Generating Coherent Sequences of Visual Illustrations for Real-World Manual Tasks

João Bordalo[1] **Vasco Ramos**[1] **Rodrigo Valério**[1] **Diogo Glória-Silva**[1]
Yonatan Bitton[2] **Michal Yarom**[2] **Idan Szpektor**[2] **Joao Magalhaes**[1]
[1]NOVA LINCS, NOVA School of Science and Technology, Portugal
[2]Google Research
jmag@fct.unl.pt, szpektor@google.com

Abstract

Multistep instructions, such as recipes and how-to guides, greatly benefit from visual aids, such as a series of images that accompany the instruction steps. While Large Language Models (LLMs) have become adept at generating coherent textual steps, Large Vision and Language Models (LVLMs) are less capable of generating accompanying image sequences. The most challenging aspect is that each generated image needs to adhere to the relevant textual step instruction, as well as be visually consistent with earlier images in the sequence. To address this problem, we propose an approach for generating consistent image sequences, which integrates a Latent Diffusion Model (LDM) with an LLM to transform the sequence into a caption to maintain the semantic coherence of the sequence. In addition, to maintain the visual coherence of the image sequence, we introduce a copy mechanism to initialise reverse diffusion processes with a latent vector iteration from a previously generated image from a relevant step. Both strategies will condition the reverse diffusion process on the sequence of instruction steps and tie the contents of the current image to previous instruction steps and corresponding images. Experiments show that the proposed approach is preferred by humans in 46.6% of the cases against 26.6% for the second best method. In addition, automatic metrics showed that the proposed method maintains semantic coherence and visual consistency across the sequence of visual illustrations.[1]

1 Introduction

When humans undertake a task with numerous intricate steps, merely reading a step description is limiting, leaving the user to imagine and infer some of the more nuanced details (Choi et al., 2022). Complementing the textual step instructions with images enhances the user experience by better communicating and representing the text semantics and ideas (Serafini, 2014).

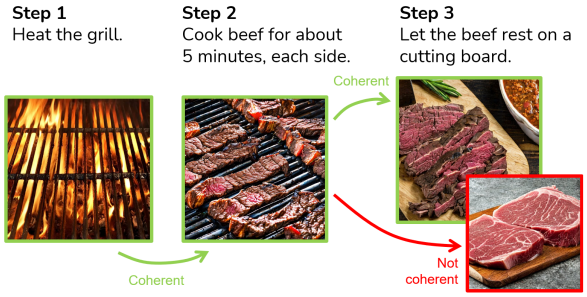

Figure 1: The properties of the elements in illustrations should remain coherent throughout the whole sequence.

Although prompt-based image generation has advanced significantly (Betker et al., 2023; Rombach et al., 2022; Saharia et al., 2022), state-of-the-art (SOTA) models such as Latent Diffusion Models (LDMs) (Rombach et al., 2022) still struggle when generating image sequences to accompany textual instruction steps (Lu et al., 2023). The challenge lies in effectively combining two key aspects: (a) accurately portraying the actions outlined in the step instructions, and (b) ensuring coherence between successive images to avoid confusing the user. Existing storytelling approaches (Feng et al., 2023; Pan et al., 2022; Rahman et al., 2023) operate mostly on linear storytelling and use synthetic cartoon datasets with explicit sequence information, i.e., the textual prompts describe the images appropriately and have no implicit co-references. These aspects limit the applicability of existing methods to real-world scenarios (Figure 1), where there is a lack of informative prompts accompanying images, and dependencies between prompts are not necessarily linear.

In this paper, we explore the generation of image sequences within two domains: recipe instructions, and Do It Yourself (DIY) guides, both showing increasing online consumption (Bausch et al., 2021;

[1]https://novasearch.github.io/generating-coherent-sequences/

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12777–12797

Brimble, 2020; Sarpong et al., 2020; Quader, 2022). In these domains, accuracy and coherence are of utmost importance to ensure that the result of all manual actions is correct, and that the user is correctly guided to the target output, Figure 1. These domains contain (i) complex sequential manual tasks of detailed actions, (ii) coherence requirements for the images accompanying the sequence step descriptions, and (iii) a non-linear sequential structure, where steps may be related to earlier steps–not necessarily the previous step.

To tackle these challenges, we propose to extend Latent Diffusion Models (Rombach et al., 2022), with an LLM decoder to semantically condition the reverse diffusion process in the sequence of steps and a copy mechanism to select the best LDM initialisation. The image generation process is conditioned on the current step and the previous steps, to increase semantic coherence. In addition, our method initializes the reverse diffusion process with a latent vector iteration copied from a previous generation process to ensure the visual coherence of the generated image. Through this dual attendance to past textual and visual items in the sequence, we aim to achieve *semantic coherence*, which pertains to the presence and persistence of objects in consecutive images, and *visual coherence*, which aims to ensure the consistency of backgrounds and visual object properties across successive images.

Extensive automatic and manual evaluations confirmed that our model outperforms strong baselines in terms of the overall quality of the generated sequence of illustrations in the cooking and DIY domains.

2 Related Work

Methods to generate sequences of images, conditioned on textual input, have been explored in the story visualization and story continuation tasks. Story Visualization aims at generating a coherent sequence of images, based on a multi-sentence paragraph or a series of captions forming a narrative (Li et al., 2019; Pan et al., 2022; Maharana et al., 2022). Story Continuation is a variant of Story Visualization, in which the generated sequence is initiated by a source image. In both tasks, generating every image independently of other images in the sequence leads to low visual coherence. In contrast, editing the previous image like in (Cong et al., 2022; Fu et al., 2020) will lead to insuficiently diverse images.

Several works addressed these tasks. Pan et al. (2022) proposed AR-LDM, a method to tackle Story Visualization and Continuation using a history-aware autoregressive latent diffusion model (Rombach et al., 2022), which encodes the history of caption-image pairs into a multimodal representation that guides the LDM denoising process. Despite the intuitive idea, the computational complexity of the conditioning network makes this approach too costly. Additionally, AR-LDM still shows room for improvement in terms of coherence. Feng et al. (2023) noted that AR-LDM conditions the current generation on all historic frames and captions equally, despite not all frames being similarly related. To tackle this limitation, they proposed ACM-VSG, a method that selectively adopts historical text-image data for the generation of the new image. The adaptive encoder automatically finds the relevant historical text-image pairs via CLIP similarity. A key difference between AR-LDM and ACM-VSG is the computational cost: while AR-LDM fine-tunes CLIP, BLIP, and the LDM, ACM-VSG trains only the cross-attention module, at a much lower cost. Rahman et al. (2022) used the full history of U-net latent vectors from all segments of the sequence, averaging these historic latent vectors in a cross-attention layer that is merged with the existing one in the LDM pipeline. In this method, image generation is conditioned on the text and on the entire set of latent vectors, with limited awareness of the visual coherence between segments.

The above approaches focus on two synthetic cartoon datasets (Kim et al., 2017; Gupta et al., 2018), with limited characters and scenes. Among these, (Pan et al., 2022) shows limited performance when applied to real-world sequences, and (Feng et al., 2023; Rahman et al., 2023) do not evaluate their solutions in a real-world scenario. Additionally, these datasets have textual descriptions of the associated images, which are rarely available for real-world complex tasks – the focus of this paper.

Finally, it is relevant to note the conceptual relation to the visual storytelling problem (Huang et al., 2016) to generate a text story from a sequence of images (Wang et al., 2020; Hsu et al., 2021) and to the extraction of news storylines (Marcelino et al., 2019, 2021). While, conceptually similar, both tasks can be seen as the inverse of the problem addressed in the current paper.

3 Illustrating Real-World Manual Tasks

We consider a set of manual tasks $\mathcal{D}$, where each task $TS \in \mathcal{D}$ is composed of a sequence of n step-by-step instructions, $TS = \{(s_1, v_1), ..., (s_n, v_n)\}$. A task step (s_i, v_i), consists of a natural language instruction s_i, and its corresponding visual instruction v_i.

Given the sequence of steps $\{s_1, \ldots, s_n\}$, our goal is to generate a sequence of images $\{v_1, \ldots, v_n\}$, in which v_i visually represents step s_i. A step s_i may be dependent on any number of previous steps, in a non-linear sequential structure (Donatelli et al., 2021). To generate each image accurately, the model needs to condition its output not only on s_i but also on previous steps $\{s_1, \ldots, s_{i-1}\}$; this way, context is preserved even when steps are ambiguous or lack information, e.g. "*Add two eggs and mix*". In addition to previous step instructions, we also need to condition on previously generated images, $\{v_1, ..., v_{i-1}\}$, to maintain the visual aspects that are only introduced in the images, such as object properties and background artefacts not mentioned in the text.

4 Sequential Latent Diffusion Model

The latent diffusion model proposed by Rombach et al. (2022) transforms the diffusion process into a low-dimensional latent space through an encoder $z = E(v)$ and recovers the real image with a decoder $v = D(z)$. The complete model is also conditioned on an input y by augmenting the U-Net backbone with a cross-attention layer to support the encoded input $\tau_\theta(y)$. The conditional LDM is learned with the loss,

$$\mathcal{L}_{LDM} = \mathbb{E}_{E(v),y,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right], \quad (1)$$

where $\epsilon \sim \mathcal{N}(0, 1)$, t is the denoising iteration, and the conditioning encoder $\tau_\theta(y)$ uses the entire set of tokens in y to condition the U-Net denoising process in the LDM backbone. Eq. 1 evidences how the conditional LDM is designed to generate one image v at a time. More relevant to our problem is the fact that the latent vectors z_t are independent across different image generations, since the reverse diffusion process iterates from T to 1 starting with a new random seed z_T for every new image generation.

