly guided by audio. This is because AV-Align assumes a strict one-to-one correspondence between peaks in the audio and motion signals—an assumption that often does not hold in natural scenarios, where motion may precede or follow audio cues.

AlignSync and RelSync. AlignSync and RelSync generally show decreasing scores as the degree of delay increases, indicating sensitivity to misalignment. However, they do not show clear differences between perfectly synchronized samples and delayed ones, especially on TheGreatestHits dataset. In addition, both metrics are designed for evaluation at 6 fps, which makes it difficult to assess the performance of models operating at higher frame rates, such as 24 fps.

We also observe cases where models without audio conditioning receive higher scores than those guided by audio (see Appendix C.4). One possible explanation is that these metrics are more effective when evaluating sequences that are simple temporal shifts of the same ground-truth content, as assumed during training. In contrast, when the evaluated sequence differs from the original ground-truth content, the metrics may no longer provide reliable scores.

CycleSync. CycleSync clearly distinguishes perfectly synchronized samples from those with temporal misalignment. Once the delay exceeds a certain threshold, the differences between misaligned cases become less pronounced. In other words, the score is not strictly monotonically decreasing.

This is due to CycleSync's use of a fixed 0.05s tolerance window to determine alignment between onset peaks in the original and reconstructed audio. While this allows for robust separation between synchronized and unsynchronized cases, it does not explicitly quantify how far misaligned peaks fall beyond the threshold.

This behavior arises because the tolerance hyperparameter is set to 0.05s, and CycleSync determines alignment between onset peaks in the original and reconstructed audio within this fixed tolerance window. While this design provides robust separation between synchronized and unsynchronized cases, it does not explicitly quantify how far misaligned peaks fall beyond the threshold. It could be addressed by incorporating multi-scale tolerance or continuous scoring mechanisms to capture varying degrees of misalignment. We leave this as future work.

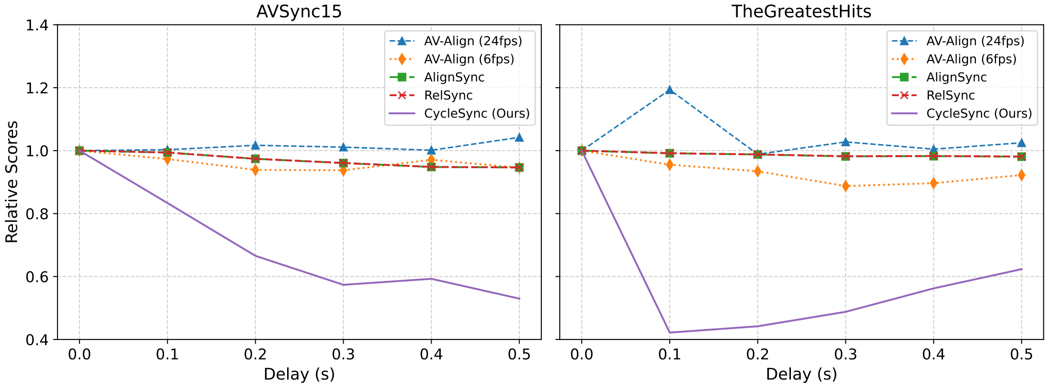


Figure 7: Comparison of relative synchronization scores under increasing audio-video delays on AVSync15 and TheGreatestHits datasets. The vertical axis denotes each metric's score normalized by its value under perfect synchronization (0.0s).

Table 5: Comparison of synchronization metric scores on the AVSync15 dataset. Parentheses show percentage change from perfect synchronization (positive = increase, negative = decrease).

Sample Type	**AV-Align (24fps)** ↑	**AV-Align** ↑	**AlignSync** ↑	**RelSync** ↑	**CycleSync** ↑
Perfect Sync	24.22 (0.0%)	**20.30** (-0.0%)	**25.04** (-0.0%)	**50.00** (0.0%)	**20.97** (0.0%)
Delay 0.1s	24.29 (+0.3%)	19.76 (-2.7%)	24.89 (-0.6%)	49.70 (-0.6%)	17.48 (-16.6%)
Delay 0.2s	24.63 (+1.7%)	19.06 (-6.1%)	24.39 (-2.6%)	48.70 (-2.6%)	13.96 (-33.4%)
Delay 0.3s	24.49 (+1.1%)	19.03 (-6.3%)	24.05 (-4.0%)	48.04 (-3.9%)	12.03 (-42.6%)
Delay 0.4s	24.25 (+0.1%)	19.71 (-2.9%)	23.74 (-5.2%)	47.41 (-5.2%)	12.43 (-40.7%)
Delay 0.5s	**25.24** (+4.2%)	19.20 (-5.4%)	23.70 (-5.4%)	47.33 (-5.3%)	11.11 (-47.0%)

