Story Visualization by Online Text Augmentation with Context Memory

Daechul Ahn1,§ Daneul Kim2 Gwangmo Song3 Seung Hwan Kim3
Honglak Lee3,4 Dongyeop Kang5 Jonghyun Choi1,†

1Yonsei University 2GIST 3LG AI Research 4University of Michigan 5University of Minnesota
{dcahn, jc}@yonsei.ac.kr flytordk98@gm.gist.ac.kr gwangmo.song@lgresearch.ai
skcruise@gmail.com honglak@eecs.umich.edu dongyeop@umn.edu

Abstract

Story visualization (SV) is a challenging text-to-image generation task for the difficulty of not only rendering visual details from the text descriptions but also encoding a long-term context across multiple sentences. While prior efforts mostly focus on generating a semantically relevant image for each sentence, encoding a context spread across the given paragraph to generate contextually convincing images (e.g., with a correct character or with a proper background of the scene) remains a challenge. To this end, we propose a novel memory architecture for the Bi-directional Transformer framework with an online text augmentation that generates multiple pseudo-descriptions as supplementary supervision during training for better generalization to the language variation at inference. In extensive experiments on the two popular SV benchmarks, i.e., the Pororo-SV and Flintstones-SV, the proposed method significantly outperforms the state of the arts in various metrics including FID, character F1, frame accuracy, BLEU-2/3, and R-precision with similar or less computational complexity.

1. Introduction

Story visualization (SV) [17] is a task of generating a sequence of images from a paragraph, i.e., a sequence of natural language sentences. It is challenging for the requirement of rendering the visual details in images with convincing background of a scene - seasonal elements, environmental objects such as table, location, and the proper character appearing, which here we refer to as context, spread across the given text sentences. Specifically, it needs to encode implicit context presented in the given sentences since each one often omit visual details (i.e., they may be spread over the sentences) necessary to generate a semantically correct im-

Figure 1. Linguistic variations in story visualization and the overview of the proposed method. (a) An example of a story data with various linguistic variations (Var.#) for each image. (b) Modeling temporal context spread across sentences by our context memory (Sec. 3.1). (c) Addressing linguistic variations by online text augmentation from each image for every epoch (Sec. 3.2). (d) Benefit of the proposed context memory (left) and online text augmentation (right).

§: work done while interning at LG AI Research. †: corresponding author.
Code: https://github.com/yonseivnl/cmota
correct [17,22,23]. However, if we use contextual information given in Sent.#1, “The woods are covered with snow”, it leads to correct image with matching background.

In addition, widely used benchmark datasets for story visualization task [9,17] provide a single text-image pair for training and inference, mostly due to the annotation budget constraint. This prevents the model from learning language variations, and thus harms the linguistic generalization performance of the model.

To address the aforementioned challenges without requiring large scale data and models, we propose a new memory scheme in bi-directional Transformer for encoding the context which generates pseudo-texts in an online fashion to address linguistic variations at inference. We call our model as Context Memory and Online Text Augmentation or CMOTA for short. We empirically validate that our model outperforms the state-of-the-art SV methods by large margins on various metrics evaluated with widely used benchmarks in the literature, i.e., Pororo-SV and Flintstones-SV.

Note that while large pre-trained models [3, 5–7, 31, 43] have shown great success in synthesizing an image or a video from a language description [13,32,38], huge computational complexity and large training data makes the models prohibited. Moreover, although we propose and evaluate the model for the standard benchmark datasets without large pretraining data trained with a large model, it would be interesting to apply and evaluate the proposal in large models for further improvement.

We summarize our contributions as follows:

- We propose a new memory architecture for Transformer to selectively make use of contexts in a story paragraph.
- For better generalization of linguistic variations in the given paragraph at inference, we generate pseudo-texts and augment them in an online fashion for richer linguistic supervision.
- Our model significantly outperforms prior arts (even some hyper-scale models) by large margins in five evaluation metrics with similar or less computational complexity.

