This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Generating Illustrated Instructions

Sachit Menon[1,2*] Ishan Misra[1] Rohit Girdhar[1]
[1]GenAI, Meta [2]Columbia University
https://facebookresearch.github.io/IllustratedInstructions

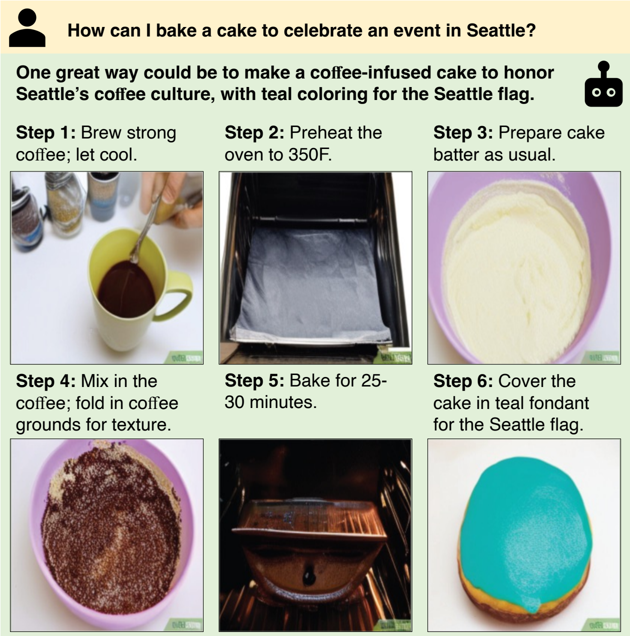
Abstract

We introduce a new task of generating "Illustrated Instructions", i.e. *visual instructions customized to a user's needs. We identify desiderata unique to this task, and formalize it through a suite of automatic and human evaluation metrics, designed to measure the validity, consistency, and efficacy of the generations. We combine the power of large language models (LLMs) together with strong text-to-image generation diffusion models to propose a simple approach called StackedDiffusion, which generates such illustrated instructions given text as input. The resulting model strongly outperforms baseline approaches and state-of-the-art multimodal LLMs; and in 30% of cases, users even prefer it to human-generated articles. Most notably, it enables various new and exciting applications far beyond what static articles on the web can provide, such as personalized instructions complete with intermediate steps and pictures in response to a user's individual situation.*

1. Introduction

The internet is a vast resource to find answers to all types of questions. It is often easy to find a webpage or a video that walks through the exact steps to achieve a user's goal. With the rise of Large Language Models (LLMs) trained on internet-scale data, users can just ask the LLM for instructions to achieve a goal. This allows the users to get answers for specific personalized queries for which there may not be an existing webpage on the internet, *e.g.*, modifying cooking recipes with user specific dietary restrictions. Moreover, if the users make a mistake when following the instructions, simple follow-up questions to the LLM can generate alternate instructions, a major advantage over static web search.

In spite of such advantages, LLMs still have one major limitation – they cannot generate visuals, which are critical for users to learn from and follow instructions for a wide range of tasks [6]. Consider, for instance, instructions that require visual inspection, *e.g.*, a recipe that requires the user to stir fry onions until golden brown, or searching for bubbles when a flat tire is submerged under water. An image accompanying such instructions can make following them significantly easier. Can we develop methods with the strengths of LLMs that can also generate such visuals?



How can I bake a cake to celebrate an event in Seattle?

One great way could be to make a coffee-infused cake to honor Seattle's coffee culture, with teal coloring for the Seattle flag.

Step 1: Brew strong coffee; let cool.

Step 2: Preheat the oven to 350F.

Step 3: Prepare cake batter as usual.

Step 4: Mix in the coffee; fold in coffee grounds for texture.

Step 5: Bake for 25-30 minutes.

Step 6: Cover the cake in teal fondant for the Seattle flag.

Figure 1. **StackedDiffusion generating Illustrated Instructions.** Given a goal (or any textual user input), StackedDiffusion produces a customized instructional article complete with illustrations that not only tells the user how to achieve the goal in words, but also shows the user by providing illustrations.

In this work, we tackle this challenge, developing models that can not only *tell* a user how to accomplish their task, but also *show* them how. We define the novel task of **Illustrated Instructions**: creating a set of steps with visualizations that showcase an approach to solve the user's task. We carefully consider the different dimensions of the problem, laying out three desiderata unique to this setting. To measure these desiderata, we develop automated metrics based on prior work in instructional article assessment [67] and image generation [50].

[*] Work done during an internship at Meta.

We propose a new model to solve the task of Illustrated Instructions that combines LLMs with a text-to-image diffusion model. We train our model on stacks of instructional images from websites such as WikiHow. The resulting **StackedDiffusion** model creates full instructional articles, complete with customized steps and a sequence of images uniquely generated to describe those steps. It leverages large-scale pretrained LLMs and finetuned text-to-image diffusion models, and employs techniques to accomplish the task without the need to introduce any new learnable parameters. This includes spatial tiling for simultaneous multi-image generation, text embedding concatenation to reduce information loss in long instructions, and a new "step-positional encoding" to better demarcate the different steps in the instructions. Thus, StackedDiffusion can generate useful illustrations even when trained even with limited instructional data (*cf*. Figure 7).

We compare StackedDiffusion with various baseline approaches based on off-the-shelf tools, and find that they all fall short, even when used in conjunction with one another. Existing T2I models are incapable of generating visuals directly from a user query. Even when given more detailed instructional text, we show that existing T2I models fail to produce images that are simultaneously faithful to the goal, the step, and consistent among each other. Given the recent introduction of multimodal LLMs [1, 28], we also compare with a recent open-source model, GILL [27], and show such models also fall short of consistently generating useful visuals along with text. We posit that our approach of leveraging the spatial priors learned by pretrained diffusion models to generate multiple images together, in conjunction with pretrained LLMs, is significantly more compute and data efficient than an approach purely based on next-token prediction, without these priors. Our thorough ablations and human evaluations show that StackedDiffusion convincingly surpasses state-of-the-art models. Our final model even outperforms *human-generated* articles in 30% of cases, showing the strong potential of our approach.

