Appendix

This appendix provides comprehensive supplementary materials to support our study. Below are brief descriptions of all the sections covered in the appendix. Please visit our project page for more visualization.

- Appendix A: **Data Examples with Annotations**
  - Presents data examples from our CookGen dataset.
  - Showcases annotated "actions" and "captions" that provide detailed multimodal information of cooking processes.
- Appendix B: **Additional Data Statistics**
  - Offers distributions of video lengths, clip lengths, and textual annotations.
  - Demonstrates the dataset's richness and suitability for long narrative video generation.
- Appendix C: **Data Evaluation Details**
  - Details our data evaluation process.
  - Includes inverse video generation results, the prompts used for video captioning, GPT-4o evaluations, and human evaluation results.
- Appendix D: **Implementation Details**
  - Outlines the implementation details of our models.
  - Provides key hyperparameters and training & inference configurations.
- Appendix E: **Action-Caption Matching Pseudo Code**
  - Includes the pseudo code for our action-caption matching algorithm.
  - Essential for aligning video clips with their corresponding annotations.
- Appendix F: **CLIP beats VAE for interleaved generation**
  - Introduces three autoencoders (EMU-2, SEED-X, SDXL-VAE) and compares reconstruction vs. generation.
  - Demonstrates that CLIP-diffusion embeddings (EMU-2, SEED-X) outperform SDXL-VAE in language-driven visual generation due to better vision-language alignment.
- Appendix G: **Generated Video Examples**
  - Showcases generated video examples.
  - Illustrates the effectiveness of our pipeline in producing long narrative videos for cooking recipes like "Fried Chicken" and "Shish Kabob."
- Appendix H: **Limitations**
  - Discusses the limitations of our approach.
  - Includes issues with noisy "actions" from automatic speech recognition and potential failure cases in video generation.

A. Data Examples with Annotations

Figures 7 and 8 shows two data examples from our CookGen dataset, annotated with high-quality descriptions that provide detailed multi-modal information of cooking processes. The examples clearly show structured annotations of key actions and corresponding visual descriptions, making the dataset ideal for generating long narrative videos.

(a) **Action:** Elise works with chicken thighs, advises to trim excess skin and fat
Caption: A person is preparing chicken on a wooden cutting board. He uses a pair of black-handled scissors to cut through the chicken pieces, which are spread out on a clear cutting mat.

(b) **Action:** She offers alternatives with chicken breast bone-in skin-on or chicken drumsticks
Caption: A person with light skin is preparing raw chicken pieces on a wooden surface. He places several pieces of chicken on a white cutting board.

(c) **Action:** Elise heats up a large skillet with two teaspoons of olive oil and a teaspoon of butter
Caption: A person is seen in a kitchen setting, holding a wooden spoon. He places a small piece of butter into a black frying pan on a gas stove.

(d) **Action:** Turn over the chicken pieces and cook for another 4 minutes Remove the chicken from the pan but keep the browned pieces in the pan
Caption: Golden-brown chicken pieces are sizzling in a black frying pan on a gas stove.

(e) **Action:** Use the remaining oil in the pan to brown the orzo Cook the orzo like a traditional rice pilaf, using the same method as before
Caption: A person is cooking rice in a black frying pan on a gas stove. He pours the rice from a glass bowl into the pan, then uses a wooden spatula to spread and stir the rice.

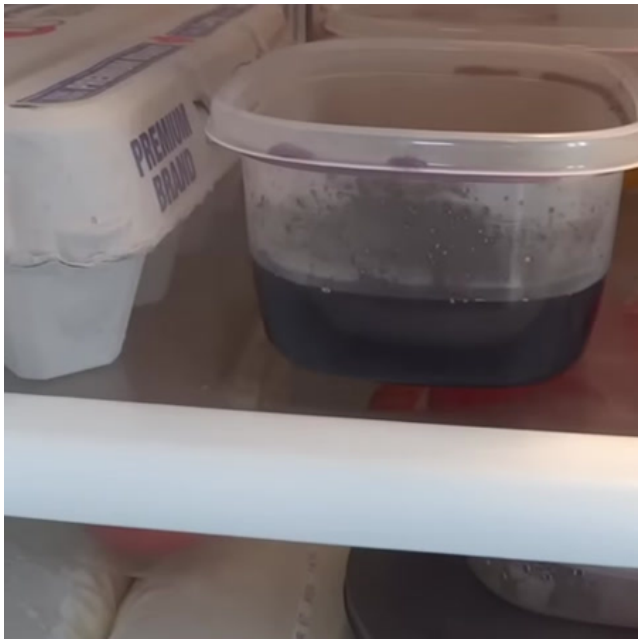
(f) **Action:** Add 2 cups of gordo's to a hot pan
Caption: A person wearing a blue shirt is cooking rice in a black frying pan on a stovetop. Using a wooden spatula, he stirs the rice, ensuring it is evenly cooked.

(g) **Action:** Combine the mixture with the orzo and cook for a few minutes until the sauce thickens
Caption: A woman is cooking on a stovetop, adding pieces of breaded chicken to a pan filled with chopped onions and rice.

