

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.*

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 investi-
gating 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 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

079 in the generated videos.

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

082 2. Related Work

083 Video generation for procedural tasks lies at the intersection
084 of procedural task understanding, diffusion-based genera-
085 tive modeling, and structured reasoning. Prior work closest
086 to what we aim to do include Generating Illustrated
087 Instructions[9], I2G[3], ShowHowTo[15]. Generating Il-
088 lustrated Instructions pairs textual steps with synthesized
089 images using diffusion models, but falls behind in consis-
090 tency. I2G provides new methods to improve consistency
091 through novel evaluation metrics. ShowHowTo generates
092 realistic and consistent sequence of frames, but only gen-
093 erates a single frame for each sub-instruction. We aim to
094 build on these works by exploring ways to generate consist-
095 ent short video clips (rather than frames) that accompany
096 the instructions. Additionally, RecipeGen [19] introduces a
097 step-aligned multimodal benchmark centered on real-world
098 recipe generation, pairing procedural text with correspond-
099 ing visual evidence.

100 Since we aim to achieve video generation through dif-
101 fusion, recent works on efficient diffusion models that aim
102 to enhance consistency are also relevant. GenHowTo[14]
103 is one such work that learns to generate action and state-
104 transformation frames conditioned on text and an ini-
105 tial image. We also aim to see motivation from recent
106 diffusion-based video synthesis models such as Wan[17]
107 and Phantom[6]. Wan provides high-fidelity text-to-video
108 and image-to-video generation via a 3D latent diffusion
109 framework, while Phantom extends it for subject-consistent
110 generation using cross-modal alignment between text, im-
111 age, and video.

112 In addition to these, there exists works like
113 VidDetours[2] and Stitch-a-Recipe/Stich-a-demo[18],
114 which tackle procedural branching and multi-step compo-
115 sition through retrieval. VidDetours retrieves alternative
116 video segments conditioned on user queries (e.g., “without
117 blender”) and temporal context, while Stitch-a-Recipe
118 assembles clips corresponding to textual instructions to
119 produce a full recipe demonstration. Though effective
120 for procedural understanding, both remain fundamentally
121 retrieval-based, limiting visual coherence and adaptability.
122 However, we are motivated by the ideas proposed by these
123 works and aim to integrate functionalities like detouring to
124 our method in the future. In this regard, two relevant works
125 are Video-Mined Task Graphs for Keystep Recognition
126 [1] and Differentiable Task Graph Learning[12]. These
127 methods demonstrate how task hierarchies and branching
128 dependencies can be mined (or trained) automatically from
129 large instructional video corpora, revealing causal relations
130 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 generated video clips must be consistent with each other and within themselves, and must individually answer each of the generated 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

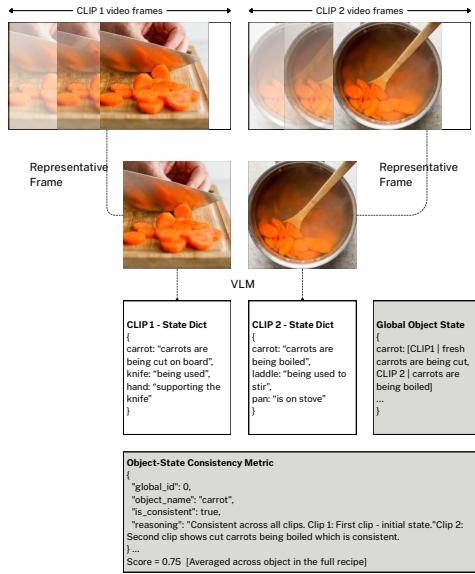


Figure 1. Object-State consistency metric on an example of boiling carrots for stew. Note that our metric is averaged across the objects in the global dictionary.

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.

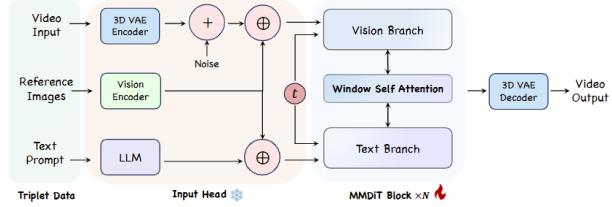


Figure 2. Phantom training framework (borrowed from [6])

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

315
316
317

318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340

341
342
343
344
345
346
347
348
349
350
351
352
353
354
355

356 in a database. Following this, we calculate the image em-
 357 beddings of these objects and the text embeddings of their
 358 descriptions, both in CLIP space [11], and take their aver-
 359 age before storing it in a FAISS index for fast retrieval. The
 360 same can be understood from the flowchart in Figure 3