Contributions: 1) We introduce the novel task of Illustrated Instructions, which requires generating a sequence of images and text that together describe how to achieve a goal (§ 3 and §§ 5.1 and 5.2), along with desiderata and metrics for this task; 2) We propose a new approach StackedDiffusion for Illustrated Instructions, with novel modifications to the modeling procedure, enabling generation of visuals suitable for instructional articles for the first time without any additional parameters (§ 4); 3) We show that our proposed method achieves strong performance on all metrics, and confirm that human evaluators prefer it over existing methods by wide margins–even surpassing ground truth images in some cases (§§ 5.3 to 5.5); 4) Finally, we showcase new abilities that StackedDiffusion unlocks, including personalization, goal suggestion, and error correction, that go far beyond what is possible with fixed articles (§ 5.6).

2. Related Work

Instructional data, tasks, and methods. In the text domain, learning language models on WikiHow [30, 72], has led to advances in tasks such as summarization [71], commonsense procedural knowledge [72, 75], question answering [11], and hierarchical reasoning [74]. In particular, Zhang et al. [72] introduce the *goal inference* task, in which a model is presented a goal text as input and asked which of 4 candidate steps is one that actually helps achieve that goal, as well as the analogous *step inference* task. In the multimodal setting, Yang et al. [67] introduce the Visual Goal-Step Inference (VGSI) dataset and task, which consider articles of interleaved text and images. In this task, a model is again presented goal text as input but is asked which of 4 candidate images is one that actually helps achieve that goal. They show representations learned on this data aid in tasks related to instructional videos [39, 58].

Learning representations from multimodal data. Multimodal data (including multimodal instructional data, such as from video [39, 58]) has proven to be a powerful source of signal for tasks such as zero-shot recognition [25, 47], text-image and text-video retrieval [3, 9, 16, 17, 34, 36, 37, 40, 60], temporal segmentation [36, 46, 56, 65, 76], activity localization [9, 33, 36, 63, 65, 77], anticipation [12, 15, 19, 52, 61], question-answering [31, 53, 64], summarization [41, 42], and even recipe personalization [14].

Existing work on instructional data has centered around *understanding*, rather than generation. We instead focus on the novel setting of generating full multimodal articles complete with text and illustrations.

Generative models. Recent work has examined text-conditioned visual generation through autogressive [8, 69] or diffusion models [4, 18, 21, 43, 48, 49, 51]. These advances have been leveraged to create text and images together with purely autoregressive [1, 70] or combined approaches [27]. Concurrent work [2, 57] enables the capability of generating multiple images together in the autoregressive framework, but requires substantial additional parameters and focus on creating images that adhere solely to the nearby text rather than enabling consistency. (Similar issues arise in the text-to-video setting; see Appendix 7.) StackedDiffusion, on the other hand, leverages the priors built into the T2I model to obtain consistency without any additional parameters.

3. Illustrated Instructions

We now formalize our task and the corresponding desiderata. The input to the system is a goal text $\mathbf{g}$. As output, we would like to produce step text $\mathbf{s}_i$ as well as step illustration $\mathbf{I}_i$ for each step i. The step text should be a natural language

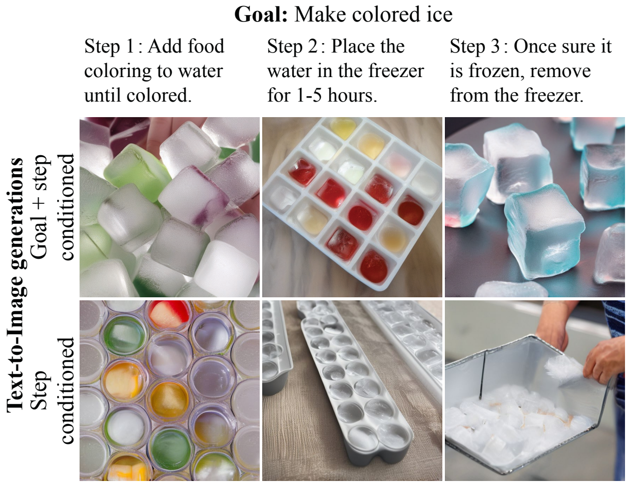

Goal: Make colored ice
Step 1 : Add food coloring to water until colored.
Step 2 : Place the water in the freezer for 1-5 hours.
Step 3 : Once sure it is frozen, remove from the freezer.
Text-to-Image generations
Goal + step conditioned
Step conditioned

Figure 2. **Failure modes of a naive approach.** A frozen T2I model is not able to capture both the goal and the step, showing only one or the other depending on how it is prompted. Further, it can not produce consistent images, leading to odd changes such as the color of the ice varying between images.

description of the step, while the step illustrations should be an image corresponding to the step.

On first glance, one might think that this task is a straightforward amalgamation of textual instruction generation and text to image generation. While these tasks are closely related and necessary substeps towards illustrated instructions, they are not sufficient. The naive approach–simply creating step text $\mathbf{s}_i$ from goal text $\mathbf{g}$, and each image $\mathbf{I}_i$ individually from each $\mathbf{s}_i$–fails to recognize the fresh new challenges that do not exist in either of these two other tasks. We identify three key requirements for useful illustrated illustrations: goal faithfulness, step faithfulness, and cross-image consistency.

The first requirement is clear: if the images do not relate to the goal, they cannot be good illustrations. This motivates the desiderata of *goal faithfulness*, which requires that each image faithfully reflects the goal text.

However, goal faithfulness alone is not enough. Consider the first row of images in Figure 2. The images all reflect the ultimate goal–creating colored ice–but fail in their role as step illustrations. This in turn motivates *step faithfulness*, which requires that each image be faithful to the step text. We see in this example that baseline T2I models fail dramatically along this metric; every image reflects the goal rather than the specifics of the step requested.