Table 6: Comparison of synchronization metric scores on TheGreatestHits dataset. Parentheses show percentage change from perfect synchronization (positive = increase, negative = decrease).

Sample Type	**AV-Align (24fps)** ↑	**AV-Align** ↑	**AlignSync** ↑	**RelSync** ↑	**CycleSync** ↑
Perfect Sync	14.84 (0.0%)	**27.27** (0.0%)	**25.07** (0.0%)	**50.00** (0.0%)	**16.52** (0.0%)
Delay 0.1s	**17.71** (+19.3%)	26.05 (-4.5%)	24.86 (-0.8%)	49.59 (-0.8%)	6.97 (-57.9%)
Delay 0.2s	14.67 (-1.2%)	25.48 (-6.6%)	24.76 (-1.2%)	49.40 (-1.2%)	7.30 (-55.8%)
Delay 0.3s	15.25 (+2.8%)	24.20 (-11.3%)	24.61 (-1.8%)	49.11 (-1.8%)	8.06 (-51.2%)
Delay 0.4s	14.91 (+0.5%)	24.45 (-10.3%)	24.63 (-1.8%)	49.15 (-1.7%)	9.29 (-43.77%)
Delay 0.5s	15.21 (+2.5%)	25.15 (-7.8%)	24.59 (-1.9%)	49.06 (-1.9%)	10.30 (-37.65%)

Table 7: **Correlation with human ratings.** CycleSync achieves the highest positive correlation with human judgments, while other metrics show weak or negative trends.

Metric	Correlation	95% CI Lower	95% CI Upper
CycleSync	**0.486**	0.053	0.919
AV-Align	0.043	-0.451	0.538
RelSync	-0.623	-1.011	-0.236
AlignSync	-0.625	-1.011	-0.238
DeSync	0.206	-0.279	0.690

Table 8: **Model ranking agreement with human ratings.** CycleSync correctly reflects human preference, ranking Syncphony above Pyramid Flow.

Model	Human Score ↑	CycleSync ↑	AV-Align ↑	RelSync ↑	AlignSync ↑	DeSync ↓
Pyramid Flow (I2V)	2.68	8.15	**24.96**	**55.36**	**27.75**	0.88
Syncphony (Ours)	**4.30**	**22.04**	21.88	50.44	25.19	**0.86**

C.3 Human alignment validation for CycleSync

Human evaluation is essential for establishing a reliable synchronization metric. To assess how well CycleSync aligns with human perception, we conduct a user study with 9 participants, who rate the sync quality of 20 videos, sampled from Pyramid Flow and Syncphony, on a 1–5 scale. We then compute Pearson correlations between the human ratings and the metric scores. We compare CycleSync against existing synchronization metrics: AV-Align (Yariv et al., 2023), AlignSync, RelSync (Zhang et al., 2024), and DeSync (Cheng et al., 2025).

Correlation with human ratings. As shown in Table 7, CycleSync shows the strongest positive correlation with human ratings ($r = 0.486$), whereas prior metrics exhibit weak or even negative correlations. DeSync also shows a positive trend ($r = 0.206$), but remains notably weaker than CycleSync.

Model ranking agreement. We further compared model-level rankings derived from each metric against human ratings (Table 8). CycleSync accurately captures the human-preferred ordering between the two models. DeSync also reflects the correct direction of preference, though with lower discriminative strength.

These results provide strong empirical evidence that CycleSync is both quantitatively sensitive to temporal misalignment and best aligned with human perception, making it a more reliable synchronization metric than existing metrics.

C.4 Results of baselines and Syncphony with existing metrics

We additionally report the performance of baseline models and ours using existing synchronization metrics on the AVSync15 and TheGreatestHits datasets in Table 9 and Table 10, respectively.