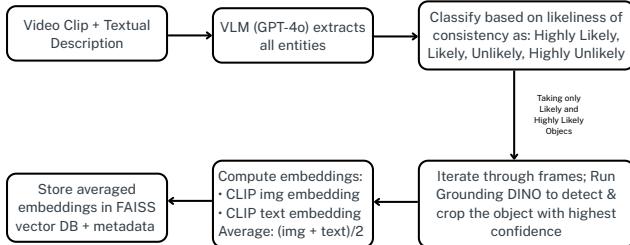


Figure 3. Memory Module Data Storage Pipeline

361 During inference time (Figure 4), we proceed in an on-
 362 line manner for generating the video clips. Before generat-
 363 ing each clip, we prompt a VLM to with the input text, pre-
 364 vious step texts, and objects used in previous steps, to pre-
 365 dict probable ‘consistent’ objects for the current step. Note
 366 that these objects can either be new objects that will come
 367 into existence in the current step, or objects that previously
 368 existed. We then use the descriptions of these predicted ob-
 369 jects, as well as previous steps, to enhance the text of the
 370 current step, again using GPT-4o. This process ensures that
 371 relevant details of the objects are integrated into the text and
 372 it is also contextualized with respect to all previous steps.
 373 Finally, we calculate the text embeddings of the predicted
 374 objects in CLIP space and use it to retrieve the closest im-
 375 age from the stored FAISS vector database. These retrieved
 376 images along with the enhanced text is fed into the Phantom
 377 model to generate the video clip. In addition to maintaining
 378 consistency, an added advantage of this method is its online
 379 nature and lack of need for training.

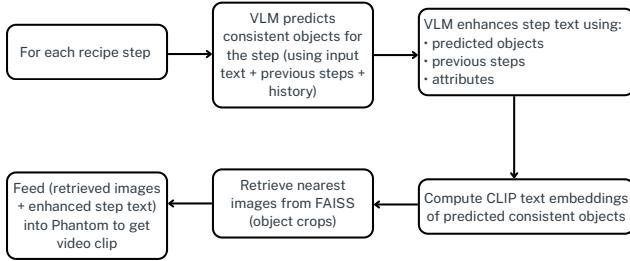


Figure 4. Memory Module Inference Pipeline

3.7. Reinforcement learning based Enhancement

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

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

4. Experimental Settings

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

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

5. Results and Discussion

412 In this section, we evaluate our proposed framework
 413 through a combination of qualitative comparisons and quan-
 414 titative metrics. Our experiments are designed to assess sev-
 415 eral key aspects of the system such as overall generation
 416 quality, temporal and object-level consistency, as well as the
 417 contribution of our memory-based inference pipeline and
 418 finetuning strategy. We conduct our evaluations for 4 set-
 419 tings - baseline Phantom model, Phantom finetuned using
 420 Diffusion Pipe, baseline Phantom enhanced with Memory
 421 Module, finetuned Phantom enhanced with Memory Mod-
 422 ule. The results are given in Table 1

5.1. Quantitative Analysis

423 From the Table 1 we observe that the frame transition met-
 424 rics like Dino L2 Distance and shot boundary metric is
 425 improved only slightly from baseline to finetuned model.
 426 However, we see that adding memory module results in a
 427 decent improvement of 36%. We believe that this is be-
 428 cause of the consistency between the reference objects se-
 429 lected by the memory module.

429 The Step consistency metric and goal consistency met-
 430 ric 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.

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

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

455 5.2. Qualitative Analysis

456 Qualitatively, our memory-augmented models demonstrate
 457 substantially improved visual coherence and object fidelity
 458 compared to the baseline. Generated videos exhibit clearer
 459 object boundaries, smoother transitions between cooking
 460 steps, and more consistent ingredient transformations, mak-
 461 ing the overall sequence easier to follow. In contrast, base-
 462 line models frequently hallucinate objects, omit relevant in-
 463 gredient changes or fail to update utensil states appropri-
 464 ately—for example, showing chopped meat as whole or ig-
 465 noring the placement of a pan. By leveraging the mem-
 466 ory module to retrieve prior object states and contextual in-
 467 formation from previous steps, our approach ensures that
 468 each generated clip aligns with both the current action and
 469 the overall goal of the recipe. This results in videos that
 470 not only maintain temporal consistency but also faithfully
 471 capture the progression of ingredients and tools, reducing
 472 abrupt or inconsistent changes across consecutive steps and
 473 producing a more realistic and instructive depiction of the
 474 procedural task.

