11-777 Final Report

Nevan Giuliani* **Maxwell Jones*** **Alex Lyons*** **Hyunwoo Park***
{ngiulian, mjones2, alyons, hp2}@andrew.cmu.edu

Abstract

In this project we attempt to expand the task of visual storytelling by producing both story captions and images for the remainder of a story just given one initial frame (image and caption). Much previous work on this task focus on generating story captions from images or story images from captions. However, we present a pipeline for doing both simultaneously while ensuring that we have coherence between the story captions, coherence between the images, and alignment between the text and images. The key insight we make in our approach is that story captions are poor inputs to diffusion models so we generate both story captions as well as descriptive captions, the latter of which is used as input in image generation.

1 Introduction and Problem Definition (1-1.25 pages)

Visual storytelling is the task of producing a narrative from a photosteam (Wang et al., 2018) . In this work, we aim to extend this to what we call the task of story generation. We aim to produce both the story captions and story images of a story given just an initial text and caption. The story images and caption should maintain the same style and themes present in the initial frame. Therefore, in order to maintain coherence between the captions, coherence between the images, and alignment amongst the text and images, we create a multimodal pipeline that leverages conditional text and image generation to ensure the stories have the desired continuity. For the text generation, we finetuned a LlaVA (Liu et al., 2023) model which allows us to generate the next story caption as well as a descriptive version of the next caption which is fed to a Stable Diffusion model for creating the next images. For the image generation we finetuned

Figure 1: Example of VIST story, initial frame in red

Storygen (Liu et al., 2024a), a learning-based autoregressive model that allows for conditioning on a text prompt as well as previous image-caption pairs. We show both qualitatively and quantitatively that our pipeline leads to generated stories with high alignment in the previously mentioned areas.

All of the work we do is using the VIST (Huang et al., 2016) dataset which contains sequences of captions and images grouped into many stories. An example of such a story is shown in Figure 1. The initial frame is boxed in red.

Note that most previous work in visual storytelling only focused on the generation of the captions or images given the other. Methods like AREL (Wang et al., 2018) and PR-VIST (Hsu et al., 2021) generate text while methods like Storygen (Liu et al., 2024a) and AR-LDM (Pan et al., 2022) generate images. Our primary contributions are as follows:

- Allowing for the simultaneous generation of images and captions while maintaining the appropriate consistency
- Using a descriptive text version of a story caption as input to the image generation model as opposed to story captions
- Giving users tools to better create/iterate on visual stories with increased flexibility and creativity options from text to image models, while maintaining consistency/editability

*Everyone Contributed Equally – Alphabetical order

2 Related Work and Background (5 papers per person)

Related Datasets

Flintstones SV One related dataset is Flintstones SV (Maharana and Bansal, 2021) created from the original Flintstones dataset (Gupta et al., 2018). The Flintstones dataset consists of videos of 25,184 densely annotated video clips created from the program The Flintstones. All the clips are 3 seconds and typical depict some interaction between the characters. The Flintstones SV dataset was created from the Flintstones dataset by sampling a single frame from each dataset and then grouping frames from adjacent clips int groups of size 5. Sampling a single frame from each clip ensures that there is little redundancy between frames but using adjacent clips ensures good coherence.

Pororo SV Another related dataset is the Pororo SV (Li et al., 2019) created from the Pororo dataset (Kim et al., 2017). The original Pororo dataset was used for the task video question answering. This dataset consisted of clips from the TV show "Pororo the Little Penguin". Each clip is one second long and has a written description associated with it. Around 40 of these clips together make a story and each story has a set of question and answers associated with it. Pororo SV was created by sampling one frame from each clip and using the text description of the clip as the caption. Five of these frames were concatenated together to give the full dataset. In total there are 15,336 stories created which get split in 13,000 for training and 2,336 for testing.

Sequential Storytelling Image Dataset (SSID) A final related dataset is Sequential Storytelling Image Dataset (Malakan et al., 2023). SSID was created in order to address some of the issues with VIST. Specifically, they note that because VIST stories were created as a collection of individual photos in Flickr albums, they lack logical coherence. Therefore, the creators of SSID decided to create stories from open-source videos from the following three topics: narrative movies, lifestyle documentaries, and media appearances. Stories were created by selecting 5 frames from the video and then using Amazon Mechanical Turk to crowd source the story annotation process. A lot of quality checking was done to ensure the storytelling style. In total 3,473 stories were created which is much smaller than VIST.

Unimodal Baselines We include a unimodal baseline that focuses on the task of generating intermediate captions. To determine how well the text alone can predict the next text in the story (without the image), we used a text-only baseline. For this baseline, the input is a set of four out of five captions corresponding to a story, and the output is a fifth caption to finish the story (Figure 2).

The idea behind this baseline is that we want to investigate how much the story continuity relies on only the text compared to the text and images together. Our hypothesis is that both the images and text provide unique information that needs to be shared to edit/manipulate one coherent story, and a poorer performance in this baseline would show that image data is in fact needed.

The Visual Storytelling dataset paper (Huang et al., 2016) provides a solution to creating a story from images in the form of an RNN network that takes in each image frame and its previous embedding and outputs story captions autoregressively. Once the transformer architecture (Vaswani et al., 2017) became more prevelant for text based tasks, story generation methods switched to encoder/decoder methods.

The MPT StoryWriter model (MosaicML) is a decoder-transformer trained on text obtained by MosaicML, and then fine-tuned on large story-based text data. To evaluate the baseline, we compared the generated caption to the original fifth caption and corresponding image. For purely text-based metrics, we used the full testing SIS-VIST subset (approximately 5500 stories), and for text-image based metrics, we used a subset of approximately 2000 stories for storage efficiency. We expect this baseline to provide a reasonable output given the previous four captions, but not match with the expected caption and images given it has no access to image data. Results are included in the results section.

Prior Work We divide the prior work broadly into two groups: image to text and text to image

Image to Text These methods focused on generating a sequence of story captions for each image to create a coherent story.

One prior work in this category is Adversarial Reward Learning (AREL) (Wang et al., 2018) whose key insight is to use reinforcement learning to learn a reward function from human demonstra-

Figure 2: Unimodal Text Evaluation Example

tions. AREL consists of two major components a policy model and a reward model. The policy model is a CNN-RNN which generates words from a vocabulary given an image to produce a story. The reward model is a CNN-based model that uses n-gram features from the text as well as visual features from the images to compute an estimated reward for each substory within a story.

Another prior work in this category is PR-VIST (Hsu et al., 2021). Pr-VIST is a framework for story generation whose key insight is to build a relational graph between all elements in the input sequence of images and then find the optimal storyline in this graph to output a final story. The method can be split into two phases: story plotting and story reworking. In the phase of story plotting, the story graphed is created by extracting objects from all the images and linking then using a prepared knowledge graph. Also in this phase a storyline is produced from the story graph using UHop (Chen et al., 2019). In the phase of story reworking, a generator is used to actual create a story from the storyline and then a discriminator is used to classify the story as good or bad. After a few cycles, the generator learns to produce stories consist with those of humans.

Text to Image These methods focused on generating a sequence of image for each story caption that reflect the text prompt well and align well with each other.