(h) **Action:** Stock is cooked until orzo has fully absorbed liquid and chicken is cooked through, about 10-12 minutes Dish is removed from heat and left to sit for five minutes Dish is sprinkled with unspecified seasoning
Caption: A delicious dish of roasted chicken pieces is presented in a black skillet, surrounded by a colorful mix of diced vegetables and grains.

Figure 7. **Data examples with annotated "actions" and "captions"**. A video of cooking recipe of *"One Pot Chicken and Orzo"*.

(a) **Action:** Hi everyone, this one's called rainbow broken glass jello
Caption: A colorful, multi-layered dessert is displayed on a black surface. The dessert features vibrant red, green, blue, and purple segments, arranged in a geometric pattern.

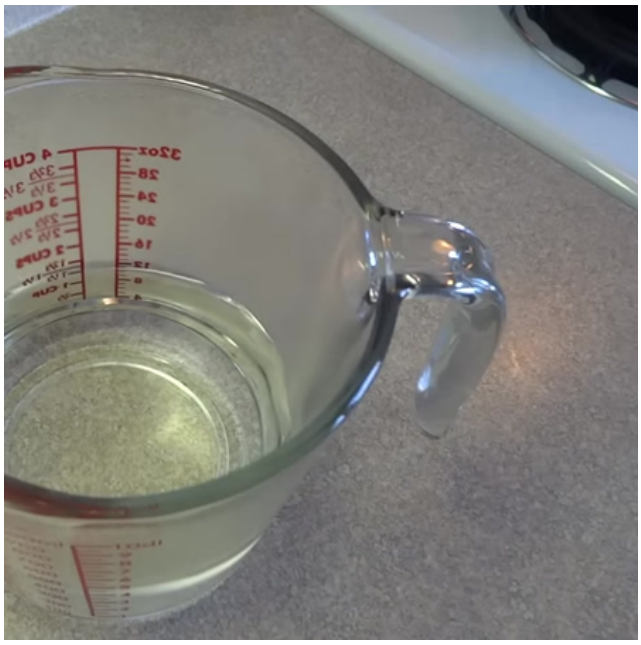


(b) **Action:** Now normally when you make jello you use two cups of boiling water, but in this case we're only using one cup because we want the jello to be extra firm
Caption: The video shows the interior of a refrigerator, focusing on the door shelf. The containers are filled with dark, blue, orange, and red liquids.

(c) **Action:** I find the easiest way to do this is to put the small container into a larger container of hot water
Caption: A person with light skin is holding a clear plastic container filled with a yellow liquid, inspecting its contents.

(d) **Action:** Loosen the edges of the Jello piece Slide the Jello piece out and cut it into cubes Cut the Jello cubes into half-inch pieces
Caption: A person is slicing a block of yellow gelatin on a wooden cutting board, cutting it into uniform strips.

(e) **Action:** Spread out the different colored Jello pieces in a 9 by 13 inch baking dish
Caption: A person is arranging colorful gelatin cubes in a glass baking dish, adjusting the placement of green, orange, purple, and black cubes.



(f) **Action:** Make a separate gelatin mixture by boiling two cups of water and adding two envelopes of gelatin
Caption: A clear glass measuring cup is placed on a countertop, containing water. A person pours a white powder into it.

(g) **Action:** Stir the sweetened condensed milk into the gelatin and water mixture
Caption: A person is vigorously whisking a creamy mixture in a clear glass measuring cup.

(h) **Action:** Let it set for several hours, then cut it into squares and serve
Caption: A glass baking dish is filled with a creamy white liquid, topped with colorful, triangular-shaped glass pieces.

Figure 8. **Data examples with annotated "actions" and "captions"**. A video of preparing *"Rainbow Broken Glass Jello"*.

