Generating Illustrated Instructions

Supplementary Material

7. Related Work: Text-to-Video Generation

Works in the text-to-video setting [5, 13, 23, 24, 38, 54, 62, 66, 73] are similar to ours in generating multiple images (frames) together, but are typically limited to short time scales, where visual content only changes in minor ways from frame to frame. Some recent works aim to generate longer videos [59, 68], however at a cost of substantial parameter overhead over their base T2I models. Many of these methods also use architectural design choices that are specific to this setting, where there is not substantial change from frame to frame, such as factorized spatial and temporal attention; this does not suit our setting, where there are much larger changes across consecutive images.

8. Tiling Comparison

While the attention operations in our architecture are capable of incorporating context across the full sequence irrespective of the stacking orientation, one might hypothesize that convolutional operations may not have a wide enough receptive field to obtain the same results with different tiling strategies. We experiment with a closer to square orientation for comparison to evaluate this hypothesis. Specifically, we compare to 8x6x4x32x32 → 8x4x(3x32)x(2x32) stacking (closer to square), and find the results are indeed similar. In particular, human evaluation gives a win rate of 45.9% for the (3x2) orientation, near even; GF is slightly higher (79.3), SF is slightly lower (60.7), and CIC is marginally higher (50.5). This suggests that the tiling strategy does not have a substantial impact on the model's performance.

9. Latent Diffusion Models Preliminaries

Latent diffusion models [49] model a data distribution in the latent space of another model, such as a VAE [26]. The latent diffusion model is trained to generate samples from the data distribution $p(x)$ in the latent space, induced by the encoder $\mathcal{E}$, via gradual denoising. These samples can then be decoded by the VAE decoder $\mathcal{D}$ to produce samples from the approximated data distribution. The objective matches the form of our Eq 3.

10. LLM inference prompt engineering

We find that simple prompting strategies suffice to have the LLM produce outputs in the format we want. Specifically, we use the following prompt:

```
{INPUT_TEXT} Write your response in the
 form of a goal "Goal: {GOAL}" followed
 by concise numbered headline steps,
each one line, without any other text.
Use at most 6 steps.
```

where {INPUT_TEXT} is any user input, such as those shown in the figures. If only a goal is provided, we provide the input text as "How can I {GOAL}?", for example "How can I make colored ice?".

11. Data

We repurpose the VGSI dataset [67] for generative purposes. The original dataset is split by step-image pairs. We note that this splits images from a single goal across training and validation. We thus recombine the data, group by associated goal, and then create new splits. Each data point we create is formed by a string of goal text as well as varying numbers of pairs of step text strings and associated images. The distribution of the number of steps is seen in Figure Figure 4. We find that the overwhelming majority of articles contain less than 6 steps. The training set consists of 95328 goals for a total of 476053 step-image pairs. The validation set, after filtering only for goals in the 'Recipe' category, consists of 1711 goals for 9473 step-image pairs.

12. Metrics

Goal Faithfulness (GF). We draw on the literature from Visual Goal-Step Inference to define a metric for goal faithfulness. The intuition is as follows. An image should, typically, be more associated with the text for the goal it was made for than for the text of other goals. We can measure this association by computing the cosine similarity between the image and the goal text with pretrained contrastive vision-language models such as CLIP [47]. However, CLIP scores are notably miscalibrated when comparing across different image-text pairs. Thus, to make the metric meaningful, we must compare the similarity scores to the similarity with other goals. We accomplish this by constructing multiple-choice questions as in VGSI [67]. For each image, we compare the CLIP similarity score with the correct goal text to the CLIP similarity scores with the texts of three other randomly selected goals. We then compute the accuracy of the model in choosing the correct goal text. While VGSI uses this metric with a fixed dataset to evaluate their vision-language models, we instead fix the vision-language model used in order to evaluate the data being generated.

Step Faithfulness (SF). As mentioned in § 3, no matter how well an image reflects the overall goal, it is useless if it does not illustrate the step it serves to illustrate. Inspired by Goal Faithfulness, we can define a similar metric for step faithfulness. An image should be more similar to the text for the step it was made for than for other steps. In particular, it should be more similar to the text than other steps with the same goal. It is worth noting that the step faithfulness metric varies in magnitude depending on the number of steps (N) in the sequence. As the number of steps increases, the chance that any individual step will have a visual that is not strongly associated with the caption increases, resulting in lower values. However, for a fixed N, comparison between models is meaningful.