475 5.3. Discussion

476 Our experimental results highlight the complementary
 477 strengths of finetuning and our memory-based inference
 478 pipeline. While finetuning alone yields only modest im-
 479 provements in low-level temporal smoothness (as seen
 480 through small gains in Dino L2 and shot boundary metrics),
 481 incorporating memory leads to a substantial boost of 36%
 482 in temporal coherence, demonstrating that consistent con-
 483 ditioning signals across steps have a far greater impact than
 484 model weight updates on such a small dataset. This finding
 485 reinforces our central thesis: for procedural tasks, the bot-
 486 tleneck is not generative fidelity but the stability and persis-
 487 tence of object-level grounding. Memory-guided retrieval
 488 provides this grounding by ensuring that each clip is gener-
 489 ated with explicit, step-relevant visual references, reducing
 490 drift and hallucination across steps.

491 Interestingly, the MCQ-based step and goal consistency
 492 metrics remain nearly constant across all model variants.
 493 We attribute this to inherent properties of the evaluation:
 494 the question–answer pairs use tightly clustered, adversarial
 495 negative options, and the model tends to overbias on the
 496 primary entity (e.g., "carrot"), making fine-grained distinc-
 497 tions between similar actions difficult. This ceiling effect
 498 suggests that these metrics may be insensitive to incremen-
 499 tal improvements and that more discriminative benchmarks
 500 may be needed for procedural action understanding in gen-
 501 erative video models.

502 The Object-State Consistency metric shows a more
 503 meaningful improvement, especially when considering its
 504 reduced variance across samples. Although the overall gain
 505 (2.5%) appears small, it is important to contextualize this
 506 within the limitations of the metric itself. Baseline models
 507 often introduced entirely new objects at each step—errors
 508 that were miscounted as "consistent" on first appearance.
 509 Our memory-enhanced models, by contrast, rely on persis-
 510 tent object retrieval and thus avoid unnecessary reintroduc-
 511 tions, leading to more stable and interpretable state transi-
 512 tions. The lower standard deviation across runs further re-
 513 flects this stability: even when absolute gains are modest,
 514 reliability is significantly higher.



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

515

6. Conclusion

516 In this work, we present a consistency-aware pipeline for
 517 generating procedural videos that leverages object-centric
 518 retrieval, multimodal reasoning and modern diffusion-based
 519 video models. By combining VLM-driven object ex-
 520 traction, Grounding DINO-based localization, CLIP-space
 521 embedding alignment and FAISS retrieval, our methods
 522 enforce visual continuity across independently generated
 523 clips, addressing one of the key limitations of current in-
 524 structional video generation systems. Importantly, our ap-
 525 proach operates entirely at inference time, requiring no ar-
 526 chitectural changes to the underlying video diffusion model
 527 and enabling compatibility with state-of-the-art generators
 528 such as Phantom-WAN.

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

540 7. Future Work

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

543 constrained by the limited scale of our fine-tuning data.
 544 Due to time and resource constraints, the current model was
 545 trained on lesser videos, which restricts its ability to gener-
 546 alize across diverse cooking styles, environments, and ob-
 547 ject configurations. We expect substantial improvements by
 548 fine-tuning on a significantly larger and more varied dataset,
 549 particularly in terms of temporal stability, fine-grained ob-
 550 ject fidelity and robustness to uncommon scenarios. Ex-
 551 panding the dataset will also allow us to better evaluate
 552 cross-recipe generalization and test the limits of our con-
 553 sistency retrieval pipeline.

554 Another major direction is the completion and integra-
 555 tion of our reinforcement-learning module. While the re-
 556 ward design and implementation are functional, full RL
 557 training remains outstanding. Incorporating GRPO-based
 558 optimization offers an appealing path forward, as it al-
 559 lows the use of flexible, non-differentiable metrics—such
 560 as identity similarity, object persistence scores, or clip-
 561 to-clip consistency metrics—to directly shape the gener-
 562 ation behavior. This could significantly reduce accumu-
 563 lated drift over long sequences and offer a principled way
 564 to impose object-centric constraints during sampling. Ad-
 565 ditionally, our VLM prompting strategy can be made more
 566 structured and hierarchical to reduce hallucinations and im-
 567 prove prediction of consistent objects. Designing more ex-
 568 plicit prompt templates, or prompting the VLM to perform
 569 step-level reasoning before prediction, may further enhance
 570 grounding and produce more coherent multi-clip instruc-
 571 tional videos.

572 **Team Roles**

573 The project was carried out collaboratively, with each
574 team member contributing to multiple aspects of the work.
575 Responsibilities included dataset preparation and prepro-
576 cessing, implementation of the memory-based inference
577 pipeline, finetuning of the video diffusion models, and de-
578 sign of quantitative and qualitative evaluation protocols.
579 Team members also jointly handled experimental analysis,
580 visualization of results, and writing of the manuscript.