On AVSync15, the fine-tuned Pyramid Flow model, which generates audio-independent but plausible motion, achieves the highest AV-Align score. A similar pattern is observed in TheGreatestHits, where the same model also obtains higher AlignSync and RelSync scores than other audio-conditioned models.

These results reveal a limitation of existing metrics, which tend to favor models that produce highly dynamic motion with plausible timing, even if that motion is not aligned with the audio signal.

In contrast, CycleSync consistently assigns the lowest scores to the same model across both datasets. This is because CycleSync penalizes mismatches in temporal structure between the original audio and the reconstructed audio from the generated video. Rather than comparing audio and motion peaks directly, CycleSync compares the temporal structure of the original and reconstructed audio signals, enabling more precise assessment of synchronization quality.

Table 9: Quantitative results on the AVSync15 dataset.

Input	Model	AV-Align ↑	AlignSync ↑	RelSync ↑	CycleSync ↑
T+A	TempoTokens (Yariv et al., 2023)	15.51	22.38	46.91	13.10
I+T	Pyramid Flow (Jin et al., 2024a)	18.85	23.65	47.56	14.25
	Pyramid Flow (fine-tuned)	**20.69**	23.97	47.76	12.34
I+T+A	AVSyncD (Zhang et al., 2024)	19.31	**24.61**	48.99	16.38
	Ours w/o ASG ($w = 0$)	20.01	24.24	48.28	15.31
	Ours w/ ASG ($w = 2$)	19.89	24.45	48.74	**16.48**
	Ours w/ ASG ($w = 4$)	20.00	24.58	**49.04**	16.26
	Groundtruth	20.84	25.10	50.00	22.15

Table 10: Quantitative results on TheGreatestHits dataset.

Input	Model	AV-Align ↑	AlignSync ↑	RelSync ↑	CycleSync ↑
I+T	Pyramid Flow (Jin et al., 2024a)	25.24	25.12	50.46	7.41
	Pyramid Flow (fine-tuned)	26.76	26.67	53.35	9.23
I+T+A	AVSyncD (Zhang et al., 2024)	23.29	26.55	53.07	9.89
	Ours w/o ASG ($w = 0$)	**27.11**	26.08	52.21	11.70
	Ours w/ ASG ($w = 2$)	26.92	26.10	52.27	**16.18**
	Ours w/ ASG ($w = 4$)	26.81	**27.04**	**54.14**	**17.71**
	Groundtruth	26.00	25.07	50.00	15.99

C.5 Limitation of CycleSync.

As a reconstruction-based metric, CycleSync relies on the quality and behavior of the underlying video-to-audio (V2A) model. The reconstructed audio may sometimes reflect dataset-level biases rather than the visual content of the input video itself.

For example, in frog videos, although only a single frog may be visible, many clips in the dataset include ambient sounds from nearby frogs. As a result, the reconstructed audio sometimes contains multiple frog sounds, regardless of the actual motion in the video. Similarly, bowling videos in the dataset often include background music, which can occasionally appear in the reconstructed audio even if it is not visually implied. Such cases may affect CycleSync scores in specific contexts. This issue could potentially be addressed by improving the V2A model or applying post-processing, which we leave for future work.

D Architectural details

D.1 Training layer selection

D.1.1 Video generation backbone

We adopt Pyramid Flow (Jin et al., 2024a) as the video generation backbone due to its efficiency and scalability in generating long, high-resolution videos. Pyramid Flow is an autoregressive diffusion transformer trained with a flow-matching objective, which sequentially synthesizes consecutive video chunks by denoising each chunk for given a previous chunk and a text prompt.

To capture temporal and spatial consistency, it employs 3D Rotary Positional Encoding (RoPE) (Su et al., 2024) within its self-attention layers, enabling the model to encode relative positions across time, height, and width. In addition, the model dynamically adjusts resolution throughout the denoising process—using low-resolution frames at early (noisier) timesteps and high-resolution frames at later (cleaner) stages—thereby reducing computational cost while maintaining visual detail.

This design enables resource-efficient training and generation, supporting high-resolution and long-duration video synthesis even under constrained computational resources.