Finally, the generated images should be consistent with each other. While the second row of images in Figure 2 are all faithful to the step text, the color of the ice (and even the style of the images, cartoon or real) changes between images. This is jarring and confusing to the reader. This motivates the final criterion of *cross-image consistency*, which requires that each image be consistent with the other images produced for a particular generation.

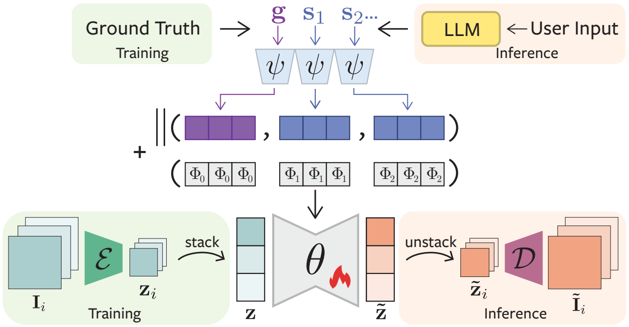


Figure 3. **Overview of StackedDiffusion.** At training time, we use the given goal and step text, and stack the encoded ground truth step-images. At inference time, we obtain the goal and step text from an LLM, and unstack denoised latents to produce the output images. See § 4 for details and notation.

4. StackedDiffusion

We propose a new architecture, StackedDiffusion, to overcome the limitations of existing text-to-image (T2I) approaches for the task of generating interleaved text and images for instructional tasks. StackedDiffusion builds upon T2I models based on latent diffusion models (LDMs) [49], which are diffusion models [48] that operate on a low-resolution 'latent' encoding of the original images. Our primary desiderata is that the images generated must be faithful to both the goal and step text. However, in initial experiments we found that simply encoding the goal and step texts joined together (as strings) leads to an uninformative encoding. For instance, generations conditioned on this combined text tended to ignore certain steps or the goal. Furthermore, the length of this combined text will likely exceed the context length limitations of the text encoders [47] commonly used in T2I models, leading to undesired truncation and information loss.

Hence, we elect to use a more general approach, applicable to text encoders with any context length. Rather than encoding the combined goal and step text to obtain the condition, we first separately encode the goal and step texts and then concatenate the encodings. This allows the model to learn to use the goal and step text independently, as well as in combination. We add a 'step-positional encoding' Φ_i to this concatenation. It is broadcast across the dimensions pertaining to a particular step, to indicate to the model where a step begins and ends. Φ_0 denotes the positional encoding reserved to indicate the goal embedding. Hence, given a text encoder ψ, and N steps in a given goal, we compute the overall text conditioning $\mathcal{C}$ as

$$\mathcal{C} = \Big|\Big| \left(\psi\left(\mathbf{g}\right) + \Phi_0, \ \Big|\Big|_{i=1}^{N} \psi\left(\mathbf{s}_i\right) + \Phi_i \right) \qquad (1)$$

where $||$ is the concatenation operation. This is shown in Figure 3 (top). This design decision is critical to obtain good goal and step faithfulness.

In addition, independently generating each image does not achieve the requirement of cross-image consistency as the model cannot exchange information across images. Thus, we generate all the $\mathbf{I}_i$ images at once, which allows the model to jointly generate the sequence and achieve cross-image consistency. The remaining question, of course, is how to accomplish this simultaneous generation such that it gives rise to cross-image consistency. Our key observation is that T2I models already have a strong prior for consistency within a single image. Could we make use of this previously-learned knowledge?

We propose a simple method to accomplish this: *spatial tiling*. As illustrated in Figure 3, the denoising U-Net is given latents $\mathbf{z}_i$ corresponding to each of the output images simultaneously, tiled spatially as if a single image. At training time, training images are encoded into the latent space as usual, then reshaped into the tiled format. In detail,

$$\mathbf{z}_i = \mathcal{E}\left(\mathbf{I}_i\right), \qquad \mathbf{z} = \mathop{\Big|\Big|}_{i=1}^{N} \mathbf{z}_i \tag{2}$$

$$L_{LDM} := \mathbb{E}_{\mathbf{z},\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}^{(t)}, t, \mathcal{C})\|_2^2\right] \tag{3}$$

where $\mathcal{E}$ is the encoder to map the image to the latent space (for instance, using a VAE [26]), L_{LDM} is the training objective for the LDM, t denotes a timestep of the diffusion process, ϵ denotes the noise added at a given timestep, and θ denotes the parameters of the learned denoising U-Net. At inference, we use classifier-free guidance [43] to generate a latent $\tilde{\mathbf{z}}$ given conditioning goal and N step texts. We split the latent into $\tilde{\mathbf{z}}_1, ..., \tilde{\mathbf{z}}_N$, and decode into generated images using a decoder $\mathcal{D}$ corresponding to the encoder $\mathcal{E}$, *i.e.* $\tilde{\mathbf{I}}_i = \mathcal{D}\left(\tilde{\mathbf{z}}_i\right)$. (We opt for tiling along a single spatial dimension for simplicity, and find that alternative tiling strategies do not significantly affect performance; see Appendix § 8.) We refer the reader to the Appendix § 9 and [49] for further details on LDMs. Given the stacked conditioning and generation operations, we refer to our final model as StackedDiffusion.

During training, goal and step text are obtained from ground truth, while at inference, a pretrained LLM is used to transform arbitrary user input text to an inferred goal and generated step texts. We describe the LLM inference procedure and prompt engineering in the Appendix § 10. We initialize the U-Net using a pretrained T2I model, and finetune all layers when training with the stacked input. Our stacked conditioning only increases the spatial resolution of the input to the U-Net for which can be modeled entirely using the existing parameters (spatial convolutions, attention) of the U-Net. We use the T2I model's text encoder (ψ), kept frozen, to encode the goal and step texts.