An initial attempt at generating coherent stories from text comes from storyGAN(Li et al., 2019), a GAN based technique where the generator is conditioned on both previous frames and previous text captions. With the advent and rapid use of diffusion models(Ho et al., 2020; Song et al., 2021; Sohl-Dickstein et al., 2015) for image generation, subsequent works use these instead for initial model weights.

AR-LDM (Pan et al., 2022) is a model that fits in this category. The model takes as input a set of story captions and outputs an image for each caption. This was the first application of diffusion models to synthesize coherent stories using previous image-caption pairs. Previous work assumed conditional independence between different frames and used only the captions to generate each image. The architecture of AR-LDM utilizes a CLIP text encoder for the current caption and a BLIP multimodal encoder for the history of all previous image-caption pairs which allow it to generate context-aware images. The output image for the current frame is fed to the BLIP encoder in an autoregressive process.

The authors evaluated AR-LDM on three datasets including the VIST dataset that we will be using for our project. To measure performance they used a combination of FID score and human evaluation. The model achieve state of the art scores on both in the tasks of story visualization and story continuation. For human evaluation, reviewers were asked to compare stories generated by AR-LDM to models such as StoryDALLE on the axes of visual quality, relevance, and consistency. AR-LDM was overwhelmingly preferred on all axes across all 3 datasets tested. In terms of the automatic evaluation, AR-LDM also achieved a much lower (better) FID score across all datasets compared to all previous methods.

StoryGen (Liu et al., 2024b) is another model in this category. It is an autoregressive image generation model that conditions on both the current text prompt and previous image-caption pairs. The conditioning on the previous image-caption pairs ensures the same style is maintained throughout the story.

Improved Visual Story Generation with Adaptive Context Modeling (Feng et al., 2023) aims to leave the single step autoregressive framework from AR-LDM while trying to minimally change architecture. In order to achieve this goal, they add a component that chooses which of the previous frames are useful for attention, then doing cross attention with that frame (in practice, they use a weighted combination of frames). They also apply a final guidance step with previous frames at every diffusion step if they deem some previous frame to be very similar to their current frame.

Relevant techniques LoRA, or low rank adapters (Hu et al., 2021) is a technique used for finetuning both large language models and text to image models (Ryu, 2023). Instead of finetuning the entire weights of a model, LoRA postulates that the update should only be in a low rank space, and to enforce this constraint, the learn matrices $\Delta W = BA$, where $W \in \mathbb{R}^{n \times m}$ is the original weight matrix, and $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$ are the trainable weights, with $r \ll n, m$ to enforce the low rank constraint.

Another relevant technique is visual instruction tuning (Liu et al., 2024b),

3 Task Setup and Data

We consider a story to be a set of pairs $\{(i_k, c_k)\}_{k=1}^{n}$ where i_k is the k-th image, c_k is the k-th caption, and n is the length of the caption. The task is given (i_1, c_1) generate $\{(i_k, c_k)\}$ for all $k > 1$. Note that most of the stories in the VIST dataset have length $n = 5$. However, the task also requires that there is alignment amongst the i_k's, alignment amongst c_k's, and strong alignment between any pair (i_k, c_k) for any k in order to ensure that there is a valid story. In order to measure the image-text alignment we used CLIP score (Radford et al., 2021). To measure the quality of the generated captions, we used roBERTa embedding similarity (Liu et al., 2019) with the ground truth captions.

As stated above we are using the VIST (Huang et al., 2016) dataset for training and evaluation of our pipeline. The dataset currently contains around 167,528 unique images and 40,155 stories which is a sequence of around four or five images that generally have similar content. The dataset was constructed by querying the Flickr API for albums that matched with terms the authors considered "storyable". The authors also ensured that the albums had between 10 and 50 images all taken with 48 hours of each other which helps ensure they have a common style. After finding these albums, they leverage the Amazon Mechanical Turk in order to crowd source the creation of stories.

Note that within VIST there are actually two separate types of annotations: Story-in-sequence (SIS) and description-in-isolation (DII). The captions in SIS are supposed to tell a story and use much more figurative language. On the other hand, DII captions provide a more descriptive depiction of each image and does not consider the context of a story. This idea of descriptive text and storytelling text is crucial to the rest of the work.

4 Baselines

Results for all baselines are shown in section 6.

4.1 Image to text baselines

4.1.1 BLIP

For the task of generating a story from a set of images we included a naive baseline in which we used BLIP to generate a caption for each given image. This method is naive for two reasons. One is that there is no notion of a coherent story as only the current image is used to generate the output caption. Therefore, it essentially reduces the task of visual storytelling to that of image captioning. The second is that BLIP outputs text that is much more descriptive in nature than the text for storytelling is supposed to be. However, in terms of generating captions aligned with the story images it does represent a valid baseline. It also gives us a sense how much models like AREL and PR-VIST, which are tailored to storytelling, improve over generic captioning models. We used a subset of around 30% of the SIS-VIST subset and evaluated the captions outputted by BLIP in two ways: the textual similarity between it and the ground truth captions and the text-image alignment with the original image. For evaluating textual similarity we used cosine distance of SRoBERTa sentence embeddings as well as the n-gram metric METEOR. For image text alignment we used CLIP score.

4.1.2 PR-VIST

A description of PR-VIST is provided in section 2. We evaluated PR-VIST on the entire SIS test set. For evaluation we again measured the text similarity to the ground truth captions using cosine distance of the sRoBERTa embeddings and METEOR. Note that because PR-VIST generates the entire output story at once rather than frame by frame, we are unable to measure the text-image alignment using CLIP score in the way that we were for some of the simpler baselines.

4.1.3 AREL

A description of AREL is provided in section 2. We used the entire SIS test set for the evaluation of AREL. As with the other text generation baselines we evaluate the similarity to the ground truth captions with cosine distance and METEOR.

4.2 Text to Image baselines

4.2.1 Independent Text-to-image methods

We also consider the task of generating images from a set of text captions. For this task, we consider the text captions as given by the descriptive caption set of the Visual Storytelling dataset. While these captions do not form a cohesive story from frame to frame, they better describe each frame, which is more conducive for use in a text-to-image model. For image generation, we consider three different Stable Diffusion models. For all methods, we consider each frame as an independent image,

First, we consider SD 1.5 (Rombach et al., 2022), the original Stable Diffusion model trained on internet scale data. Next, we consider the second, larger version SDXL (Podell et al., 2023), which produces higher fidelity images at the cost of higher compute. Finaly, we use Stable Diffusion Turbo (Sauer et al., 2023), a one step model distilled from the original Stable Diffusion model (Rombach et al., 2022). We chose this model as a final test in order to speed up image generation while not sacrificing a large amount of image quality, since we evaluate on the test set, containing thousands of images.

4.2.2 Image editing and insertion

To produce realistic newly generated captions based on the stories within the testing data, we utilized prompt engineering with ChatGPT. We formatted each set of individual captions into one coherent story, and inputted it along with additional information specific to the baseline's requirement. For the editing baseline, we asked ChatGPT to replace the second sentence with another sentence that matches the context of the rest of the story. For the insertion baseline, we asked ChatGPT to generate a new sentence between the second and third captions that makes sense in the story. These were then used as "edited" or "inserted" captions respectively to guide their respective image generation tasks.