Cross-Image Consistency (CIC). Finally, a key component of the Illustrated Instructions task is that a set of images is generated rather than a single image. These images are not simply independent. If a particular set of ingredients is shown for "gather the ingredients," then the same ingredients should be shown for "mix the dry ingredients." More generally, we would like to avoid jarring inconsistencies between images in the same sequence, such as objects changing color or number, etc. We can measure this by computing the DINO similarity between the images for each step. We use DINO [7] as the measure of visual similarity over CLIP as it reflects *visual* over *semantic* features, which are more relevant for consistency across images. This has been noted in work for personalized image generation, such as DreamBooth [50]. CLIP features, in particular, tend to be invariant to the aspects of the image that are most relevant for consistency, such as color, number of objects, style, and more.

FID. We compute FID with respect to the ground truth dataset – specifically, the 9473 images of the validation set are used as the reference distribution. We use the clean-fid [45] to avoid common pitfalls in FID computation.

Human Evaluation. We show the outputs of each model being compared in a two-way comparison side by side, as shown in Figure 11. For human evaluation, we select a random subset of 140 goals that are fixed across evaluations from the validation set. We ask annotators to select which of the two articles they prefer (or if they are tied). As in previous work [20], we find that providing criteria for evaluators to consider during evaluation and requiring a justification for the ultimate decision leads to substantially higher quality evaluations. We use 3 annotators per comparison. To ensure quality of annotation, we selected annotators with the Amazon Mechanical Turk "Master" qualification that had a $> 95\%$ approval rate with at least 1000 prior tasks completed. For each method, we consider how many goals had a strict majority of annotators pick that method's generations. Tied cases are removed prior to win rate computation. We report the percentage of majority wins for each method. We find that on average 80% of annotators agreed with the decision for a given sample.

13. Additional Diffusion Baselines

We include comparisons with two additional baselines here for completeness: a model using a joint encoding for the goal and step text, and a model that only uses the step text as conditioning.

The goal+step joint-encoding ablation obtains high GF (91.6), similar to the frozen model that also operates on a joint-encoding, reflecting that the goal text dominates the embedding. Finetuning, however, improves SF (50.2) over the frozen model. The CIC increases slightly (51.6) as the influence of the goal information on all images is reduced.

The step-only baseline, as expected, yields much lower GF (59.5) as it does not have the goal information; but much higher SF (77.2). The cross-image consistency (CIC) worsens substantially (53.1) as there is no shared goal information to tie the steps together. This tradeoff leads to overall similar performance to the separate-encoding model in human evals, strongly under-performing our final model.