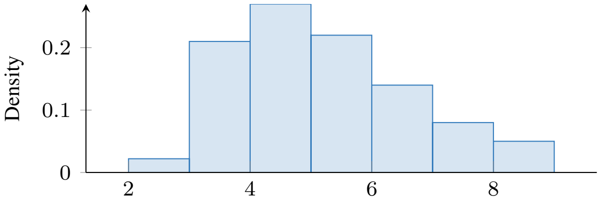


Figure 4. **Illustrated Instructions data.** The histogram shows the distribution of step counts in the data. We find that more than 80% of articles consist of 6 or fewer steps.

Since the spatially stacked latent has a spatial resolution comparable to the ones used for high resolution image generation, we encounter issues observed by prior work [32] for high resolution training. Specifically, [32] finds that high resolution latents retain substantial information about the input being noised with typical noise addition schedules, even at the final diffusion timestep $t = T$. This results in an unintentional distribution shift between train time (when all observed inputs contain signal) and test time (when the first inputs are pure noise with no signal). We address this by adjusting the training diffusion noise schedule such that the signal-to-noise ratio (SNR) at the final diffusion timestep T is zero [32]. This ensures enough noise is added during training so as to mitigate this difference between train and test time usage.

Implementation Details. We build upon a text-to-image latent diffusion model [49] trained on a large in-house image-text dataset. It leverages a VAE [26] to map the images to a 4D latent space, with a 8× reduction in spatial resolution. The U-Net largely follows the implementation from [49], with minor architectural modifications. As conditioning, it uses the Flan-T5-XXL text encoder [10]. We train StackedDiffusion for 14000 steps using the AdamW optimizer [35] with a learning rate of 10^{-4}, weight decay of 0.01, and gradient clipping at ℓ_2 norm of 1.0. For classifier-free guidance, we use a conditioning dropout rate of 0.05. We choose to generate at most $N = 6$ images simultaneously, because as we will see in § 5.1, the vast majority of the data available has 6 or fewer steps. For shorter training sequences, we pad with empty frames and dummy step texts, and for longer, we drop the extra steps and images. We ablate this choice of N in § 5.4. Hence in practice, for batch of 8 sets of 6 steps at 256px resolution, we get an encoded latent of $8 \times 6 \times 4 \times 32 \times 32$. We concatenate the latents spatially into $8 \times 4 \times (6 * 32) \times 32$. The denoising U-Net is finetuned to denoise these stacked images, leveraging what it has previously learned about spatial consistency, and adapts to this new setting. At inference time, steps are generated using a pretrained LLM [44], prompted to generate at most N steps (§ 10). To generate illustrations, noise is sampled in the shape of the tiled latent with N steps, and the steps generated by the LLM are padded with dummy steps if less

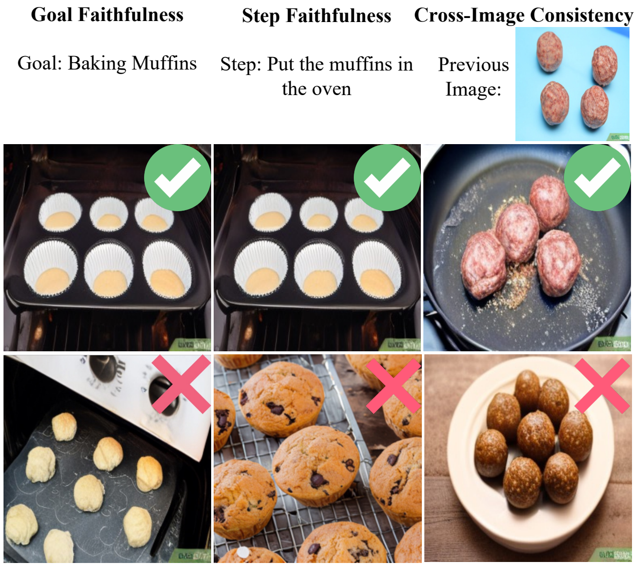


Figure 5. **Metrics.** We introduce three metrics, one for each desideratum presented in § 3. Goal faithfulness: the second image does not show muffins. Step faithfulness: the second image does not show the step. Cross-image consistency: the second image shows a different number of meatballs with different visuals.

than N. The resulting noise is denoised using the U-Net as usual, and finally reshaped to obtain the N output images upon decoding. See Appendix § 15 for further details.

5. Experiments

We train and evaluate StackedDiffusion on web-based instructional data. We compare the generations to multiple established baselines and ground truth, using automatic metrics and human evaluations. We now describe the data, metrics, baselines, and the key results. Finally, we demonstrate some new applications that StackedDiffusion enables, showing how it goes beyond standard instructional articles.

5.1. Illustrated Instructions Dataset

We introduce a new dataset to train and evaluate models for the Illustrated Instructions task, by repurposing the Visual Goal-Step Inference (VGSI) dataset [67]. VGSI consists of WikiHow articles, each of which have a high-level goal, 6-10 natural language steps (see Figure 4), and associated image illustrations. We observe that the same data can be used for generating instructional articles. It can provide signal for how the goal and step texts should map to output images, and how images should match with each other.

For evaluation, we construct a held-out set from this same data. Used directly, however, the data is not well suited as many of the tasks are too high-level to be useful for consistent illustration. For instance, “How to Start a Business in North Carolina” may have steps “Brainstorm ideas” and “Go to the courthouse” that have no shared visual content. As such, we filter the data for evaluation to the “Recipes” category, which always has a visually clear end state and where illustrations must be consistent with each other. Please see Appendix § 11 for more details.

	Human (↑)	GF (↑)	SF (↑)	CIC (↓)	FID (↓)
T2I (Frozen)	22.0	92.9	43.2	51.3	69.3
T2I (Finetuned)	33.3	**78.8**	52.4	51.5	53.5
StackedDiffusion	**(ref)**	74.3	**61.5**	**50.7**	**39.5**
Ground Truth	82.5	81.7	73.7	50.6	(N/A)

Table 1. **Comparison to baselines**. Human evaluation is reported as win rate vs our full StackedDiffusion model. GF corresponds to goal faithfulness accuracy, SF to step faithfulness accuracy, CIC to cross-image consistency, and FID to Fréchet Inception Distance.