B. Additional Data Statistics

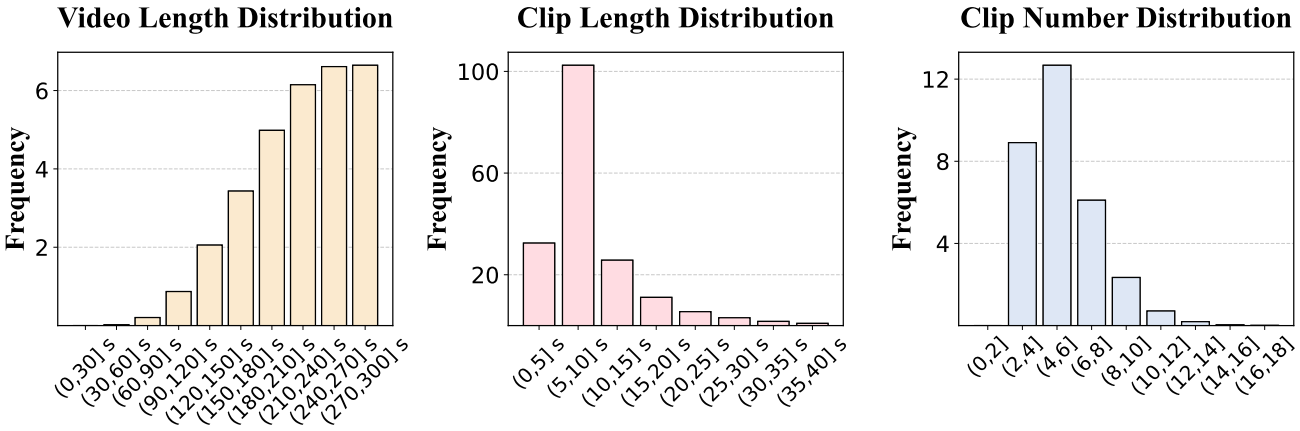


Figure 9. **Statistics on the video data.** We do statistics on the video lengths of the collected whole videos, the clip lengths of the scene-cut video clips, and the number of clips selected for each video.

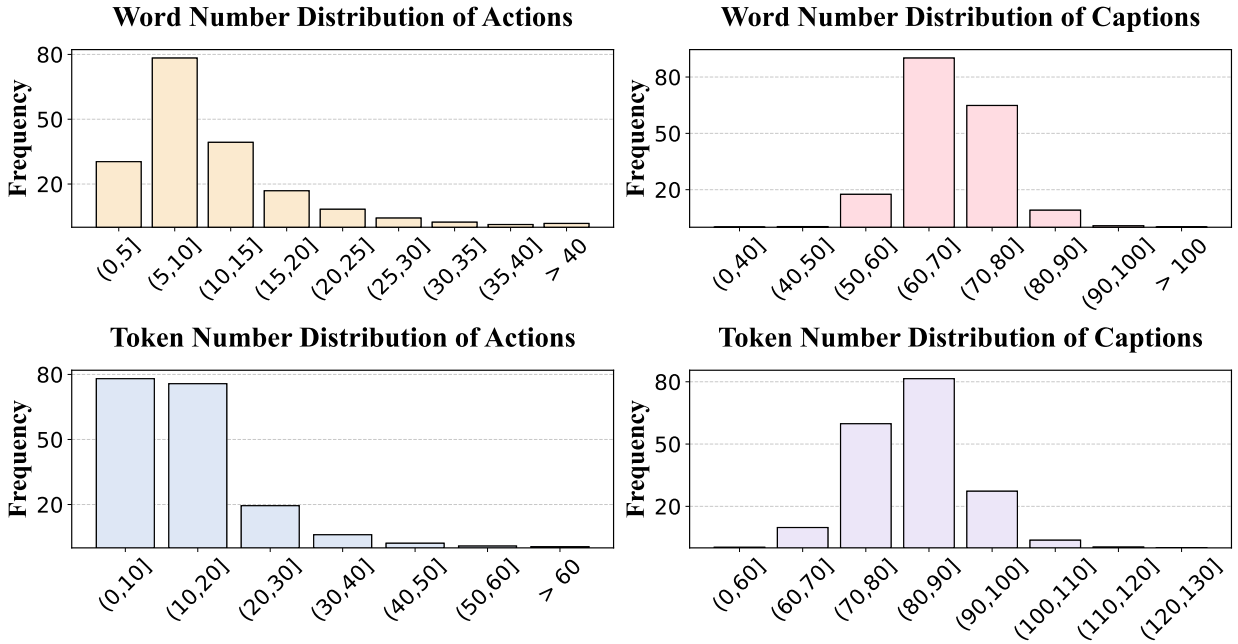


Figure 10. **Statistics on the text annotations.** We do statistics on the number of words and tokens (Llama [48] tokenized) of annotated "actions" and "captions," respectively.

The statistics in Figure 9 and Figure 10 demonstrate the high quality and suitability of our dataset for long narrative video generation. Figure 9 reveals that the video lengths range broadly, with most videos falling between **30** and **150** seconds. Clip lengths are primarily distributed between **5** and **30** seconds, ensuring manageable segments for modeling. Additionally, the majority of videos contain **4** to **12** clips, providing a balanced structure for narrative flow. Figure 10 shows that the word counts for "actions" predominantly range from **10** to **25**, while "captions" range from **40** to **70**. Token distributions further highlight their richness, with "actions" having **20** to **60** tokens and "captions" extending up to **120** tokens. These detailed annotations ensure well-aligned and contextually rich representations of the video content.

Overall, the dataset's design ensures coherent sequences of actions and captions with reasonable clip and video lengths, making it well-suited for generating high-quality, long-form narrative videos.

C. Annotation Quality Reverification Details

High-quality captions are essential for narrative visual generation. To verify the quality of our annotations, we build an evaluation pipline of inverse generation (§C.1) and visual understanding through VLM experts (§C.2).