2. Related Work

Story visualization. StoryGAN [17] is one of the recent methods that utilized a story-level discriminator to improve global consistency in generated images. CP-CSV [39] disentangles figure and background information to enhance character consistency. In order to improve global semantic matching between paragraph and generated image sequence, DuCo-StoryGAN [23] presents a pre-trained video captioner as an auxiliary loss along with other design improvements on top of StoryGAN. More recently, VLC-StoryGAN [22] utilizes constituency parse-trees and common sense knowledge to improve consistency and an object-level feedback loop to improve image quality. Another recent work VP-CSV [2] is a two-stage approach, i.e., 1) character generation and 2) background completion, using Transformer model to address this task. We discuss these in more detail in the supplementary material for the space sake. But the prior arts largely neglect to encode the story narrative, which is our primary contribution here.

Very recently, Maharana et al. propose a new task setup of story continuation; using first image as a condition. They fine-tune the large model DALL-E [32] for the SV, which they call StoryDALL-E [24]. Ours differs from it in several aspects as follows. We have explicit memory connections between adjacent image generators using the context memory to globally encode the sentences. In contrast, [24] utilizes global story embedding as additional input for understanding context. Further, while it is huge in size (1.3B parameters), trained with 14 million text-image pair, ours are much larger (97M parameters) and outperforms it in multiple metrics by large margins; even in the image quality metric, FID. Please refer to Sec. 4.2 for empirical comparisons.


Recently, high-quality video generation models are proposed, including GODIVA [46] and NUWA [47]. Furthermore, recent studies generate high-resolution videos, making sequential frames in high-quality [11,38]. However, most of state-of-the-art text-to-video model [11,38] generates a video from a single sentence, mostly having consistent backgrounds. Very recently, there is a method proposed to generate a video from a long paragraph [45]. Although they generate the video in a long time horizon, they require both a mega-scale model trained with huge data and a detailed paragraph where each sentence is describing the scene that are close in time. In contrast, story visualization requires generating frames arbitrarily distant in time (i.e., so-called ‘key-frames’) corresponding to different sentences, requiring to generate an image sequence that have contextually convincing background.

Text-to-image generation. Text-to-image generation is a sub-problem of story visualization, with literature focusing on semantic relevance and resolution improvements. Recently, text-based image synthesis has been greatly improved with the help of a vast amount of training data with a hyper-scale model including DALL-E [32] and its successor DALL-E2, CogView [5] and Make-A-Scene [8] using a sketch input.

Although text-to-image generation models generates very high-quality images, it may lack encoding the context, metaphorical sentences spread across multiple sentences. In
addition, naively using state-of-the-art text-to-image generation models is computationally prohibited. For example, diffusion-based models [31, 34] have hyper-scale model size, (e.g., Imagen [34] parameter count of 2-B, DALL-E2 [31] parameter count of 3.5-B) making it non-trivial for applying it in a wide range of inference scenarios that may not have the sufficient computing resource. Here, we consider relatively light architectures as our base model for computational efficiency. More discussion are in the supplement.

**Transformer using memory.** To mitigate context fragmentation issue [4], i.e., losing long-term dependency over a data stream in context, there are efforts to encode long-term contexts in generating an image sequence. [4, 15] adopt a recurrent path into transformer architecture. Specifically, the modeling of new data segments is conditioned on historical hidden states produced in the previous time step and uses highly summarized memory states. Unlike the conventional memory architectures, we propose a novel memory that has a dense connection from the past with attentive weighting schemes for their better usage (Sec. 3.1).

**Online augmentation.** While offline data augmentation that prepares data outside of learning process is prevalent in many literature [1,12,19], online-augmentation that depends on training is seldom explored especially in story visualization task [40]. [35, 40] propose an online augmentation for image classification and VQA task. [26] uses bi-level optimization in image classification. In medical domain, [48] investigate the online augmentation for personalized image based Histopathology diagnosis [48]. Note that we show the benefit of online augmentation over the offline augmentation in the SV for the first time in this literature.

### 3. Approach

There are multiple challenges in story visualization, including (1) generating semantically natural images without artifacts from a description, (2) encoding the context (e.g., consistent background of a scene - seasonal elements, environmental objects such as table and location, and consistent characters) spread across the sentences in a given paragraph [17] and (3) addressing the linguistic variation at the inference time (i.e., a given text description may not be in different writing style).