5.2. Metrics

Having established three desiderata in § 3, we now turn to the question of how to evaluate them. In addition to evaluating the quality of images using FID [22], we propose three metrics, one for each desideratum, that can be used to evaluate the faithfulness and consistency of the generated article. Finally, given the limitations of automatic metrics for evaluating generative modeling tasks [55], we use human evaluations as our primary metric for overall quality. We illustrate the metrics in Figure 5 and briefly describe them next. Please see the Appendix § 12 for more details.

Goal Faithfulness (GF) measures how well the generated image is associated to the goal text. We evaluate this by constructing multiple-choice questions (MCQ) as in VGSI [67]. For each generated image, we compare its CLIP similarity [47] with the correct goal text vs the similarity with the texts of three other randomly selected goals. We compute the accuracy of the model in choosing the correct goal text.

Step Faithfulness (SF) measures how faithfully the generated image illustrates the step it aims to depict. An image should match the text for the step it was made for, more than other steps. We measure this using CLIP similarity and a MCQ task similar to goal faithfulness, where the image should have higher CLIP similarity with the corresponding step text than the other step texts within the same goal.

Cross-Image Consistency (CIC) evaluates how consistent the generated images for a goal are with each other and penalizes jarring inconsistencies across the images, such as objects changing color or number. For instance, if a particular set of ingredients are shown for “gather the ingredients,” then the same ingredients should be shown for “mix the dry ingredients.” We measure this by computing the average ℓ_2 distance between DINO [7] embeddings of the images for each step, as this considers the similarity of visual (rather than purely semantic) features. Like prior work [50], we

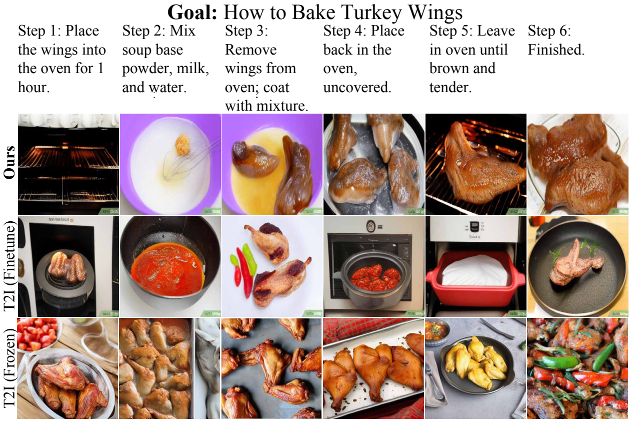

Goal: How to Bake Turkey Wings
Step 1: Place the wings into the oven for 1 hour.
Step 2: Mix soup base powder, milk, and water.
Step 3: Remove wings from oven; coat with mixture.
Step 4: Place back in the oven, uncovered.
Step 5: Leave in oven until brown and tender.
Step 6: Finished.
Ours
T2I (Finetune)
T2I (Frozen)

Figure 6. **Baseline comparison.** StackedDiffusion images are preferred overwhelmingly over baselines. The frozen baseline tends to only produce images showing the goal, while the finetuned baseline produces images that are more faithful to the step text, but have no visual features in common.

also found that CLIP features do not work well for image similarity as they are invariant to critical aspects such as color, number of objects, style *etc*. DINO's self-supervised pretraining objective leads to features that are sensitive to the visual aspects particular to an image, rather than being invariant to aspects not captured by category or language.

Human Evaluation is finally used to confirm that the automated evaluation corresponds to actual human preferences. We provide fully rendered articles from each model to human evaluators on Amazon Mechanical Turk (AMT), and ask them to choose which article they prefer, or if they are tied. We then consider the win rate as the proportion for which a majority of evaluators selected the given method compared to the articles produced by StackedDiffusion. A win rate of 50 denotes the methods are tied, whereas < 50 denotes the method is worse than StackedDiffusion.

5.3. Baselines

We now describe some baseline approaches based on existing state-of-the-art image generation models, and compare them to StackedDiffusion in Table 1.

T2I (frozen): The most obvious choice for a baseline is to simply use a pretrained and frozen text-to-image (T2I) model. We use our in-house T2I model that is also used to initialize StackedDiffusion. To ensure this baseline gets same information as StackedDiffusion, we prompt the model with a concatenation of the goal with each step, $\mathbf{g} \,||\, \mathbf{s}_i$, for each i, and produce a single image $\mathbf{I}_i$. The sequence then is composed of N independent generations.

We find that these images are faithful to the goal text, but fail to be faithful to the step text and lack consistency. The goal faithfulness is substantially higher than even the ground truth (92.9% vs 81.7%). This is because the model is not capable of reasoning about what a particular step towards a goal should look like. Instead, it simply generates an image that matches words in the input text, as illustrated in the first row of Figure 2. This results in all the images for *e.g.* "Make Colored Ice" showing the actual finished ice rather than any of the intermediate steps. In other words, the step text is eclipsed by the goal text.

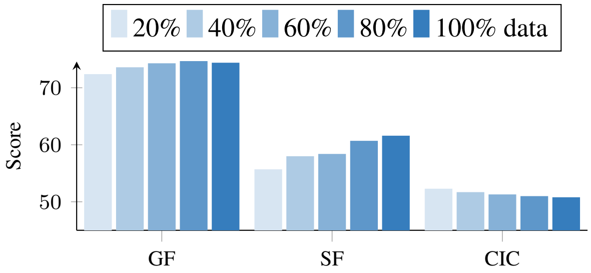


Figure 7. **Effect of training data.** We find that having more training data improves StackedDiffusion's faithfulness and consistency. However even with 20% data it performs well, thanks to its effective use of pretrained T2I model weights.

T2I (finetuned): We finetune a T2I model with the goal and step text embeddings concatenated together as the condition, again producing N independent generations. We find that this model is more faithful to the step text than the frozen T2I model, but still substantially less than our final model (52.4% vs 61.5%). This suggests the goal still overshadows the step text, but to a lesser extent. The goal faithfulness is also lower than for the frozen T2I model, as this model is trained to create images for steps rather than goals. The cross-image consistency is low, similar to the frozen model, as the images are still generated independently.