C.1. Inverse Video Generation

This evaluation is motivated by the understanding that high quality captions, when combined with ground truth keyframes, more effectively reconstruct the original videos. We evaluate the dataset's ability to reconstruct original videos using the annotated captions, with and without conditioning with ground truth keyframes. For this evaluation, we assess the validation set (∼5,000 video clips). We measure reconstruction quality using FVD [49]. The results, shown in Table 12, indicate that our captions capture sufficient semantic information, enabling effective representation of the original videos. When generating with ground-truth keyframes, the video quality is very high and closely aligned with the original videos, as shown by the low FVD score (116.3). Without keyframes, the captions alone still provide reasonable alignment. Examples of reconstructed videos are included in the supplementary materials.

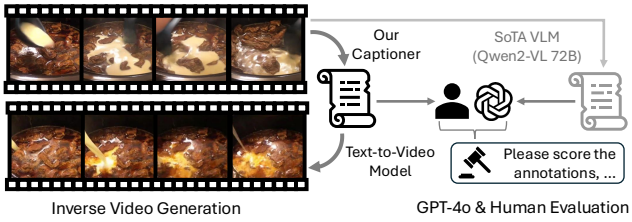


Figure 11. **Annotation quality evaluation pipeline**. We verify our annotation quality through a pipeline of two major aspects: 1) inverse video generation 2) GPT-4o and human evaluation.

Validation Set	w/. GT keyframe	W/o. GT keyframe
# Clips	FVD	FVD
5504	**116.3**	561.1

Figure 12. **High-quality captions enable inverse video generation.** We utilize the annotated captions and actions to inversely generate video clips using a pretrained Text-to-Video diffusion model. Higher reconstruction fidelity (i.e., similarity to the original videos) indicates superior captions and actions. Our inversely generated videos achieve very low FVD scores compared to the original videos, highlighting the high quality of our annotations.

	GPT-4o Evaluation		Human Evaluation	
Score (0-100)	Qwen2-VL-72B	**Ours**	Qwen2-VL-72B	**Ours**
	98.0	95.2	79.3	**82.0**

Table 8. **Caption Quality Evaluation**. We compare the caption quality between our captioner and the Qwen2-VL-72B model by both GPT-4o and human annotators. Our model achieves competitive results despite a much smaller model size.

C.2. Semantic Consistency across VLM Experts

GPT-4o & human evaluation. We evaluate the quality of our captions using both GPT-4o and six human annotators, in which we ask humans and GPT-4o to rate our dataset provided captions according to two criteria: the coverage of video elements and the absence of hallucinations in the caption. Following [59], hallucination refers to the model generating content absent or unsupported by the video, such as incorrect details about objects, actions, or counts.

To demonstrate the quality, we compare our captions with those generated by a state-of-the-art open-source VLM (Qwen2-VL-72B). As shown in Table 8, our dataset's captions receive a decent score of 95.2 out of 100, showing slightly better alignment with rigorous human evaluation than the Qwen2-VL-72B model. Results from both human evaluators and GPT-4 assessments indicate that the dataset contains high-quality captions.

C.3. Inverse Video Generation Results

As discussed in Appendix C.1, high-quality captions, especially with ground truth keyframes, enable effective video reconstruction. We compare ground truth video frames with inversely generated frames using the GT first keyframe and annotated captions, as shown in Figures 13 to 15. The reconstruction aligns well with the narrative, accurately capturing actions, though patterns and interactions differ slightly from the original video. This shows that while the captions convey crucial information for reconstruction, they lack finer visual details, a limitation for current vision-language models and human annotators.

For example, in Figure 13, the ground truth shows a hand pouring creamy liquid into a slow cooker and stirring, while the generated frames replicate the actions with slight differences in texture and liquid mixing. Similarly, in Figure 15, the ground truth shows a face drawn with cream on orange liquid, but the generated frames vary in precision and interaction details.

These examples highlight the captions' strength in preserving narrative flow while exposing gaps in capturing fine-grained visual detail.

Figure 13. **Left:** Ground truth, **Right:** Inverse generation with GT keyframe. **Caption:** Chunks of meat are simmering in a dark-colored slow cooker. A hand pours a creamy liquid into the pot, causing the liquid to mix with the meat and broth. The mixture bubbles and thickens as the liquid is added. The person stirs the contents with a black spoon, ensuring the ingredients are well combined. The slow cooker continues to cook the meat, which appears tender and well-cooked.

Figure 14. **Left:** Ground truth, **Right:** Inverse generation with GT keyframe. **Caption:** A person wearing a black sleeve is whisking a creamy mixture in a clear glass bowl. The mixture appears to be a batter or dough, gradually becoming smoother and more uniform. The person's left hand holds the bowl steady on a light-colored countertop. The whisking motion is consistent and thorough, ensuring the mixture is well-blended. The background is plain, focusing attention on the mixing process.