14. Training/Inference Cost

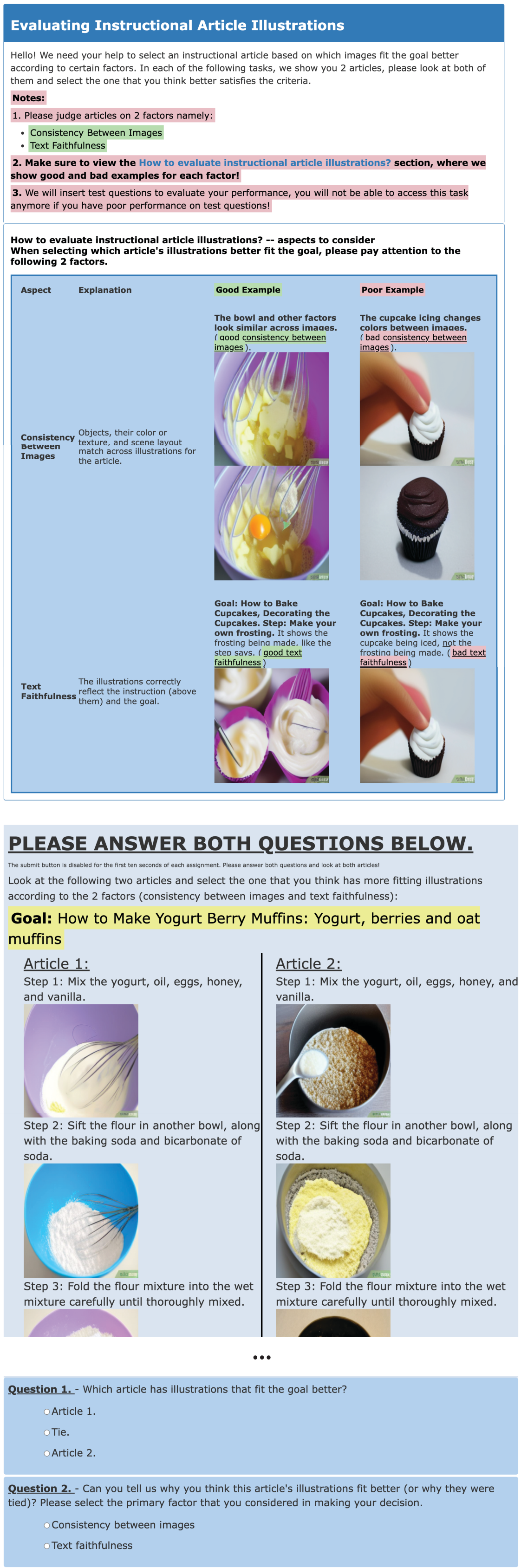
StackedDiffusion needs only 1 node with 8 GPUs to train, making it easily accessible and reproducible for the broader research community. Spatial tiling allows every step to be generated in parallel, maximally utilizing the GPU computation power. Hence, in spite of attending to features for all steps at once, StackedDiffusion takes about the same inference time as a single-step model that requires N forward passes to generate all the steps. To train to convergence, StackedDiffusion took 9.7 hours, only marginally higher than single-step finetuning (7 hours) on 8 A100 GPUs.

15. StackedDiffusion implementation Details

The U-Net used in the in-house model we build on largely follows the architecture used in [49], with a T5-XXL text encoder. The output channels of each block are changed to $(320, 640, 1280, 1280)$. The GroupNorm epsilon value is 10^{-5}. The input text condition uses additional projection and attention layers before being fed into the U-Net as in [49]. Specifically, this consists of an attention block with 4096 dimensions followed by layer normalization, a linear projection from 4096 to 1280 dimensions, and another layer normalization, and a final linear layer from 1280 to 1280 dimensions.

16. GILL Comparison Details

We first tried to use GILL [29] directly with the default settings other than the maximum number of output tokens being set to 512 so as to be an appropriate length for a full article. We enable both generation and retrieval for GILL.



Evaluating Instructional Article Illustrations

Hello! We need your help to select an instructional article based on which images fit the goal better according to certain factors. In each of the following tasks, we show you 2 articles, please look at both of them and select the one that you think better satisfies the criteria.

Notes:

1. Please judge articles on 2 factors namely:

- Consistency Between Images
- Text Faithfulness

2. Make sure to view the How to evaluate instructional article illustrations? section, where we show good and bad examples for each factor!

3. We will insert test questions to evaluate your performance, you will not be able to access this task anymore if you have poor performance on test questions!

How to evaluate instructional article illustrations? -- aspects to consider
When selecting which article's illustrations better fit the goal, please pay attention to the following 2 factors.

Aspect	Explanation	Good Example	Poor Example
Consistency Between Images	Objects, their color or texture, and scene layout match across illustrations for the article.	**The bowl and other factors look similar across images.** (good consistency between images).	**The cupcake icing changes colors between images.** (bad consistency between images).
Text Faithfulness	The illustrations correctly reflect the instruction (above them) and the goal.	**Goal: How to Bake Cupcakes, Decorating the Cupcakes. Step: Make your own frosting.** It shows the frosting being made, like the step says. (good text faithfulness)	**Goal: How to Bake Cupcakes, Decorating the Cupcakes. Step: Make your own frosting.** It shows the cupcake being iced, not the frosting being made. (bad text faithfulness)

PLEASE ANSWER BOTH QUESTIONS BELOW.