For the task of image editing, we use SD Turbo (Sauer et al., 2023). We add 50 percent noise to the original image, and denoise with the new caption produced by chat GPT. Again, we use the descriptive captions as a base. For the task of new panel insertion, we use SDXL Turbo (Sauer et al., 2023) for image generation using the new captions provided by ChatGPT. Of note for the image generation methods is that they do not condition on other images in the story, as the original stable

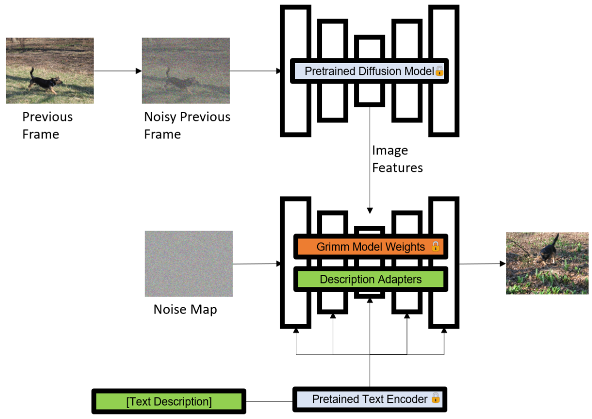

Figure 3: Architecture for Intlligent Grimm (**?**) with LoRA added for better description learning

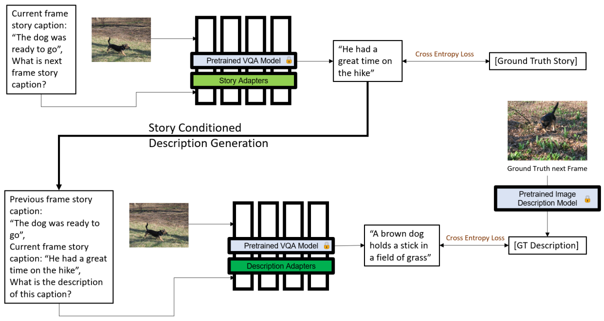

Figure 4: Architecture for VQA learning. Note that descriptions are generated in a story-aware manner

diffusion methods have no ability to condition on other images, and image conditioning methods on stable diffusion (Mou et al., 2023) usually control style/edge maps as opposed to our tasks. Competitive baselines that condition on previous story frames are very computationally costly for training, indicating that it is not trivial to alter a stable diffusion model to take in previous frames of a story and output a new frame.

5 Proposed Model (>1 page)

We created a three component system. The first component takes in the previous frame(s) and text for the current frame, and outputs the next frame (Figure 3). The second component takes in the previous frame and story text and outputs story text. The third component takes in the previous frame, story, caption, and newly generated story caption, and generates a description (Figure 4). In previous SOTA work for frame generation, the current frame's story label along with used as an input to a diffusion model to create the next image. Our key insight is that the story label may not be the optimal label for Image generation. In Figure 4, the current story label is “He had a great time on the hike”. While this makes sense in the context of the story, it does not fit as the input to a diffusion model outputting a brown dog with a stick in his mouth. To mitigate this mismatch, we decided finetuning the VQA model LlaVA (Liu et al., 2023) to output BOTH a story text and a descriptive text for every frame. We actually finetuned two separate models, one that outputs the descriptive text and one that outputs the the story text. In the final pipeline, the descriptive text will then be used along with previous frames to condition a diffusion model producing the next image.

The diffusion model we used was StoryGen (Liu et al., 2024a) Since we build off of previous work that takes story captions as input the the diffusion model (Liu et al., 2024a). Since the diffusion model requires rich text to generate high quality images, the story captions in this work were very long and descriptive, which may not be optimal in all settings. To mitigate this, we also finetune the diffusion model with descriptive text from the ground truth descriptive labels instead of the story text. Secondly, StoryGen introduces a mechanism for previous image conditioning. In order to get features from the previous images for use by the diffusion model, the previous images themselves are slightly noised and denoised with their own descriptive text by the same finetuned model, and the model features from previous frame denoising are used as conditioning for the current frame denoising. A new cross attention block is introduced for this task. For all experiments, we pass in the previous three frames features as conditioning, and pad with blank images if there are less than three images prior to the current image being generated.

5.1 Loss functions

For finetuning the two LlaVA models we used cross entropy loss on the output text given by the model with the ground truth text. For finetuning the story text model, the model was prompted with the the image and story caption for the previous frame and asked what the next caption should be. The prompt configutation is shown below .

```
{
"id": unique_id,
"image": f"{img}.jpg",
"conversations": [
  {
   "from": "human",
   "value": "Current frame story
    caption: '" + story_text_curr
    + "', What is the next frame
    story caption?"
  },
  {
    "from": "gpt",
     "value": story_text_next
  }
]
}
```

For finetuning the descriptive text model, the model was prompted with the previous image, previous story caption, and the story caption for the current frame. The model is asked what the descriptive text for the current frame should be. Note that when training we use the ground truth sis captions for the story caption of the current frame rather than the other LlaVA model. Therefore, the training of the two models is completely separate. The prompt configuration is shown below.

```
{
"id": unique_id,
"image": f"{img}.jpg",
"conversations": [
  {
     "from": "human",
     "value": "Previous frame
     story caption: \'" +
     story_text_curr + "\',
     Current frame story caption:
     \'"  + story_text_next
     + "\', What is the
     description of this
     caption?"
  },
  {
     "from": "gpt",
     "value": description_text
  }
]
}
```

For finetuning the StoryGen model, we follow the finetuning procedure of IntelligentGrim(Liu et al., 2024a) and standard diffusion models (Rombach et al., 2022). Given a noisy input frame and text caption, the diffusion model denoises the input to yield a clean image. From here, Mean Squared Error loss is applied w.r.t. the ground truth clean image, and model parameters are updated. We fine-tune using the Visual Storytelling dataset (Huang et al., 2016) for 50000 steps with a batch size of 12 across 3 NVIDIA A5000 machines.

5.2 Changes to training data

When preprocessing the training dataset we filtered out any stories that did not have exactly 5 frames. Additionally, we resized all the images to 512×512 and converted them to jpegs for storage reasons.

5.3 Hyperparameters and their effects

LlaVA Model Hyperparameters

- Training data size: For both the story text LlaVA model and descriptive text LlaVA model we trained once where we used 2,000 examples from the VIST train set and many times where we used 10,000 example from the VIST train set. We then evaluated on the full VIST evaluation dataset by measuring the SRoBERTa cosine similarity to the ground truth stories. We found that the models trained on 10,000 examples performed better as seen in results section.

- Temperature: We also varied the temperature governing the shape of the distribution used to sample the tokens when generating the output. We got the best results with the smallest temperatures or a very sharp distribution over the output tokens.

Storygen Model Hyperparameters

- Classifier Free Guidance: Since we are using a diffusion model, a common hyperparameter is classifier free guidance (Ho and Salimans, 2022). This allows a tradeoff between diversity of images generated with a text prompt and image quality, where quality increases and diversity decreases as the guidance scale increases.