4.1 Sequence Context Decoder

Generally, textual step descriptions describe what the user should do at a specific step of their manual task. These descriptions do not make accurate captions of the accompanying step images as they often contain information that is not visually representable, such as temporal information, *"Cook for 10 minutes"*, or multiple actions, *"Chop the rosemary, dice the carrots, and peel the cucumber."* Additionally, it is also common for steps to not be self-contained, as they depend on the previous step descriptions for context.

Original Step: "Slide Dutch baby onto serving platter or cutting board, or serve straight from pan, if desired. Top with fresh berries and sprinkle with lemon juice. Dust with confectioner's sugar and (...)"

No Context: "A person is sprinkling sugar on top of a pancake."

Current Step as Context: "A person is sprinkling confectioner's sugar on top of a Dutch baby pancake."

Figure 2: Example captions generated by InstructBLIP. In the "No Context" example, the model only receives the image. In the "Current Step as Context" example, the model receives the image plus the "Original Step".

To overcome this, for each step s_i, we use a decoder-only model, φ, which we call Sequence Context Decoder, to transform the step and its context into a visual caption c_i which describes the contents of the image v_i. To ensure the generated captions are contextually relevant, we adopt a middle-ground approach and consider the target step s_i and a context window of w steps. Formally, we define the decoder

$$c_i = \varphi(s_i, \{s_{i-1}, \ldots s_{i-w}\}) \quad (2)$$

to generate a contextual caption c_i from its step description s_i and context.

The decoder $\varphi(\cdot)$ is trained similarly to an image caption generator, but instead of receiving images as input, it receives the step and its context. By training the model to output image captions for the original images that we are trying to replicate, the model learns to generate texts that are more appropriate as image generation prompts. The objective now is learning how to generate better image generation prompts, instead of training the image generation module.

To train the decoder model $\varphi(\cdot)$, we generated contextual captions for each image in dataset $\mathcal{D}$ using InstructBLIP (Li et al., 2022). To achieve richer and contextualized captions, we prompted InstructBLIP with additional context, conditioning the caption on the recipe steps, in addition to the image. Figure 2 shows example captions generated by the model, given a real data point in the dataset.

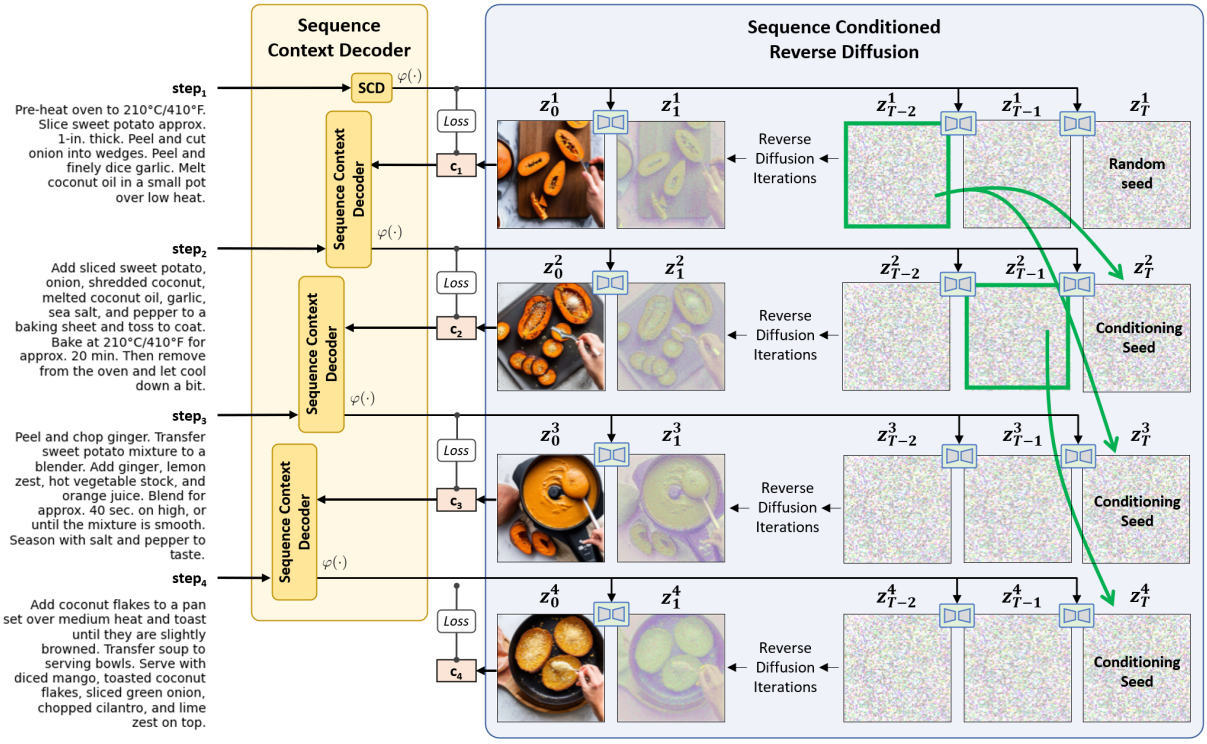

Figure 3: The proposed method uses the sequence context decoder to maintain semantic coherence. The reverse diffusion process uses a conditioning seed z_T^i that is copied from a previous step and iteration z_k^j. See Equation 3.

4.2 Sequence Conditioned Reverse Diffusion

To maintain visual coherence among images in the sequence, we need to condition the current generation on the previous images. Image-to-image (Meng et al., 2022) generation follows this principle, but new images are too strongly influenced by previous ones and do not correctly integrate the new aspects present in the step description.

Following the rationale of conditioning every reverse diffusion process on previous processes, we propose to leverage latent vector iterations from early reverse diffusion processes. This leads us to the final formulation of the proposed method,

$$\mathcal{L}_{SLDM}(s_i) = \mathbb{E}_{E(v_i),s_i,\epsilon,t}\Big[\|\epsilon - \epsilon_\theta(z_t^i, t, \tau_\theta(c_i = \varphi(s_i, \{s_{i-1}, \ldots, s_{i-w}\})))\|_2^2\Big] \quad (3)$$

where for each s_i a new reverse diffusion process starts with a conditioning seed z_T^i copied from a previous step s_j with $j < i$ and a latent vector iteration k corresponding to the latent vector z_k^j, as illustrated in Figure 3. Next, we describe the details of how these two variables are determined.

4.2.1 Random and Fixed Seeds

To ground the generation, a straightforward method is to set a fixed seed for every step i in the sequence, $z_T^i = c^{te}$. By fixing the initial seed, we aim to improve the coherence between generated images. The first two columns in Figure 5 demonstrate this approach. We observed a greater homogeneity between generated images when using a fixed seed. Hence, all step illustrations share the same random seed, to achieve more coherent scenes and backgrounds but without the capacity to select the optimal starting seed.

4.2.2 Conditioned Initialisation

While using a fixed seed can improve the results, we argue that a better solution is achieved by using latent vectors from previous reverse diffusion processes. In particular, *latent vectors that have already been semantically conditioned on past steps*. Figure 3 shows how the latent vector representations z_t^i evolve with increasing iterations, until they arrive at the final image $v_i = D(z_0^i)$ for step s_i. These latent representations already contain meaningful information about the image (Mao et al., 2023), which could be leveraged to improve the coherence of the following generations. To achieve this, we need to carefully select which step to

choose, to use as input seed for the next step image generation.

A step s_i may be dependent on any previous step $j < i$, $\{s_{i-1}, \ldots, s_1\}$. To select the optimal initialization of the reverse diffusion process for step s_i, we start by determining the most similar step s_j as the

$$\arg\max_j sim(s_i, s_j \in \{s_{i-1}, \ldots s_1\}) \quad (4)$$

where $sim(\cdot)$ represents CLIP text similarity. If this similarity score is above a predefined threshold η, we use s_j to extract the latent vector. If no step s_j has a similarity score above η, we generated image v_i with the shared random seed.

The reverse diffusion process progressively iterates over the latent vectors towards the final image. This means that conditioning the reverse diffusion process on latent vector iterations from a later iteration, i.e. a highly denoised latent vector, would force the resulting image to be very close to the previous one. To decide how strongly we want to condition v_i on the step s_j, we select the k^{th} latent vector iteration as

$$k = n \cdot (sim(s_i, s_j) - \eta)/(1.0 - \eta) \quad (5)$$

where n is the maximum number of reverse diffusion iterations that we consider.

This brings us to the target reverse iteration vector z_k^j which will be used as a starting seed z_T^i in Eq 3 when calculating $\mathcal{L}_{SLDM}(s_i)$. Figure 3 illustrates the whole process: the proposed method captures the visual aspects that should be in the image, and the linked denoising latent vector provide the seed to generate a step image.