The submit button is disabled for the first ten seconds of each assignment. Please answer both questions and look at both articles!

Look at the following two articles and select the one that you think has more fitting illustrations according to the 2 factors (consistency between images and text faithfulness):

Goal: How to Make Yogurt Berry Muffins: Yogurt, berries and oat muffins

Article 1:

Step 1: Mix the yogurt, oil, eggs, honey, and vanilla.

Step 2: Sift the flour in another bowl, along with the baking soda and bicarbonate of soda.

Step 3: Fold the flour mixture into the wet mixture carefully until thoroughly mixed.

Article 2:

Step 1: Mix the yogurt, oil, eggs, honey, and vanilla.

Step 2: Sift the flour in another bowl, along with the baking soda and bicarbonate of soda.

Step 3: Fold the flour mixture into the wet mixture carefully until thoroughly mixed.

•••

Question 1. - Which article has illustrations that fit the goal better?

○ Article 1.
○ Tie.
○ Article 2.

Question 2. - Can you tell us why you think this article's illustrations fit better (or why they were tied)? Please select the primary factor that you considered in making your decision.

○ Consistency between images
○ Text faithfulness

Figure 11. **Human evaluation setup.**

We experimented with prompts such as “Please write me a step-by-step article for the goal GOAL” and many variants of it. Despite this, we found that the generations were always substantially shorter than a full article, likely due in part to the short length of the training data GILL has seen. We experimented with different values for the temperature as well as other parameters, but were unable to produce longer outputs that would adequately illustrate a goal. The text quality of these outputs was also very low. For instance, when prompted about the goal “How to Make Apple Pie,” GILL responded “I can’t imagine why you would want to do this.”. Typically, the outputs would comprise text for a single step or a single image, and we did not observe any instances of multiple.

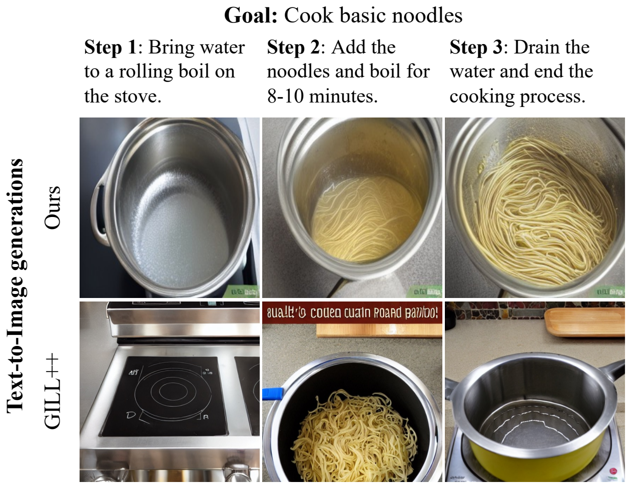

Goal: Cook basic noodles
Step 1: Bring water to a rolling boil on the stove.
Step 2: Add the noodles and boil for 8-10 minutes.
Step 3: Drain the water and end the cooking process.
Text-to-Image generations
Ours
GILL++

Figure 12. **Comparison with our GILL++ .**

Because of this, we elected to try to surmount the language modeling difficulties of GILL and see how well it can perform when given text from the same LLM used for our experiments, which we call “GILL++”. This would evaluate the image generation and retrieval capabilities of GILL on their own. (To some extent, this does defeat the purpose of a multimodal generative model; it reflects the limitations of current generative models.) We performed LLM inference as described in Appendix § 10. We provided the generated text to GILL with the prompt “Please illustrate the step: {STEP} for the goal: {GOAL}”. We also experimented with various values of the “sf” (scaling factor) inference parameter, which the GILL codebase comments: “increase to increase the chance of returning an image”. However, the model did not always produce an image even when increasing this value. The eventual parameters we arrived at for GILL++ were a scaling factor of 1.4, temperature of 0.6, *top_p* of 0.95, and all other parameters left the same. A qualitative comparison can be seen in Figure 12. We find that the image generations still remain inferior to ours even when the text is fixed.

17. Retrieval Baselines

We implement two separate retrieval-based baselines that use real images for the illustrations of each step. We describe these both below.

17.1. LLM + CLIP

The LLM+CLIP baseline is a combination of text generated by an LLM and real images retrieved from that text. We prompt the LLM the same way as for the final StackedDiffusion model. After obtaining these steps, we retrieve the top image from the training dataset using CLIP similarity. For each step, we rank all images in the training set by their CLIP similarity to that step’s text, and use the most similar image for each. Each of these images can come from any disparate goal in the training set.

17.2. Goal Retrieval

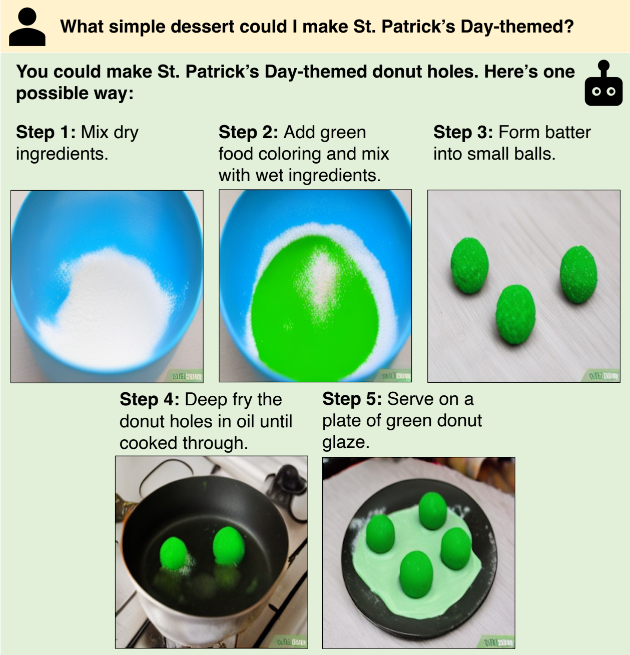
For the goal retrieval baseline, we pick the closest full article from the training set to compare to. The intent of this baseline is to show that the model does not simply reproduce the closest article it saw during training but rather generates new articles. We use the CLIP text similarity compare the generated goal text to the training set. We then pick the closest article to the generated goal text. We use the steps from this article as the steps for the generated article. This baseline is similar to the LLM + CLIP baseline, but instead of using the LLM to generate the steps, we use the steps from the closest article in the training set and simply use the images associated with those ground-truth steps.

18. Additional Qualitative Results

We present additional qualitative results in Figure 13, Figure 14, Figure 15, Figure 16, Figure 17, Figure 18, Figure 19, and Figure 20. In particular, we note that Figure 20 shows that StackedDiffusion can generate articles outside of the recipe domain, such as for furniture repair.

19. Failure Cases

In addition, we present examples of failure cases in Figs. 21 and 22. We observe one failure mode in being *over-consistent*: while Step 3 says to mix in ‘another’ bowl, the model ends up producing a bowl that looks similar to the bowl in Step 2. We also observe a case where the model finds consistency difficult to produce: geometric arrangements seem to be considered similarly, making a 3x3 grid of cupcakes into a 2x4 grid.



What simple dessert could I make St. Patrick's Day-themed?

You could make St. Patrick's Day-themed donut holes. Here's one possible way:

Step 1: Mix dry ingredients.

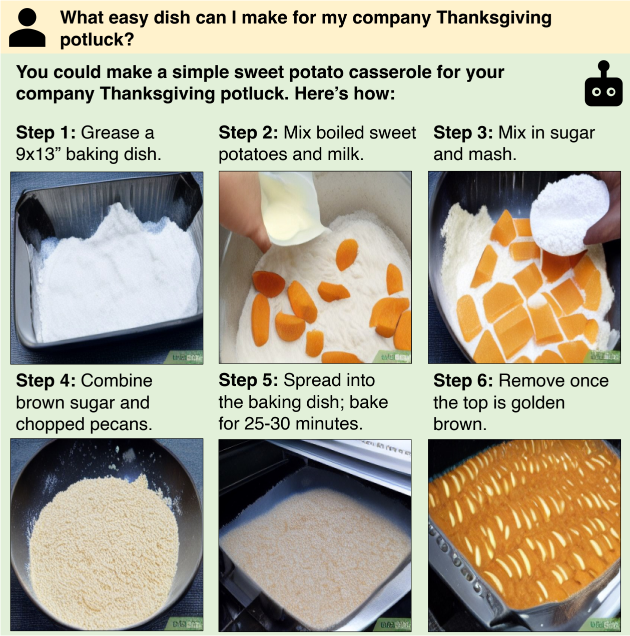
Step 2: Add green food coloring and mix with wet ingredients.

Step 3: Form batter into small balls.

Step 4: Deep fry the donut holes in oil until cooked through.

Step 5: Serve on a plate of green donut glaze.

Figure 13. **Additional personalization.** The steps illustrate donut holes customized to be green for St. Patrick's day.



What easy dish can I make for my company Thanksgiving potluck?

You could make a simple sweet potato casserole for your company Thanksgiving potluck. Here's how:

Step 1: Grease a 9x13" baking dish.

Step 2: Mix boiled sweet potatoes and milk.

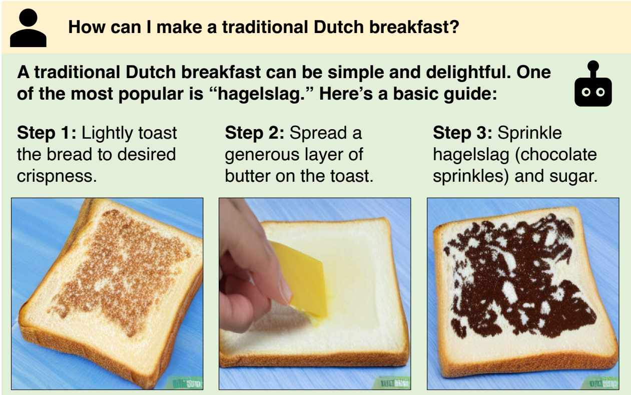
Step 3: Mix in sugar and mash.

Step 4: Combine brown sugar and chopped pecans.

Step 5: Spread into the baking dish; bake for 25-30 minutes.

Step 6: Remove once the top is golden brown.

Figure 14. **Additional goal suggestion.** StackedDiffusion suggests the goal of making a casserole and provides illustrated instructions for it.



How can I make a traditional Dutch breakfast?

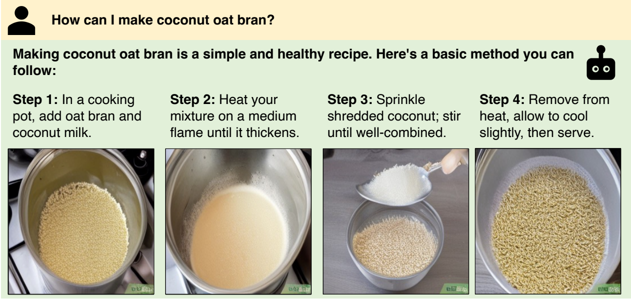
A traditional Dutch breakfast can be simple and delightful. One of the most popular is "hagelslag." Here's a basic guide:

Step 1: Lightly toast the bread to desired crispness.

Step 2: Spread a generous layer of butter on the toast.

Step 3: Sprinkle hagelslag (chocolate sprinkles) and sugar.

Figure 15. **Additional goal suggestion.** The world knowledge of the LLM allows it to inform the user about hagelslag.



How can I make coconut oat bran?

Making coconut oat bran is a simple and healthy recipe. Here's a basic method you can follow:

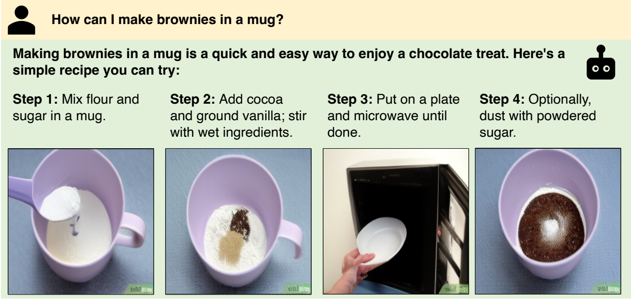
Step 1: In a cooking pot, add oat bran and coconut milk.

Step 2: Heat your mixture on a medium flame until it thickens.

Step 3: Sprinkle shredded coconut; stir until well-combined.

Step 4: Remove from heat, allow to cool slightly, then serve.

Figure 16. **Additional article generation.** Coconut oat bran is a variant on an existing dish (not found in the training set) that StackedDiffusion can illustrate.



How can I make brownies in a mug?

Making brownies in a mug is a quick and easy way to enjoy a chocolate treat. Here's a simple recipe you can try:

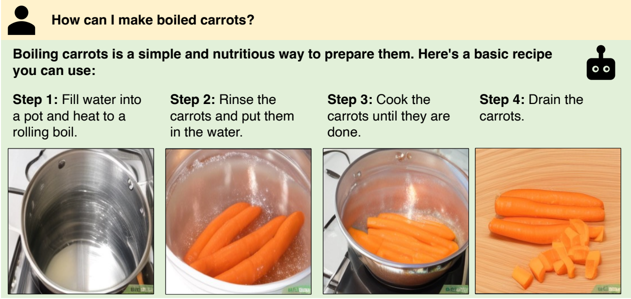
Step 1: Mix flour and sugar in a mug.

Step 2: Add cocoa and ground vanilla; stir with wet ingredients.

Step 3: Put on a plate and microwave until done.

Step 4: Optionally, dust with powdered sugar.

Figure 17. **Additional article generation.** The mug shows the consistency StackedDiffusion enables.



How can I make boiled carrots?

Boiling carrots is a simple and nutritious way to prepare them. Here's a basic recipe you can use:

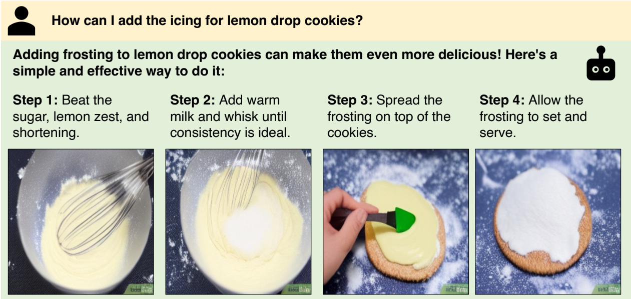
Step 1: Fill water into a pot and heat to a rolling boil.

Step 2: Rinse the carrots and put them in the water.

Step 3: Cook the carrots until they are done.

Step 4: Drain the carrots.

Figure 18. **Additional article generation.**



How can I add the icing for lemon drop cookies?

Adding frosting to lemon drop cookies can make them even more delicious! Here's a simple and effective way to do it:

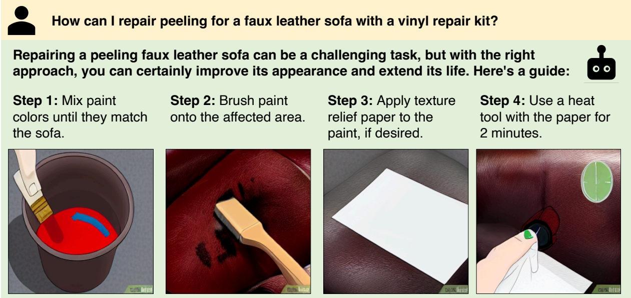
Step 1: Beat the sugar, lemon zest, and shortening.

Step 2: Add warm milk and whisk until consistency is ideal.

Step 3: Spread the frosting on top of the cookies.

Step 4: Allow the frosting to set and serve.

Figure 19. **Additional article generation.**



How can I repair peeling for a faux leather sofa with a vinyl repair kit?