StackedDiffusion achieves substantially improved step faithfulness and cross-image consistency, while maintaining high goal faithfulness. We show an illustrative example in Figure 6. Unlike the baselines, the cross-image consistency is very close to that of the ground truth data; as generating them together allows for influence between them during the generation process. We also find that the FID of our generated images is substantially lower than that of other models, indicating that our generated images are more similar to the ground truth data. Finally on human evaluations, StackedDiffusion clearly outperforms them both by a preference of 78% and 66.6% respectively. Perhaps most notably, when compared to the ground truth images, human evaluators still picked StackedDiffusion 18.5% of the times, suggesting that our model generates some illustrations even better than the ground truth data. Further baselines can be found in Appendix § 13, with a comparison of training and inference costs in Appendix § 14.

5.4. Ablations

Step Count. As discussed in § 4, we elect to use a maximum step count of 6 as it allows sufficient data coverage

	Win Rate		Win Rate
T2I (Frozen)	**33.8**	T2I (Finetuned)	**34.0**
GILL [27]++	**3.9**	LLM + CLIP	**15.6**
Goal Retrieval	**33.1**	Ground Truth	70.0

Table 2. **System-level comparison to prior work.** Human evaluation results using fully generated outputs. Reported as human evaluation win rate of the associated method over StackedDiffusion (hence, higher is better for the reported method).

and most articles fall under this value. We find that human evaluators prefer this model to the shorter step count model of length 4. It is slightly preferred over the model trained with a longer step count of 8, likely as there is insufficient data of long lengths, limiting any advantage that might be gained in longer length generations.

Baseline	4 Step (↑)	8 Step (↑)
6 Step	32.1	46.9

Training data. In Figure 7, we randomly sub-sample varying proportions of the total data, and train StackedDiffusion on each of those subsets. As the amount of training data increases, it generates more faithful and consistent images. However even with less data it performs well, owing to its effective use of the pretrained T2I model's initialization.

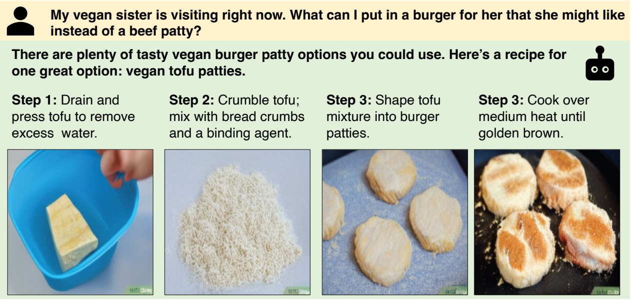
Importance of 0SNR. We experiment with the 0SNR technique [32] due to the high spatial dimensionality of the latents. We find that not using 0SNR results in substantially reduced step faithfulness, dropping from 61.5% to 52.8%.

Importance of Step-Positional Encoding. We also evaluate the importance of the step-positional encoding, Φ_i. This gives the model a critical cue for which parts of the condition correspond to which steps. We find that not using the step-positional encoding results in substantially reduced step faithfulness as well, dropping from 61.5% to 49.8%.

5.5. Comparison to prior work

The previous metrics were computed with respect to fixed ground truth goal and step texts to provide direct comparisons. However, they do not encompass the full scope of our model's capabilities: the flexibility of the generated text is one of StackedDiffusion's core strengths. To evaluate in a closer setting to this real-world usage, we examine the quality of the articles generated by the full system, where the steps are generated with a LLM [44]. We perform system-level comparisons using human evaluators, including the quality of the generated text and the quality of the images created from the generated text.

We show our results in Table 2. We first compare to the baselines introduced in § 5.3, and see results similar to Table 1. We see that even with LLM-generated text, StackedDiffusion outperforms baseline T2I based approaches, either frozen or finetuned, by more than 66% win-rate.



My vegan sister is visiting right now. What can I put in a burger for her that she might like instead of a beef patty?

There are plenty of tasty vegan burger patty options you could use. Here's a recipe for one great option: vegan tofu patties.

Step 1: Drain and press tofu to remove excess water.

Step 2: Crumble tofu; mix with bread crumbs and a binding agent.

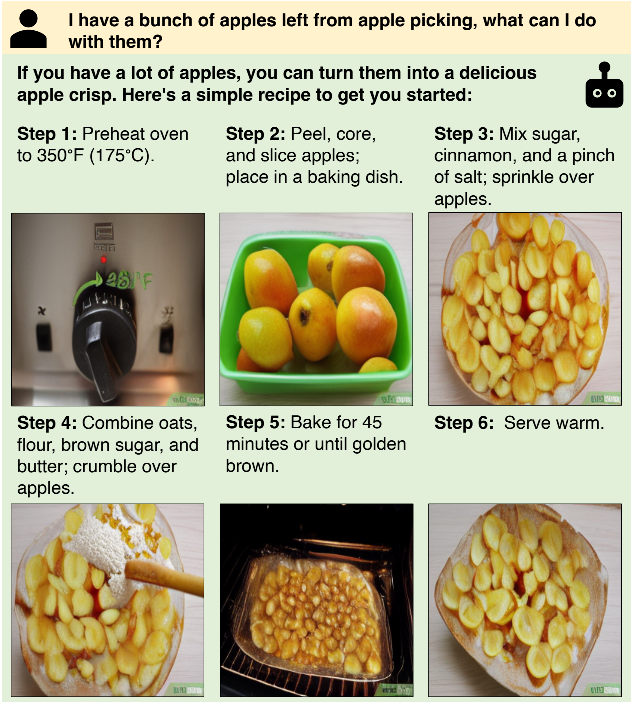
Step 3: Shape tofu mixture into burger patties.

Step 3: Cook over medium heat until golden brown.

Figure 8. **Personalization.** StackedDiffusion enables instructions personalized far beyond what is possible with preexisting articles.