For photo-realistic image generation from a language description, we recently witness unprecedented improvements in the quality of generated images by the help of large scale model [3,5–7,31,43]. To leverage the large model’s benefits for the story visualization task, we may use it to generate an image sequence by gradually concatenating sentences to visualize the story (i.e., use the first sentence to generate the first image, use a concatenated sentence of the first and the second to generate the second, and so on). Although this may generate a single photo-realistic image, it is not able to capture the context sparsely spread across the sentences as story progresses (see supplement for more discussion).

A more involved way of using the large model for SV is to fine-tune it for the downstream SV or using a video generation models. However, as the most high-performing models are huge in size and the codes and the pre-trained models are not publicly available, the computational cost and reproducibility reduces practicality.

For a given dataset without using large extra training data, we propose a new memory module to better encode past information with a Transformer by an attentively weighted densely connected architecture (Sec. 3.1). We further propose to generate pseudo-texts during the learning process to augment (online-augmentation) for better linguistic generalization without requiring large external data by learning the bi-directional Transformer [14, 30, 33] in both directions of generating images from texts and vice versa as illustrated in Fig. 2 (Sec. 3.2).

**Base model.** As shown in Fig. 2, we use bi-directional (i.e., multi-modal) transformer that iteratively generates images and texts in both ways. Similar to [32], the image tokens are sequentially predicted from the input text sequences by the Transformer. Then, the decoder of the VQ-VAE [42] translates the predicted image tokens into image sequence. The text tokens are also sequentially predicted from the input image token sequence by the same Transformer.

Particularly, for the bi-directional multi-modal generation, i.e., generating simultaneously text and image from the unified architecture, we add two embeddings; a positional embedding for absolute position between tokens and a segment embedding for distinguishing source and target. Tokens of a text ($\{t_1, \ldots, t_m\}$) and an image ($\{z_1, \ldots, z_n\}$) ($m, n$: # of tokens for text and image) are fed into the Trans-
Figure 3. Proposed memory module for the bi-directional Transformer. Memory module updates current memory $M_t$ by utilizing memory attention and GRU operation with propagated memory, $M_{t-1}$, and previously calculated hidden state, $H_t^l$. To produce image, we propose a novel memory attention (Attn.) mask denoted by blue squares to select only text tokens for less bias to previously generated image tokens. Within the memory attention mask, white-colored squares filter out the content, i.e., image (img) token, whereas the blue squares allow the text (txt) token to propagate through for memory interaction.

former to predict tokens in the other modality in multiple epochs with the following objective function written as:

$$
\mathcal{L}_{j,t_21} = \sum_{k=1}^{m} - \ln p_j(z_k | t_1, \ldots, t_m, z_1, \ldots, z_{k-1}),
$$

$$
\mathcal{L}_{j,t_21} = \sum_{k=1}^{m} - \ln p_j(t_k | z_1, \ldots, z_n, t_1, \ldots, t_k-1),
$$

$$
\mathcal{L}_j = \mathcal{L}_{j,t_21} + \lambda_1 \mathcal{L}_{j,t_21},
$$

where $p_j(\cdot)$ is a likelihood of $j$-th generated tokens in one modality given the other modality, $L_{j,t_21}$ is a loss corresponding to $j$-th text-to-image generation (i.e., negative log likelihood), $L_{j,t_21}$ is vice versa. $\lambda_1$ is a balancing hyperparameter.

To train the model, we iteratively train the generation model in each modality multiple times per each epoch.

3.1. Context Memory

To encode the context and calculate the propagated memory, as depicted in Fig. 3, we first apply cross attention between the current hidden state, $H_t^l \in \mathbb{R}^{T_c \times d}$ calculated from $(l-1)$-th transformer layer and memory state at time $(t-1)$, $M_{t-1} \in \mathbb{R}^{T_M \times d}$ ($T_c$: # tokens, $T_M$: # memory states, $d$: hidden state dimension). We then obtain $S_t = \text{Attn}(M_{t-1}, H_t^l, H_t^l)$, where $\text{Attn}(Q, K, V) := \text{Softmax} \left( \frac{QK^T}{\sqrt{d_q}} \right) V$ and $d_q$ is a query dimension ($Q$), with the memory attention mask depicted in the blue box in Fig. 3.