- Image guidance: this parameter controlled the degree to which we condition on the previous frame when generating the new frame. Similar to classifier free guidance, image guidance allows tradeoffs between diversity of the next frame and consistency with previous frames. As seen in the results section varying this parameter did not affect the similarity to the ground truth images too much and only slightly increased the similarity to previous images.

6 Results (1 page)

Text Generation: Ours vs Baselines All of the baseline and model text results are shown in Table 1. Compared to MPT, our models outperform in all three metrics. One of the likely reasons for this is that MPT is unimodal (only text) and our model uses both caption and image information. Therefore, our model has more story context, and therefore can produce content more closely aligning with the original story. Compared to BLIP, our model performed worse for METEOR and CLIP, but performed better for SRoBERTa. One explanation for the underperformance on the first two metrics is that it is likely the images themselves contain enough story objects that act as key words in the story, and thus more closely align with the story. However, unlike the first two metrics, the SRoBERTa metric contains information from all the story captions. Therefore, unlike BLIP our model has the ability to carry information across frames, and thus using more story context is able to achieve a higher SRoBERTa score. Compared to the two SOTA models, our model performs worse on METEOR and better on SRoBERTa. As will be shown later, our model sometimes produces more generic outputs, which could explain why when comparing single frames our model performs worse compared to the SOTA models for METEOR. However, similarly to the BLIP case, since our model is based on a well-established VLM, it is able to more effectively capture story-wide information, which is shown by how the story-wide metric SRoBERTa shows our model performs better.

Text Generation: Autoregressive vs Non-Autoregressive When comparing the text results generated autoregressively compared to non-autoregressively, the performance decreases. However, this is expected given that since both the captions and images are only grounded to the GT data via the first frame and caption, the model has creative freedom to generated a completely new story. Therefore, while the non-autoregressive results are forced to be closer to the ground truth captions, the new captions follow a story that only starts the same as the ground truth. Thus, when comparing the autoregressive results against the ground truth, the similarity metrics decrease as expected. They

Methods	METEOR ↑	Avg CLIP Cos Sim. ↑	SRoBERTa Cos Sim. ↑
MPT (MosaicML)	0.066	0.201	0.208
BLIP (Li et al., 2022)	0.090	0.289	0.266
AREL (Wang et al., 2018)	0.353	-	0.478
PR-VIST (Hsu et al., 2021)	0.176	-	0.462
Ours - 2k (temp = 0.2)	0.071	0.239	0.555
Ours - 10k (temp = 0.2)	0.076	0.240	0.586
Ours - 10k (temp = 0.5)	0.072	0.236	0.567
Ours - 10k (temp = 1)	0.072	0.235	0.570
Ours - 10k (temp = 2)	0.042	0.216	0.446
Ours - 2k (Autoregressive)	0.054	-	0.357

Table 1: Text based metrics

Methods	CLIP Text-Image Sim ↑	Previous Image Sim ↑	Ground Truth Image Sim ↑	Total Story Image Sim ↑
Baseline Images	0.295	.679	N/A	.675
SDXL-Turbo (GT caption image generation)	0.316	0.572	0.688	0.591
SD-1.5	0.309	0.575	0.681	0.589
IntelligentGrimm (no finetuning)	0.278	0.592	0.650	616
Ours (Non-autoregressive, image_guidance=5.0)	0.300	0.646	0.724	0.642
Ours (Non-autoregressive, image_guidance=3.5)	0.306	0.639	0.726	0.632
Ours (Fully-autoregressive, image_guidance 5.0)	0.196	0.626	0.581	0.740

Table 2: Image Generation Results

are only calculated because if they were orders of magnitude lower, then most likely something would be wrong with the content generation itself (for example, not creating valid sentences). Since the autoregressive results aren't grounded to any ground truth data, they should be mainly evaluated using the intrinsic metrics and visually by example.

Image Generation Metrics We consider a few metrics when measuring story performance for image generation. The first is similarity to the description text provided. Stories should follow the description provided by either a user or another model for the generation process. The next is previous image similarity. A cohesive story should have subsequent frames similar to their previous values. We also measure ground truth image similarity, with the goal being to be as similar as possible to the ground truth images. Finally, we measure total story similarity, where we take the average similarity between any two frames in a given story. For all similarity metrics, we consider cosine similarity between CLIP embeddings.

Image Generation: Baselines We consider a fewWe consider a few baselines when running image generation. The most naive baseline is simply running stable diffusion 1.5 (Rombach et al., 2022) or SDXL-turbo (Sauer et al., 2023) on ground truth descriptions. Since these models are very powerful, we expect to have high text to image similarity, but then worse performance on statistics that measure story cohesiveness. As expected, the text to image similarity for these models are on par with other results from this section, however they suffer when measuring metrics like ground truth similarity, previous image similarity, and total story image similarity. We also measure the baseline images themselves, to get an understanding of optimal total story/previous image similarities.

We consider a few different versions of our model. In the Non-autogregressive case, we condition each subsequent frame generation with the ground truth images/descriptions, and pass in the ground truth description to the model. We see that both image guidances produce higher text-image similarity, previous images similarity, and ground truth image similarity when compared to baselines. We also see a higher total story image similarity, where the total story consists of 4 frames generated in this way plus the original ground truth frame.

Secondly, the higher image guidance method performs slightly better with respect to all story coherence metrics, but overall performs very similarly to the other method.

We also comsider the intelligentGrimm (Liu et al., 2024a), but without our finetuning component. Surprisingly, the ground truth image similarity for this method actually performs worse that the baseline methods that don't condition on previous images. We surmise this is because the intelligentgrimm method is finetuned on cartoon-type data, so as a result images may come out looking unrealistic, also lowering clip similarity to the description text. This is further proven by results in Figure 10. With that being said, images are still more similar to the previous image than the non-conditioning baseline, indicating that this method without finetuning is still taking into account the previous frames.

In the autoregressive case, we condition each subsequent frame with the previously generated frames and captions, and generate next captions with our own llava finetuned model. As a result, over the course of the story we see much lower correspondence to the ground truth images and ground truth text descriptions, but comparable values to our other method versions for previous image similarity. Surprisingly, the total story image similarity metrics is significantly higher in this case. This could be because the model is unlikely to switch scenes between frames which happens fairly regularly in the dataset itself, causing many frames to look similar to each other. See Figure 11 for examples.

7 Analysis (2 pages)

This section should include plots. For example, how key metrics vary with a specific hyperparameter, task complexity, etc.

7.1 Intrinsic Metrics

These are not the task itself (they might overlap with auxiliary losses) but are skills the model should have.

Intrinsic Metric 1 In order to determine how aligned a story is with the goal of storytelling instead of just giving text descriptions, we decided to include a metric counting the number of pronouns in the generated captions. Thus for each model, we computed the average number of pronouns in each SIS story. The results are reported in Table 3. The hypothesis is that models which just describe the content of the images will include fewer pronouns than models which have been trained to output coherent stories. The average number of pronouns per story in the SIS dataset was 2.73.

- MPT: Significantly more pronouns than the ground truth stories
- BLIP: Significantly fewer pronouns than ground truth stories. This makes sense as the model aims to describe the image rather than tell a story.
- AREL: Similar number of pronouns in generated stories as the ground truth.
- PR-VIST: Slightly more pronouns that the ground truth.
- Our Method: On average slightly less pronouns than the ground truth.