D.1.2 Training layer selection in video backbone

Pyramid Flow consists of 24 transformer blocks. To identify which layers to fine-tune, we individually skip each of the 24 transformer blocks during inference and observe the effects on image-to-video (I2V) generation (see Figures 10 and 11).

Skipping early blocks (0–7) significantly degrades appearance, often causing artifacts in the background and object structure. In contrast, skipping later blocks (8–23) mostly preserves the appearance of the input image (first frame) in the generated video, primarily affecting the motion. This suggests that early blocks are critical for preserving the input's appearance, whereas later blocks are responsible for refining motion. This separation aligns with the architecture: early blocks use separate attention weights for text and video, while later blocks share them. Based on this functional and structural separation, we fine-tune only the last 16 blocks (8–23) with minimal impact on the pretrained model's visual fidelity.

D.2 Audio RoPE

D.2.1 Implementation details

To encode the temporal structure of audio features explicitly, we apply **Rotary Positional Encoding (RoPE)** to inject relative temporal information directly into the cross-attention mechanism, as illustrated in Figure 3(a).

We first obtain video latents $\{\boldsymbol{z}_l\}_{l=0}^{L_{\text{video}}}$ from a VAE encoder, where each $\boldsymbol{z}_l$ represents a compressed spatiotemporal feature at the l-th position in the video sequence. Simultaneously, we extract audio features $\{\boldsymbol{a}_i\}_{i=0}^{L_{\text{audio}}}$ from a DenseAV encoder, capturing the temporal and semantic structure of the audio input.

To align these modalities, we divide the audio sequence into local segments corresponding to each video latent. For each target video latent $\boldsymbol{z}_l$, we define the corresponding audio segment $\mathbb{A}_l$ as:

$$\mathbb{A}_l = \{\boldsymbol{a}_i \mid i \in [\alpha(l-\Delta), \alpha(l+\Delta)]\}, \tag{7}$$

where α is a scaling factor mapping video indices to audio indices (accounting for the different sequence lengths), and Δ determines the width of the temporal window(we set Δ=1).

Then, we apply Audio RoPE to the audio segments. The procedure is as follows:

Step 1. Assign positional indices.

- Each video latent $\boldsymbol{z}_l$ is assigned 3D coordinates (l, h, w) representing its temporal and spatial location within the video sequence.
- For each audio segment $\mathbb{A}_l$, the constituent audio features are assigned linearly interpolated temporal indices within the range $[l-(\Delta+0.5), l+(\Delta+0.5)]$, such that the center of the segment aligns exactly with l.

Step 2. Project into query and key spaces.

$$\boldsymbol{q}_l = \boldsymbol{W}_Q \boldsymbol{z}_l, \quad \boldsymbol{K}_l = \{\boldsymbol{W}_K \boldsymbol{a}_i \mid \boldsymbol{a}_i \in \mathbb{A}_l\}, \tag{8}$$

where $\boldsymbol{W}_Q$ and $\boldsymbol{W}_K$ are learnable linear projection matrices.

Step 3. Apply RoPE rotations.

$$\boldsymbol{q}_l^{\text{rope}} = \text{RoPE}(\boldsymbol{q}_l, (l, h_l, w_l)), \quad \boldsymbol{K}_l^{\text{rope}} = \{\text{RoPE}(\boldsymbol{W}_K \boldsymbol{a}_i, (t_i, 0, 0)) \mid \boldsymbol{a}_i \in \mathbb{A}_l\}, \tag{9}$$

where t_i denotes the interpolated temporal index assigned to each $\boldsymbol{a}_i$.

Step 4. Compute cross-attention between video latent $\boldsymbol{z}_l$ and audio segment $\mathbb{A}_l$:

$$\text{Attention}(\boldsymbol{z}_l, \mathbb{A}_l) = \text{Softmax}\left(\frac{\boldsymbol{q}_l^{\text{rope}}(\boldsymbol{K}_l^{\text{rope}})^\top}{\sqrt{d}}\right)\boldsymbol{z}_l, \tag{10}$$

Table 11: Ablation of Audio RoPE.

Model Variant	**CycleSync** ↑
w/o Audio RoPE	$14.41_{\pm 1.40}$
w/ Audio RoPE	$15.31_{\pm 1.49}$

where $\boldsymbol{z}_l = \{\boldsymbol{W}_V \boldsymbol{a}_i \mid \boldsymbol{a}_i \in \mathbb{A}_l\}$ is the set of value projections of the audio features, and d is the dimension of the projected space.

By explicitly injecting temporally aligned positional cues into both video and audio features, our model captures the sequential structure of audio signals more effectively, leading to improved synchronization between generated video motion and corresponding audio events.

D.3 Ablation study

We conduct ablation experiments to examine the effect of Audio RoPE. The results in Table 11 indicate that using Audio RoPE leads to higher synchronization quality compared to the model without it. Without applying RoPE to audio features, the model frequently exhibits misalignments, with motions often preceding or lagging behind the corresponding audio cues. In contrast, applying RoPE to the audio features results in tighter temporal alignment between motion and sound events, enabling the model to better capture the sequential structure of the audio input. Additional ablation examples are included in the supplementary materials (Appendix G).

E User study

To assess the perceptual quality of our generated videos, we conducted a user study comparing our method with the state-of-the-art Audio-to-Video model AVSyncD (Zhang et al., 2024). We select AVSyncD as the sole baseline in the user study, as other baselines generate noticeably unsynchronized motion. The evaluation focused on three aspects: synchronization with audio, image quality, and frame consistency.

The study was conducted using all 150 test videos from the AVSync15 dataset. These were divided into five subsets of 30 videos each, with each subset assigned to two participants (10 participants total). For every video, participants were shown two versions—one generated by our model and one by AVSyncD based on the same audio input and initial image. Participants were asked to answer the three questions for each video pair:

- **Synchronization:** Which video is better synchronized with the audio in terms of motion timing?
- **Image Quality:** Which video has better image quality in terms of realism and clarity?
- **Frame Consistency:** Which video is more visually consistent across frames, without flickering or sudden jumps?

As illustrated in Figure 9, participants evaluated video pairs using a web interface showing both videos and three corresponding questions.

The results, summarized in Figure 8, show that our model was consistently preferred: 74% for synchronization, 90% for image quality, and 94% for frame consistency.

These results demonstrate that our model is consistently favored by human evaluators across all three aspects. This further validates the effectiveness of our synchronization mechanisms and the visual fidelity of our methods.

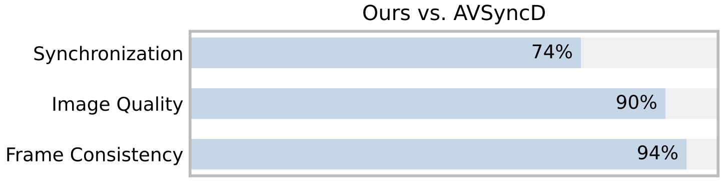


Figure 8: Human preference rates (%) for our method over ASyncD across three evaluation criteria.

F Additional video samples

F.1 Samples from Syncphony

Please see the top sections of the attached HTML file (`"Html_Suppl/index.html"`) for generated videos from Syncphony.

G Ablation samples

Please refer to Section **Ablations** in the attached HTML file (`"Html_Suppl/index.html"`), which includes ablation results for Motion-aware Loss, Audio Sync Guidance, and Audio RoPE.

H Comparison samples

Please refer to Section **Comparison** in the attached HTML file (`"Html_Suppl/index.html"`), which compares videos generated by our model, AVSyncD Zhang et al. (2024), and Pyramid Flow (fine-tuned), a variant of our model without audio cross-attention layers.

I Implementation and experimental details

I.1 Why Image-to-Video backbone?

We also applied our method to a Text-to-Video (T2V) model, AnimateDiff (Guo et al., 2023), and trained it on the AVSync15 dataset, which contains limited 1,350 training samples. We found the model generates motion aligned with audio, but it shows overfitting, with limited diversity in appearance. This is because T2V models have to generate both appearance and motion without a reference image. With a small dataset, it becomes difficult to produce diverse appearances, and even harder to learn various audio-driven motion patterns.

In contrast, Image-to-Video (I2V) models, such as Pyramid Flow (Jin et al., 2024a), are conditioned on an initial image and focus on predicting motion rather than full appearance. This simplifies the learning process and reduces the risk of overfitting. For these reasons, we adopt the I2V model as our video generation backbone.