In particular, we apply a memory mask in the attention operation to select text tokens as memory content (depicted as blue-shaded grid cells in the right of Fig. 3) because including image content as memory could be strong constraint for text-to-image generation (see empirical studies of the proposed mask in the supplementary material). Then we feed the $S_t$ and $M_{t-1}$ to the GRU in order to compute the information to be stored in the memory, $M_t$, propagated to $(t+1)$.

Memory connection. Following [4, 15], we stack multiple Transformer’s layers, depicted in Fig. 4-(b). Compared to conventional memory updating architectures with serial hidden state connections in all layers (Fig. 4-(a)), our approach only has memory connections in the last layer, which we call it as partial-level memory augmented (PMA). This is because the later layers (closer to the last layer) achieve better representations with higher-level features, abstract and structured representations [28, 29, 41] with improved computational efficiency, which is unlike prior works (see Sec. 4.2.2 and supp for further discussion).

Additionally, not all historic information is equally important for generating an image at a time step. Similar to the masked self-attention [43], we attentively weight the past information for better modeling of sparse context as:

$$
\bar{M}_{t:1} = \text{Attn}(M_{t-1}, M_{t-2:1}) \bar{M}_{t:1},
$$

$$
\bar{M}_{t:1} = [M_{t-1}; \bar{M}_{t:1}], \quad (3 \leq t \leq 5),
$$

$$
H_t^l = \text{Attn}(H_t^l, [H_t^l; \bar{M}_{t:1}], [H_t^l; \bar{M}_{t:1}]),
$$

where $\bar{M}_{t:1}$ is a masked self-attention feature from previous transformer layer.
where $M_{t-1}$ is a memory at time $(t - 1)$ and $[1 : (t - 2)]$ refers to the concatenation from time 1 to $(t - 2)$ as depicted in Fig. 4-(b). By doing so, we fuse the contextual information into current hidden state $H_t$. At $t = 2$, we use $M_1$, instead of $M_2$ as the $M_2$ is only available. We call it as attentively weighted memory (AWM).

### 3.2. Online Text-Augmentation

Vedantam et al. argue that multiple descriptions for an image help generalization as they address language variations presented in descriptions [44]. Hence, a number of image captioning datasets [10, 20, 37] provide multiple natural language directives, obtained by multiple human annotators, in both training and evaluation splits. But the SV benchmark datasets [17] provide only a single sentence per an image.

To address the linguistic variations of a text input at inference process, we first consider to generate a pseudo text by a well-trained image-to-text generation model, which we refer it as offline-augmentation [16]. But the offline augmentation generates only a single sentence, which may not provide sufficient diversity. Instead, we propose to generate multiple pseudo-texts and augment them in an online fashion when training our model to increase the diversity. We call it online text augmentation, depicted in Fig. 5.

Thanks to our bi-directional multi-modal architecture, we can naturally integrate the process of generating pseudo-texts to the process of learning image-to-text model and the text-to-image generation model as depicted in Fig. 5. In the early epochs, less meaningful sentences are generated, but as training progresses, more meaningful sentences are generated (see orange box in Fig. 5). As a side-product, by supervising the model learning with intermediate goals at each time step, we expect to expedite the convergence of gradient flow. (2) generating an image for every epoch. (3,4) learn the text-to-image model with $L_{CE}$. In early epochs, it generates less meaningful pseudo-text (thus discard them), but produces pseudo-texts matching with the input image as the training progresses.

Pororo-SV dataset for story visualization task without data overlap. Following prior work [22], we use Flintstones-SV which was originally exploited in the text-to-video synthesis task [9].

### Metrics.

For the evaluation metrics, we use Fréchet Inception Distance (FID) (used in [22, 39]) for visual quality of the generated images, character classification score (Char. F1, Frm. Acc.) [17] for character consistency, and global semantic matching (B-2/3, R-Prec.) [22, 23] between generated image sequence and story paragraph. We elaborate on the details of the datasets and implementation in the supplementary material for the sake of space.

### Baselines.

We compare our method to a number of prior arts including state-of-the-art story visualization methods such as StoryGAN [17], CPCSV [39], DuCo-StoryGAN [23], VLC-StoryGAN [22] and VP-CSV [2] for both quantitative and qualitative analysis in the story visualization setting. Moreover, we compare [24] in the story continuation setting.

### Implementation details.

To train our model, we first tokenize the text and image inputs. Particularly, the image inputs are encoded by the encoder of the VQ-VAE [42] into image tokens and decoded by the decoder of the VQ-VAE for generating image sequence. Once the inputs are tokenized, we append two special embedding tokens to the text and image tokens. These tokens signify the beginning of each modality, denoted as ‘SOS’ (start of sentence) and ‘SOI’ (start of image). Subsequently, we add two embeddings: (1) a positional embedding to indicate absolute position of each token and (2) a segment embedding to distinguish source and target modality, e.g., for text-to-image generation task, the source corresponds to the text while the target corresponds to the image. Through the utilization of segment embeddings to distinguish between source and target, we enable the model to generate target data from source data.
We utilize them as supplementary supervisions to address linguistic diversity of a sentence at inference.

\[4.2.1 \text{ Comparison with the State of the Arts}\]

In Table 1, we summarize the performance of the prior arts and the proposed CMOTA in Pororo-SV [17] and Flintstones-SV [9] dataset in the various metrics. By default, all methods generate 64 × 64 images along with the higher resolution 128 × 128 images. CMOTA outperforms existing methods by a large margin in both benchmarks in most of metrics.

The very recent method VP-CSV [2] performs on par to our CMOTA while ours still outperforms in FID, BLEU and R-precision, implying that CMOTA generates high quality images maintaining global semantic matching between story paragraph and images better than the VP-CSV. The lower performance than the VP-CSV in Char. F1 score and Frm. Acc. is attributed to the specialized character-centric module that only focuses the model to generate accurate characters for each image at the expense of the other performance metrics. In addition, our CMOTA-HR model even outperforms the VP-CVS [2] by large margin without the specialized character-centric module.

\[4.2. \text{ Quantitative Analysis}\]

\[4.2.1 \text{ Comparison with the State of the Arts}\]

In Table 1, we summarize the performance of the prior arts and the proposed CMOTA in Pororo-SV [17] and Flintstones-SV [9] dataset in the various metrics. By default, all methods generate 64 × 64 images along with the higher resolution 128 × 128 images. CMOTA outperforms existing methods by a large margin in both benchmarks in most of metrics.

The very recent method VP-CSV [2] performs on par to our CMOTA while ours still outperforms in FID, BLEU and R-precision, implying that CMOTA generates high quality images maintaining global semantic matching between story paragraph and images better than the VP-CSV. The lower performance than the VP-CSV in Char. F1 score and Frm. Acc. is attributed to the specialized character-centric module that only focuses the model to generate accurate characters for each image at the expense of the other performance metrics. In addition, our CMOTA-HR model even outperforms the VP-CS [2] by large margin without the specialized character-centric module.

\[4.2. \text{ Quantitative Analysis}\]

\[4.2.1 \text{ Comparison with the State of the Arts}\]

In Table 1, we summarize the performance of the prior arts and the proposed CMOTA in Pororo-SV [17] and Flintstones-SV [9] dataset in the various metrics. By default, all methods generate 64 × 64 images along with the higher resolution 128 × 128 images. CMOTA outperforms existing methods by a large margin in both benchmarks in most of metrics.