Intrinsic Metric 2 One important aspect of the text generation for each image is that it should read like a story, instead of just a description of the image. Therefore, we compared the words used in the generated text to the ground truth texts. We aimed to do two comparisons: A comparison to the story-based captions (SIS), and the description-based captions (DII). In the ideal case, the generated text will be very similar to the SIS text given they represent the type of story-based descriptions we desired. Furthermore, the similarity to the DII should be worse, but not too poor given they are describing the same concepts.

In order to compare the words used, we utilized the KL-divergence between the words used across all the captions between the different sets of text. Specifically, we computed the KL-divergence between generated text and SIS captions, and between the generated text and the DII captions. Additionally, we applied smoothing to adjust for the cases where words appear in one set of text but not the other to prevent the divergence from reaching infinity. We evaluated the five text-producing models using this metric: MPT, BLIP, AREL, PR-VIST, and our best finetuned LlaVA model with the training set size of 10,000 and temperature of 0.2. The overlap between these distributions can be seen in Figures 5, 6, 7, 8, 9.

The model with the best (smallest) KL-divergence was MPT with scores of 0.84 and 1.71 for the SIS and DII comparisons respectively. Its good performance for SIS makes sense, given that its only input is other SIS captions. Since it is a GPT model, which aims to predict the appropriate text response given text input, it should predict words similar to the input words. Furthermore, the switch from using story-based words to more descriptive words appropriately causes a drop in similarity when comparing the generated text to the DII captions.

The next best performing model was our model with a score of 1.45 for SIS and 1.40 for DII. For DII, our model conditions on the previous story text, previous image, and current story text. Having access to all these pieces of information likely enhanced its ability to produce descriptive text that both matched the style of the previous descriptive text and matches the current story caption which is why the DII score is quite good. The SIS performance is also very strong likely due to the same reasons as MPT.

The BLIP model produced scores of 6.75 and 2.27 for SIS and DII respectively. The poor similarity between the generated captions and SIS captions is most likely due to the fact that BLIP is an image description model, meaning that it won't have "story-like" words (such as pronouns as mentioned previously). Consequently, it makes sense that the DII captions were more similar given that they are description-based rather than story-based.

AREL produced scores of 4.53 and 6.10 for SIS and DII respectively. As expected, the SIS score was better than the DII score since the model is trained to produce story-like output. However, its poor overall performance in both SIS and DII can be explained by common undesired behavior. First, the model often produced text that repeated itself, leading to an unrealistic story. Second, many of the outputted captions followed the format of "This is a picture of [word]." This is both unstory-like, and also is even unlike a description given that descriptions usually describe the image, instead of just stating that it is describing an object. Examples of both of these behaviors will be shown in the qualitative analysis for text baselines later.

Finally, PR-VIST produced scores of 2.21 and 3.15 for SIS and DII respectively. Similarly to MPT and AREL, the model is designed to produce story-like text, and therefore makes sense that the SIS similarity was better than the DII similarity. One notable difference between the generated text and ground truth text that could partially explain a loss in performance was the fact that instead of using proper nouns, PR-VIST outputted text to fill in. For example, instead of saying a woman's name, it would output "[female]."

Overall, it seems like the main factor that affected the word distribution similarity was if other words from the same distribution were used as input, given that MPT and our model were the only ones that did this, and thus greatly outperformed the three other models. Therefore, it is imperative that our model utilizes training text from SIS and DII efficiently.

Intrinsic Metric 3 In order to determine if generated images generally look realistic and come from a similar distribution to ground truth images, we use bounding box detection as a proxy. Specifically, we consider a bounding box-specific metric:

1. bounding box number. Specifically, we measure the number of bounding boxes in each image and compare results via histograms

We note that our method does not heavily reduce the number of bounding boxes found in stable diffusion images, which indicates that the images being generated still fall under the distribution of natural images given descriptions. This is a good sign that the generated stoires are coherent as standalone individual images. We see that SDXL-Turbo produces slightly more bounding boxes than other methods, which may be a direct result of the fact that it is a higher fidelity model trained on a larger and more carefully curated dataset. Finally, we see that the intelligentGrimm method performs worst in terms of bounding box predictions, which could be a result of cartoonish outputs.

Methods	Pronouns↑	Word Divergence (SIS/DII) ↓	Objects ↑
MPT (MosaicML)	6.93	0.84/1.71	-
BLIP (Li et al., 2022)	0.45	6.75/2.27	-
AREL (Wang et al., 2018)	2.32	4.53/6.10	-
PR-VIST (Hsu et al., 2021)	3.85	2.21/3.15	-
SD 1.5 ((Rombach et al., 2022))	-	-	4.58
SDXL-Turbo ((Sauer et al., 2023))	-	-	4.69
IntelligentGrimm ((Liu et al., 2024a))	-	-	3.99
Ours (Non-autoregressive, image_guidance=5.0)	-	-	4.53
Ours (Non-autoregressive, image_guidance=3.5)	-	-	4.54
Ours (Fully-autoregressive, image_guidance 5.0)	-	-	4.57
Ours - 2k (temp = 0.2)	1.75	1.57/1.93	-
Ours - 10k (temp = 0.2)	1.71	1.45/1.40	-
Ours - 10k (temp = 0.5)	1.97	1.05/-	-
Ours - 10k (temp = 1)	1.96	1.02/-	-
Ours - 10k (temp = 2)	6.21	0.56/-	-
Ours (Autoregressive)	1.89	3.46/2.76	-

Table 3: Intrinsic Metrics for Baselines and Our Models

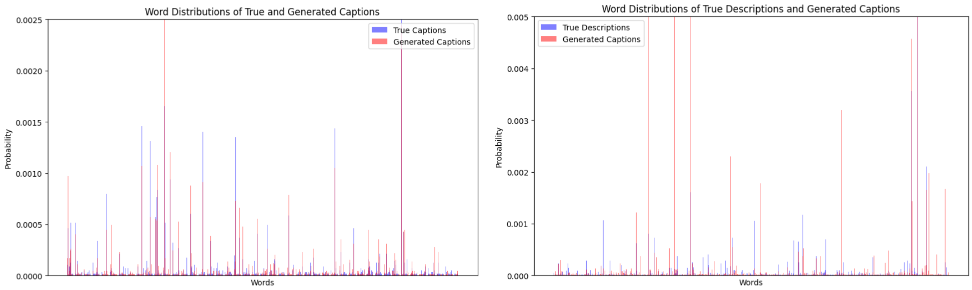

Figure 5: Overlap in word distributions between MPT-generated captions and SIS (left) and DII (right).

