

Diffusion-Based Procedural Video Generation for Instructional Videos

Agrim Jain, Krishanu Saini, Shabari S Nair

Paper ID

Abstract

001 *Instructional content for procedural tasks—especially*
002 *cooking—has been widely explored using text–image*
003 *pipelines, yet single images lack the bandwidth to convey*
004 *fine-grained actions. Retrieval-based video approaches of-*
005 *fer richer visuals but often yield temporally inconsistent*
006 *“stitched” clips. Despite advances in video generation, no*
007 *existing method provides step-wise instructional feedback*
008 *with generated short videos that maintain temporal coher-*
009 *ence and object consistency across multiple steps. We study*
010 *this problem, analyze the challenges inherent to multi-step*
011 *video generation, and propose methods to address them.*
012 *To evaluate performance, we propose metrics for temporal*
013 *smoothness, semantic alignment, and intra/inter-clip object*
014 *stability. We also develop a training-free inference frame-*
015 *work with a retrieval-based external memory module that*
016 *enforces object-centric consistency across and within steps.*
017 *Experiments on HowTo100M show improvements in tem-*
018 *poral coherence and notable gains in object-state stability,*
019 *demonstrating the effectiveness of memory-augmented gen-*
020 *eration for instructional cooking videos.*

021 1. Introduction

022 In today’s world, instructional videos describing how to
023 perform certain procedural tasks have populated the inter-
024 net. Consequently, they have proved to be very effective
025 mediums of learning, especially when it comes to tasks like
026 cooking. There has been a lot of literature when it comes to
027 generating such instructional content in the domain of cook-
028 ing. [9], [15], [3] focus on generating textual instructions
029 accompanied by aligned images that illustrate instructions
030 for executing a procedural task (like cooking). However, a
031 single image per instruction is often not descriptive enough
032 and has too low of a bandwidth to serve as effective educa-
033 tional content for all audiences. There also exists another
034 pillar of works like [18], [2] that retrieve from an existing
035 database of clips/images to illustrate their generated feed-
036 back/instructions. However, we have found such ‘stitched’
037 videos to be very erratic and lack consistency, making them

rather difficult to follow even if they are well aligned with
the task. An natural solution to this is to generate video
rather than retrieving existing one. However, to the best
of our knowledge, no method currently exists that provide
illustrative feedback/instructions for a procedural task that
contain text accompanied by short video clips that are gen-
erated (not retrieved). We hypothesize that this is not only
due to compute complexity in generation, but also other
issues like dearth of data, lack of consistency and realism
across frames and clips. We aim to close this gap by inves-
tigating how we can generate such video clips for cooking
recipes that are consistent and well aligned with the task.

We begin by trying to adapt existing text-to-video mod-
els like Phantom [6] out-of-the box to the tasks and inves-
tigate its shortcomings. Following that we propose a series
of metrics aimed at quantifying various aspects of generated
video clips like smoothness and inter clip transition (DINO
L2 Distance, Shot Boundary Detection), alignment with the
text and recipe (Step Consistency, Goal Consistency), and
consistency of objects within and between different clips in
a recipe (Object-State Consistency). We then design an in-
ference framework with an external memory module meant
to enforce object consistency, and perform evaluations in
different settings (with and without finetuning, with and
without memory module based inferencing). Upon evalua-
tion on the HowTo100M dataset, our approach delivers upto
36% improvement in temporal coherence and notable gains
in object-state consistency, underscoring the effectiveness
of object-centric memory for multi-step video generation.

Our major contributions can be summarized as follows:

- Investigated how diffusion models can be adapted to the task of generating instructional video clips for cooking recipes, analyzed its shortcomings, and conducted qualitative and quantitative studies on the same. As per our knowledge, our study is the first of its kind.
- Proposed new metrics to effectively quantify various aspects of generated video clip, like smoothness, alignment with the text and recipe, and consistency of objects within and between different clips in a recipe.
- Designed a training-free retrieval-based memory Memory based inference pipeline for enforcing object consistency

in the generated videos.