The very recent method VP-CSV [2] performs on par to our CMOTA while ours still outperforms in FID, BLEU and R-precision, implying that CMOTA generates high quality images maintaining global semantic matching between story paragraph and images better than the VP-CSV. The lower performance than the VP-CSV in Char. F1 score and Frm. Acc. is attributed to the specialized character-centric module that only focuses the model to generate accurate characters for each image at the expense of the other performance metrics. In addition, our CMOTA-HR model even outperforms the VP-CS [2] by large margin without the specialized character-centric module.

\[4.2. \text{ Quantitative Analysis}\]

\[4.2.1 \text{ Comparison with the State of the Arts}\]

In Table 1, we summarize the performance of the prior arts and the proposed CMOTA in Pororo-SV [17] and Flintstones-SV [9] dataset in the various metrics. By default, all methods generate 64 × 64 images along with the higher resolution 128 × 128 images. CMOTA outperforms existing methods by a large margin in both benchmarks in most of metrics.

The very recent method VP-CSV [2] performs on par to our CMOTA while ours still outperforms in FID, BLEU and R-precision, implying that CMOTA generates high quality images maintaining global semantic matching between story paragraph and images better than the VP-CSV. The lower performance than the VP-CSV in Char. F1 score and Frm. Acc. is attributed to the specialized character-centric module that only focuses the model to generate accurate characters for each image at the expense of the other performance metrics. In addition, our CMOTA-HR model even outperforms the VP-CS [2] by large margin without the specialized character-centric module.

\[4.2. \text{ Quantitative Analysis}\]

\[4.2.1 \text{ Comparison with the State of the Arts}\]

In Table 1, we summarize the performance of the prior arts and the proposed CMOTA in Pororo-SV [17] and Flintstones-SV [9] dataset in the various metrics. By default, all methods generate 64 × 64 images along with the higher resolution 128 × 128 images. CMOTA outperforms existing methods by a large margin in both benchmarks in most of metrics.

The very recent method VP-CSV [2] performs on par to our CMOTA while ours still outperforms in FID, BLEU and R-precision, implying that CMOTA generates high quality images maintaining global semantic matching between story paragraph and images better than the VP-CSV. The lower performance than the VP-CSV in Char. F1 score and Frm. Acc. is attributed to the specialized character-centric module that only focuses the model to generate accurate characters for each image at the expense of the other performance metrics. In addition, our CMOTA-HR model even outperforms the VP-CS [2] by large margin without the specialized character-centric module.
Table 1. Quantitative comparison with the state of the art. On the test split of Pororo-SV and Flintstones-SV. # Params. refers to the number of trainable parameters. Char. F1 refers to character F1 score. Frm. Acc. refers to frame accuracy. R-Prec. refers to R-Precision. ↓ indicates ‘lower the better’ and ↑ indicates ‘higher the better’. Experiments are done in both $64 \times 64$ and $128 \times 128$ resolutions. ‘HR’ refers to its high resolution ($128 \times 128$) version. † indicates our results with author’s implementation (https://github.com/adymaharana/VLCStoryGan). ‡ indicates the absolute values that we computed by the given relative values in [22].

4.2.3 Comparison with Large-Scale Models

Using the proposed partial-level connected memory architecture only with the vanilla transformer (Tr.), the generation performance improves in all metrics compared to the Tr. Also, if we compare it with all-level connected memory architecture [15], PMA shows better performance in all metrics as in Tab. 3 (see supplementary material for more experimental analysis for various memory connection schemes). Employing attentional weighted memory (AWM) improves performance overall with slight degradation in FID ($+0.12$). The bi-directional learning scheme (Bi.) improves performance in all metrics except the character classification related metrics, i.e., Char.F1 and Frm. Acc. It is because there are captions that does not contain characters’ name, just mentioning characters as ‘friends’, making the model difficult to generate specific characters.

Benefit of the text augmentation. By augmenting the generated text in offline manner (i.e., using an image captioning model pretrained on the dataset of interest), we observe overall performance increase in all metrics. We hypothesize its reason as that the pseudo-text may convey contextual information such as characters’ existence or description of background, thereby making the generation model to be robust to language variation as shown in a first row of Tab. 4. By the online augmentation of pseudo texts (i.e., gradually updating text generator, thus generating a diverse and gradually varying set of pseudo-texts), the performance further improves, as shown in Tab. 2 and Tab. 4.

4.2.2 Ablation Studies

Benefit of the proposed architectural components. To investigate the benefit of each proposed architectural component, we build up the full model with the components from the vanilla, single-directional transformer without memory as summarized in Tab. 2.