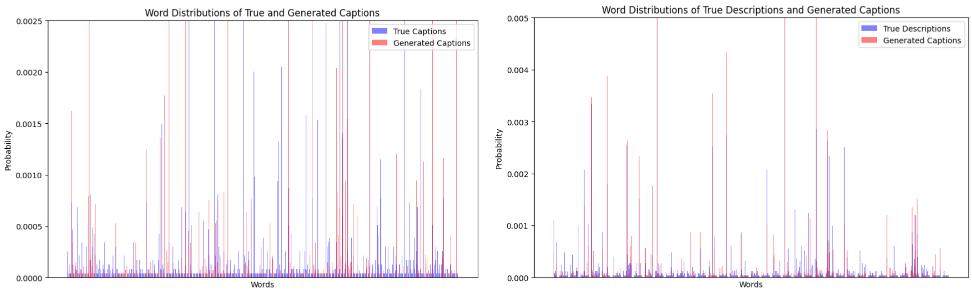

Figure 6: Overlap in word distributions between BLIP-generated captions and SIS (left) and DII (right).

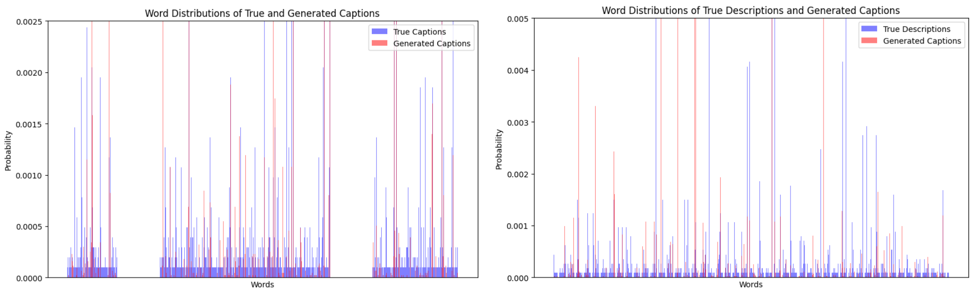

Figure 7: Overlap in word distributions between AREL-generated captions and SIS (left) and DII (right).

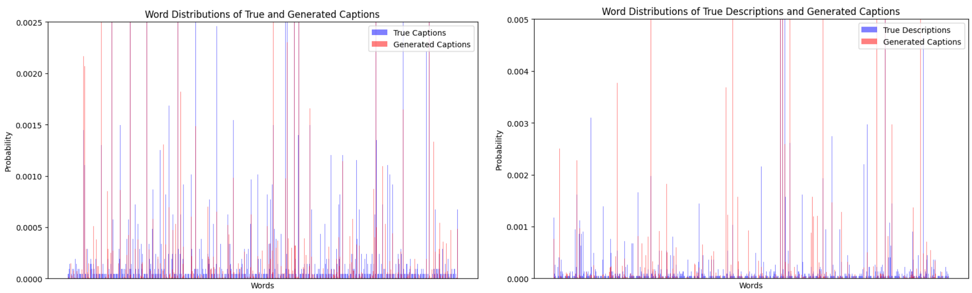

Figure 8: Overlap in word distributions between PR-VIST-generated captions and SIS (left) and DII (right).

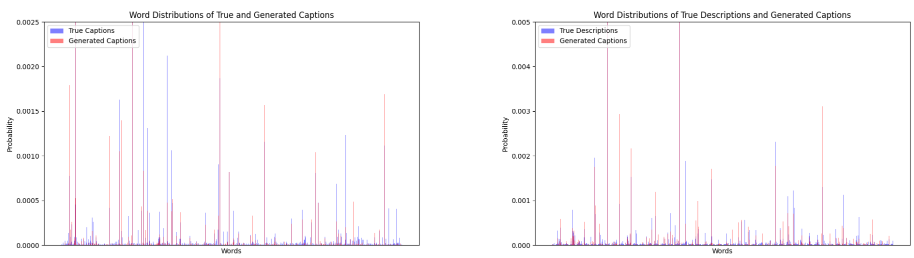

Figure 9: Overlap in word distributions between our model (10k, temp=0.2) captions and SIS (left) and DII (right).

7.2 Qualitative Analysis and Examples (full page tables – multiple pages for most projects)

In Figure 10, we consider generating the third frame of a story from the VIST validation set. when using Stable Diffusion 1.5 (Rombach et al., 2022) and SDXL-Turvo (Sauer et al., 2023), we condition only on the generation prompt. For our method and IntelligentGrimm(Liu et al., 2024a), we also condition on the previous frames/captions. First, we notice that the methods that only condition on the description have no notion of story coherence. In the first row, both SD 1.5 and SDXL-Turbo fail to output a woman consistent with the previous frame. However, some frames do not need to be aware of previous frames to generate a coherent output, as can be seen with these two baselines in the second and third row, where they output reasonable frames as compared to the ground truth third story frame. Next, note that intelligentGrimm was trained to output stories as seen in comic books or other media, and as a result can tend to output cartoonish images, as can be seen in rows 2 and 3. Finally, our method succeeds in keeping the clothing and hair type of the woman the same in row one of the image, as well as the hat on the boy in the last row. However, since our model is still based on SD 1.5 and not SDXL, it can struggle with multi-person scenes, as can be seen by the poor quality of the peoples faces and the background in row 4, as compared with the more coherent image produced by SDXL-Turbo. In figure 11 we examine our full pipeline starting from a single image/caption, where we generate both next captions and next images for the remaining 4 frames autoregressively. Looking at the two stories our model generated in Figure 11, the first thing to notice is that we are able to generate consistent lighting/scenes/characters in the first part of the story. With this being said, we see that there were many repeated or very similar captions. In the story on the bottom, the caption changes from “The crowd was getting excited” in frame 2 to “The crowd was getting more excited” in frame 3. This is due to the autoregressive nature of the model and conditioning to heavily on the previous caption. Perhaps if we conditioned on all the previous captions instead of just the most recent one, we would get more diversity in captions as we move along the story. In the second story, the fact that the scene is a horse race is lost after frame 2, since only a picture of a crowd is shown and the llava text description model is only conditioned on a single prevous frame. This causes the failure case of switching to a baseball game. From here, the scene switches from a baseball game to a football game, highlighting the fact that our method can sometimes generate inconsistent stories across frames.