I.2 Training and inference settings

We train our model using 4 NVIDIA RTX 3090 GPUs (24GB each) with a total batch size of 32. Training takes 34 hours to reach 33,000 steps. For all experiments, we use 30 denoising steps. We follow Pyramid Flow in setting the classifier-free guidance (CFG) strength to 7.0 for the first latent and 4.0 for the rest. For Audio Sync Guidance, we use $w = 2$, where $w = 0$ disables the guidance.

Inference time for a 5-second video (with pre-encoded audio and text features) is as follows:

- Audio Guidance: 2 min 53 sec
- w/o Audio Guidance: 2 min 01 sec
- w/o Audio Layers: 1 min 43 sec

At least 16 GB of GPU memory is required to generate 5-second videos.

I.3 Training and evaluation datasets

We use two datasets for training and evaluation:

- **AVSync15 (Zhang et al., 2024):** 1,350 videos for training and 150 for testing. For evaluation, we linearly extract 3 clips per video, resulting in 450 evaluation clips.
- **TheGreatestHits (Owens et al., 2016):** 733 videos for training and 244 for testing, resulting in 732 evaluation clips.

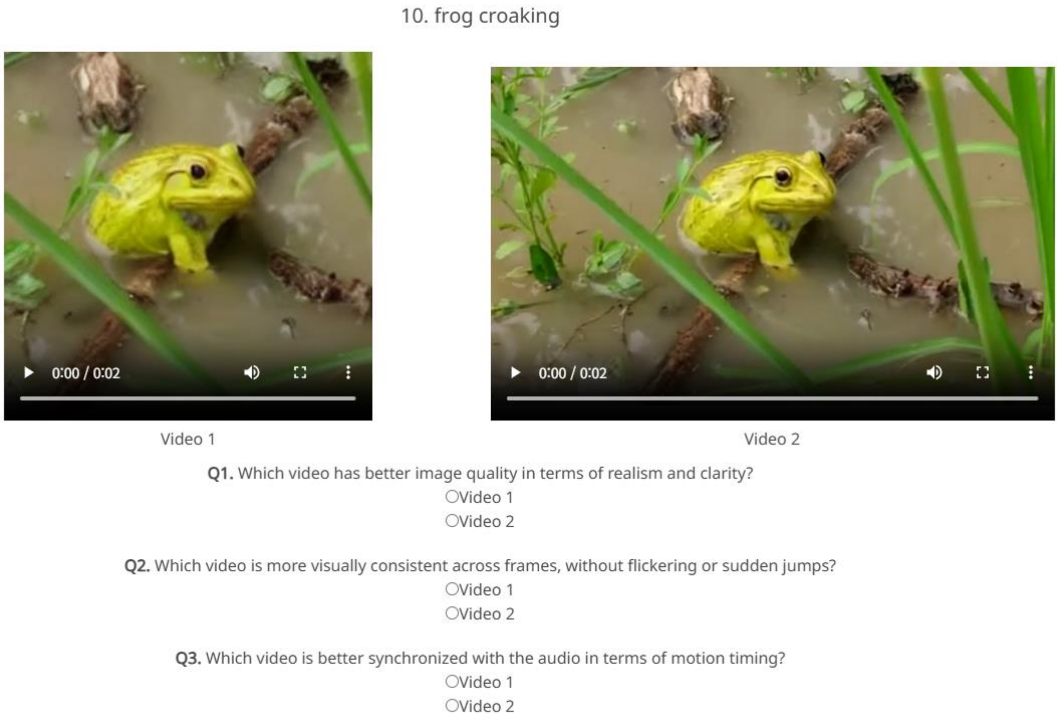
During training, we randomly sample clips from different temporal regions of each video to improve generalization to various audio-motion alignments.

J Applicability of Syncphony techniques to other modalities

While Syncphony focuses on audio-to-video generation, we believe the proposed techniques are applicable to other modalities.

Motion-aware Loss, by amplifying learning signals in high-motion regions, encourages the model to focus on dynamic cues that reflect physically grounded movements. This can benefit tasks like audio-to-3D animation, text-to-video, and text-to-3D, where generating realistic motion is essential.

In contrast, Audio Sync Guidance are designed to improve synchronization between audio and motion. This technique is applicable to tasks such as audio-to-3D animation, provided that the model adopts an attention-based architecture with functionally well-separated layers, which enables clean injection of audio signals into the network.