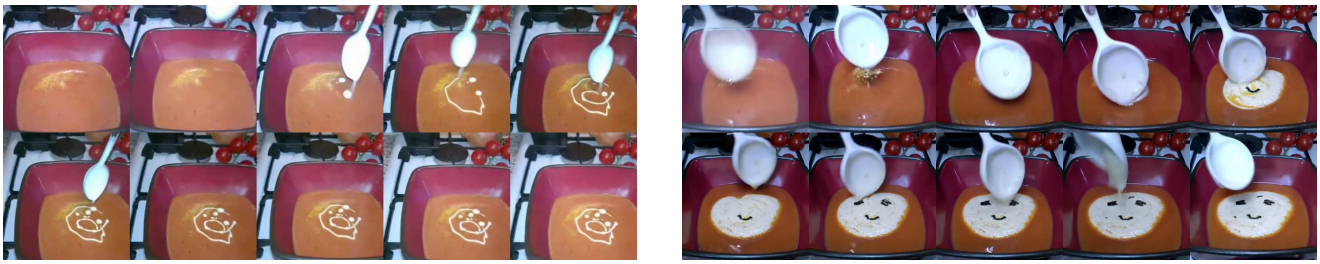


Figure 15. **Left:** Ground truth, **Right:** Inverse generation with GT keyframe. **Caption:** A red bowl filled with a thick, orange liquid is placed on a stovetop. A woman's hand, holding a white spoon, appears and begins to draw on the surface of the liquid. She creates a face with white cream, adding details to the eyes and mouth. The background shows a granite countertop with a bunch of red tomatoes and a white pot. The woman continues to add finishing touches to the face.

C.4. Prompt for Video Captioning

Below is the prompt we designed to effectively caption video clips and also for benchmarking VLMs, ensuring detailed and accurate descriptions while avoiding redundancy:

```
You are an expert in describing videos and catching the sequential motions from video frames.

For the given ten video frames, you need to generate a detailed good description within five
sentences / 80 words. Please do not include the word 'frame' or 'frames' in your answer. If
the gender of a person is clear, use 'he' or 'she' instead of they. Do not describe a single
motion/action twice like 'xxx continues doing yyy'. Don't assume actions like discussion or
```

```
having a conversation unless it is very clear in the frames. Describe the video given the
frame sequence. Describe both the appearance of people (gender, clothes, etc), objects,
background in the video, and the actions they take.
```

Listing 1. **Video Captioning Prompt**

C.5. GPT-4o Evaluation on Captions

Below is the evaluation prompt designed to objectively assess the quality of video captions generated by a Large Multimodal Model (LMM), focusing on coverage and hallucination.

```
Your role is to serve as an impartial and objective evaluator of a video caption provided by
a Large Multimodal Model (LMM). Based on the input frames of a video, assess primarily on
two criteria: the coverage of video elements in the caption and the absence of hallucinations
in the response. In this context, 'hallucination' refers to the model generating content not
present or implied in the video, mainly focused on incorrect details about objects, actions,
counts, temporal order, or other aspects not evidenced in the video frames.

To evaluate the LMM's response:
    Start with a brief explanation of your evaluation process.
    Then, assign a rating from the following scale:
        Rating 6: Very informative with good coverage, no hallucination
        Rating 5: Very informative, no hallucination
        Rating 4: Somewhat informative with some missing details, no hallucination
        Rating 3: Not informative, no hallucination
        Rating 2: Very informative, with hallucination
        Rating 1: Somewhat informative, with hallucination
        Rating 0: Not informative, with hallucination
Do not provide any other output symbols, text, or explanation for the score.
```

Listing 2. **GPT-4o Evaluation Prompt**

C.6. Human Evaluation on Captions

Matching Tier	Action (Important Info.)	Object (Important Info.)	Score
Very Match	Good Coverage, No Hallucination	Good Coverage, No Hallucination	**100**
Good Match	Good Coverage, Limited Hallucination	Good Coverage, Limited Hallucination	**85**
Somehow Match	Fair Coverage, Some Hallucination	Fair Coverage, Some Hallucination	**70**
Not Match	Little Coverage or High Hallucination	Little Coverage or High Hallucination	**0**

Table 9. **Human Evaluation Matching Rules.** Captions are rated based on coverage and hallucination levels, using four matching tiers.

We assess the quality of our captions through evaluations by six human annotators, who rate the captions based on two key criteria: the coverage of video elements (such as objects and actions) and the absence of hallucinations, defined as generating content unsupported or absent in the video [59]. As shown in Table 8, our captions achieve a high human evaluation score of **82.0**, surpassing the state-of-the-art open-source VLM (Qwen2-VL-72B) score of **79.3**. These results demonstrate the superior quality of our captions, which are more aligned with human preferences and exhibit better narrative accuracy.

For evaluation, annotators rate the captions across four tiers—Very Match, Good Match, Somehow Match, and Not Match—based on consistency with video content. The scoring rubric, detailed in Table 9, considers both coverage and hallucination levels. Our captioner consistently achieves high scores in the top tiers, validating its reliability and quality for narrative video generation.