As shown in the figure, our method outperforms prior arts [22, 23] by large margins. Although we did not have any special module for character information preserving, unlike [2], our model maintains such information over multiple sentences, even outperforming the image quality denoted as FID by $-13.38$.

SV dataset and normalized character frequency in train split dataset as illustrated in Fig. 7. Especially, it is interesting to compare with [2] as it is explicitly proposed for that purpose. As shown in the figure, our method outperforms prior arts [22, 23] by large margins. Although we did not have any special module for character information preserving, unlike [2], our model maintains such information over multiple sentences, even outperforming the image quality denoted as FID by $-13.38$.

**Figure 7. Per-character Classification F1-score on the Test Split of Pororo-SV Dataset.** Normalized character (Char.) frequency is shown in red line. It represents the relative number of characters compared to the count of Pororo in training dataset. While DuCo/VC-StoryGAN shows degraded performance as the character frequency decreases, CMOTA outperforms them across all the characters.

### Dataset | Resolution | Methods | # Params. | FID↓ | Char. F1↑ | Frm. Acc.↑ | BLEU-2/3↑ | R-Prec.↑
--- | --- | --- | --- | --- | --- | --- | --- | ---
Pororo-SV  | $64 \times 64$ | StoryGAN [17] | - | 158.06 | 18.59 | 9.34 | 3.24 / 1.22 | 1.51 ± 0.15 |
| | | CP-CSV [39] | - | 140.24 | 21.78 | 10.03 | 3.25 / 1.22 | 1.76 ± 0.04 |
| | | DuCo-StoryGAN [23] | 101M | 96.51 | 38.01 | 13.97 | 3.68 / 1.34 | 3.56 ± 0.04 |
| | | VLC-StoryGAN [22] | 100M | 84.96 | 43.02 | 17.36 | 3.80 / 1.44 | 3.28 ± 0.00 |
| | | VP-CSV [2] | - | 65.51 | **56.84** | **25.87** | 4.45 / 1.80 | 6.95 ± 0.00 |
| | | CMOTA (Ours) | 96.6M | **52.13** | 53.25 | 24.72 | 4.58 / 1.90 | 7.34 ± 0.03 |
| | $128 \times 128$ | VLC-StoryGAN-HR† [22] | 102.6M | 97.08 | 40.36 | 17.17 | 3.89 / 1.58 | 3.47 ± 0.03 |
| | | CMOTA-HR (Ours) | 103.7M | **52.77** | **58.86** | **28.89** | 5.45 / 2.34 | **16.36 ± 0.05** |
Flintstones-SV  | $64 \times 64$ | StoryGAN [17] | - | 127.19 | 46.20 | 32.96 | 13.87 / 7.83 | 1.72 ± 0.18 |
| | | DuCo-StoryGAN [23] | 101M | 78.02 | 54.92 | 36.34 | 15.48 / 9.17 | 2.64 ± 0.17 |
| | | VLC-StoryGAN† [22] | 100M | 72.87 | 58.81 | 39.18 | - | - |
| | | CMOTA (Ours) | 96.6M | **36.71** | **79.74** | **66.01** | **19.85 / 12.98** | **10.50 ± 0.35** |
| | $128 \times 128$ | CMOTA-HR (Ours) | 103.7M | 54.81 | 86.44 | 74.06 | 22.18 / 15.17 | 25.71 ± 0.70 |

The bi-directional learning scheme (Bi.) improves performance in all metrics except the character classification related metrics, i.e., Char.F1 and Frm. Acc. It is because there are captions that do not contain characters’ names.”

**Figure 7. Per-character Classification F1-score on the Test Split of Pororo-SV Dataset.** Normalized character frequency is shown in red line. It represents the relative number of characters compared to the count of Pororo in training dataset. While DuCo/VC-StoryGAN shows degraded performance as the character frequency decreases, CMOTA outperforms them across all the characters.
We discuss more about this in the supplementary material. Worse than the StoryDALL-E thanks to the huge size of Benefit of the online augmentation. Table 4. Benefit of the proposed memory connection scheme. Table 3. Benefit of the proposed components. On the test split of Pororo-SV Dataset. As a baseline, we use a single directional transformer-based text-to-image generation model. We gradually add each of proposed components to investigate its effect for improving performance, i.e., partial-level memory connection (Sec. 3.1), attentively weighted memory (Sec. 3.1), bi-directional training (Sec 3.2), and online-augmentation (Sec. 3.2).