581 **Acknowledgements**

582 We acknowledge the support of the GenAI computing
583 server whose H100 infrastructure enabled all our fine-
584 tuning experiments and large-scale code execution. We
585 would also like to thank Kumar Ashutosh (PhD student
586 under Prof. Grauman) for his invaluable guidance in un-
587 derstanding his prior work, as well as for providing in-
588 sights into the improved HowTo100 dataset from VidDe-
589 tours. We are also grateful to Jounghin An, our course TA,
590 for his thoughtful discussions during topic selection and for
591 helping us refine the project direction. Finally, we extend
592 our thanks to Professor Kristen Grauman for her feedback
593 and guidance, which shaped the conceptual direction of this
594 work.

595 **References**

- 596 [1] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Tri-
597 antafyllos Afouras, and Kristen Grauman. Video-mined task
598 graphs for keystep recognition in instructional videos, 2023.
599 2
- 600 [2] Kumar Ashutosh, Zihui Xue, Tushar Nagarajan, and Kristen
601 Grauman. Detours for navigating instructional videos, 2024.
602 1, 2
- 603 [3] Jing Bi, Pinxin Liu, Ali Vosoughi, Jiarui Wu, Jinxi He, and
604 Chenliang Xu. i^2g : Generating instructional illustrations via
605 text-conditioned diffusion, 2025. 1, 2
- 606 [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou,
607 Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerg-
608 ing properties in self-supervised vision transformers, 2021. 3
- 609 [5] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace,
610 Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon,
611 Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang,
612 and Sergey Tulyakov. Panda-70m: Captioning 70m videos
613 with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
614 *Recognition*, 2024. 4
- 616 [6] Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Ji-
617 awei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu.
618 Phantom: Subject-consistent video generation via cross-
619 modal alignment, 2025. 1, 2, 3, 4
- 620 [7] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao
621 Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang,
622 Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marry-

- 623 ing dino with grounded pre-training for open-set object de-
624 tection, 2024. 4
- 625 [8] Xiangyu Meng, Zixian Zhang, Zhenghao Zhang, Junchao
626 Liao, Long Qin, and Weizhi Wang. Identity-grpo: Optimiz-
627 ing multi-human identity-preserving video generation via re-
628 enforcement learning, 2025. 2, 5
- 629 [9] Sachit Menon, Ishan Misra, and Rohit Girdhar. Generating
630 illustrated instructions, 2024. 1, 2
- 631 [10] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac,
632 Makarand Tapaswi, Ivan Laptev, and Josef Sivic.
633 Howto100m: Learning a text-video embedding by watching
634 hundred million narrated video clips, 2019. 2
- 635 [11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya
636 Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,
637 Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen
638 Krueger, and Ilya Sutskever. Learning transferable visual
639 models from natural language supervision, 2021. 3, 5
- 640 [12] Luigi Seminara, Giovanni Maria Farinella, and Antonino
641 Furnari. Differentiable task graph learning: Procedural ac-
642 tivity representation and online mistake detection from ego-
643 centric videos, 2025. 2
- 644 [13] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective
645 deep network architecture for fast shot transition detection,
646 2020. 3
- 647 [14] Tomáš Souček, Dima Damen, Michael Wray, Ivan Laptev,
648 and Josef Sivic. Genhowto: Learning to generate actions
649 and state transformations from instructional videos, 2024. 2
- 650 [15] Tomáš Souček, Prajwal Gatti, Michael Wray, Ivan Laptev,
651 Dima Damen, and Josef Sivic. Showhowto: Generating
652 scene-conditioned step-by-step visual instructions, 2025. 1,
653 2
- 654 [16] Tdrussell. Tdrussell/diffusion-pipe: A pipeline parallel train-
655 ing script for diffusion models. 4
- 656 [17] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao,
657 Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianx-
658 iao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jin-
659 gren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao,
660 Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang,
661 Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei
662 Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui,
663 Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang,
664 Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu,
665 Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu
666 Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yi-
667 tong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun
668 Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi
669 Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open
670 and advanced large-scale video generative models, 2025. 2
- 671 [18] Chi Hsuan Wu, Kumar Ashutosh, and Kristen Grauman.
672 Stitch-a-demo: Video demonstrations from multistep de-
673 scriptions, 2025. 1, 2
- 674 [19] Ruoxuan Zhang, Jidong Gao, Bin Wen, Hongxia Xie, Chen-
675 ming Zhang, Hong-Han Shuai, and Wen-Huang Cheng.
676 Recipegen: A step-aligned multimodal benchmark for real-
677 world recipe generation, 2025. 2