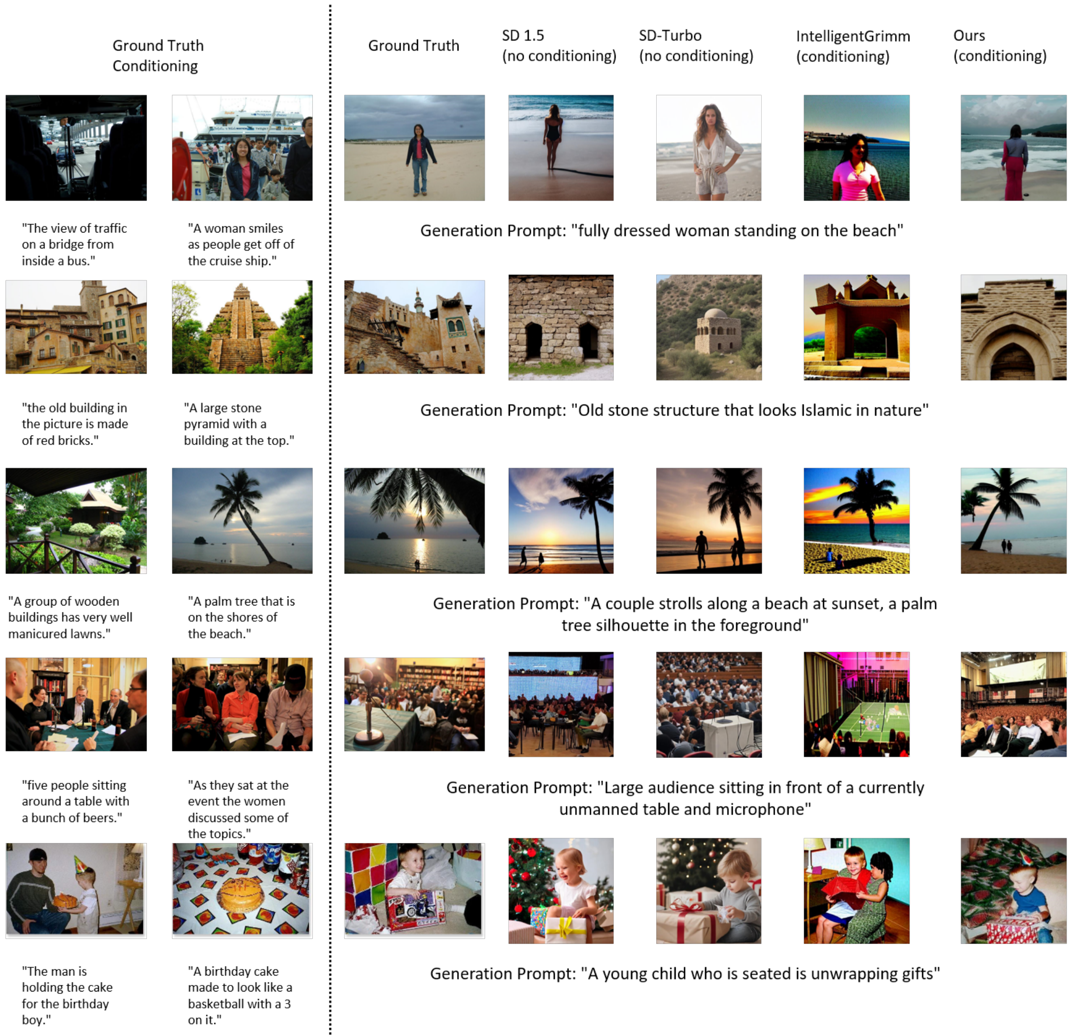

Figure 10: Comparison to baselines. We consider two ground truth story images with captions as input, and consider the output with a new ground truth story description from all models

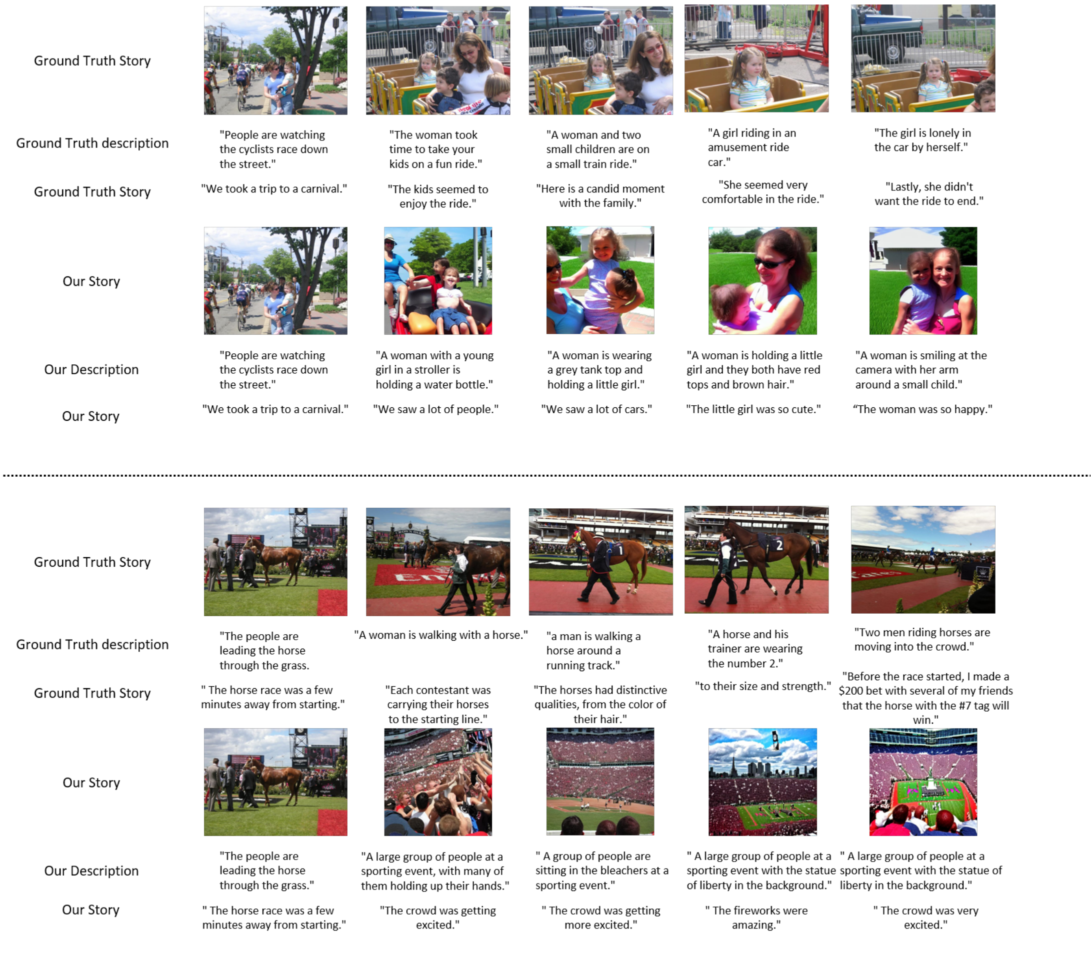

Figure 11: Autoregressive Story Generation compared to baseline

Example Input (Using GT VIST Caption and Image)	10k (temp = 0.2)	10k (temp = 0.5)	10k (temp = 1)	10k (temp = 2)	Comparison/Failures
 "Current frame story caption: 'We gathered up some fire works to set off at dark.', What is the next frame story caption?"	"We had a great time."	"There was a bird watching us as we set up the fireworks."	"The kids had a great time setting them off."	"There we are, having lots of blanks all set! And the colors on them"	In this case, when the temperature was too low (t=0.2), the text was very generic and didn't include anything specific to the story. When the temperature was mid-range (t=0.5 and t=1), the stories were both creative and specific to the current context (fireworks). When the temperature was too high (t=2), the caption started to sound too conversational compared to story-like, and was too long so it got truncated.
 "Current frame story caption: 'Some of the parents gathered in the backyard to supervise other children who wanted to swim.', What is the next frame story caption?"	"The children were having fun swimming in the pool."	"The children were having fun on the boats."	"A little girl and her mom looked on as they watched the other children swim."	"And they were there while the twilight fell over them from time passed to time untold and they all wanted one more turn on tha swingset outside under its yellow twilights"	The t=0.2 result gave a reasonable continuation of the story based on the caption and image. The t=0.5 and t=1 results started to give misinformation relative to the image, such as the fact that a boat could fit in that pool, or that there was a little girl on the deck. The t=2 caption was very long and convoluted.
 "Current frame story caption: 'These people went to a lot of expense and time to entertain us!', What is the next frame story caption?"	"The lights are so bright that it is hard to see the house."	"This is a beautiful sight!"	"The lights are on and the people are enjoying the display."	"This light bulb tree light shows off the winter theme beautifully."	The t=0.2 result goes against the input premise that they are enjoying the decorations. The t=0.5 and t=1 captions are okay, but somewhat generic. The t=2 caption continues the excitement theme from the input caption, and mentions specifically the tree in the image at the forefront.