Additionally, we compare StackedDiffusion to recent state-of-the-art approaches in multimodal generation, as in principle those can also produce instructional text combined with illustrations. Specifically, we compare to GILL [29]– a model trained to generate sequences of interleaved text and images. While other similar approaches have been proposed [1, 28], GILL is the best open-source model we could access. Using GILL directly, however, results in a human evaluation win rate of 0%. We posit this was for three primary reasons. First, the GILL model was unable to generate long enough generations to suffice as an article in comparison to our generations, despite our best efforts in modifying the generation parameters. Secondly, the model often did not produce any illustrations with the text, although we tried various prompting tactics. Thirdly, the text or images produced were often not of high enough quality to actually reflect the goal or steps. We thus introduce an alternative, which we call GILL++. This uses the (superior) text produced by the same language model as in our method, but passes each step text as input to GILL to generate the corresponding image. This results in a human evaluation win rate of 3.9% against StackedDiffusion. See Appendix § 16 for details on prompting GILL, and our GILL++ baseline.

Next, we compare StackedDiffusion to a retrieval based approach, denoted by LLM+CLIP. Here we retrieve the closest image in the training set according to CLIP similarity to each of the steps created by the LLM (with each concatenated to the goal text, similar to how the frozen T2I model is used). This retrieval-based metric is not suitable for the automated metrics previously introduced as it is based on the same underlying similarity scores as the automated metrics, but we introduce it here for human evaluation. We find that StackedDiffusion is preferred overwhelmingly over this retrieval-based approach, despite the retrieval-based approach using real images from the training set. Similarly, we compare to another retrieval-based approach which retrieves the full article with the most similar goal text to the input goal, denoted Goal Retrieval. We find StackedDiffusion strongly outperforms this as well. These results together suggest that StackedDiffusion is able



I have a bunch of apples left from apple picking, what can I do with them?

If you have a lot of apples, you can turn them into a delicious apple crisp. Here's a simple recipe to get you started:

Step 1: Preheat oven to 350°F (175°C).

Step 2: Peel, core, and slice apples; place in a baking dish.

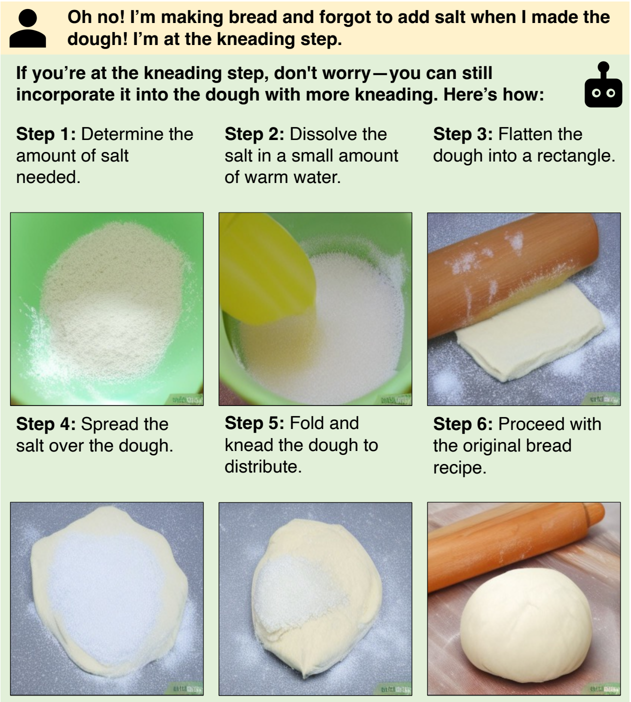
Step 3: Mix sugar, cinnamon, and a pinch of salt; sprinkle over apples.

Step 4: Combine oats, flour, brown sugar, and butter; crumble over apples.

Step 5: Bake for 45 minutes or until golden brown.

Step 6: Serve warm.

Figure 9. **Goal suggestion and inference.** StackedDiffusion can suggest what goal might be most relevant given other information.



Oh no! I'm making bread and forgot to add salt when I made the dough! I'm at the kneading step.

If you're at the kneading step, don't worry—you can still incorporate it into the dough with more kneading. Here's how:

Step 1: Determine the amount of salt needed.

Step 2: Dissolve the salt in a small amount of warm water.

Step 3: Flatten the dough into a rectangle.

Step 4: Spread the salt over the dough.

Step 5: Fold and knead the dough to distribute.

Step 6: Proceed with the original bread recipe.

Figure 10. **Error correction.** StackedDiffusion provides updated instructions in response to unexpected situations, like a user error.

to generate novel instructional articles that one can not simply retrieve from the training corpus.

Finally, we compare StackedDiffusion to the ground truth. Surprisingly again, we find that the human annotators pick our generations 30% of times even compared to ground truth. Note that these articles are handwritten and manually illustrated for the purpose of illustrating these goals. This shows the strong promise of our proposed approach. We believe with the rapid improvements in training data and generative modeling techniques, future iterations of our model could be indistinguishable or even better than manually created illustrated instructional articles.

5.6. Applications

Perhaps the biggest strength of a generative approach is its ability to handle unique user queries that may not fit into the boundaries posed by static articles on the web. Hence, we demonstrate the capabilities of the full system, combining the strengths of text generation capabilities from the language model with StackedDiffusion.

Personalized instruction. A user can provide any situational information specific to their circumstances and obtain an article customized to that situation. This is not possible with fixed, existing articles. For example, in Figure 8, the user can specify a diet and obtain an article for the food they want that adheres to that diet.

Goal suggestion. As StackedDiffusion accepts flexible input text, a user can provide higher-level information about what they want to do and obtain an article with a suggested specific goal that matches what the user would like to do. In Figure 9, the user describes their situation (having many apples) and StackedDiffusion suggests a goal that matches their situation (making apple crisps).