D. Implementation Details

We provide the training and inference hyperparameters for the interleaved auto-regressive model and the visual-conditioned video generation model in Table 10 and Table 12, respectively. The interleaved auto-regressive model is trained on images with a resolution of 448×448, using a batch size of 512 and bfloat16 precision. It employs AdamW as the optimizer, with a peak learning rate of 2×10^{-4} and a cosine decay schedule, training for 2,500 steps. Training context pairs vary between 2 and 8, while inference always uses 8 pairs for consistency. The visual-conditioned video generation model processes video data at a resolution of $448 \times 448 \times T$, with a batch size of 64 and bfloat16 precision. It uses AdamW with a peak learning rate of 1×10^{-5} and a constant decay schedule, training for 20,000 steps to handle temporal conditioning effectively.

Configuration	Setting
Image resolution	448×448
Optimizer	AdamW
Optimizer hyperparameters	$\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-6}$
Peak learning rate	2×10^{-4}
Learning rate schedule	Linear warm-up, cosine decay
Gradient clip	1.0
Total training steps	$2,500$
Warm-up steps	200
Batch size	512
Numerical precision	bfloat16
Training context pairs	[2, 8]
Inference context pairs	8

Table 10. **Implementation details of the interleaved auto-regressive model.**

Configuration	Setting
Image Resolutions	$448 \times 448 \times 4$
Optimizer	AdamW
Optimizer hyperparameters	$\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$
Peak learning rate	5×10^{-4}
Learning rate schedule	Linear warm-up, constant
Gradient clip	1.0
Total training steps	$5,000$
Warm-up steps	500
Batch size	16
Optional Masking Probability	0.25
Numerical precision	bfloat16

Table 11. **Implementation details of the rolling context conditioned render.**

Configuration	Setting
Image/Video resolution	$448 \times 448 \times T$
Optimizer	AdamW
Optimizer hyperparameters	$\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$
Peak learning rate	1×10^{-5}
Learning rate schedule	Linear warm-up, constant
Gradient clip	1.0
Total training steps	$20,000$
Warm-up steps	$1,000$
Batch size	64
Numerical precision	bfloat16

Table 12. **Implementation details of the visual-conditioned video generation model.**

E. Action-Caption Matching Pseudo Code

The action-caption matching algorithm detailed in Algorithm 1 aligns video clips with actions based on temporal overlap and specific rules. It uses the Intersection over Union (IoU) to measure the overlap between the time intervals of video clips and actions. A match is identified if either the IoU exceeds 0.5 or all of the following conditions are met: the start time difference (`start_diff`) is less than 5 seconds, the clip's end time exceeds the action's end time, and the IoU is greater than 0.2.

The algorithm processes each video iteratively. For each video, it retrieves all associated actions and their time intervals. Then, for each clip in the video, it calculates the IoU with every action and evaluates the matching conditions. Valid matches, along with their metadata (clip info and descriptions), are stored in a list $\mathcal{M}$. This systematic approach ensures that the matched actions and captions are temporally consistent, providing high-quality annotations for keyframe visual states.

Algorithm 1 Pseudo code for action-caption matching.

function IoU($[s_1, e_1], [s_2, e_2]$)
    intersection $\leftarrow \max(0, \min(e_1, e_2) - \max(s_1, s_2))$
    union $\leftarrow \max(e_1, e_2) - \min(s_1, s_2)$
    if union > 0 **then**
        return $\frac{\text{intersection}}{\text{union}}$
    else
        return 0
    end if
end function
Initialize an empty list $\mathcal{M} \leftarrow []$
for all $v \in \mathcal{V}$ **do**
    $v_{\text{id}} \leftarrow v$.id
    if $v_{\text{id}} \in \mathcal{A}$ **then**
        $\mathcal{A}_v \leftarrow \mathcal{A}[v_{\text{id}}]$
        action_times $\leftarrow \mathcal{A}_v$.times
        action_descriptions $\leftarrow \mathcal{A}_v$.descriptions
        for all $c \in v$.clips **do**
            $[s_c, e_c] \leftarrow c$.start_end
            for all $a \in$ action_times **do**
                $[s_a, e_a] \leftarrow a$
                start_diff $\leftarrow |s_c - s_a|$
                iou $\leftarrow$ IoU($[s_c, e_c], [s_a, e_a]$)
                if (start_diff $< 5 \wedge e_c > e_a \wedge$ iou > 0.2) $\vee$ iou > 0.5 **then**
                    Create match: $\mathcal{M} \leftarrow \mathcal{M} \cup m$
                end if
            end for
        end for
    end if
end for
return $\mathcal{M}$

F. CLIP beats VAE for interleaved generation.

We experiment with three different auto-encoded visual latent spaces for regression: the EMU-2 [44] CLIP-Diffusion autoencoder, the SEED-X CLIP-Diffusion autoencoder, and the KL Variational autoencoder (VAE) used by SDXL. Both SEED-X and EMU-2 use a CLIP vision encoder and a finetuned SDXL diffusion model as the decoder for encoding visual latent. From appendix Figure 16, we observe that SDXL-VAE achieves the best reconstruction quality. However, in terms of visual generation quality, as shown in Table 13, the CLIP-Diffusion based autoencoders significantly outperform VAE (*i.e.*, **+12.2** CLIP-T score and **256.6** better FID). This suggests that CLIP embeddings are more suitable for interleaved visual generation compared to VAE's latent space. This is reasonable, as SDXL-VAE is not aligned with language and lacks semantics.