Table 2. Benefit of the proposed components. On the test split of Pororo-SV Dataset. As a baseline, we use a single directional transformer-based text-to-image generation model. We gradually add each of proposed components to investigate its effect for improving performance, i.e., partial-level memory connection (Sec. 3.1), attentively weighted memory (Sec. 3.1), bi-directional training (Sec 3.2), and online-augmentation (Sec. 3.2).

4.3. Human Preference Study

We conduct a larger scale (than the prior arts) human survey using the Amazon Mechanical Turk platform to qualitatively compare the generated image sequence by our method over VLC-StoryGAN [22] on three criteria, i.e., visual quality, temporal consistency, and semantic relevancy between generated images and descriptions, following [17, 22, 23, 39]. Unlike the previous arts that employ 2 to 9 human judges [2, 17, 22, 23, 39], we employ 100 judges for statistically more reliable results.

As shown in the top table in Tab. 6, it is clear that human subjects prefer the generated images by our model over those from VLC-StoryGAN [22] on winning ratio(%). We discuss more details about this study in the supplementary material.
CMOTA and CMOTA-w/o-Memory, we use 30 randomly sampled image sequences on Pororo-SV test set to conduct A/B test following [39]. Particularly, the annotators need to consider the temporal consistency with semantic relevance between generated image sequence and captions. As depicted in the bottom table in Tab. 6, we observe that 61.2% of annotators prefer our CMOTA with memory module over the one without it. We can conclude that the memory module has a great deal in creating temporally consistent image sequence with respect to semantic relevancy.

4.4. Qualitative Analysis

We now qualitatively analyze the quality of generated image sequences in Fig. 8 on the Pororo-SV dataset’s test split. The top row shows the image sequence of ground truth, the two rows (2-3) contain prior works [17, 23] and the final row is the image sequence generated by CMOTA and its high resolution version (CMOTA-HR). CMOTA generates a semantically more plausible image sequence with better visual quality, compared to prior works [22, 23]. Particularly, the CMOTA-HR demonstrates the ability to generate visual details more effectively even as the size of the image increases, while maintaining similar semantics.

Furthermore, we qualitatively investigate the advantage of using memory architecture and summarize the results in Fig. 9. As shown in the figure, the proposed memory architecture generates a semantically more plausible image sequence with proper context, e.g., background; the CMOTA without memory fails to capture proper background context (shown in dotted red box) since a single sentence could be interpreted in many ways. In contrast, CMOTA (with the context memory) generates an image sequence with plausible background with characters preserved without an explicit character-centric model.

5. Conclusion

We propose to better encode semantic context, e.g., plausible background and characters, for story visualization using a new memory architecture in a multi-modal bi-directional Transformer. We further propose an online text augmentation training scheme to generate pseudo-text descriptions as an intermediate supervision for addressing linguistic diversity in the texts at inference. The proposed method generates a temporally coherent and semantically relevant image sequence for each sentence in the given text paragraph.

The proposed method outperforms prior works by a large margin on various metrics on the two popular SV benchmark datasets, and also outperforms some of hyper-scale models in multiple semantic understanding metrics. Although computationally prohibited and the pre-trained models are not publicly available, it would be intriguing to apply the proposed memory module and the online augmentation scheme to a large model.

Acknowledgement. This work is partly supported by the NRF grant (No.2022R1A2C4002300) 25%, IITP grants (No.2020-0-01361, AI GS Program (Yonsei University) 5%, No.2021-0-02068, AI Innovation Hub 5%, 2022-0-00077 15%, 2022-0-00113 15%, 2022-0-00959 15%, 2022-0-00871 10%, 2022-0-00951 10%) funded by the Korea government (MSIT).
References


[38] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. ArXiv, 2022. 2


[46] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions, 2021. 2