Table 4: Qualitative Analysis and Examples for Story Caption Model Outputs

8 Future work and Limitations (1 page)

One area in which our model doesn't work well is remembering long-term story information. For caption generation, our model currently only uses information from the previous frames, and not all the frames before. Therefore, if information is both lost in the caption and not kept in the image itself, then that information is lost for the rest of the story. To improve on maintaining information long-term, we train the whole model autoregressively instead of frame-by-frame, and for each input use all the captions from the previously generated frames. This would lead to less stability when training, but would allow information the model to have the capacity to manage long term information from the story.

In addressing the challenge of maintaining long-term story coherence in visual storytelling, State Space Models (SSMs) present a promising avenue. SSMs are particularly adept at efficiently modeling long sequences. By conceptualizing each frame of a story as a state influenced by both its previous state and a control input (in this case, the narrative elements or actions depicted in each frame), SSMs can effectively capture the dynamic changes throughout the story. This approach allows for a more nuanced tracking of narrative elements, ensuring that details are neither lost in transition between frames nor misrepresented as the story progresses. Moreover, the inherent structure of SSMs facilitates the modeling of hidden states that may not be directly observable through the images or captions but are crucial for the continuity and coherence of the storyline. By integrating SSMs, we can enhance our model's ability to remember and correctly reference key story elements such as character developments, thematic shifts, and plot advancements over longer sequences, thereby resolving one of the critical limitations of our current system.

Another limitation our model has it is has trouble using names in the story captions. When the story starts with mentioning a specific person by name, the model usually ignores this information in the next frame and refers to them as a generic person instead. This produces more generic and less story-like descriptions. One possible method for improving on this issue would be to augment the VIST training data to include more names. For example, one could go through each training example and replace pronouns with made up names, and therefore providing the model with more examples of how to transfer name information when generating new captions.

To further enhance our model's ability to consistently use and remember names throughout a story, integrating Named Entity Recognition (NER) technology presents a promising solution. By incorporating NER into the model's training process, it can be trained to identify and track named entities across the sequence of captions. This will enable the model to recognize names as important textual elements that need to be preserved across frames. Implementing NER would not only ensure that names are maintained in the narrative but would also improve the overall continuity and personalization of the generated stories. Additionally, by enhancing the model's capability to handle named entities accurately, we can foster richer and more complex narrative structures where character development is pivotal, thereby significantly improving the storytelling quality.

Another area of future work would be to try to train a single multimodal model to ouput both the descriptive and story text rather than having separate models for each task. Currently, the model for generating the descriptive text has access to the story text at the current frame but the model for generating the story text does not have access to the descriptive text. Perhaps, this asymmetry is leading to poor story captions and training a model to output simultaneously may prove to be better. It may ensure greater alignment among the resulting text and could potentially reduce the amount of compute required for training.

We also would have liked to evaluate our model on some of the other storytelling datasets such as Flintstones SV, Pororo SV, or SSID (all discussed in section 2). The stories in the VIST dataset were created from indiviudal photos in flickr albums while these other datasets sampled frames from videos and contain more consistency in the characters between frames. Our autoregressive strategy might have performed better on the stories in these other datasets where there is more alignment among the images and each frame is fairly predictive of the next.

9 Ethical Concerns and Considerations (unintentional, malicious, and dual-use)

With the visual storytelling task, there are not necessarily a multitude of malicious or dual-use cases for the pipeline we built. There are some potential ethical concerns in terms of representation in the generative models. As with any generative models, any bias in the pre-training or finetuning data might also be present in the generated images or text. As shown in (Luccioni et al., 2023) , text-to-image systems have been shown to under-represent marginalized groups across race, gender, and age.

However, because are our auto-regressively generates the remainder of a story given the initial image and caption this might be less of a concern. We hope that the conditioning on the previous frame causes the model to output characters and text similar to the first frame that they provided which would help alleviate some of the bias in training data.

In addition to representation issues, the model could theoretically generate offensive content in the images or captions. Mitigating this would likely require reinforcement learning from human feedback (RLHF) which could potentially be time consuming but would be necessary if a story generation tool was ever client-facing.

10 Team member contributions

Nevan Contributed sections 1-5 and 9 of the writeup and also helped with others, brainstormed methods for text evaluation

Maxwell Setup the environment for and fine-tuned the Intelligent Grimm model for the image generation section of the project, generated all images from the model for evaluation, generated all images for baseline methods and ran metrics for the baseline methods. Ran the full model pipeline for doing both text generation and using that text generation for image generation, and made figures for results that required images. Wrote the sections that relate to anayslsis of image generation results and the results themselves. +-+Also made the table for image generation. Wrote code to calculate bounding boxes and analyzed the results in the doc.

Alex Ran and computed metrics for the MPT baseline. Created LLaVA fine-tuning, evaluation, and visualization code, and ran all the LLaVA procedures that didn't involve image generation. Ran the text metrics for the model, and analyzed the text based results and limitations.

Hyunwoo Wroked on descriptive caption generation for images and insertion prediction baseline

References

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering.

Zhangyin Feng, Yuchen Ren, Xinmiao Yu, Xiaocheng Feng, Duyu Tang, Shuming Shi, and Bing Qin. 2023. Improved visual story generation with adaptive context modeling. *arXiv preprint arXiv:2305.16811*.

Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem, and Aniruddha Kembhavi. 2018. Imagine this! scripts to compositions to videos.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Chi-Yang Hsu, Yun-Wei Chu, Ting-Hao 'Kenneth' Huang, and Lun-Wei Ku. 2021. Plot and rework: Modeling storylines for visual storytelling.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 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.

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

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.

Chang Liu, Haoning Wu, Yujie Zhong, Xiaoyun Zhang, Yanfeng Wang, and Weidi Xie. 2024a. Intelligent grimm – open-ended visual storytelling via latent diffusion models.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Alexandra Sasha Luccioni, Christopher Akiki, Margaret Mitchell, and Yacine Jernite. 2023. Stable bias: Analyzing societal representations in diffusion models.

Adyasha Maharana and Mohit Bansal. 2021. Integrating visuospatial, linguistic and commonsense structure into story visualization.

Zainy M. Malakan, Saeed Anwar, Ghulam Mubashar Hassan, and Ajmal Mian. 2023. Sequential vision to language as story: A storytelling dataset and benchmarking.

MosaicML. Introducing mpt-7b: A new standard for open-source, commercially usable llms. `https://www.databricks.com/blog/mpt-7b`.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. 2023. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*.

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

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

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*, pages 10684–10695.

Simo Ryu. 2023. Lora-stable diffusion. `https://github.com/cloneofsimo/lora`.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling.