ShowHowTo: Generating Scene-Conditioned Step-by-Step Visual Instructions

Tomáš Souček[1] Prajwal Gatti[2] Michael Wray[2] Ivan Laptev[3] Dima Damen[2] Josef Sivic[1]
[1]CIIRC CTU [2]University of Bristol [3]MBZUAI
tomas.soucek@cvut.cz
https://soczech.github.io/showhowto/

Figure 1. Given an input image (left) and ordered step-by-step textual instructions for a task (top), ShowHowTo generates an image sequence of visual instructions. Rows 1 and 2 demonstrate the generation of visual instructions for two recipes starting from the same input image. Rows 2 and 3 show the generation of visual instructions for the same recipe but conditioned on different input images. ShowHowTo generates scene-consistent (*e.g.*, consistency in the person and cutting board) and temporally consistent image sequences (*e.g.*, the bowl of tortilla chips or plate of chicken skewers) that faithfully capture the instructions (*e.g.*, cutting, frying, brushing, adding *etc.*).

Abstract

The goal of this work is to generate step-by-step visual instructions in the form of a sequence of images, given an input image that provides the scene context and the sequence of textual instructions. This is a challenging problem as it requires generating multi-step image sequences to achieve a complex goal while being grounded in a specific environment. Part of the challenge stems from the lack of large-scale training data for this problem. The contribution of this work is thus three-fold. First, we introduce an automatic approach for collecting large step-by-step visual instruction training data from instructional videos. We apply this approach to one million videos and create a large-scale, high-quality dataset of 0.6M sequences of image-text pairs. Second, we develop and train ShowHowTo, a video diffusion model capable of generating step-by-step visual instructions consistent with the provided input image. Third, we evaluate the generated image sequences across three dimensions of accuracy (step, scene, and task) and show our model achieves state-of-the-art results on all of them. Our code, dataset, and trained models are publicly available.

1. Introduction

With the immense success of large vision-language models and the rise of wearable devices, we rapidly approach the era of personalized visual assistants. This technology promises to help us in a variety of everyday tasks and numerous scenarios, such as preparing a Michelin-star dish, taking care of plants, or fixing a bicycle. Unlike generic instructional videos, visual assistants will provide guidance

[1]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.
[2]School of Computer Science, Machine Learning and Computer Vision (MaVi) Research Group, University of Bristol, UK.
[3]Mohamed bin Zayed University of Artificial Intelligence.

and feedback for specific environments and task variations.

Besides being useful for people, automated visual guidance has been recently explored and shown to be beneficial in robotics. For example, [9, 32, 43, 65] generate images of intermediate goals and use them as guidance for manipulation policies. Other recent methods [8, 19, 38, 57] derive robotics policies from videos specifically generated for target tasks and environments.

When comparing the ability to generate textual step-by-step instructions vs. generating visual instructions, one can see a sharp contrast. State-of-the-art LLMs can reliably provide personalized step-by-step text-only instructions. However, translating such instructions into images and videos still presents considerable challenges. This is because current video generation models, despite their impressive progress over the past years, focus on producing relatively short clips [14, 46, 49, 63] whereas image generation models only produce one image at a time.

Recent attempts to generate visual instructions either synthesize a single step [35, 36, 53] or are not contextualized to the user's specific environment [12, 41, 45]. In other words, such methods may generate plausible images for each step, however, such images will represent arbitrary settings and can feature tools or ingredients unavailable to the user. In robotics, temporally inconsistent guidance may imply physically implausible demonstrations, resulting in unsuccessful learning of policies. To address this issue, we focus on generating step-by-step visual instructions *conditioned on an input image* from the user, which we assume showcases their starting position—the environment, ingredients, tools, *etc*., as illustrated in Figure 1.

This paper makes the following contributions. (1) We introduce the problem of generating a sequence of visual instructions conditioned on an input image. (2) We introduce a fully automatic approach to collect step-by-step visual instruction training data from in-the-wild instructional videos, creating a large-scale, high-quality dataset of 0.6M step-by-step instruction sequences of 4.5M image-text pairs. (3) We train a video diffusion model capable of generating sparse step-by-step visual instructions consistent with the input image. (4) We evaluate our generated sequences across three aspects (step, scene, and task) and show our model achieves state-of-the-art results on all of them.

2. Related Work

Datasets of visual instructions. The scale and quality of the training data play a key role in visual instruction generation. Many available datasets combine instructional or egocentric videos and manual temporal annotation of individual steps [3, 51, 54, 70] or use professional illustrations [62]. Yet the requirement of manual annotations makes these sources of data hard to scale to novel tasks and environments. To alleviate the need for manual annotations, self-supervised methods have been developed to automatically obtain key steps from in-the-wild videos [21, 39, 52, 58, 59]. These key steps can then be used for visual instruction generation [53]. Recently, the improved capabilities of large language models [2, 4, 18, 20, 33] allowed for solely using video narrations to produce temporal captions, key steps, and instructions [36, 37, 50]. We build on these works to automatically obtain key steps from videos; however, in contrast to the related works, our dataset is constructed completely automatically, is composed of individual instruction frames instead of temporal intervals, and contains significantly fewer errors. Additionally, we focus on the entire domain of instructional videos, not only cooking.

Conditional video generation. Recently, diffusion models [28] have seen a surge in popularity for generative tasks, including video generation [40, 64]. Initial works extended a U-Net model using space-time factorization for the generation of videos in pixel space [29–31]. With the large cost of generating video, others [11, 15, 23, 27, 40, 64, 69] instead use an auto-encoder to model videos within the latent space, reducing significantly the number of parameters and memory requirements. Video generation models are often conditioned on textual prompts [16, 23, 24, 29, 30, 67], images that act as initial frames [60, 68], or both to generate a sequence of frames [7, 10, 11, 17, 25, 31, 34, 56, 69]. It has also been shown that for temporal consistency, a combination of both image conditioning and textual conditioning is important [55, 56]. However, these methods focus on relatively short video clips and are not able to generate long multi-step sequences of fine-grained instructions that take minutes to execute.

Generating step-by-step instructions. Both step-by-step visual instructions and continuous videos consist of sequences of frames, yet the instruction sequences differ from the videos significantly. While videos often contain only small pixel-level frame-to-frame changes [61], visual instruction sequences often contain large semantic (*e.g*., *raw* $\rightarrow$ *cooked*) and viewpoint (*e.g*., *inside* $\rightarrow$ *outside*) changes from one key frame to another [41]. Obtaining sufficient training data for visual instructions presents a significant challenge. Therefore, Phung *et al*. [45] generate step-by-step visual instructions using a pretrained text-conditioned image diffusion model with shared attention across steps to ensure consistency in the generated image sequences, while Menon *et al*. [41] use illustrations drawn by artists from WikiHow [62] as the training data for text-to-image-sequence generation. Other works using image-conditioned models can generate step-by-step sequences by iterative generation [12, 35, 36, 53]. In contrast, our method generates step-by-step visual instructions all at once, attending across steps to generate the full image sequence, including the input, which results in superior quality and consistency.

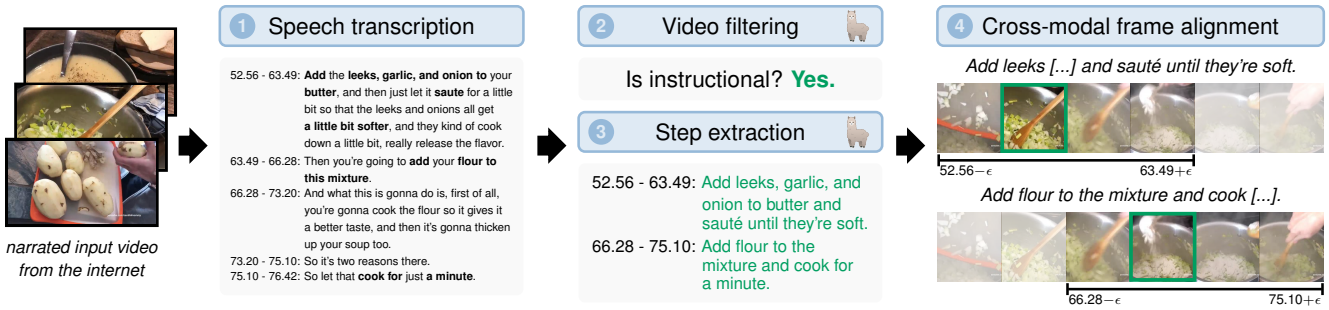

Figure 2. **Our automatic approach for creating the ShowHowTo dataset**—a large-scale instructional dataset consisting of step-by-step instruction sequences of image-text pairs to perform diverse HowTo tasks. Examples of step-by-step textual instructions and the corresponding frames are highlighted in green.

3. Building Large-Scale ShowHowTo Dataset

Learning to generate visual instructions requires a large-scale dataset that captures the rich diversity of real-world tasks and their step-by-step execution. However, manually creating such a dataset is prohibitively expensive and time-consuming, limiting the dataset's scale and coverage. We address this challenge by introducing an automated approach that leverages the natural alignment between narrations and visual demonstrations in instructional videos from the web to mine high-quality sequences of image-text pairs.

Using our proposed approach, we construct a large-scale dataset containing over half a million instruction sequences of image-text pairs spanning 25,026 diverse HowTo tasks. These sequences cover diverse domains including cooking (*e.g.*, *make strawberry crumb bars*, *prepare an avocado margarita*), home improvement (*e.g.*, *stain a cabinet*, *create a tire garden*), assembly (*e.g.*, *set up a 10×10 tent*, *tie a ring sling*), DIY crafts (*e.g.*, *make a bracelet*, *make a fairy glow jar*) and many more. We note that our data collection approach does not require *any* manual annotation, which is an important aspect to enable scaling.

3.1. Automatic Dataset Collection

Our approach takes as input a narrated instructional video for a specific task. First, it extracts a sequence of key steps in the form of concise, free-form textual instructions from the video's narration. Then, it associates each step with the corresponding keyframe in the video. The output is an ordered sequence of image-text pairs. This is a very challenging task due to the high level of noise, the possible misalignment of the narration and the visual content as well as the sheer variety of visual appearance and the spoken natural language in the input internet videos.

To tackle these challenges, we design a four-stage approach, illustrated in Figure 2: (1) The narration of the input internet instructional video is transcribed into sentences with corresponding timestamps. (2) The transcribed narration is verified to be instructional and removed if not. (3) The key instruction steps are extracted from the transcript along with their approximate temporal bounds. (4) A representative frame for each instruction step is selected through cross-modal alignment. By applying this approach to videos from HowTo100M [42], we obtain 578K high-quality sequences of image-text pairs with approximately eight steps per video on average.

Formally, we define our dataset as a collection of instruction sequences of image-text pairs. Each sequence $\{(I_i, \tau_i)\}_{i=0}^{n}$ represents an ordered set of steps required to accomplish a specific task $\mathcal{T}$. It consists of pairs of images I_i and the corresponding natural language instructions τ_i, with n denoting the number of steps in the sequence. Next, we describe the four stages in detail.

Speech transcription. Accurate transcription (ASR) of spoken narrations is a key strength of our approach, as these transcripts capture the instructor's step-by-step guidance that we use to align with the video. HowTo100M provides 1.2 million web instructional videos, but we forego the original transcripts generated using the YouTube API due to noise [26, 37]. Instead, we use WhisperX [6], a state-of-the-art speech recognition model, to obtain high-quality transcriptions with accurate timestamps from videos. We provide comparisons of the original transcripts and the improved ones in the supplementary material [1].

Filtering of irrelevant videos. We find that many HowTo100M videos are non-instructional, containing product reviews, vlogs, movie clips, *etc*. This noise may stem from the original data collection process, which relied on keyword-based web crawling and is susceptible to false positives due to inaccurate metadata. We leverage video transcripts as a strong signal for identifying instructional content and use a recent LLM (Llama 3.1 [20]) to filter the videos. We verify the reliability of this process through the evaluation on a labeled subset. Detailed analysis, qualitative results, and the prompts used for querying the LLM are provided in the supplementary material [1].

Step extraction. We observe that, in instructional videos, the key steps necessary to achieve a particular task are very

often mentioned in the narration, even if they are not well-aligned with what is shown in the frame [26]. Building on this, we prompt an LLM to extract the instructional steps from the narration transcripts in the format of WikiHow step-by-step guides, providing exemplars in the prompt to guide the extraction. Somewhat surprisingly, the LLM not only correctly extracts the key steps from the transcripts, but the model is also able to associate each step with the correct temporal intervals from the transcript, even if the step spans over multiple narrations. See Figure 2 for an example, and the supplementary material [1] for additional details and the prompt used.

Cross-modal frame alignment. For each instructional step, our goal is to identify a single representative frame that best demonstrates the instruction visually. While contrastive models [22, 47, 66] can align text instructions with frames, we observe that naive text matching across thousands of video frames leads to noisy results. Therefore, we limit the alignment to the identified instruction step temporal interval, expanded by $\epsilon = 15$ seconds to allow for some level of misalignment between the narrations and visual demonstrations [26]. Given these expanded intervals, we compute text-frame similarity scores using DFN-CLIP [22] and select the best alignment that satisfies the temporal ordering of the steps. We provide more details about the matching process in the supplementary material [1].

3.2. Dataset Statistics

In total, after filtering, the dataset contains 578K unique sequences of image-text pairs, with a total of 4.5M steps, averaging 7.7 ($\pm$ 2.8) steps per sequence, and 11.4 ($\pm$ 4.7) words per step. The sequence lengths vary from 1 to 26 steps, with 97.6% of sequences being 2 to 16 steps long. From task information provided by HowTo100M, the dataset contains instructions for 25K HowTo tasks across several categories, such as cooking, home and garden improvement, vehicles, personal care, health, and more. We include a comparative table to other datasets and further dataset analysis in the supplementary material [1].

4. ShowHowTo Model and Training Procedure

Given a user-provided image I_0, such as a photo of ingredients or tools on a table, our goal is to generate a sequence of images $\{\hat{I}_i\}_{i=1}^{n}$ of any length n based on the number of required steps, that guides the user to achieve an intended task $\mathcal{T}$, such as a cooked chicken tikka masala dish. Our goal is to generate images $\hat{I}_i$ to match the user-provided context, *i.e.*, to be grounded in the user's environment by utilizing the specific objects, tools, and workspace from the input image I_0. We achieve this goal by training a diffusion model conditioned on the input image I_0 along with the step-by-step textual instructions $\{\tau_i\}_{i=0}^{n}$ that fulfill the intended task $\mathcal{T}$ in any number of steps $1 \leq n \leq 15$.

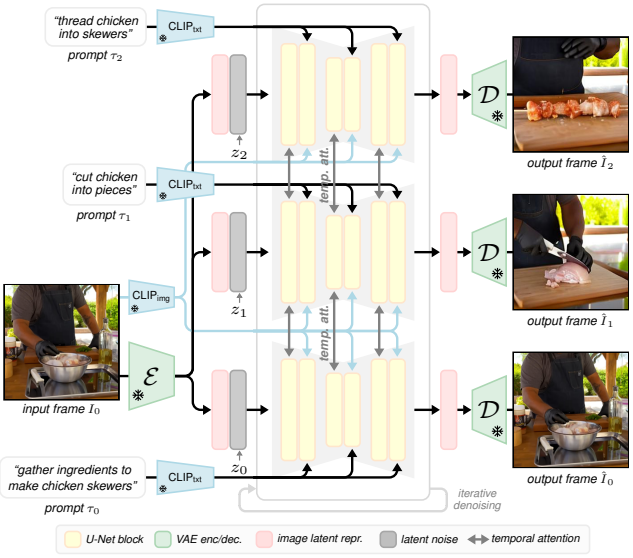

Figure 3. **Model architecture.** Given an input frame I_0 (left) and a variable number of text instructions τ_i describing each step, our diffusion model generates visual instructions $\hat{I}_i$ that correctly follow the prompts τ_i and are consistent with the input image I_0.

We build on recent progress in diffusion models for video generation [56]. However, there are the technical challenges of (a) how to inject the multi-step instruction guidance and (b) how to generate variable length sequences. We address these challenges in the next paragraphs.

Architecture. Our model, shown in Figure 3, is based on a latent video diffusion model [56] composed of a U-Net encoder and decoder, each with interleaving spatial and temporal attention layers. The input image I_0 is projected into the latent space via the VAE encoder $\mathcal{E}$, and is concatenated to each frame of the random noise z_i to form the model's input. The U-Net progressively denoises the input latent sequence while attending to all images in the sequence to ensure the generated images are temporally consistent and aligned with the input image. For better conditioning on the input image, the U-Net also contains cross-attention layers that attend to a feature representation of the input image directly. To guide the generation process to the desired visual instruction sequence, each frame i in the sequence attends to its prompt τ_i via cross-attention layers of the U-Net. In the ablations, we show that separate text conditioning for each frame in the sequence is instrumental for generating high-quality step-by-step visual instructions.

Training. We initialize the model from the pretrained checkpoint [56] trained on WebVid10M [5] for image animation and fine-tune the entire U-Net weights on our dataset. In contrast to training on videos, where the output video is commonly of a fixed length (*e.g.*, 16 frames in the case of [56]), step-by-step instructions have a variable se-

quence length. To ensure our model can generate variable-length sequences, we vary the sequence length during training by sampling from our sequences. For efficient computation, the length is varied over different batches but is kept the same across all samples in a single batch. If the dataset sequence is longer than the desired length k, we randomly sample the starting frame and use the next consecutive k frames as the model's target. We verify and further discuss these choices in Section 5.3 and provide the implementation details in the supplementary material [1].

5. Experiments

We first introduce our evaluation setup in Section 5.1. In Section 5.2, we compare our method to current state-of-the-art quantitatively on the test set and through a user study. Section 5.3 analyzes key designs of our method through ablation studies. Finally, Section 5.4, showcases qualitative results of our method. For implementation details and additional results, see the supplementary material [1].

5.1. Evaluation details

Dataset. We construct train and test splits from our dataset of 578K samples. Our test set comprises 3,964 sequences from 200 tasks covering the distribution over task categories in the full dataset. To ensure sample quality, we prioritize samples with high DFN-CLIP alignment scores as measured in our dataset creation pipeline (Section 3). For zero-shot evaluation of our method, we also use a random subset of 1442 non-illustrated instructional sequences[1] from the WikiHow-VGSI dataset [62] as an additional test set.

Evaluation metrics. We evaluate our model by measuring the correctness and consistency of the generated visual instructions using similar metrics as Menon *et al*. [41]. We describe the used metrics next with full details available in the supplementary material [1]. **(1) Step Faithfulness** [41] measures whether each generated image $\hat{I}_i$ correctly depicts its corresponding text instruction τ_i. It is computed as the zero-shot accuracy of the DFN-CLIP model where the generated image $\hat{I}_i$ is classified into classes $\{\tau_i\}_{i=0}^n$ of all text instructions of the sequence. **(2) Scene Consistency** measures whether the generated image $\hat{I}_i$ consistently captures the scene from the input image I_0 (*e.g*., the same utensils are used on the same kitchen countertop as in the input image). Intuitively, a generated image $\hat{I}_i$ is considered scene-consistent if it visually matches any frame from its source video $\{I_i\}_{i=1}^n$ (excluding the input image to avoid trivial copy solution). Therefore, for each generated image $\hat{I}_i$, the most similar image according to the DINOv2 [44] score is retrieved from the test set images. The metric then measures if the retrieved image is from the same video as the input.

[1] See the project website for the list of selected sequences.

Method	ShowHowTo			WikiHow [62]	
	Step Faithf.	Scene Consist.	Task Faithf.	Step Faithf.	Scene Consist.
(a) InstructPix2Pix [13]	0.25	0.17	0.25	0.32	0.12
(b) AURORA [35]	0.25	0.33	0.24	0.33	**0.15**
(c) GenHowTo [53]	0.49	0.13	0.27	0.60	0.06
(d) Phung *et al*. [45]	0.36	0.03	0.38	0.46	0.04
(e) StackedDiffusion [41]	0.43	0.02	**0.42**	0.57	0.07
(f) **ShowHowTo**	**0.52**	**0.34**	**0.42**	**0.72**	0.12
(g) *Random*	0.19	0.00	0.01	0.26	0.00
(h) Stable Diffusion [48]†	0.70	0.03	0.44	0.84	0.03
(i) *Copy of the input image*	0.19	0.62	0.39	0.26	0.26
(j) *Source sequences*	0.50	1.00	0.56	0.60	1.00

† Generation not conditioned on the input image.

Table 1. **Comparison with state-of-the-art on the ShowHowTo and the WikiHow datasets**. Out of all the visual instruction generation methods, our method best follows the input prompts while being consistent with the input image.

(3) Task Faithfulness measures how well the generated sequence $\{\hat{I}_i\}_{i=1}^n$ represents its intended task. It is measured as the zero-shot accuracy of the DFN-CLIP model where the generated sequence's averaged feature vector is classified into all 200 test set tasks. In contrast to Menon *et al*. [41], the generated sequences are evaluated holistically rather than per-step, as often steps are not unique to a task (*e.g*., "knead dough" step appears in both "Make sourdough bread" and "Make pizza" tasks), and the classification is done into all test set tasks rather than a small random subset, providing a more robust evaluation.

5.2. Comparison with the State-of-the-Art

Compared methods. We compare ShowHowTo to state-of-the-art methods for visual instruction generation as well as various baselines. For image-to-image methods **(a-c)**, we generate the visual instructions sequence by iteratively using the last generated image as the input for the next step generation to achieve temporal consistency. **(a) InstructPix2Pix** [13] is trained to manipulate input images according to a text prompt by training on synthetic paired image data. In contrast, **(b) AURORA** [35] is trained on a manually curated dataset of image pairs from videos, while **(c) GenHowTo** [53] extracts the image pairs for training from instructional videos automatically.

Methods that generate image sequences **(d-e)** do not accept an input image, therefore, we apply the common input masking approach, where the first denoised frame of the sequence is replaced by the noised ground truth frame in each step of the generation. The **(d) Phung *et al*.** [45] method generates consistent sequences of visual instructions by attending to all frames in the sequence in the spatial attention layers. As it is a training-free method, we reimplement it and use it with the Stable Diffusion backbone [48]. **(e) StackedDiffusion** [41] is trained on WikiHow illustrated image sequences. It generates the image sequence as a single tiled image. Similarly to the related work, we evaluate

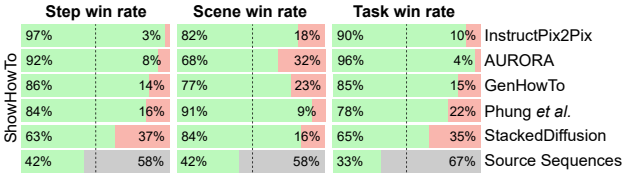

Figure 4. **User study results.** Win rates of the ShowHowTo method against baselines from pairwise forced decision user evaluations, divided into step, scene, and task. Values larger than 50% indicate ShowHowTo is preferred over the other methods (right).

all methods in zero-shot setup without finetuning.

Lastly, we show **(g) Random** lower bound and various naive baselines **(h-i)**. **(h) Stable Diffusion** [48] is a text-conditioned generative method with no input image conditioning that generates each image independently, the **(i) Copy** baseline uses the input image as the output for any prompt. As an upper limit, **(j) Source sequences** uses the original dataset frames corresponding to the text prompts.

Quantitative results. We show the results of different methods on our ShowHowTo test set as well as on a subset of WikiHow sequences [62] in Table 1. The methods trained to perform localized edits **(a-b)** generate outputs fairly consistent with the input image (see the Scene Consistency metric), yet they fail to properly capture the instructions described by the text prompts (evidenced by the Step Faithfulness metric). On the other hand, methods for generating sequences of visual instructions **(d-e)** model the instructions well according to the input text prompts, but they perform poorly in scene consistency. In contrast, our method **(f)** generates visual instructions that are consistent with the input scene and correctly capture the action specified by the prompt. Our method even generates images that are more faithful to the input textual instructions than the dataset sequences **(j)** (see the Step Faithfulness metric). There are two reasons for this: objects in real images can appear small or occluded, impacting CLIP matching, and sometimes steps do not appear visually in the video.

Additionally, in the supplementary material [1], we test our method in a zero-shot setup on the GenHowTo benchmark [53] and report additional quantitative metrics on the ShowHowTo dataset.

User study. We present a user study with 9 participants evaluating sequences from 100 randomly sampled tasks from our test set. Each participant compared 50 ShowHowTo sequences with baselines using three criteria: (1) Step Faithfulness (*Which sequence better follows the steps?*), (2) Scene Consistency (*Which sequence is more likely to come from the same video?*), and (3) Task Faithfulness (*Which sequence accurately depicts the instructions for the task of* `[task]`*?*). As shown in Figure 4, ShowHowTo outperforms all baselines. Notably, users preferred our generations over sequences from source videos in 42% of cases for both step and scene metrics, which may be attributed to instructional videos not showing good views of steps at times and the high quality of our generation. Lower task faithfulness scores against the source sequences suggest room for improvement in future methods. More details are in the supplementary material [1].

Text conditioning type	Step Faithf.	Scene Consist.	Task Faithf.	Average
1 prompt (concatenated step prompts)	0.21	0.29	0.38	0.29
1 prompt (summarized step prompts)	0.20	0.30	0.40	0.30
1 prompt per step (τ_0 = `'an image'`)	0.51	0.30	**0.42**	0.41
1 prompt per step (**ShowHowTo**)	**0.52**	**0.34**	**0.42**	**0.43**

Table 2. **Ablation of step conditioning**. The per-frame conditioning of ShowHowTo is instrumental in generating visual instructions faithful to the textual instructions.

Model training data	Step Faithf.	Scene Consist.	Task Faithf.	Average
WikiHow-VGSI [62]	**0.55**	0.12	0.30	0.32
HowToStep [37]	0.39	0.33	0.29	0.34
ShowHowTo (*food* videos only)	0.51	0.32	0.37	0.40
ShowHowTo	0.52	**0.34**	**0.42**	**0.43**

Table 3. **Ablation of the training data** as measured on the ShowHowTo test set. Our training dataset yields significant improvement over the manually curated WikiHow as well as the closely related HowToStep due to the quality of our instructions.

5.3. Ablations

We evaluate the key design decisions of our proposed method, *i.e.*, the model conditioning, training data, and variable sequence length training, in the next paragraphs. Furthermore, additional performance analysis of the trained model is available the supplementary material [1].

Text model conditioning. We evaluate how different types of text conditioning affect model performance. For video models, it is common to provide a single text prompt for conditioning. However, visual instructions vary substantially from one another, possibly requiring different conditioning. We construct a single prompt for each sequence by concatenating all step prompts and by summarizing the step prompts using an LLM [20]. In Table 2, we show that our choice of separate prompt per step significantly outperforms both of the single prompt variants. Additionally, we demonstrate that using the free-form step description for τ_0 outperforms the fixed prompt `'an image'`.

Training data. In Table 3, we analyze the impact of different training datasets by comparing our dataset with two related instructional datasets of similar scale and task coverage: (i) HowToStep [37], which contains automatically extracted video-text sequences from cooking videos, and (ii) WikiHow-VGSI [62], which consists of manually created image-text sequences from WikiHow articles, where the images primarily consist of digitally drawn illustrations. To train on the HowToStep dataset, we select the mid-

Training sequence length	Step Faithf.	Scene Consist.	Task Faithf.	Average
$\leq$ 4 steps	0.47	**0.39**	0.40	0.42
$\leq \leq$ 8 steps (**ShowHowTo**)	0.52	0.34	**0.42**	**0.43**
$\leq \leq$ 8 steps, randomly sampled	0.51	0.32	0.41	0.41
$=$ 8 steps	0.56	0.26	**0.42**	0.41
$\leq$ 16 steps	**0.57**	0.26	**0.42**	0.41

Table 4. **Ablation of the training sequence length** as measured on the ShowHowTo test set. We compare the performance of our model when trained on different sequence lengths.

dle frame of each video segment as the visual instruction frame. We observe significantly worse performance caused both by the lack of precise instruction frame information as well as very noisy video segments (*e.g.*, the dataset contains *'Thank you for watching!'* segments which are not instructional). The performance is also worse when compared to the model trained only on the *food*-related ShowHowTo sequences that are extracted from the very same videos as the HowToStep sequences, indicating a superiority of our sequence extraction process. Training on WikiHow-VGSI yields higher Step Faithfulness score, likely due to the quality of manually matched images and prompts. However, the overall performance remains significantly below our approach, as the model primarily learns from illustrated images, resulting in less consistent scene generation.

Variable sequence length. Instructions are often of variable length, therefore, one of the key model requirements is to support generating image sequences of different lengths. While attention-based architectures allow for any sequence length, the question is how to train such a model. We test training the model on variable sequences of up to 4 frames, 8 frames, and 16 frames. We also train the model on sequences of length 8 only. In Table 4, we show that training on shorter sequences up to 4 frames results in high Scene Consistency but low Step Correctness, while training on sequences up to 16 frames is the opposite. This can be attributed to the fact that short sequences often keep the same background across the whole sequence, encouraging the model to preserve the background at the expense of the prompt. Longer sequences, on the other hand, have more variation of the background, *e.g.*, as the task moves from the counter to the hob. The model is thus less likely to enforce the background during inference. Lastly, we also show that it is important to train always on consecutive sequences of visual instructions. If a subset of visual instructions from a video is sampled randomly (with the temporal ordering preserved), the scene consistency is decreased (Table 4, row 3).

5.4. Qualitative Results

Qualitative results in Figure 1, Figure 6, and additional figures in the supplementary material [1] demonstrate the key strengths of the ShowHowTo model. It consistently preserves the scene as well as various objects, tools, and in-

Figure 5. **Qualitative comparison** using the input image (left) and the textual instructions (top) for the task of *making a calzone*. The images from the source video are shown in the first row. Except for ShowHowTo, methods either struggle to preserve the input scene or to produce coherent steps.

gredients from the user-provided input image (*e.g.*, pot and vegetables in Figure 6, first row). It dynamically adjusts the viewpoint to emphasize key actions. Similarly to vanilla text-to-image and text-to-video models, our model can also introduce plausible task-relevant objects (*e.g.*, knives or bowls) if these objects are not present in the user-provided input image. Notably, the model effectively adapts human poses to demonstrate various object manipulations (*e.g.*, flower arranging in Figure 6, third row).

Figure 5 shows qualitative comparisons with related methods. Methods for generating instructional sequences (StackedDiffusion [41] and Phung *et al.* [45]) fail to preserve the scene from the input image. For example, they

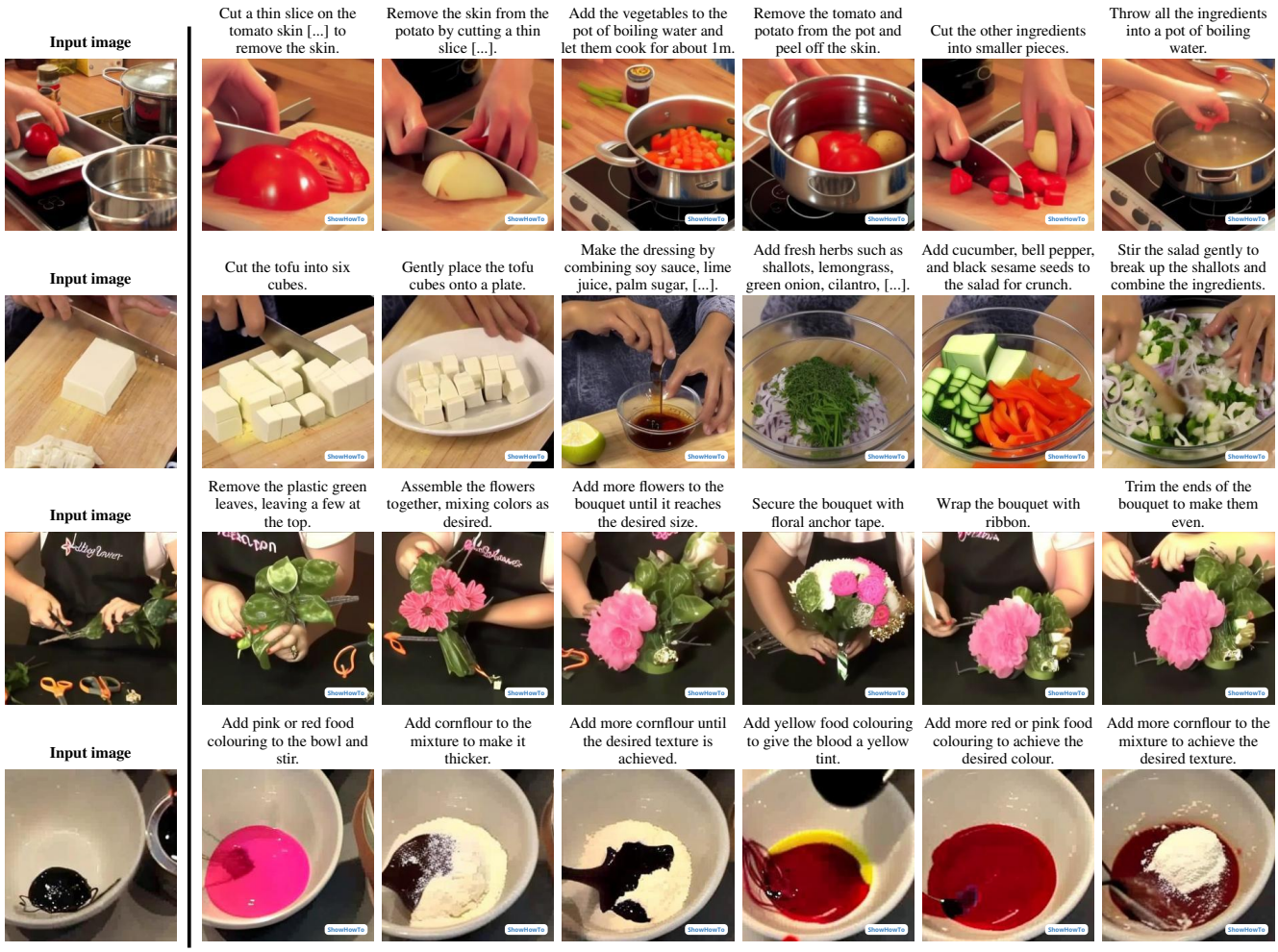

Figure 6. **Qualitative results of our method** for sequences from the test set. Given the input image (left) and the textual instructions (top), ShowHowTo generates step-by-step visual instructions while maintaining objects from the input image (*e.g.*, the cooking pot and the ceramic bowl in rows one and four) as well as among generated images (*e.g.*, glass bowl in the second row).

generate blue, black, or wooden kitchen countertop instead of the glossy white one from the input image. On the other hand, image-to-image approaches (GenHowTo [53], AURORA [35], and InstructPix2Pix [13]) perform minimal edits and propagate errors through the output image sequence due to iterative generation (*e.g.*, the persistent floating dough generated by the GenHowTo method).

Limitations. While our method can generate complex step-by-step visual instructions conditioned on the input image, it inherits the limitations of the models it is based on and introduces new limitations stemming from the novel source of training data. ShowHowTo model can struggle to maintain object states across many frames; for example, it can generate an image with raw meat after it was cooked in previous steps. Though the model often correctly generates common objects from instructional videos, for rare objects, such as electrical components, the model may generate objects in physically impossible configurations. Please see the supplementary material [1] for failure case examples.