Some results from our experiments are given in this drive link: [Link](#).

2. Related Work

Video generation for procedural tasks lies at the intersection of procedural task understanding, diffusion-based generative modeling, and structured reasoning. Prior work closest to what we aim to do include Generating Illustrated Instructions[9], I2G[3], ShowHowTo[15]. Generating Illustrated Instructions pairs textual steps with synthesized images using diffusion models, but falls behind in consistency. I2G provides new methods to improve consistency through novel evaluation metrics. ShowHowTo generates realistic and consistent sequence of frames, but only generates a single frame for each sub-instruction. We aim to build on these works by exploring ways to generate consistent short video clips (rather than frames) that accompany the instructions. Additionally, RecipeGen [19] introduces a step-aligned multimodal benchmark centered on real-world recipe generation, pairing procedural text with corresponding visual evidence.

Since we aim to achieve video generation through diffusion, recent works on efficient diffusion models that aim to enhance consistency are also relevant. GenHowTo[14] is one such work that learns to generate action and state-transformation frames conditioned on text and an initial image. We also aim to see motivation from recent diffusion-based video synthesis models such as Wan[17] and Phantom[6]. Wan provides high-fidelity text-to-video and image-to-video generation via a 3D latent diffusion framework, while Phantom extends it for subject-consistent generation using cross-modal alignment between text, image, and video.

In addition to these, there exists works like VidDetours[2] and Stitch-a-Recipe/Stich-a-demo[18], which tackle procedural branching and multi-step composition through retrieval. VidDetours retrieves alternative video segments conditioned on user queries (e.g., “without blender”) and temporal context, while Stitch-a-Recipe assembles clips corresponding to textual instructions to produce a full recipe demonstration. Though effective for procedural understanding, both remain fundamentally retrieval-based, limiting visual coherence and adaptability. However, we are motivated by the ideas proposed by these works and aim to integrate functionalities like detouring to our method in the future. In this regard, two relevant works are Video-Mined Task Graphs for Keystep Recognition [1] and Differentiable Task Graph Learning[12]. These methods demonstrate how task hierarchies and branching dependencies can be mined (or trained) automatically from large instructional video corpora, revealing causal relations among actions. As a future direction, we aim to adopt a

similar task-graph formulation to encode procedural dependencies and enable detours when users modify instructions mid-generation.

Another relevant line of work is Identity-GRPO [8], which introduces a gradient-regularized policy optimization objective to preserve subject identity across generated video sequences. By reinforcing identity-consistent features during sampling, it significantly reduces temporal drift without retraining the underlying diffusion model. While focused on human identity, its key insight that explicit consistency signals can be injected at inference time aligns with our aim of enforcing object-level consistency without modifying the base video generator.

3. Methodology

In this section, we will define the task of procedural video generation, explaining the video consistency metrics and explaining our post-training strategies for aligning our model to our task.

3.1. Task Formulation

We are given the individual textual steps T_i for a recipe, and we are required to generate a video clip V_i pertaining to each of these steps. The generate video clips must be consistent with each other and within themselves, and must individually answer each of the generated the steps as well as the overarching goal of the recipe. A naive strategy to this would be using an out of the box text-to-video model. However we noticed several drawbacks with such an approach:

- Models often hallucinate, giving unnatural outputs
- There is a lack of relevant details and artifacts
- There is a lack of consistency in shape, color and form of objects that occur within the same step and between different steps (since all clips are generated independently)

These drawbacks form the base of our finetuning and inference method which we detail in the upcoming sections.

3.2. Dataset Preparation

We prepared an extensive dataset tailored for our video generation process. We use the HowTo100[10] dataset for this task. However, the captions and the videos in the dataset contain a lot of distracting content which would not help the model learn. So taking inspiration from [2], we summarize the text transcriptions through an LLM and only store the video clip durations that have meaningful cooking steps. As of now, our dataset consists of 270 recipes with each video having 7-8 clips on average for that recipe and the corresponding text prompts.

3.3. Metrics

Procedural video generation for Instructional Videos is a task that has not been widely addressed and hence lack

proper metrics on which the model can be evaluated. This subsection elaborates on the metrics we design to evaluate consistency of the clips in procedural video generation step by step and understand how each approach has an impact on the videos generated. The proposed metrics defined are as follows:

DINO L2 Distance: To quantify frame-to-frame visual consistency at transition points (between the end of one clip and start of next) in generated videos, we measure the L2 distance between DINO[4] feature embeddings of consecutive frames at such points. DINO provides robust, semantically meaningful representations, making it well-suited for capturing subtle changes in appearance and structure. For each pair of successive frames, we extract their DINO embeddings and compute the Euclidean (L2) distance, which reflects the magnitude of visual variation between them. These distances are then averaged across all transition points to produce a single consistency score. Lower values indicate smoother temporal transitions and higher perceptual stability, whereas larger distances suggest abrupt or inconsistent changes in the generated content.

Shot Boundary Detection: To evaluate whether a generated video maintains consistency with the previous clip, we compute the shot-change probability using TransNetV2[13], a state-of-the-art deep network for shot boundary detection. A high probability indicates a strong discontinuity, suggesting that the generated video diverges noticeably from the reference, whereas a low probability implies a smooth, consistent transition. This provides a complementary measure of temporal and visual coherence.

Step Consistency: To assess whether a generated video clip faithfully reflects the semantic content of a target recipe step, we first uniformly sample a fixed number of frames from the generated video and extract CLIP [11] image embeddings for each frame. These embeddings are averaged to obtain a single video-level representation, normalized to ensure comparability. We then prompt an LLM (GPT-4o) to generate 5 text prompts very similar to the ground truth step, but also differing in meaningful ways. For example, for a ground truth step 'peel the carrot', a generated text could be 'slice the carrot'. These 5 generated texts would serve as hard negatives. We then calculate the similarity scores between the text embeddings of these 6 texts and the video embedding. Finally we take the softmax of the scores across the 6 texts, and the score corresponding to the ground truth text would serve as the final metric. This metric provides a fine-grained way to evaluate whether the procedural video captures the specific step it is expected to depict.

Goal Consistency: To evaluate whether the series of generated video clips successfully achieves the intended final outcome of the recipe, we compute a CLIP-based goal consistency score using the last frame of the last video clip. We provide description of all the steps in the recipe to an

LLM (GPT-4o) and prompt it to generate a concise description of what an image of the final dish of this recipe would look like. We then output the similarity score between this description and the last frame of the final generated video clip. The reasoning behind this is that a instructional recipe video should end with showing how the final dish looks like, and that should be a reasonably good indication of whether all the steps in the recipe have been correctly followed or not. Higher similarity indicates that the completed video aligns well with the intended goal, whereas lower similarity suggests that the final visual outcome deviates from the expected result. This metric directly assesses whether the generated video "finishes" the step in a way that is semantically faithful to the prompt, making it a strong indicator of success for procedural video generation.

Object-State Consistency: To evaluate cross-clip consistency in procedural video generation, we introduce an object-state tracking metric that monitors how the visual properties of key objects evolve across recipe steps. For each generated clip, we extract a representative middle frame and identify all visible objects using a VLM. Each detected object is then assigned a short, GPT-generated state description (e.g., "whole carrot on board", "pan heating on stove"). Across clips, objects are matched by name rather than ID, and their states are compared against a global state dictionary that tracks the most up-to-date description of each object. If an object's state remains the same or changes in a way that is compatible with the cooking step's textual prompt (e.g., "carrot becomes sliced" after a "slice the carrots" step), we classify the object as consistent and update the global state accordingly. When a state change contradicts the instruction or appears unjustified, the system flags the object as inconsistent, with GPT providing a brief explanation of why the transition is semantically invalid. Repeating this process across all clips provides a fine-grained measure of temporal and procedural coherence, and yields a final consistency score reflecting the proportion of objects whose trajectories remained logically aligned with the recipe workflow. This metric directly captures whether the video model preserves the evolving physical state of ingredients and tools—a core requirement for generating accurate and instructive cooking sequences.

3.4. Phantom Model Architecture

The Phantom [6] architecture is designed as a unified framework for generating subject-consistent videos from both single and multiple reference images. It is built upon a pre-existing video foundation model that uses a Multimodal Diffusion Transformer (MMDiT) structure. The architecture consists of two main parts: an "input head" that processes the various inputs, and the core MMDiT module where the cross-modal learning occurs. The input head uses separate encoders for each type of input: a 3D VAE for the

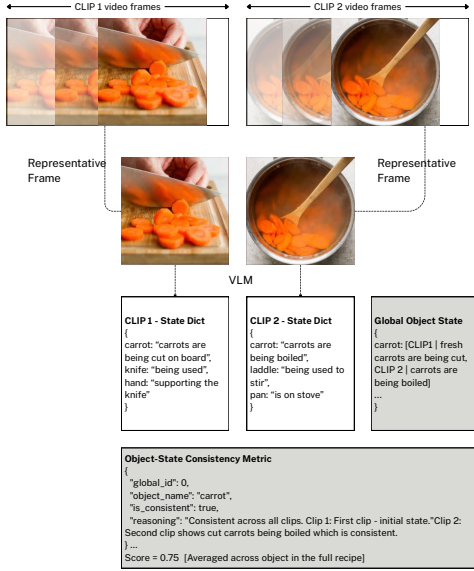


Figure 1. Object-State consistency metric on an example of boiling carrots for stew. Note that our metric is averaged across the objects in the global dictionary.

video, a Large Language Model (LLM) for the text prompt, and critically, a dual-encoder system for the reference image(s) which uses both a VAE and a CLIP image encoder. This dual approach captures both low-level details (from the VAE) and high-level semantic meaning (from CLIP) of the subject.

Once the inputs are encoded, their features are strategically combined and fed into the two branches of the MMDiT module. The low-level VAE features from the reference image are concatenated with the video features in the visual branch, while the high-level CLIP features from the image are concatenated with the text prompt’s features in the text branch.

Training: The training architecture of phantom is shown in figure 2. Phantom uses triplet loss used for training the model which is a carefully constructed set of text-image-video combinations designed to teach the model cross-modal alignment. For each training instance, the process starts with a video clip. A detailed text caption is generated for this video, describing the subjects, their actions, and the environment. The key innovation is how the corresponding image for the triplet is selected. Instead of simply extracting a frame from the source video, the system finds an external image that contains the same subject described in the text caption. This “cross-pairing” forces the model to learn the essential identity of the subject from the image and combine it with the motion and scene instructions from the text, rather than just “copy-pasting” the subject along with its original background or lighting from a source frame. This method is crucial for preventing what the au-

thors call “image content leakage” and ensures the model genuinely learns to animate a subject in new contexts based on textual commands.

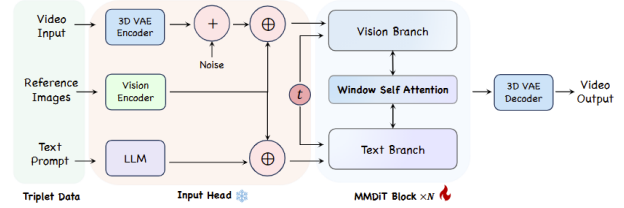


Figure 2. Phantom training framework (borrowed from [6])

3.5. Fine tuning

Phantom wan is trained on a general cross modal dataset Panda-70M [5]. This is a generic model and from our zero-shot results we realized it’s not suitable for cooking task.

Phantom has trained their models using a triplet dataset to demonstrate consistency over humans. This loss metric however leads to loss of important ingredient information while cooking and produces very animated and unreal images for food objects. In order to align the model with cooking task we prepared a dataset of image-text pairs using the curated HowTo100M dataset. We use the Diffusion-Pipe [16] framework to train our Phantom model with LoRA for 20 epochs.

Diffusion-Pipe: It is a modular, pipeline-parallel training architecture for diffusion models, decomposing the end-to-end workflow into interchangeable components—such as text encoding, denoising, scheduling, and decoding—connected through a unified pipeline interface. This design supports efficient scaling across devices while enabling flexible component swapping and rapid experimentation. The framework is compatible with modern diffusion models, including Wan, Phantom-WAN and Hunyuan-Video, making it a suitable backbone for our training setup.

3.6. Memory based Inference Pipeline

In order to solve the problem of consistency in object, we propose a inference pipeline to track and extract possible ‘consistent’ object to feed into Phantom. For each clip and its corresponding textual description in the training set we first use a VLM (GPT-4o) to extract all the objects within it with detailed descriptions, and further classify each object as ‘Highly Likely’, ‘Likely’, and ‘Unlikely’ in terms of how probable that object is to remain consistent. For example, a stove would be classified as ‘Highly Likely’, but a tomato would be classified as ‘Unlikely’. For each of the objects classified as ‘Likely’ or ‘Highly Likely’, we iterate through all the frames of the clip and use Grounding DINO [7] to find the crops. For each of the objects we then store the crop with the highest confidence, along with its description

in a database. Following this, we calculate the image embeddings of these objects and the text embeddings of their descriptions, both in CLIP space [11], and take their average before storing it in a FAISS index for fast retrieval. The same can be understood from the flowchart in Figure 3

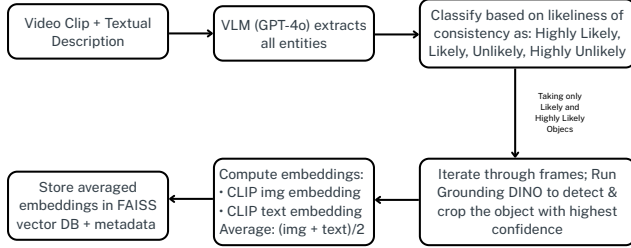


Figure 3. Memory Module Data Storage Pipeline

During inference time (Figure 4), we proceed in an on-line manner for generating the video clips. Before generating each clip, we prompt a VLM to with the input text, previous step texts, and objects used in previous steps, to predict probable ‘consistent’ objects for the current step. Note that these objects can either be new objects that will come into existence in the current step, or objects that previously existed. We then use the descriptions of these predicted objects, as well as previous steps, to enhance the text of the current step, again using GPT-4o. This process ensures that relevant details of the objects are integrated into the text and it is also contextualized with respect to all previous steps. Finally, we calculate the text embeddings of the predicted objects in CLIP space and use it to retrieve the closest image from the stored FAISS vector database. These retrieved images along with the enhanced text is fed into the Phantom model to generate the video clip. In addition to maintaining consistency, an added advantage of this method is its online nature and lack of need for training.

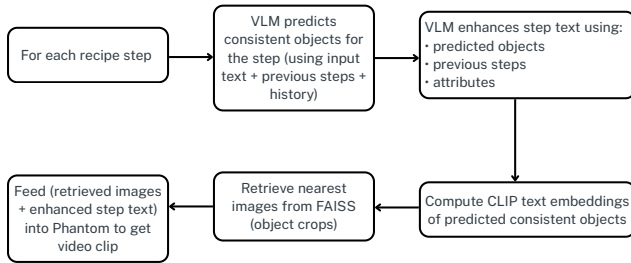


Figure 4. Memory Module Inference Pipeline

3.7. Reinforcement learning based Enhancement

Identity-GRPO [8] is a human-feedback-driven optimization pipeline designed to refine multi-human, identity-preserving video generation. The method leverages human preference signals to construct a reward model and then op-

timizes the generator using a GRPO-based reinforcement learning loop. While Identity-GRPO focuses on aligning generation quality with human-judged identity fidelity, our setting differs in both objective and supervision. Instead of relying on human evaluation or RLHF to train the reward model, we plan to use an automatic object state consistency metric as the reward function. This metric captures whether an object maintains coherent attributes and temporal state across generated frames, enabling reinforcement learning that targets fine-grained temporal stability rather than identity preservation. Consequently, our RL setup shares the iterative optimization structure of Identity-GRPO but replaces human preference data with a fully automated, state-consistency-based reward model tailored for object-centric video generation. However, due to time and resource constraints, we were unable to run this training. We hope to take this up as a promising future direction.

4. Experimental Settings

For finetuning using Diffusion Pipe we have used 5000 image-text extracted from HowTo100M. The finetuning has been done for 20 epochs with LoRA adapter (rank - 16, alpha - 16), and Adam optimizer. All evaluation has been done on 100 recipes from the HowTo100M dataset. During inference, denoising has been done for 50 steps at 16 fps. Each generated clip is of 5 seconds.

All the experiments have been done on one H100 GPU and T4 from Google Colab.

5. Results and Discussion

In this section, we evaluate our proposed framework through a combination of qualitative comparisons and quantitative metrics. Our experiments are designed to assess several key aspects of the system such as overall generation quality, temporal and object-level consistency, as well as the contribution of our memory-based inference pipeline and finetuning strategy. We conduct our evaluations for 4 settings - baseline Phantom model, Phantom finetuned using Diffusion Pipe, baseline Phantom enhanced with Memory Module, finetuned Phantom enhanced with Memory Module. The results are given in Table 1

5.1. Quantitative Analysis

From the Table 1 we observe that the frame transition metrics like Dino L2 Distance and shot boundary metric is improved only slightly from baseline to finetuned model. However, we see that adding memory module results in a decent improvement of 36%. We believe that this is because of the consistency between the reference objects selected by the memory module.

The Step consistency metric and goal consistency metric are based on the fidelity of the generated video to solve

Model	L2 Dino Dist.	Shot Boundary	Step Consistency	Goal Consistency	Object-State Consistency (mean/std)
Baseline (Phantom)	0.172	0.68	0.1675	0.2490	0.825 / 0.143
Finetuned	0.167	0.57	0.1676	0.2503	0.825 / 0.132
Baseline + Memory Module	0.141	0.46	0.1678	0.2380	0.830 / 0.109
Finetuned + Memory Module	0.110	0.44	0.1679	0.2415	0.846 / 0.106

Table 1. Quantitative comparison of different model configurations across consistency metrics.

the MCQs which are adversarially selected to be very similar. The results we obtain on this metric does not improve which we believe is because the options are quite similar (hard negatives); for e.g., for the question "what action is being performed on the carrot?", the options were "the carrot being peeled", "the carrot being cut into long pieces", "the carrot being chopped". These options are quite similar and we suspect our model overfits on the entity carrot. So the result for all these video generation models is not much better than a random guess.

Our object-state consistency metric also has only a 2.5% increase which seems quite low. However, This is most probably because of an inherent limitation of our metric: in most baseline videos the model generated completely new object every step which results in new items in our state dictionary which were counted as consistent on their first appearance (as there was no wrong state change) and do not appear afterwards in the video clips (which is actually incorrect). Also, the standard deviation of baseline is higher compared to improved models showing a lack in consistency.

5.2. Qualitative Analysis

Qualitatively, our memory-augmented models demonstrate substantially improved visual coherence and object fidelity compared to the baseline. Generated videos exhibit clearer object boundaries, smoother transitions between cooking steps, and more consistent ingredient transformations, making the overall sequence easier to follow. In contrast, baseline models frequently hallucinate objects, omit relevant ingredient changes or fail to update utensil states appropriately—for example, showing chopped meat as whole or ignoring the placement of a pan. By leveraging the memory module to retrieve prior object states and contextual information from previous steps, our approach ensures that each generated clip aligns with both the current action and the overall goal of the recipe. This results in videos that not only maintain temporal consistency but also faithfully capture the progression of ingredients and tools, reducing abrupt or inconsistent changes across consecutive steps and producing a more realistic and instructive depiction of the procedural task.

5.3. Discussion

Our experimental results highlight the complementary strengths of finetuning and our memory-based inference pipeline. While finetuning alone yields only modest improvements in low-level temporal smoothness (as seen through small gains in Dino L2 and shot boundary metrics), incorporating memory leads to a substantial boost of 36% in temporal coherence, demonstrating that consistent conditioning signals across steps have a far greater impact than model weight updates on such a small dataset. This finding reinforces our central thesis: for procedural tasks, the bottleneck is not generative fidelity but the stability and persistence of object-level grounding. Memory-guided retrieval provides this grounding by ensuring that each clip is generated with explicit, step-relevant visual references, reducing drift and hallucination across steps.

Interestingly, the MCQ-based step and goal consistency metrics remain nearly constant across all model variants. We attribute this to inherent properties of the evaluation: the question-answer pairs use tightly clustered, adversarial negative options, and the model tends to overbias on the primary entity (e.g., "carrot"), making fine-grained distinctions between similar actions difficult. This ceiling effect suggests that these metrics may be insensitive to incremental improvements and that more discriminative benchmarks may be needed for procedural action understanding in generative video models.

The Object-State Consistency metric shows a more meaningful improvement, especially when considering its reduced variance across samples. Although the overall gain (2.5%) appears small, it is important to contextualize this within the limitations of the metric itself. Baseline models often introduced entirely new objects at each step—errors that were miscounted as "consistent" on first appearance. Our memory-enhanced models, by contrast, rely on persistent object retrieval and thus avoid unnecessary reintroductions, leading to more stable and interpretable state transitions. The lower standard deviation across runs further reflects this stability: even when absolute gains are modest, reliability is significantly higher.



Figure 5. Video generation for a recipe: beef stew with quail eggs. Generated by our fine tuned with memory module.

6. Conclusion

In this work, we present a consistency-aware pipeline for generating procedural videos that leverages object-centric retrieval, multimodal reasoning and modern diffusion-based video models. By combining VLM-driven object extraction, Grounding DINO-based localization, CLIP-space embedding alignment and FAISS retrieval, our methods enforce visual continuity across independently generated clips, addressing one of the key limitations of current instructional video generation systems. Importantly, our approach operates entirely at inference time, requiring no architectural changes to the underlying video diffusion model and enabling compatibility with state-of-the-art generators such as Phantom-WAN.

While our results are preliminary due to limited fine-tuning data and compute, they demonstrate that integrating structured object information and step-aware reasoning meaningfully improves consistency in procedural video synthesis. Our system establishes a foundation upon which richer capabilities such as task-graph-based branching, RL-driven consistency rewards, and more sophisticated prompting strategies can be built. We hope this work serves as a step toward flexible and coherent generation of multi-step instructional videos that better reflect real-world procedural tasks.

7. Future Work

Although our initial results demonstrate the feasibility of consistency-aware procedural video generation, they are

constrained by the limited scale of our fine-tuning data. Due to time and resource constraints, the current model was trained on lesser videos, which restricts its ability to generalize across diverse cooking styles, environments, and object configurations. We expect substantial improvements by fine-tuning on a significantly larger and more varied dataset, particularly in terms of temporal stability, fine-grained object fidelity and robustness to uncommon scenarios. Expanding the dataset will also allow us to better evaluate cross-recipe generalization and test the limits of our consistency retrieval pipeline.

Another major direction is the completion and integration of our reinforcement-learning module. While the reward design and implementation are functional, full RL training remains outstanding. Incorporating GRPO-based optimization offers an appealing path forward, as it allows the use of flexible, non-differentiable metrics—such as identity similarity, object persistence scores, or clip-to-clip consistency metrics—to directly shape the generation behavior. This could significantly reduce accumulated drift over long sequences and offer a principled way to impose object-centric constraints during sampling. Additionally, our VLM prompting strategy can be made more structured and hierarchical to reduce hallucinations and improve prediction of consistent objects. Designing more explicit prompt templates, or prompting the VLM to perform step-level reasoning before prediction, may further enhance grounding and produce more coherent multi-clip instructional videos.

Team Roles

The project was carried out collaboratively, with each team member contributing to multiple aspects of the work. Responsibilities included dataset preparation and preprocessing, implementation of the memory-based inference pipeline, finetuning of the video diffusion models, and design of quantitative and qualitative evaluation protocols. Team members also jointly handled experimental analysis, visualization of results, and writing of the manuscript.

Acknowledgements

We acknowledge the support of the GenAI computing server whose H100 infrastructure enabled all our finetuning experiments and large-scale code execution. We would also like to thank Kumar Ashutosh (PhD student under Prof. Grauman) for his invaluable guidance in understanding his prior work, as well as for providing insights into the improved HowTo100 dataset from VidDetours. We are also grateful to Jounghbin An, our course TA, for his thoughtful discussions during topic selection and for helping us refine the project direction. Finally, we extend our thanks to Professor Kristen Grauman for her feedback and guidance, which shaped the conceptual direction of this work.

References

- [1] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos, 2023. 2
- [2] Kumar Ashutosh, Zihui Xue, Tushar Nagarajan, and Kristen Grauman. Detours for navigating instructional videos, 2024. 1, 2
- [3] Jing Bi, Pinxin Liu, Ali Vosoughi, Jiarui Wu, Jinxi He, and Chenliang Xu. i^2g : Generating instructional illustrations via text-conditioned diffusion, 2025. 1, 2
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 3
- [5] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 4
- [6] Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment, 2025. 1, 2, 3, 4
- [7] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. 4
- [8] Xiangyu Meng, Zixian Zhang, Zhenghao Zhang, Junchao Liao, Long Qin, and Weizhi Wang. Identity-grpo: Optimizing multi-human identity-preserving video generation via reinforcement learning, 2025. 2, 5
- [9] Sachit Menon, Ishan Misra, and Rohit Girdhar. Generating illustrated instructions, 2024. 1, 2
- [10] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips, 2019. 2
- [11] 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, 2021. 3, 5
- [12] Luigi Seminara, Giovanni Maria Farinella, and Antonino Furnari. Differentiable task graph learning: Procedural activity representation and online mistake detection from ego-centric videos, 2025. 2
- [13] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection, 2020. 3
- [14] Tomáš Souček, Dima Damen, Michael Wray, Ivan Laptev, and Josef Sivic. Genhowto: Learning to generate actions and state transformations from instructional videos, 2024. 2
- [15] Tomáš Souček, Prajwal Gatti, Michael Wray, Ivan Laptev, Dima Damen, and Josef Sivic. Showhowto: Generating scene-conditioned step-by-step visual instructions, 2025. 1, 2
- [16] Tdrussell. Tdrussell/diffusion-pipe: A pipeline parallel training script for diffusion models. 4
- [17] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wenteng Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. 2
- [18] Chi Hsuan Wu, Kumar Ashutosh, and Kristen Grauman. Stitch-a-demo: Video demonstrations from multistep descriptions, 2025. 1, 2
- [19] Ruoxuan Zhang, Jidong Gao, Bin Wen, Hongxia Xie, Chenming Zhang, Hong-Han Shuai, and Wen-Huang Cheng. Recipegen: A step-aligned multimodal benchmark for real-world recipe generation, 2025. 2