5 Experimental Methodology

Dataset. We collected a dataset consisting of publicly available manual tasks in the recipes domain from AllRecipes. We also considered DIY manual tasks from WikiHow, in an out-of-domain evaluation. Each manual task has a title, a description, a list of ingredients/resources, and a sequence of step-by-step instructions, which may or may not be illustrated. Since we want to illustrate the steps of a task, we focus on manual tasks which are fully illustrated, as we can use these images as ground-truth for training and evaluating our methods. In total, we used 1100 tasks, with an average of 5.06 steps per task, resulting in 5562 individual steps.

We found that recipes with long steps descriptions or many steps, were difficult to tackle and produced worse results. This pointed towards a refinement of the dataset, so that our approach could better focus on the issue of coherence, instead of tackling other problems.

When a step description is too long, it contains too much information, often with multiple actions, which is hard to represent in a single image. Adding to this issue, the CLIP (Radford et al., 2021) text encoder, used in the Stable Diffusion (Rombach et al., 2022) model, truncates the input text at 77 tokens. We filtered the recipes that had steps that were too long. A second problem arises when a recipe has too many steps, as it is difficult to produce coherent illustrations over such a long sequence of steps. To mitigate this concern, we limited the number of steps in a recipe from 4 to 6 steps. In a final stage of refining the dataset, we removed any steps that did not contain actions which we could illustrate, such as steps merely saying *"Enjoy!"*.

Contextual Caption Generation. As previously described, we provide InstructBLIP (Li et al., 2022) with additional context to produce contextualized captions. We experimented with different context lengths and decided to rule out experiments that gave InstructBLIP the full context, i.e., all previous step instructions, as this led to very long outputs that often repeated irrelevant information from the input context, instead of describing the image. We addressed the issue of long input prompts, by using a context window, as described in Section 4.1. This allows us to give the model additional context while mitigating possible errors in the generated training caption. When generating the image captions used to train the Sequence Context Decoder, we produced two sets of captions: long and short. The prompt to InstructBLIP consists of the additional context window followed by *"Given the steps, give a short description of the image. Do NOT make assumptions, say only what you see in the image."*. We generate captions with a window of 2 steps–short captions–and with a window of 3 steps–long captions. Finally, we generate a long and a short caption for every image, v_i in the dataset.

Model Details. To train the sequence context decoder model, we fine-tuned an Alpaca-7B model for 10 epochs on a single A100 40Gb GPU. We used a cross-entropy loss and a cosine learning rate

scheduler, starting at $1e^{-5}$. The batch size was set to 2, with a gradient accumulation step of 4. The dataset had a total of 5562 step-caption pairs, from which we used 80% for training and 180 examples for testing. We used a frozen Stable Diffusion 2.1 (Rombach et al., 2022) for image generation.

Human Annotations. To evaluate our models we ran three annotation jobs, and, in all cases, annotators were allowed to provide feedback on the generation errors. See Appendix D for details. In the first job, annotators inspected 30 sequences of images generated with different methods (withheld from annotators) and selected the 3 best sequences, out of a total of 5 sequences. Besides this selection, we provided an additional *No good sequence* label, for when no sequence of images was of good quality. The second job aimed at obtaining finer-grained annotations for two methods that performed best. Annotators compared the proposed method against the second best method (latent 1) and were asked to select the preferred one or to indicate a tie. After these two initial human annotation jobs, we decided to compare the proposed method to the real-world image sequences and asked annotators to rate the two sequences. This is by far the most challenging setting where real-world image sequences have the natural visual coherence that we wish to achieve.

Automatic Metrics. To assess image sequences generated by the proposed method, we measured the semantic correctness and the sequence coherence with CLIPScore (Hessel et al., 2021) and with the novel DreamSim metric (Fu et al., 2023), respectively. DreamSim measures the similarity between two images in terms of their foreground objects and semantic content, while also being sensitive to colour and layout. This is particularly well suited to our task, as we wish to maintain visual coherence across the entire sequence.

6 Results and Discussion

In this section, we start by comparing the proposed method to latent diffusion models, SD2.1 (Rombach et al., 2022), and non-LDM models, i.e. aMUSEd (Patil et al., 2024) and DALL·E Mini (Dayma et al., 2021), with automatic metrics. For the ablation studies, we experimented with using (1) a **random seed** for all steps of the sequence, (2) a **fixed seed** for all steps of the sequence, (3) a **fixed latent vector iteration** from the previous step, represented by **Latent** k, where k is the fixed iteration, and (4) the proposed method that selects a latent vector iteration, z_i^j from the previous denoising steps as a starting seed.

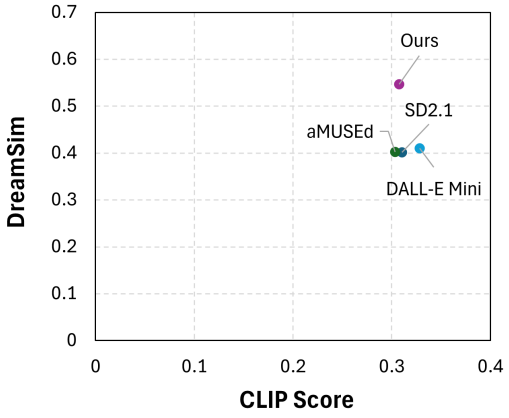

Figure 4: Automatic evaluation of image sequence. CLIP-Score (Hessel et al., 2021) measures the alignment between the step and the image. DreamSim (Fu et al., 2023) measures similarity between visual illustrations in the sequence.

6.1 Automatic Evaluation

Figure 4 shows that the proposed method can improve the coherence of image sequences (as measured by DreamSim), while maintaining the same text-to-image generation capabilities (as measured by CLIPScore). This result is particularly important because it clearly shows that it is possible to maintain key visual and semantic traits from specific iterations of the reverse-diffusion process – an LDM property that we leverage in this paper.

6.2 Sequence Generation Results

Recipes Domain. To validate our initial hypothesis, we start by analysing its performance in the recipes domain. Our goal is to assess how conditioning the reverse diffusion process of each step affects the overall results. Table 1 provides the complete set of results across all competing methods. It is clear how using latent vector iterations and random seeds supports the intuition behind our method: a manual task is composed of continuous and independent actions, which should be conditioned by latent vector iterations and by random seeds, respectively. Building on this observation, we calibrated the η parameter of the proposed method, using human evaluation, as reported in annex in Table 2.

To compare the proposed method to the best performing method (Latent T-1), we asked annotators to select the best sequence out of two side-by-side

Method	Best (%)	Second Best (%)	Third Best (%)
Random seed	17.70	**41.20**	13.00
Fixed seed	<u>29.40</u>	17.70	<u>33.30</u>
Latent T-1	**33.30**	17.70	**37.04**
Latent T-2	17.70	<u>23.50</u>	16.70
Img-to-Img	2.00	0.00	0.00

Table 1: Annotation results for the evaluation of the various methods of maintaining visual coherence. Annotators picked *No Good Sequence* in 18.99% of the sequences; we report the results for the remaining 81.01%.

η	Best (%)	Second Best (%)	Third Best (%)
0.70	19.20	12.80	14.90
0.65	12.80	14.90	**25.50**
0.60	19.20	23.40	21.30
0.55	14.90	23.40	21.30
0.50	**34.04**	**25.53**	17.02

Table 2: Annotation results for the sequences generated with different threshold values, η. Annotators picked *No Good Sequence* in 20.34% of the generations; we show results for the remaining 79.66%.

Method	Recipes (seen)	DIY (unseen)
Proposed method (wins)	**46.67**	**30.00**
Second best (wins)	26.67	23.33
Tie	10.00	16.67
No good sequence	16.67	30.00

Table 3: Annotation results of the comparison between our proposed method and the winning method from Table 1 (Latent T-1).

sequences (see Appendix D for details about the annotation instructions). Results reported in Table 3 show that in 46.7% of the cases, human annotators preferred our method over the alternative and in 10.0% of the cases the two methods were equally good. According to the annotations, the proposed method was equal or better than the second best baseline with an agreement of 70% across the full set of tasks and annotators. This confirms our hypothesis and supports the importance of selectively conditioning the denoising process on the previously generated steps of the sequence.

DIY Domain. To assess the generalisation of the proposed method, we evaluated its performance in an unseen domain: DIY tasks. With the results of our human annotation study, we observed that the transition from generating recipe images to DIY tasks has shown promising results. Results in Table 3 show that in 30.0% of the tasks, neither method produced satisfactory results. For the tasks that were correctly illustrated, we see that annotators preferred our method in 30.0% of the tasks, compared to 23.3% for the second-best approach. Additionally, 16.7% of comparisons resulted in a tie between the two methods. Although we see limitations in this domain, the results show that our approach is capable of generalizing to an unseen domain, producing satisfactory image sequences.

Generated vs Ground-Truth Sequences. We compared the quality of generated image sequences against the ground-truth sequences and asked human annotators to rate each sequence in a 5 point Likert scale. This is a particularly challenging setting because the real image sequences are photos taken by humans in a real-world setting, where sequence coherence is naturally captured. Table 4 shows that our method achieves over 60% of the ground-truth score, with the ground-truth sequences only 0.42 points below the maximum score.

Method	Average Rating
Proposed method	2.93 ± 1.14
Ground-truth	4.58 ± 0.79

Table 4: Human annotation results for the comparison of the proposed method with ground-truth images.

Qualitative Analysis. To illustrate how different conditioning methods affect the quality of generated sequences, we present several examples in Figure 5 and in Appendix E.2. In Figure 5, we can see that by using a fixed seed for all steps, we are able to preserve the background and add objects for each specific step. We can also see that the image-to-image method conditions the generations too strongly. Figure 5 also shows that our method is capable of preserving and recall key visual artefacts from several steps back in the sequence. We believe this is a distinctive and fundamental feature of the proposed method.

In the DIY out-of-domain experiment, Figure 6 shows a strong generalization to tasks involving

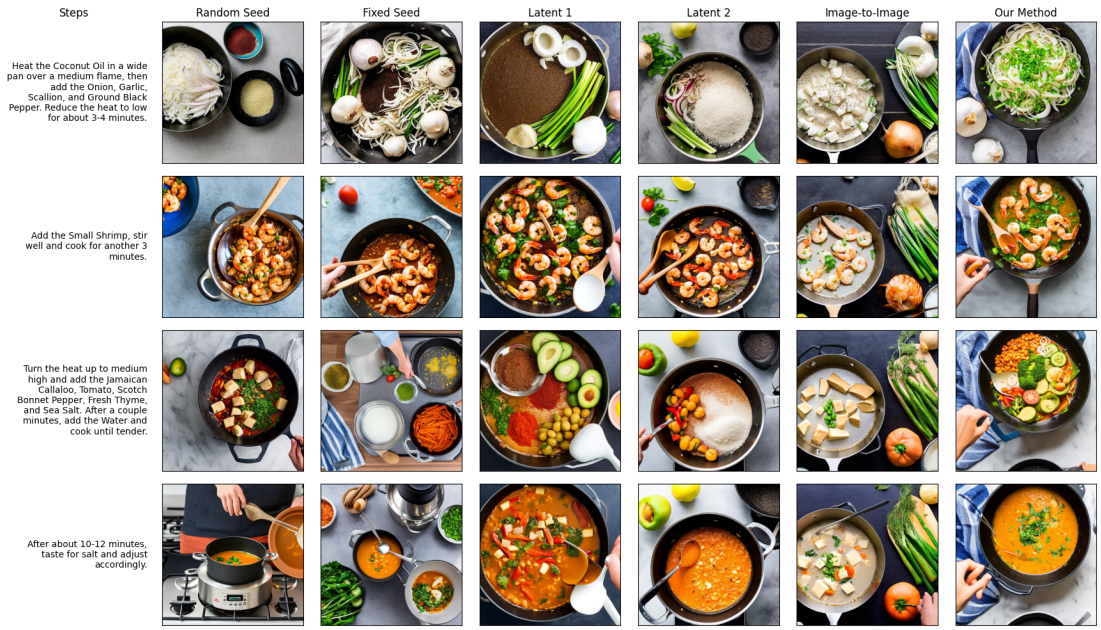

Figure 5: Examples of recipe illustrations with different methods for maintaining visual coherence.

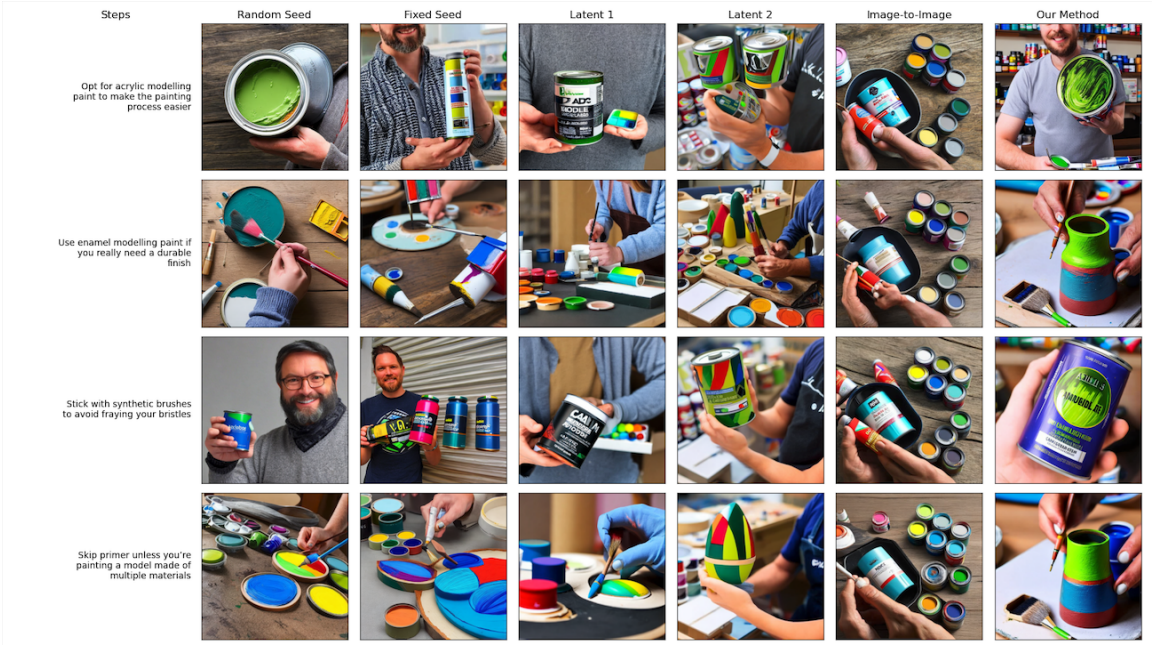

Figure 6: Examples of DIY illustrations with different methods for maintaining visual coherence.

simple object utilisation, such as using a broom for cleaning or a brush for painting. While fixed latent vector iteration methods show some memorization capability, the image-to-image generations are too biased on previous generations, and random seeds lead to very diverse generations. In this unseen domain, the proposed method encounters considerable challenges when tasked with more intricate activities, like performing a car's oil change with its complex mechanical components, or engaging in tasks that involve philosophical or introspective elements, see Appendix E.2 for visual examples. It is worth noting that these limitations are inherited from the core image generation method, which struggles to handle fine-details.

6.3 Sequence Conditioned Reverse Diffusion

To better understand the strength of our **visual coherence** hypothesis, we conducted an experiment where all the images of the sequence are generated from the latent vectors of a fixed iteration from the previous task step. Specifically, we use the latent

vector iteration k, with $k \in \{1, 2, 5, 10, 20, 49\}$, of the previous task step as the starting point of the current reverse diffusion process.

Our empirical analysis of these generations, from which some examples can be seen in annex in Figure 7, confirms our initial hypothesis showing that using latent vector iterations from previous steps provides a good result in many cases. Additionally, this experiment evidences the strength of latent vector iterations as conditioning signals in a reverse diffusion process. This experiment shows that later iterations add a very strong bias to the generation emphasising the importance of selecting the best conditioning seed from previous reverse diffusion iterations and processes. An extreme case of this phenomenon is observed in the image-to-image method in Figure 5.

6.4 Sequence Context Decoder

We assessed the capacity of the sequence context decoder of maintaining the **semantic coherence** by manually annotating the decoder output for six settings with different context lengths and caption lengths, Table 5. We considered three context lengths: the shortest one considers the current step, while the longest, shows two steps and a caption. For the captions, we used both *short* and *long* captions, as detailed in Section 5.

Table 5 shows the results for the different evaluation settings. These results indicate that the model attains the best semantic coherence with a context window of 2 and with short captions. We observed that captions generated with short contexts tend to lack some information, while captions generated with longer contexts introduced too much information, which was often noisy. This is aligned with a recent study (Liu et al., 2023) that highlights the fact that current large language models do not robustly make use of information in long input contexts. These results indicate that additional context needs to be carefully considered and curated, as the model is not able to filter out excess information. In annex E.1 we provide more insight into the performance of the decoder.

7 Conclusions

In this paper, we addressed the problem of illustrating complex manual tasks and proposed a framework for generating a sequence of images that illustrate the manual task. The framework is composed of a novel **sequence context decoder** that preserves the **semantic coherence** of a sequence of actions by transforming it into a visual caption. The full sequence illustration framework is completed by a **sequence conditioned reverse diffusion** process that uses a latent vector iteration from a past image generation process to maintain **visual coherence**.

Sequence Context	Captions	Avg. Rating
$\{s_n\}$	*short*	3.00
$\{s_n\}$	*long*	3.23
$\{s_n, c_{n-1}\}$	*short*	**3.68**
$\{s_n, c_{n-1}\}$	*long*	<u>3.56</u>
$\{s_n, s_{n-1}, c_{n-2}\}$	*short*	3.41
$\{s_n, s_{n-1}, c_{n-2}\}$	*long*	3.35

Table 5: Sequence Context Decoder results for different context lengths, configurations and caption type.

Automatic and human evaluations in the target domain demonstrated the strong performance of the framework in generating coherent sequences of visual instructions of manual tasks. The proposed method was preferred by humans in 46.6% of the cases against 26.6% for the second best method. In addition, automatic measures, also confirmed that our method maintains visual and semantic coherence while illustrating a complex manual task. The generalization to unseen domains was successfully validated in the DIY domain with positive results.

8 Risks and Limitations

While we conducted a thorough set of experiments and validations, we acknowledge that more experiments could shed more light in some aspects. First, we did not experiment with larger LLMs to discover the impact of scaling. Second, we also did not consider that steps may dependent on multiple previous steps. Third, we did not consider contexts larger than 3 steps. Finally, in terms of risks, we acknowledge that our work could potentially be used to generate false information.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. This work was partially supported by Amazon Alexa Prize TaskBot, by a Google Research Gift and by NOVA LINCS ref. UIDB/04516/2020 (https://doi.org/10.54499/UIDB/04516/ 2020).

References

Cleber Lemes Bausch, Gabriel Sperandio Milan, Ana Paula Graciola, Luciene Eberle, and Suélen Bebber. 2021. The covid-19 pandemic and the changes in consumer habits and behavior. *Revista Gestão e Desenvolvimento*, 18(3):3–25.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*.

Lucy Brimble. 2020. More than 70% of adults use social media for recipes instead of cookbooks, survey finds. *Independent UK*.

Jason Ingyu Choi, Saar Kuzi, Nikhita Vedula, Jie Zhao, Giuseppe Castellucci, Marcus Collins, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein. 2022. Wizard of tasks: A novel conversational dataset for solving real-world tasks in conversational settings. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 3514–3529. International Committee on Computational Linguistics.

Gaoxiang Cong, Liang Li, Zhenhuan Liu, Yunbin Tu, Weijun Qin, Shenyuan Zhang, Chengang Yan, Wenyu Wang, and Bin Jiang. 2022. Ls-gan: Iterative language-based image manipulation via long and short term consistency reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4496–4504, New York, NY, USA. Association for Computing Machinery.

Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phúc Le Khac, Luke Melas, and Ritobrata Ghosh. 2021. Dall·e mini.

Lucia Donatelli, Theresa Schmidt, Debanjali Biswas, Arne Köhn, Fangzhou Zhai, and Alexander Koller. 2021. Aligning actions across recipe graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6930–6942, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhangyin Feng, Yuchen Ren, Xinmiao Yu, Xiaocheng Feng, Duyu Tang, Shuming Shi, and Bing Qin. 2023. Improved visual story generation with adaptive context modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4939–4955, Toronto, Canada. Association for Computational Linguistics.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. 2023. Dreamsim: Learning new dimensions of human visual similarity using synthetic data.

Tsu-Jui Fu, Xin Wang, Scott Grafton, Miguel Eckstein, and William Yang Wang. 2020. SSCR: Iterative language-based image editing via self-supervised counterfactual reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4413–4422, Online. Association for Computational Linguistics.

Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem, and Aniruddha Kembhavi. 2018. Imagine this! scripts to compositions to videos. In *European Conference on Computer Vision*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7514–7528. Association for Computational Linguistics.

Chi-yang Hsu, Yun-Wei Chu, Ting-Hao Huang, and Lun-Wei Ku. 2021. Plot and rework: Modeling storylines for visual storytelling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4443–4453, Online. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.

Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. 2017. Deepstory: Video story qa by deep embedded memory networks.

Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C. H. Hoi. 2022. Lavis: A library for language-vision intelligence.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6329–6338.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Eric Wang, and William Yang Wang. 2023. Multimodal procedural planning via dual text-image prompting.

Adyasha Maharana, Darryl Hannan, and Mohit Bansal. 2022. Storydall-e: Adapting pretrained text-to-image transformers for story continuation. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, page 70–87, Berlin, Heidelberg. Springer-Verlag.

Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. 2023. Guided image synthesis via initial image editing in diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 5321–5329. ACM.

Gonçalo Marcelino, David Semedo, André Mourão, Saverio Blasi, João Magalhães, and Marta Mrak. 2021. Assisting news media editors with cohesive visual storylines. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 3257–3265, New York, NY, USA. Association for Computing Machinery.

Gonçalo Marcelino, David Semedo, André Mourão, Saverio Blasi, Marta Mrak, and Joao Magalhaes. 2019. A benchmark of visual storytelling in social media. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, page 324–328, New York, NY, USA. Association for Computing Machinery.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Xichen Pan, Pengda Qin, Yuhong Li, Hui Xue, and Wenhu Chen. 2022. Synthesizing coherent story with auto-regressive latent diffusion models.

Suraj Patil, William Berman, Robin Rombach, and Patrick von Platen. 2024. amused: An open MUSE reproduction. *CoRR*, abs/2401.01808.

Shams Bin Quader. 2022. How the central sydney independent musicians use pre-established 'online diy'to sustain their networking during the covid-19 pandemic. *The Journal of International Communication*, 28(1):90–109.

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. 2021. Learning transferable visual models from natural language supervision.

Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. 2022. Make-a-story: Visual memory conditioned consistent story generation.

Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. 2023. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2493–2502.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

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, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

David Sarpong, George Ofosu, David Botchie, and Fintan Clear. 2020. Do-it-yourself (diy) science: The proliferation, relevance and concerns. *Technological Forecasting and Social Change*, 158:120127.

Frank Serafini. 2014. *Reading the visual: An introduction to teaching multimodal literacy*. Teachers College Press.

Ruize Wang, Zhongyu Wei, Piji Li, Qi Zhang, and Xuanjing Huang. 2020. Storytelling from an image stream using scene graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9185–9192.

A Model training

We chose Alpaca-7B as our base model, as it is an open-source instruction-tuned model, and there are implementations using LoRA (Hu et al., 2022), which reduces the computational cost during training. We encourage future work to study the impacts of scaling the LLM. We experimented with different hyperparameters, namely different learning rates, and learning rate schedulers, to find the most suitable ones for our problem. We point out that we are using the loss of the models as the main criterion for future experiments, as we do not have an automatic metric for evaluating model behaviour. The loss is not always indicative of the model's performance in the task at hand, as we verified empirically.

Training Details	
Base Model	Alpaca-7B
Training Time	$\approx 10h$
Epochs	10
Loss Function	Cross-Entropy
Weight Decay	0.01
Model Max Length	400
Batch Size	2
Gradient Accumulation Steps	4
Effective Batch Size	8
Learning Rate	$1e^{-05}$
Learning Rate Scheduler	Cosine
Optimizer	AdamW
Adam β_1	0.900
Adam β_2	0.999
Adam ϵ	$1e^{-08}$
LoRA	
LoRA Rank	8
LoRA α	32
LoRA Dropout	0.1

Table 6: Training parameters for the best model.

Since the cosine scheduler has greater variability between runs, due to different number of epochs leading to different loss curves, we decided to run further experiments with the next best-performing model. We experimented with varying the weight decay parameter but found that there were no significant differences in the loss curves for the three weight decay values.

For the aforementioned runs, we used $\beta_1 = 0.900$ and $\beta_2 = 0.999$ as the AdamW optimizer's hyperparameters. As a final test, we changed these to the values proposed by Ouyang et al. (2022), $\beta_1 = 0.900$ and $\beta_2 = 0.950$. We did not see any improvement in the loss curve.

Based on these results, we fine-tuned our Alpaca-7B models for 10 epochs on a single A100 40Gb GPU. We used a cross-entropy loss, a cosine learning rate scheduler, starting at $1e^{-5}$. Our batch size was 2, with a gradient accumulation step of 4, leading to an effective batch size of 8. The dataset had a total of 5562 examples; we used 80% for training and the remaining for evaluation. Figure 6 summarizes the training information for our best-performing model.

B Fixed Latent Iteration Generation

As mentioned in Section 6.3, we conducted an empirical analysis of generations using a fixed latent vector iteration from the previous step. This analysis helped us understand how different latent vector iterations impact the generation of the following image. An example from this analysis is shown in Figure 7.

C Negative Prompts

We found problems which were not related to the prompts or the concepts we were trying to generate, but general problems in image generation, i.e., tiled images, or deformed hands. In order to reduce some of these common problems, present in Stable Diffusion generations, we used negative prompts. A negative prompt steers the generation away from the concepts present in it. This string of text is added to the end of the original prompt.

In the negative prompt, we included undesirable concepts such as *human* or *hands*, and also included some additional concepts, following common practices. Our final negative prompt follows: *negative_prompts = ["hands", "human", "person", "cropped", "deformed", "cut off", "malformed", "out of frame", "split image", "tiling", "watermark", "text"].*

D Human Annotations

In this Section, we present examples of the annotation tasks.

The human annotation pool consisted of 3 PhD students and 5 MSc students. 25% of the annotators were women and 85% were men.

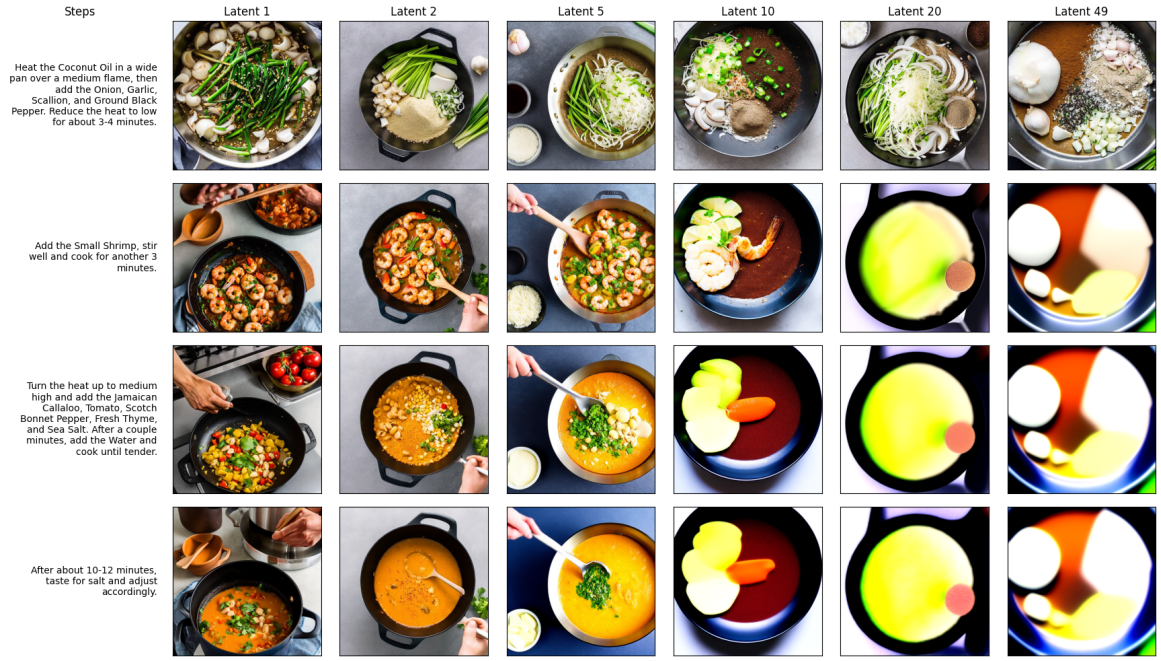

Figure 7: Maintaining visual coherence through the use of different memory latent vectors.

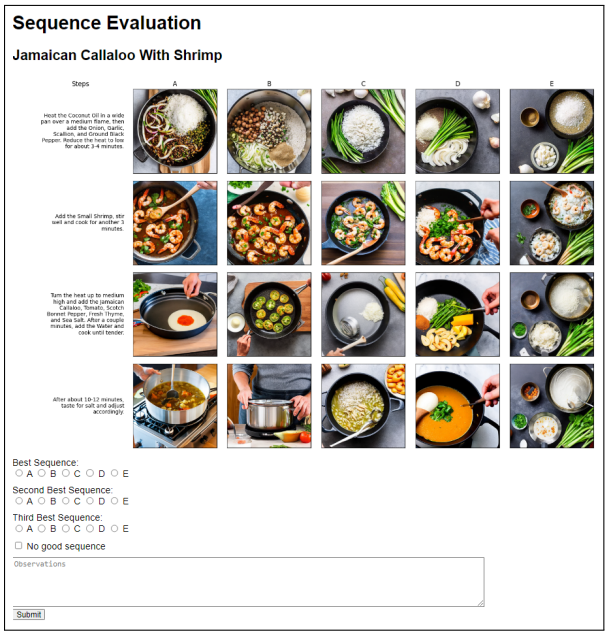

Figure 8: Annotation of the comparison between visual coherence methods.

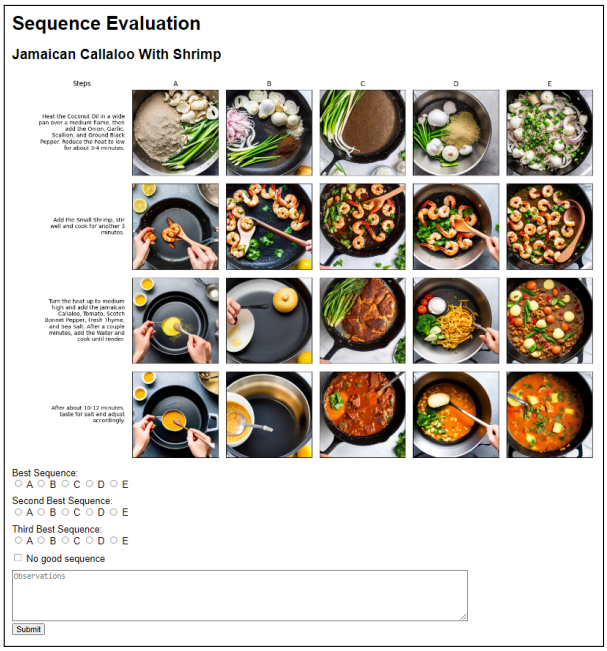

Figure 9: Annotation of the comparison between different heuristic thresholds.

Figure 8 shows the annotation task to choose the best visual coherence maintaining method. The annotators saw 5 sequences: *Random Seed*, *Fixed Seed*, *Latent 1*, *Latent 2*, and *Image-to-Image*. They were then asked to pick the best, second best, and third best sequences. They could also indicate that there were no good sequences, by checking the *No good sequence* checkbox. Additionally, they could leave an observation, in the appropriate text area. Figure 9 shows the annotation task to tune the threshold of our method. The annotators had to pick between 5 sequences, generated with different values of threshold: 0.50, 0.55, 0.60, 0.65, 0.70. They were asked to pick the best, second best, and third best sequences. They could also indicate that there were no good sequences, by

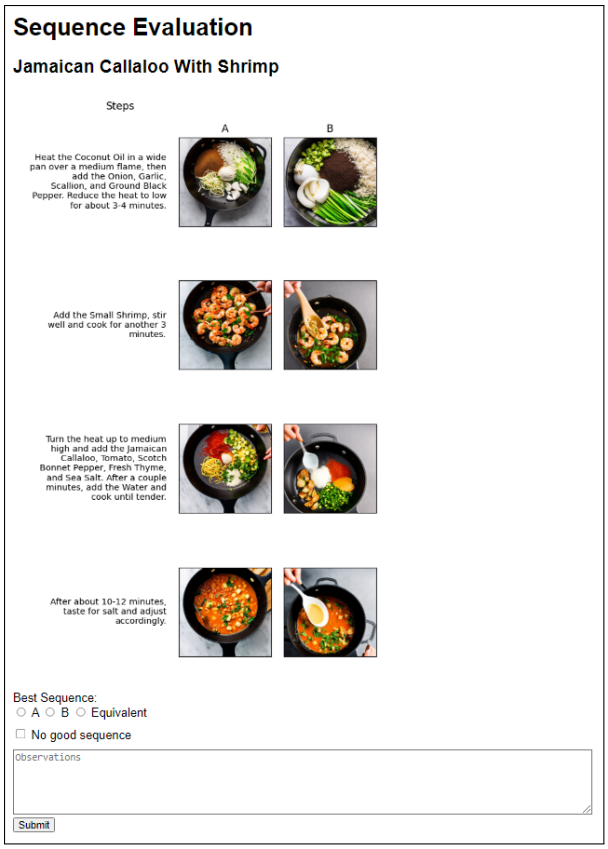

Figure 10: Annotation of the comparison between our method and the best visual coherence method.

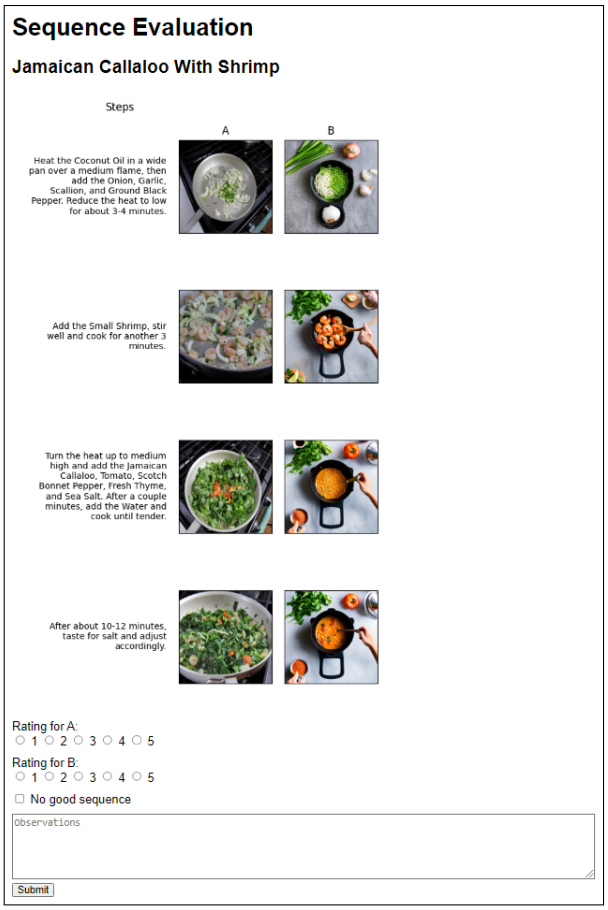

Figure 11: Annotation of the comparison between our method and the ground-truth images.

checking the *No good sequence* checkbox. They could also leave an observation, in the appropriate text area. Figure 10 shows the annotation task to choose between our method and the winning visual coherence maintaining method. The annotators saw 2 sequences, one generated with *Latent 1* and another with our method. They had to choose the win sequence, or deem them equivalent. If there was no good sequence, they could check the *No good sequence* checkbox. They could also leave an observation. Figure 11 shows the annotation task to rate sequences generated with our method and the ground-truth images, from a scale of 1 to 5. Additionally, the annotators could select that there was no good sequence, or leave an observation. Figure 12 shows the annotation guidelines for the task to rate sequences generated with our method and the ground-truth images.

E Failure Analysis of Extra Examples

E.1 Sequence Context Decoder

We further analysed the errors of the best method and present the results in Table 7. Hallucinations occurred in 3.9% of the generations, and the LLM copied the input into the output in 7.2% of the cases. Finally, it is interesting to see that the input was too complex in 6.2% of the cases, i.e., describing more actions than what is possible to depict in the image.

Error type	%
Hallucinations	3.9%
Complex step with many actions	6.2%
Copied input	7.2%

Table 7: The contribution of each error type to the overall performance.

E.2 Qualitative Analysis

Table 8 shows some example generations from the Sequence Context Decoder, each highlighting a particular behaviour of the model. In Example 1, we can see the model correctly identifying the ingredients from the context, caption$_{n-2}$, going two steps back, and integrating them in the final output. It also recognizes the plate from step$_n$ as the

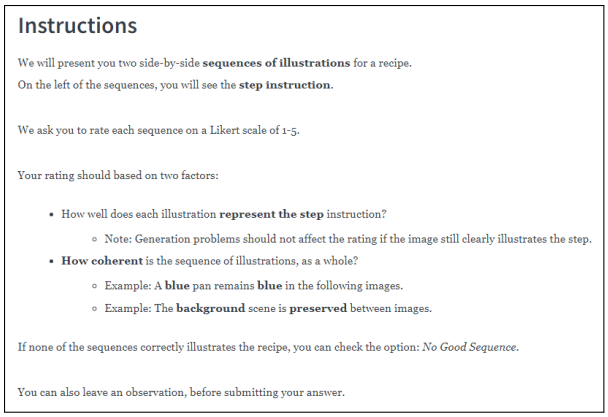

Figure 12: Annotation guidelines for the comparison between our method and the ground-truth images.

object containing the ingredients. This shows the potential in giving the model additional context to generate richer prompts. In Example 2, we can see that, despite being able to maintain the *red apples*, the model makes no explicit reference to their state, chopped up. This is still a limitation, which may lead to a wrongful representation of intact apples. In Example 3, we want to highlight two main aspects of the generation: we can see the model adding the *bowl of soup* from the context to the prompt, maintaining semantic coherence. We can also see that the model kept *lime juice*. This is correct, from the point of view of the task at hand, but shows the lack of understanding of what can be perceived in an image. We reason that this knowledge should come from the pretraining of the model, and not from our fine-tuning to this task. Example 4 shows an example of a depiction that is mostly correct, but misses a step of the sequence. The representation of the saucepan with black tea in it is plausible, but step_n indicates the saucepan should be removed from the heat. Finally, in Example 5, we see a very long step, with various actions. In this case, we consider it plausible for the model to pick one of these actions. This is a better result than attempting to represent them all, which would lead to an inadequate prompt. Despite this, this specific generation lacks some context, as the word *vegetables* is generic; it is important for the generated prompts to be specific, containing the ingredients mentioned in the context.

E.3 Challenges of Sequential Image Synthesis

To analyse the off-the-shelf behaviour of Stable Diffusion in the task at hand, we devised a few generation baselines. It is important to understand how the generation model used the text in the prompts, and where, and how, it failed. To do this, we assessed the performance of three image generation baselines in our target problem: (i) the current step instruction, (ii) the current step instruction plus the previous step instruction, and (iii) the current step instruction plus all previous step instructions. We generated images based on: (i) the current step instruction, (ii) the current step instruction plus the previous step instruction, and (iii) the current step instruction plus all

Isolated errors. We found problems which appeared consistently throughout different generations. These problems were not tied to the prompts or the concepts we were trying to generate, but downfalls of the image generation model. Among such examples were tiled images, and deformed objects, hands, and faces.

Representing Long Prompts. We found that long prompts were detrimental to the generations. These models are not prepared to represent many actions or objects, with some context being lost in the generation. Furthermore, these models fail to illustrate distant relationships in the texts, Figure 1.

Hence, training an LDM in this setting is not realistic due to the limitations of existing models. First, the amount of required data is not easily obtained, second, current LDMs are not designed to support long inputs, and third they do not support the notion of sequential generation.

Example 1: The Sequence Context Decoder was able to dereference the objects from caption$_{n-2}$ and create the correct caption for step$_n$.	
Caption$_{n-2}$	A wooden cutting board with ingredients for a salad including **avocado**, **crab meat**, and **grapes**
Step$_{n-1}$	Dress all ingredients except the arugula with one fourth cup dressing, dress the arugula separately with two tablespoon dressing.
Step$_n$	Plate the arugula first. Combine the rest of the ingredients, and add on top.
Output	A plate of salad with **avocado**, **crab meat**, and **grapes** on top.

Example 2: The state of objects is not always preserved in the generated caption.	
Caption$_{n-1}$	A purple plastic cup is sitting on a wooden table. Inside the cup, there is a mixture of black pepper, olive oil, brown sugar, curry powder, salt, and white vinegar. There are also some sultanas floating in the mixture. The cup is covered with plastic wrap.
Step$_n$	Get White Rice in a big bowl. Add in Celery and Scallion. In order to add a bit of color and some sweetness, **chop up Red Apple** and dust that up. Toss them together.
Output	A bowl of white rice is being prepared with celery, scallions, and **red apples**

Example 3: Some objects are correctly added to the generated caption.	
Caption$_{n-1}$	In the image, there is a pot of soup on the stove with chicken and vegetables in it. There is also a person's hand reaching into the pot to add more ingredients to the soup.
Step$_n$	Serve with White Rice on the bottom and Lime Wedge on the side. Or with tortillas for dipping into the soup.
Output	In the image, there is a **bowl of soup** with a tortilla in it. The soup has chicken, vegetables, and **lime juice** in it.

Example 4: Sequence of actions was not fully captured by the model.	
Step$_{n-1}$	Bring Water to boil in a medium saucepan.
Step$_n$	Once water has boiled, remove saucepan from heat and add the Black Tea. Steep for 15 minutes.
Output	a saucepan with black tea in it on a stove top

Example 5: When steps provide plenty of context, using a single step is sufficient.	
Step n	Halve fennel, remove stalk, and cut into fine strips. Finely dice onion. Peel garlic. Clean zucchini and bell pepper and cut into bite-sized pieces. Combine vegetables with rosemary, thyme, and some of the olive oil in a plastic bag. Close, shake, and marinate for approx. 20 min.
Output	A woman is cutting up vegetables on a cutting board with a knife and a peeler.

Table 8: Qualitative analysis of the Sequence Context Decoder results.

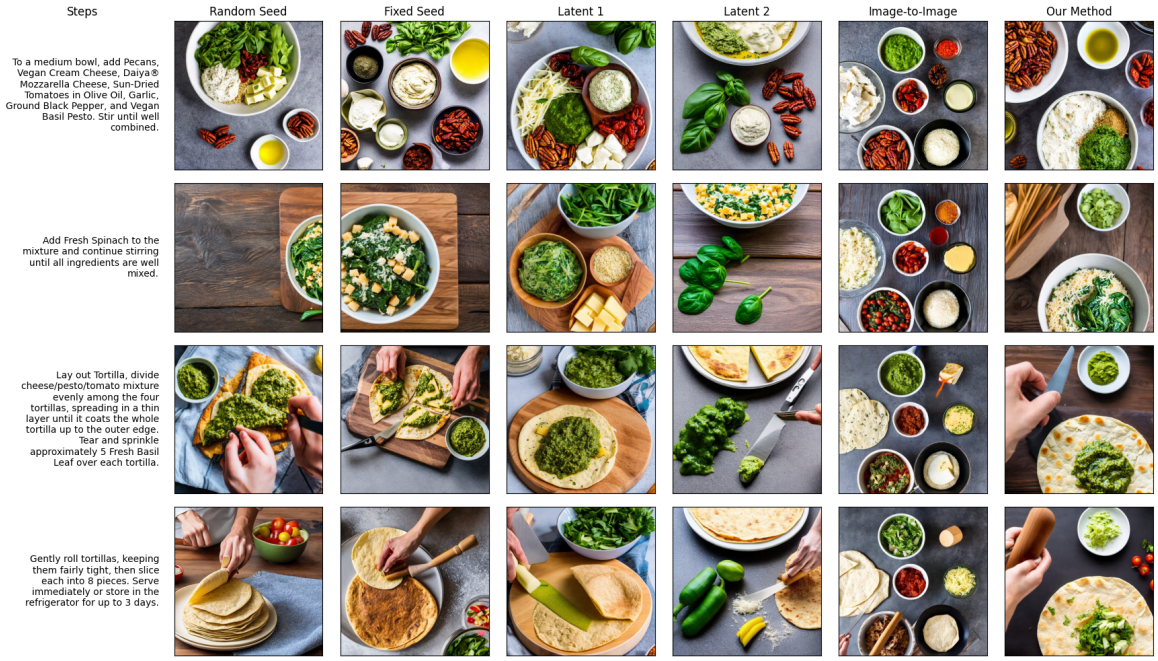

Figure 13: Examples of recipe illustrations with different methods for maintaining visual coherence.

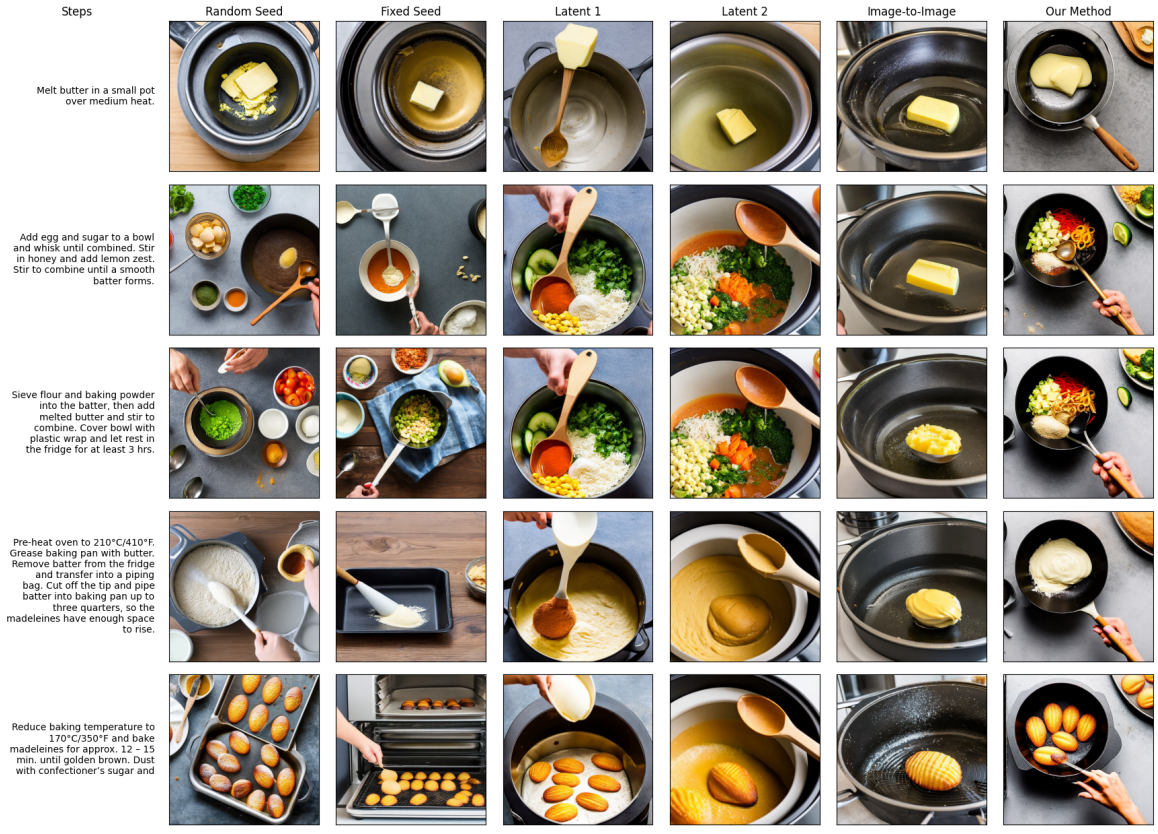

Figure 14: Examples of recipe illustrations with different methods for maintaining visual coherence.

Figure 15: Examples of recipe illustrations with different methods for maintaining visual coherence.

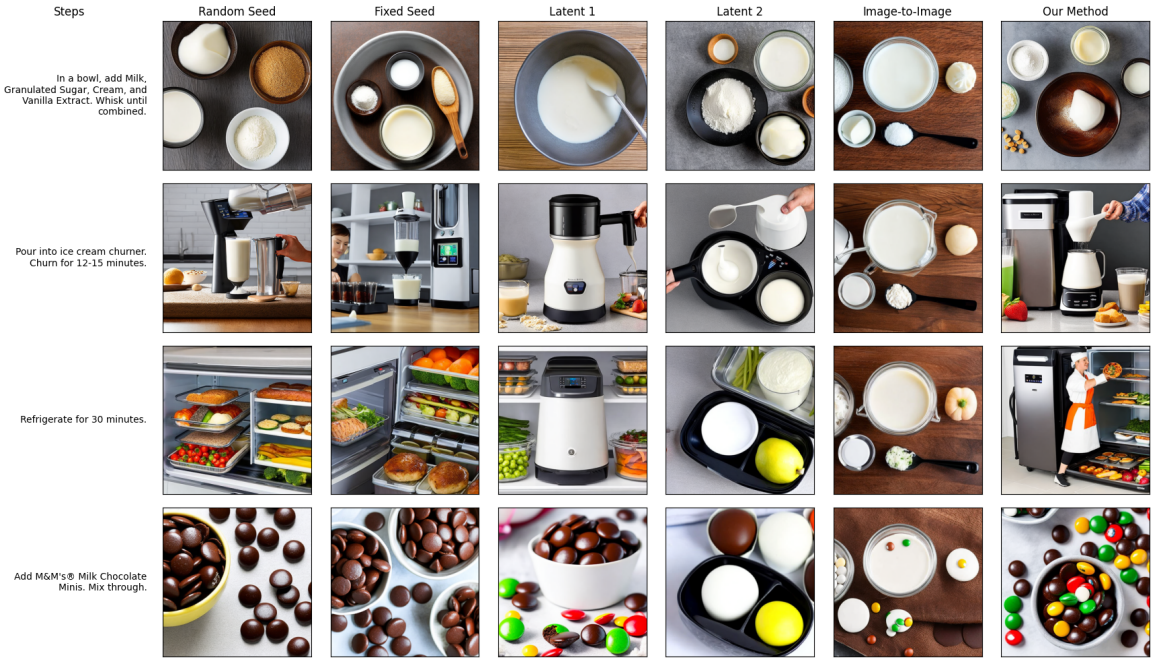

Figure 16: Examples of recipe illustrations with different methods for maintaining visual coherence.

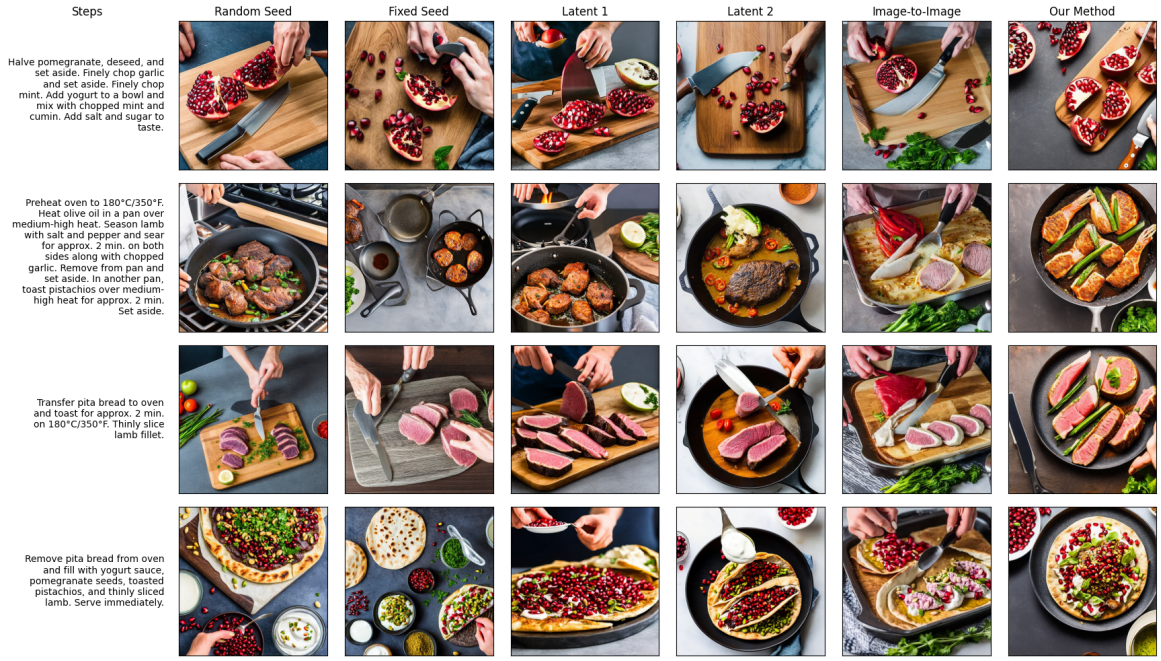

Figure 17: Examples of recipe illustrations with different methods for maintaining visual coherence.

Figure 18: Examples of recipe illustrations with different methods for maintaining visual coherence.

Figure 19: Examples of task illustrations with different methods for maintaining visual coherence.

Figure 20: Examples of task illustrations with different methods for maintaining visual coherence.

Figure 21: Example of a task that is very challenging to illustrate. We can see how the generated images still capture some of the more challenging elements of the steps, such as "Make a note of the longitude and latitude" in step 4, with the images showing a pen.

Figure 22: Examples of task illustrations with different methods for maintaining visual coherence.