Error correction. Oftentimes, a user in the course of performing a task may make a mistake. Fixed articles provide no avenue for recourse. However, StackedDiffusion can adapt to user error and create alternative instructions that best correct for and accomodate the user's situation. See Figure 10 for an example of this. A fixed article would not provide any alternative paths that adapt as the user performs the action. This highlights the new avenues opened up by our generative approach to this task.

6. Limitations and Conclusions

While StackedDiffusion achieves strong performance in generating Illustrated Instructions, we note that there still remains a gap with ground truth data. Future work might examine how to obtain instructional data at greater scale and how this influences StackedDiffusion. Leveraging the latest T2I architectures to improve the faithfulness and quality of our generations would be another promising avenue for future work. Finally, leveraging improvements in text-to-video techniques to generate video clips describing each step would further improve the usability of such a system.

References

[1] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

[2] Emanuele Aiello, Lili Yu, Yixin Nie, Armen Aghajanyan, and Barlas Oguz. Jointly training large autoregressive multimodal models. In *ICLR*, 2024.

[3] 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] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[5] 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.

[6] Russell N. Carney and Joel R. Levin. Pictorial illustrations still improve students' learning from text. *Educational Psychology Review*, 14(1):5–26, 2002.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021.

[8] Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Patrick Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-To-Image Generation via Masked Generative Transformers. In *ICML*, 2023.

[9] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. In *ICCV*, 2021.

[10] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[11] Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. Joint Learning of Answer Selection and Answer Summary Generation in Community Question Answering. In *AAAI*, 2020.

[12] Eadom Dessalene, Chinmaya Devaraj, Michael Maynord, Cornelia Fermuller, and Yiannis Aloimonos. Forecasting action through contact representations from first person video. *PAMI*, 2021.

[13] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *ICCV*, 2023.

[14] Bahare Fatemi, Quentin Duval, Rohit Girdhar, Michal Drozdzal, and Adriana Romero-Soriano. Learning to substitute ingredients in recipes. *arXiv preprint arXiv:2302.07960*, 2023.

[15] Antonino Furnari and Giovanni Maria Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *PAMI*, 2020.

[16] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020.

[17] Valentin Gabeur, Arsha Nagrani, Chen Sun, Karteek Alahari, and Cordelia Schmid. Masking modalities for cross-modal video retrieval. In *WACV*, 2022.

[18] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022.

[19] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *ICCV*, 2021.

[20] 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.

[21] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[23] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv preprint arXiv:2210.02303*, 2022.

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

[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[26] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

[27] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models. In *NeurIPS*, 2023.

[28] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *ICML*, 2023.

[29] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding Language Models to Images for Multimodal Generation. In *NeurIPS*, 2023.

[30] Mahnaz Koupaee and William Yang Wang. WikiHow: A Large Scale Text Summarization Dataset. *arXiv preprint arXiv:1810.09305*, 2018.

[31] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020.

[32] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *arXiv preprint arXiv:2305.08891*, 2023.

[33] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning To Recognize Procedural Activities with Distant Supervision. In *CVPR*, 2022.

[34] Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang. Hit: Hierarchical transformer with momentum contrast for video-text retrieval. In *ICCV*, 2021.

[35] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019.

[36] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[37] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021.

[38] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023.

[39] 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.

[40] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *CVPR*, 2021.

[41] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *NIPS*, 2021.

[42] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. Tl; dw? summarizing instructional videos with task relevance and cross-modal saliency. In *ECCV*, 2022.

[43] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

[44] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[45] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

[46] AJ Piergiovanni, Anelia Angelova, Michael S Ryoo, and Irfan Essa. Unsupervised action segmentation for instructional videos. *arXiv preprint arXiv:2106.03738*, 2021.

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

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022.

[50] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *CVPR*, 2023.

[51] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *NIPS*, 2022.

[52] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *ICCV*, 2019.

[53] Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. Look before you speak: Visually contextualized utterances. In *CVPR*, 2021.

[54] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-Video Generation without Text-Video Data. *arXiv preprint arXiv:2209.14792*, 2022.

[55] George Stein, Jesse C Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, J Eric T Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *arXiv preprint arXiv:2306.04675*, 2023.

[56] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[57] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In *ICLR*, 2024.

[58] 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.

[59] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable Length Video Generation from Open Domain Textual Descriptions. In *ICLR*, 2022.

[60] Wenzhe Wang, Mengdan Zhang, Runnan Chen, Guanyu Cai, Penghao Zhou, Pai Peng, Xiaowei Guo, Jian Wu, and Xing Sun. Dig into multi-modal cues for video retrieval with hierarchical alignment. In *IJCAI*, 2021.

[61] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. *arXiv preprint arXiv:2201.08383*, 2022.

[62] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023.

[63] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *ICCV*, 2021.

[64] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021.

[65] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. In *ICCV*, 2021.

[66] Mengjiao Yang, Yilun Du, Bo Dai, Dale Schuurmans, Joshua B Tenenbaum, and Pieter Abbeel. Probabilistic adaptation of text-to-video models. *arXiv preprint arXiv:2306.01872*, 2023.

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

[68] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, Jianlong Fu, Gong Ming, Lijuan Wang, Zicheng Liu, Houqiang Li, and Nan Duan. NUWA-XL: Diffusion over Diffusion for eXtremely Long Video Generation. *arXiv preprint arXiv:2303.12346*, 2023.

[69] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[70] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multimodal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023.

[71] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *ICML*, 2019.

[72] Li Zhang, Qing Lyu, and Chris Callison-Burch. Reasoning about Goals, Steps, and Temporal Ordering with WikiHow. In *EMNLP*, 2020.

[73] 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.

[74] Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. Show Me More Details: Discovering Hierarchies of Procedures from Semi-structured Web Data. In *ACL*, 2022.

[75] Yilun Zhou, Julie Shah, and Steven Schockaert. Learning Household Task Knowledge from WikiHow Descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, 2019.

[76] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

[77] Dimitri Zhukov, Jean-Baptiste Alayrac, Ivan Laptev, and Josef Sivic. Learning actionness via long-range temporal order verification. In *ECCV*, 2020.