6. Conclusion

This work explores, for the first time, generating environment-specific visual instructions to accomplish a user-defined task. We introduce a fully automated and scalable pipeline to create a dataset of 578K instructional image-text sequences from online videos, *without requiring any manual supervision*. Using this data, we train the ShowHowTo model to generate contextualized step-by-step visual instructions. Experiments demonstrate superior ability to generate accurate, scene-consistent instructional steps across various HowTo tasks, outperforming existing methods. We believe this work opens new avenues for personalized guidance in assistive technologies and step-by-step goal generation for robot planning.

Acknowledgements

We acknowledge VSB – Technical University of Ostrava, IT4Innovations National Supercomputing Center, Czech Republic, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (grant ID: 90254).

Research at the University of Bristol is supported by EPSRC UMPIRE (EP/T004991/1) and EPSRC PG Visual AI (EP/T028572/1). Prajwal Gatti is partially funded by an uncharitable donation from Adobe Research to the University of Bristol.

This research was co-funded by the European Union (ERC FRONTIER, No. 101097822 and ELIAS No. 101120237) and received the support of the EXA4MIND project, funded by the European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement N° 101092944. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Supplementary material (appendix) for the paper. https://arxiv.org/abs/2412.01987. 3, 4, 5, 6, 7, 8

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[3] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. *NeurIPS*, 2024. 2

[4] Anthropic. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku. https://www.anthropic.com/news/3-5-models-and-computer-use, 2024. 2

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 4

[6] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. Whisperx: Time-accurate speech transcription of long-form audio. *INTERSPEECH*, 2023. 3

[7] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024. 2

[8] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. 2

[9] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023. 2

[10] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[11] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 2

[12] João Bordalo, Vasco Ramos, Rodrigo Valério, Diogo Glória-Silva, Yonatan Bitton, Michal Yarom, Idan Szpektor, and Joao Magalhaes. Generating coherent sequences of visual illustrations for real-world manual tasks. *arXiv preprint arXiv:2405.10122*, 2024. 2

[13] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 5, 8

[14] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 2

[15] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2

[16] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024. 2

[17] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *ICLR*, 2023. 2

[18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 2023. 2

[19] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *NeurIPS*, 2023. 2

[20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2, 3, 6

[21] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *CVPR*, 2023. 2

[22] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023. 4

[23] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *ICCV*, 2023. 2

[24] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 2

[25] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2024. 2

[26] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *CVPR*, 2022. 3, 4

[27] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 2

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2

[29] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[30] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 2022. 2

[31] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *CVPR*, 2022. 2

[32] Xuhui Kang and Yen-Ling Kuo. Incorporating task progress knowledge for subgoal generation in robotic manipulation through image edits. *arXiv preprint arXiv:2410.11013*, 2024. 2

[33] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022. 2

[34] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023. 2

[35] Benno Krojer, Dheeraj Vattikonda, Luis Lara, Varun Jampani, Eva Portelance, Christopher Pal, and Siva Reddy. Learning action and reasoning-centric image editing from videos and simulations. *arXiv preprint arXiv:2407.03471*, 2024. 2, 5, 8

[36] Bolin Lai, Xiaoliang Dai, Lawrence Chen, Guan Pang, James M Rehg, and Miao Liu. Lego: Learning egocentric action frame generation via visual instruction tuning. *arXiv preprint arXiv:2312.03849*, 2023. 2

[37] Zeqian Li, Qirui Chen, Tengda Han, Ya Zhang, Yanfeng Wang, and Weidi Xie. Multi-sentence grounding for long-term instructional video. 2024. 2, 3, 6

[38] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024. 2

[39] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. In *ICCV*, 2023. 2

[40] Kangfu Mei and Vishal Patel. Vidm: Video implicit diffusion models. In *AAAI*, 2023. 2

[41] Sachit Menon, Ishan Misra, and Rohit Girdhar. Generating illustrated instructions. In *CVPR*, 2024. 2, 5, 7

[42] 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. In *ICCV*, 2019. 3

[43] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *NeurIPS*, 2018. 2

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024. 5

[45] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Coherent zero-shot visual instruction generation. *arXiv preprint arXiv:2406.04337*, 2024. 2, 5, 7

[46] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 5, 6

[49] RunwayML. Gen-3 alpha. 2024. 2

[50] Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. Howtocaption: Prompting llms to transform video annotations at scale. In *ECCV*, 2024. 2

[51] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. *NeurIPS*, 2024. 2

[52] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Multi-task learning of object states and state-modifying actions from web videos. *TPAMI*, 2024. 2

[53] Tomáš Souček, Dima Damen, Michael Wray, Ivan Laptev, and Josef Sivic. Genhowto: Learning to generate actions and state transformations from instructional videos. In *CVPR*, 2024. 2, 5, 6, 8

[54] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 2

[55] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *NeurIPS*, 2024. 2

[56] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*, 2024. 2, 4

[57] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. 2

[58] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *CVPR*, 2024. 2

[59] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *ICCV*, 2023. 2

[60] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 2023. 2

[61] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *ICLR*, 2024. 2

[62] Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. Visual goal-step inference using wikihow. In *EMNLP*, 2021. 2, 5, 6

[63] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

[64] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *CVPR*, 2023. 2

[65] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023. 2

[66] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 4

[67] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *IJCV*, 2024. 2

[68] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 2

[69] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 2

[70] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 2