Text-Driven Stylization of Video Objects

Sebastian Loeschcke[1], Serge Belongie[2], Sagie Benaim[2]

[1]Aarhus University, [2]University of Copenhagen

Abstract.

We tackle the task of stylizing video objects in an intuitive and semantic manner following a user-specified text prompt. This is a challenging task as the resulting video must satisfy multiple properties: (1) it has to be temporally consistent and avoid jittering or similar artifacts, (2) the resulting stylization must preserve both the global semantics of the object and its fine-grained details, and (3) it must adhere to the user-specified text prompt. To this end, our method stylizes an object in a video according to two target texts. The first target text prompt describes the global semantics and the second target text prompt describes the local semantics. To modify the style of an object, we harness the representational power of CLIP to get a similarity score between (1) the local target text and a set of local stylized views, and (2) a global target text and a set of stylized global views. We use a pre-trained atlas decomposition network to propagate the edits in a temporally consistent manner. We demonstrate that our method can generate consistent style changes over time for a variety of objects and videos, that adhere to the specification of the target texts. We also show how varying the specificity of the target texts and augmenting the texts with a set of prefixes results in stylizations with different levels of detail. Full results are given in the supplementary and in full resolution in the project webpage: `https://sloeschcke.github.io/Text-Driven-Stylization-of-Video-Objects/`.

Keywords: Video Editing, Text-Guided Stylization, CLIP

1 Introduction

Manipulating semantic object entities in videos using human instructions requires skilled workers with domain knowledge. We seek to eliminate these requirements by specifying a desired edit or stylization through an easy, intuitive, and semantic user instruction in the form of a text-prompt.

However, manipulating video content semantically is a challenging task. One challenge is in generating consistent content or style changes in time, that adhere to the target text specification. Another challenge is to manipulate the content of an object such that it preserves the content of the original video and the global semantics while also adhering to fine-grained details in the target text.

In recent years, advances in computational methods emerged that enable manipulation of appearances and style in images and allow novice users to perform realistic image editing. These methods include manipulation tools that use natural language (text prompts) to express the desired stylization of images or
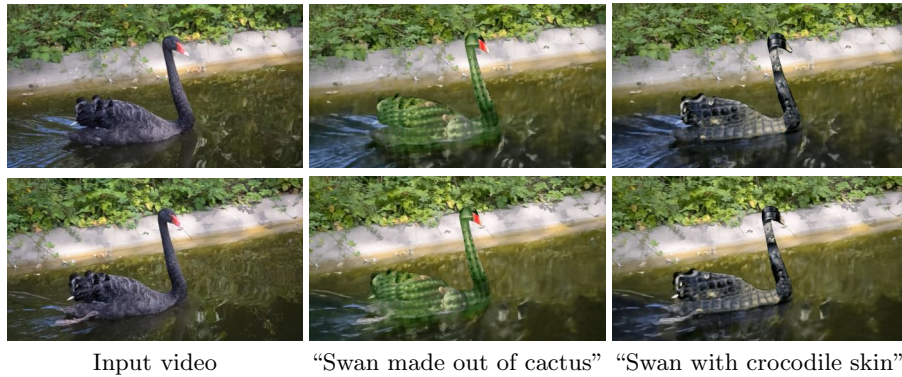
Input video   "Swan made out of cactus" "Swan with crocodile skin"

Fig. 1: Two representative video frames and the edited video frames together with the global target text.

3D objects [18,6]. The text-driven manipulation is facilitated by recent developments in models for joint embeddings of text and images, e.g the Contrastive Language Image Pretraining (CLIP [23]) model. Instead of manipulating images or 3D objects, we use CLIP in the context of video manipulation. This is not a straightforward task since simply maximizing the semantic (CLIP-based) similarity between a valid target text and each 2D frame in the video often leads to degenerate solutions. Also, applying methods for image manipulation to each frame in a video results in edits that lack temporal consistency.

The recently introduced Neural Layered Atlases (NLA) work [14], demonstrates the ability to separate a moving object in a video from its background by decomposing the video into a set of 2D atlases. Each atlas provides a unified image representation of an object or background over the video. Edits applied to the image representation are automatically mapped back to the video in a temporally consistent manner. However, editing an image still requires manual effort and editing skills from the user. Another problem with this approach is that the 2D atlas representation can be hard to edit due to local deformations.

We propose a method for performing intuitive and consistent video editing with multiple capabilities by using the representational power of CLIP to express a desired edit through a text-prompt. An example could be to change the style of a swan swimming in a lake according to a target text: "A swan with cactus skin." Text is easily modifiable and allows users to express complex and abstract stylizations intuitively. Using text to express edits reduces the need for manual editing skills and also avoids the problems related to deformation in the 2D atlas representations. An illustration is shown in Fig. 1.

To apply temporally consistent edits to an object in a video, our method uses the atlas decomposition method presented in NLA [14]. We train a generator on a single input video by sampling local and global views of each frame in the video and applying various augmentations to each view. Our method uses a global loss that compares each global view with a global target text and a local

loss that compares each local view with a local target text. The global loss then focuses on the global semantics and the local views focus on the fine-grained details. To regularize our learning, we use a sparsity loss that encourages sparse representation and a temporal triplet loss that encourages frames that are close in time to be similar in CLIP's embedding space.