10. frog croaking

Video 1 Video 2

Q1. Which video has better image quality in terms of realism and clarity?
○Video 1
○Video 2

Q2. Which video is more visually consistent across frames, without flickering or sudden jumps?
○Video 1
○Video 2

Q3. Which video is better synchronized with the audio in terms of motion timing?
○Video 1
○Video 2

Figure 9: Screenshot of the user study interface of each video pair with questions.

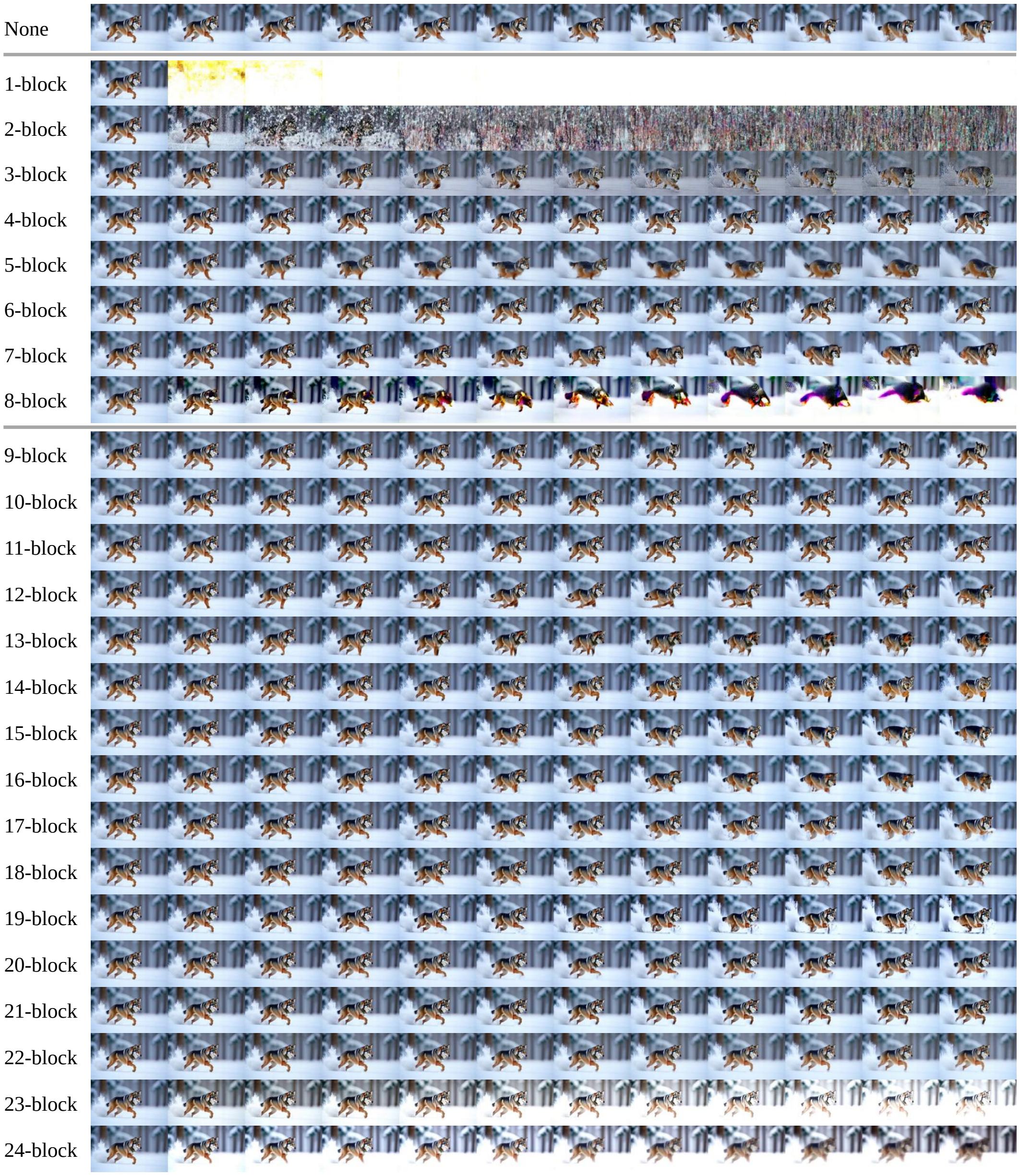


Figure 10: Frame results of skipping each transformer block individually.

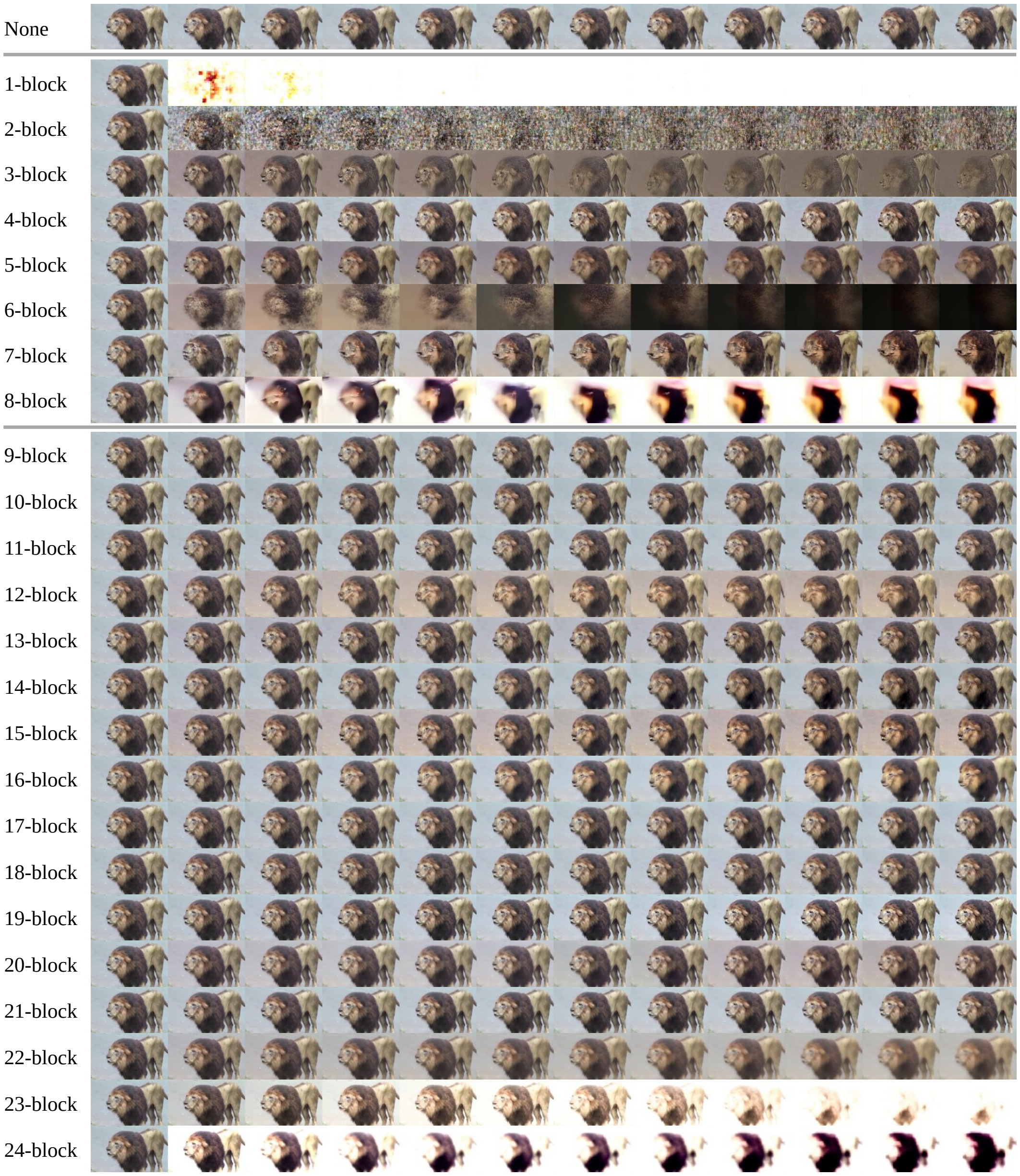


Figure 11: Frame results of skipping each transformer block individually.