Repairing a peeling faux leather sofa can be a challenging task, but with the right approach, you can certainly improve its appearance and extend its life. Here's a guide:

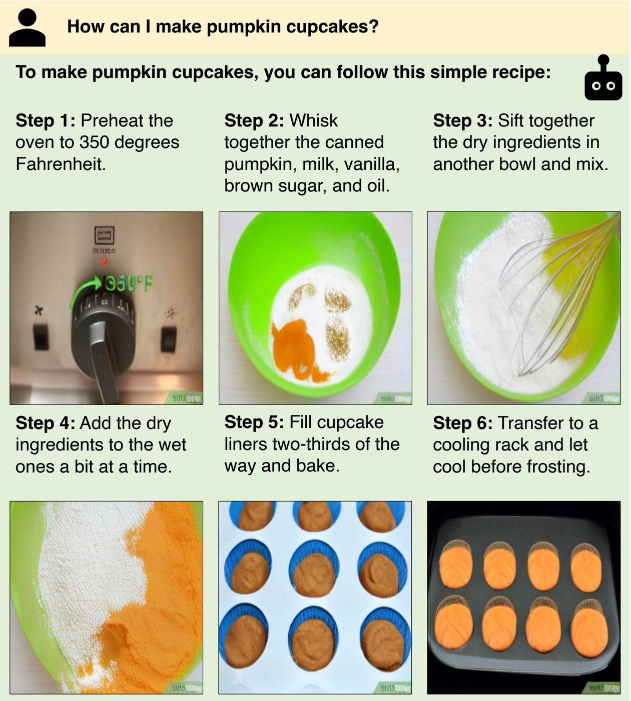
Step 1: Mix paint colors until they match the sofa.

Step 2: Brush paint onto the affected area.

Step 3: Apply texture relief paper to the paint, if desired.

Step 4: Use a heat tool with the paper for 2 minutes.

Figure 20. **Additional article generation outside of recipes.**



How can I make pumpkin cupcakes?

To make pumpkin cupcakes, you can follow this simple recipe:

Step 1: Preheat the oven to 350 degrees Fahrenheit.

Step 2: Whisk together the canned pumpkin, milk, vanilla, brown sugar, and oil.

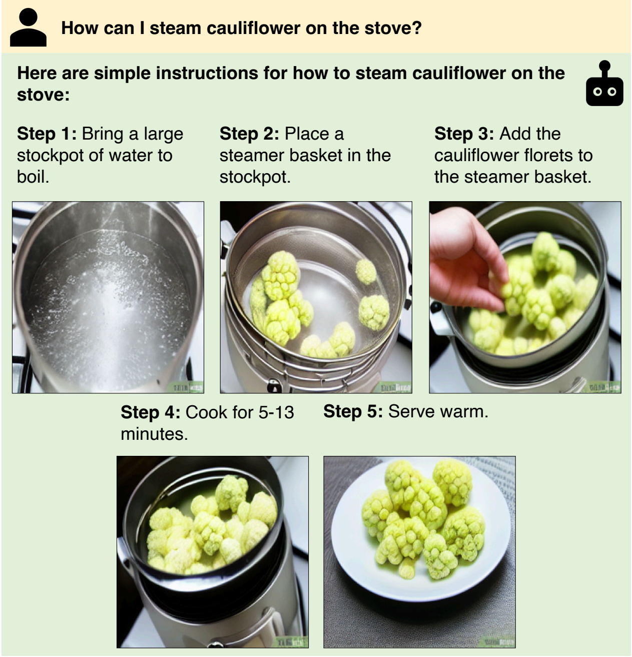
Step 3: Sift together the dry ingredients in another bowl and mix.

Step 4: Add the dry ingredients to the wet ones a bit at a time.

Step 5: Fill cupcake liners two-thirds of the way and bake.

Step 6: Transfer to a cooling rack and let cool before frosting.

Figure 21. **Example failure case.** We note two failure modes. First, the model is over-consistent, producing a bowl in Step 3 that looks similar to the bowl in Step 2 when it should be another bowl. Second, the model appears to consider geometric arrangements as similar to each other even if the numbers in each differ, leading to inconsistency between Step 5 and Step 6.



How can I steam cauliflower on the stove?

Here are simple instructions for how to steam cauliflower on the stove:

Step 1: Bring a large stockpot of water to boil.

Step 2: Place a steamer basket in the stockpot.

Step 3: Add the cauliflower florets to the steamer basket.

Step 4: Cook for 5-13 minutes.

Step 5: Serve warm.

Figure 22. **Example failure case 2.** An additional example displaying over-consistency at the cost of step faithfulness. Step 2 already displays cauliflower in the in the steamer basket, but the text only describes it being added in Step 3.