We demonstrate that our method results in natural and consistent stylizations of objects for a diverse set of videos and target texts. We show how varying the specificity of both the local and global target texts varies the stylization and how augmenting the target texts with neutral prefixes can result in more detailed stylizations. We also demonstrate that our global loss focuses on the global semantics and the local losses on the fine-grained details.

2   Related Work

Our work is related to video editing works and to text-based stylization works.

2.1   Video editing

Unlike images, editing or stylizing objects in videos requires the ability to handle temporal consistency. One natural approach is to propagate the edits from one frame to the next as in Video Propagation Networks [10]. Other approaches use optical flow to propagate edits made on a few key-frames [28]. These approaches work well when there is a clear correspondence between frames, but have difficulties, e.g., when the video contains occlusions.

To address occlusion challenges, recent work has used deep learning approaches, e.g. self-supervised methods for learning visual correspondence from unlabeled videos [9,29] or methods that decompose a video into a 2D representation [14,30]. Our work uses the representation proposed by Neural Layered Atlases (NLA) [14], which decomposes a video into a set of layered 2D atlases. Each atlas provides a unified representation of the appearance of an object or background throughout the video. However, NLA only allows for basic manual editing. We use NLA's atlas separation method for objects in videos, but unlike NLA, we allow for text-driven stylization.

2.2   Text-based stylization

Our work bears similarities to recent image and 3D manipulation techniques that edit style and appearances through natural language descriptions. These descriptions are often embedded with the Contrastive Language Image Pretraining (CLIP) [23] model, a multi-modal embedding model that learns an image-text embedding space. Recent work used CLIP together with pretrained generative networks for image editing and stylization [1,21,3,5,6,8,15]. For example, Style-CLIP [21], and StyleGAN-NADA [8] both use a pretrained StyleGAN [12] and CLIP to perform image editing, either by using CLIP to control a latent code or

to adapt an image generator to a specific domain [8]. Usually, pretrained generators only work well for the specific domain they are trained on. In contrast, our method does not require a pretrained generator. We train our own generator on the set of video frames we wish to stylize.

Other examples of semantic text-guided manipulation in the context of 3D objects include 3DStyleNet [31], a method for changing the geometric and texture style of 3D objects and ClipMatrix [11] which uses text-prompts to create digital 3D creatures. Another line of recent work uses joint-embedding architectures [26,25,24,19] for image generation, e.g., DALL-E [25] and its successor, DALL-E 2 [24], which can also be used for stylizing images. DALL-E 2 uses a two-stage model, where a CLIP image embedding is generated using a text prompt and a decoder is then used to generate an image conditioned on the generated image embedding. Training joint embedding architectures requires enormous datasets and many training hours. Instead of training on a large dataset, we train on a set of frames for a single video and use augmentations to extract many different views of the input frames. As opposed to all the abovementioned techniques, we work on videos.

Another line of work [7,17] uses CLIP without relying on a pretrained generator, e.g. Texts2Mesh [18]. In Text2Mesh, the CLIP embedding space is used to enable text-driven editing of 3D meshes. Text2Mesh uses a multi-layer perceptron (MLP) to apply a stylization to (x, y, z)-coordinates of an input mesh. The neural optimization process of the MLP is guided by a semantic loss that computes the similarity between multiple augmented 2D views embedded with CLIP and a target text. Similarly, we do not rely on a pretrained generator.

Our work was developed concurrently to Text2Live [2], which shares many of the same goals and methods as our work. Similarly to our method, Text2Live uses a pretrained Neural Layered Atlases (NLA) model to separate a moving object in a video from its background. Text2Live train a generator to apply text-driven local edits to a single frame and use the NLA model to map the edits back to the input video in a temporally consistent manner. In contrast to our approach, Text2Live does not directly generate the edited output. Instead, it generates an edit layer that is composited with the original input.

3   Method

We wish to apply natural and temporally consistent stylizations to objects in videos using a natural language text prompt as guidance. To change the style of an object to conform with a target text prompt in a temporally consistent manner, we build on top of a Layered Neural Atlas method [14], which separates the appearance of an object in a video from its background. We then use a pretrained text-image multimodal embedding of CLIP [23] in a set of objectives. Minimizing this set of objectives aims at matching the style of a foreground object in a video with that of a target text. The objectives include a global and local objective. The global objective focuses on the global semantics by maximizing the similarity between the global views and a target text that relates

to the underlying content. Instead, the local objective focuses on the fine-grained details, by maximizing the similarity between the local views with a target text that relates to local semantics of the stylization. We add a sparsity loss, similar to a L_1-regularization term, that encourages the predicted foreground color values to be minimal. Additionally, we add a temporal triplet loss that encourages the embeddings of frames that are close in time to also be close in CLIP's embedding space. We begin by describing the method of CLIP [23] and that of Neural Layered Atlases (NLA) on which our method is based. We then describe the training and loss formulations used by our method.