Method	Autoencoder Style	VL Aligned.	Recon. Ability	CLIP-T	FID
SDXL-VAE	Variational U-Net	✗	High	13.2	286.6
EMU-2	CLIP-Diffusion	✓	Medium	**25.4**	76.7
SEED-X	CLIP-Diffusion	✓	Low	25.1	**30.1**

Table 13. **Visual latent spaces for visual regression.** The VAE latent space is challenging for auto-regressive models to regress in a single step due to its limited correlation with language. In contrast, the language-aligned latent spaces (EMU-2 and SEED-X) allow for easier and effective regression in an interleaved manner.

Figure 16. **Auto-encoded results with different latent spaces.** While SEED-X and EMU-2 both use a CLIP vision encoder and a diffusion model (*i.e.* finetuned SDXL) as decoder for autoencoding visual latents, SEED-X is semantic-biased and EMU-2 keeps much more visual details. SDXL-VAE shows the best image reconstruction ability, however, the latent space is not aligned with language (*i.e.* without pretraining on image-text pairs like CLIP).

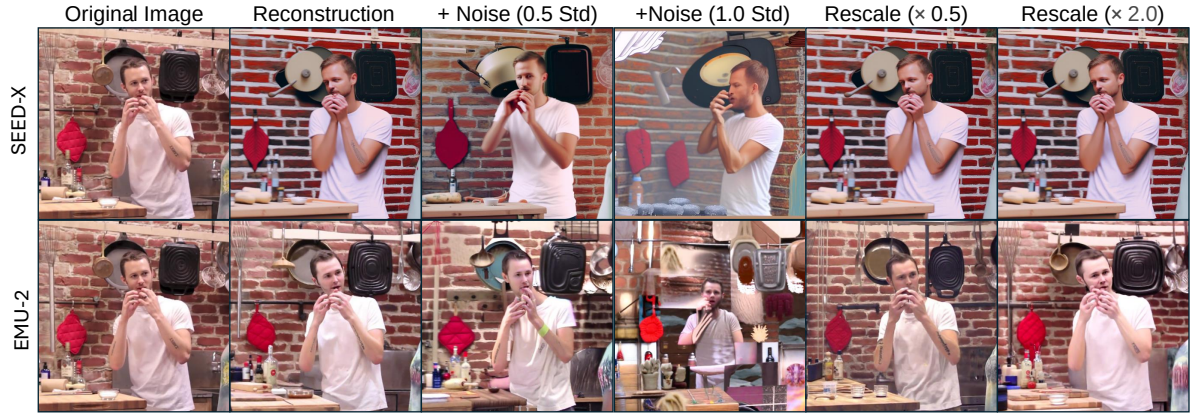


Figure 17. **Both Scale and Direction Matters.** We experiment with pseudo regression errors by altering latent direction and scale using Gaussian noise and scaling factors. Reconstruction results confirm that preserving both scale and direction is important for latent regression.

G. Generated Video Examples

Figures 18 and 19 present two examples of long narrative video generation for cooking "Fried Chicken" and "Shish Kabob," illustrated step-by-step. The generation process begins with our interleaved auto-regressive director, which generates keyframe visual embeddings and their corresponding captions. These embeddings and captions are then used as conditions for the video generation model, which produces high-quality video clips that effectively narrate the cooking process and emphasize the crucial "action" information. The resulting video clips demonstrate excellent performance in capturing the step-by-step cooking instructions. All video clips are also included in the supplementary materials for further review.

(a) **Action:** Add raw chicken pieces and seasoning to a bowl of flour.

(b) **Action:** Mix yogurt or buttermilk with seasoning in a bowl.

(c) **Action:** Dip chicken pieces into the batter to coat evenly.

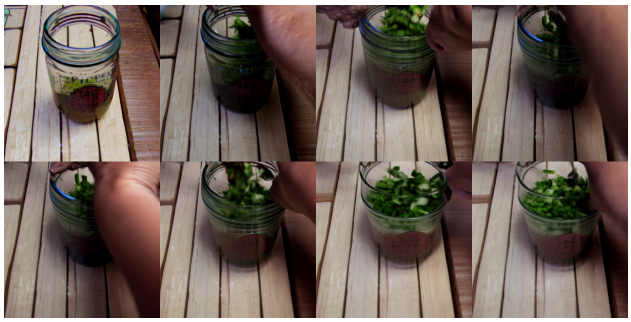
(d) **Action:** Coat the battered chicken in the flour mixture.

(e) **Action:** Fry the coated chicken in hot oil until crispy and golden.

(f) **Action:** Sprinkle seasoning on the fried chicken and serve.

Figure 18. **Video generation example.** Our pipeline effectively accomplishes long narrative video generation by producing six essential steps (*i.e.*, video clips) for cooking "Fried Chicken." It delivers a clear, structured, and instructional step-by-step narrative, showcasing the model's capability to generate coherent and comprehensive videos.

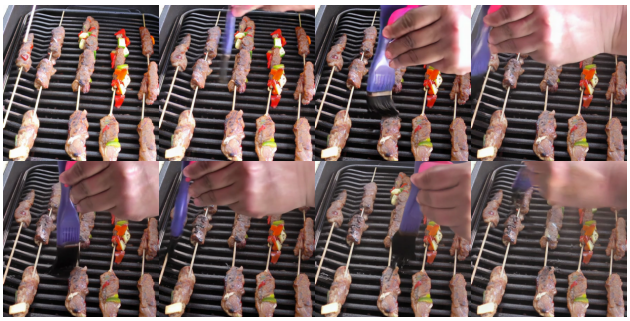
(a) **Action:** Mix chopped vegetables in a glass bowl.



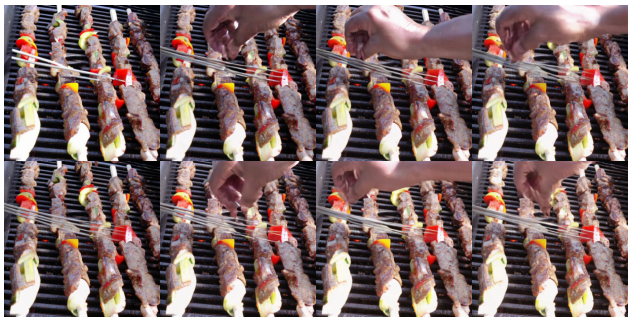
(b) **Action:** Add seasoning to the mixture of chopped vegetables.

(c) **Action:** Thoroughly mix the seasoned vegetable mixture.

(d) **Action:** Add chicken pieces to vegetable and chicken mixture.



(e) **Action:** Brush oil onto the skewered chicken and vegetable kebabs.



(f) **Action:** Place the prepared chicken and vegetable kebabs onto a grill.

(g) **Action:** Drizzle olive oil over the chicken and vegetable kebabs.

(h) **Action:** Check on the grilling skewered chicken and vegetable kebabs.

Figure 19. **Video generation example.** Our pipeline successfully generates eight crucial steps (*i.e.*, video clips) to prepare the dish "Shish Kabob." This showcases a clear, structured, and instructional step-by-step narrative, demonstrating the model's capability to produce coherent and comprehensive video content.

H. Limitations

H.1. Noisy “Actions" from ASR

While our CookGen dataset provides high-quality visual and contextual annotations, the action annotations derived from automatic speech recognition (ASR) have notable limitations. ASR-generated text often contains noise, resulting in action descriptions that are incomplete, ambiguous, or not directly informative for capturing the crucial steps in cooking processes. For instance, in Figure 8(a), the action annotation *“Hi everyone, this one’s called rainbow broken glass jello”* offers little value for understanding the cooking process, while another annotation in Figure 8(b) *“Now normally when you make jello you use two cups of boiling water”* provides vague guidance without specific details about the method. Such noisy annotations fail to align with the detailed and instructive nature of cooking instructions, which require precision and clarity to guide long narrative video generation effectively. This limitation underscores the importance of refining action annotations to improve their informativeness and utility for modeling cooking tasks.

H.2. Failure Cases

While our method generates high-quality long narrative videos, there are instances where the model fails to produce meaningful cooking steps, and the rendered video clips contain unrealistic or irrelevant content due to hallucination.

Auto-regressive Director: Repeated “Steps”. Figure 20 illustrates a failure case where the auto-regressive director repeatedly generates similar visual embeddings, resulting in redundant and uninformative cooking video clips. For example, in the provided frames, the generated steps involve repeatedly cutting the salmon fillet, which adds little value to the narrative and fails to progress meaningfully. This issue is a known limitation of auto-regressive models, often caused by a lack of diversity in the embedding generation process. A potential solution is to introduce penalties for repeated embeddings or add constraints to encourage greater variability in visual outputs.

(a) **Action:** Cutting away the salmon fillet from the backbone

(b) **Action:** Slicing the salmon fillet into even pieces

Figure 20. **Failure Case.** Auto-regressive model could generate repeated “Steps”, which is not informative to viewer.

Video Generation Model: Unrealistic Hallucination. Unrealistic hallucination occurs when a video generation model produces content inconsistent with the intended narrative. In Figure 21(a), the action ”placing the fried chicken into an oven set to preheat” is misrepresented as frying chicken in a pan, with an unrealistic increase in the quantity of chicken, showing a lack of object continuity. In Figure 21(b), the action ”adding a drizzle of sauce to a plate of grilled skewers” introduces an illogical appearance of new grilled food items, deviating from the intended action and disrupting narrative coherence.

(a) **Action:** Placing the fried chicken into a oven set to preheat

(b) **Action:** Adding a drizzle of sauce to a plate of grilled skewers

Figure 21. **Failure Case.** Video generation model could make unrealisc hallucination to generate things from “air”.