3.1 CLIP

CLIP is a multi-modal embedding method that trains an image encoder E_{img} and a text encoder E_{txt} to match between the embeddings of corresponding image-text pairs using a contrastive loss formulation. This loss formulation optimizes the similarity between corresponding image-text pair T and I. More specifically, I and T are first embedded:

$$I_{emb} = E_{img}(I) \in \mathbb{R}^{512}, \quad T_{emb} = E_{txt}(T) \in \mathbb{R}^{512}$$

The similarity between I and T is then measured by $\text{sim}(I_{emb}, T_{emb})$ where $\text{sim}(a, b) = \frac{a \cdot b}{|a||b|}$, is the cosine similarity.
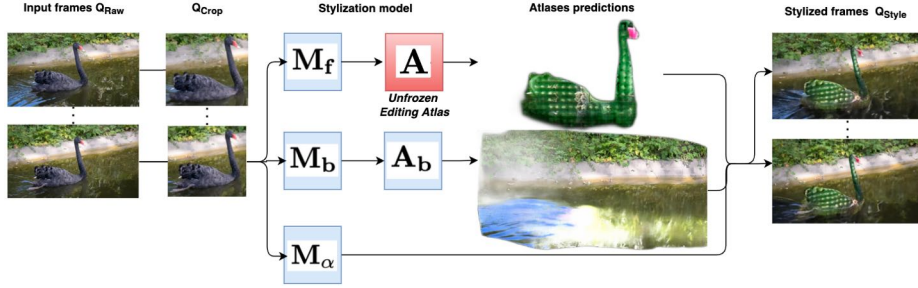
3.2 Neural Layered Atlases (NLA)

Fig. 2: **Our stylization pipeline.** In the first step, we train the network using the NLA procedure [14] to reconstruct input video frames. We then finetune the editing atlas A using our approach described in Fig. 3. In our stylization pipeline, we create a set of cropped input video frames Q_{Crop}. This set is passed through a stylization model to create a foreground and background atlas. A set of stylized frames Q_{Style} is produced by α-blending the predicted atlases. All weights of the MLPs are frozen except for the editing atlas MLP A which is finetuned. A closer look at how our editing atlas is trained is given in Fig. 3.
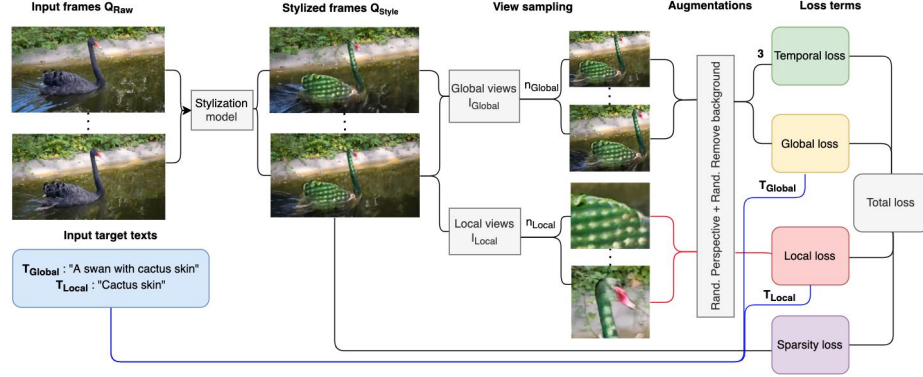
Fig. 3: **Finetuning the editing atlas.** As described in the stylization pipeline (Fig. 2), we use our stylization pipeline to get a set of stylized frames Q_{Style}. To this end, we finetune our editing atlas, which is part of the stylization model. We sample n_{Global} global views I^{Global} and n_{Local} local views I^{Local}. The set of images I^{Global} are then augmented using random perspectives and random background removal and used together with the global target text T_{Global} to compute a global loss (Eq. 3). Three global augmented images are used to compute the temporal loss (Eq. 4). Similarly, the I^{Local} images are augmented and used together with the local target text T_{Local} to compute the local loss (Eq. 2). Lastly, a sparsity loss (Eq. 5) is computed from the stylized frames Q_{Style}.

Neural Layered Atlases (NLA) [14] decompose a video into a set of layered 2D atlases. Each atlas provides a unified representation of the appearance of an object or the background throughout the video. NLA use two mapping networks M_f and M_b, where each takes a pixel and time location (x, y, t) in the video as input and outputs the corresponding 2D (u, v)-coordinate in each atlas:

$$M_f(p) = (u_f^p, u_f^p), \quad M_b(p) = (u_b^p, u_b^p)$$

An atlas network A takes the predicted 2D (u, v)-coordinate as input and outputs the atlas's RGB color at that location. Additionally, all pixel coordinates are fed into the Alpha MLP network M_α which outputs the opacity of each atlas at that location. The RGB color can then be reconstructed at each pixel location by alpha-blending the predicted atlas points according to the opacity value predicted by M_α.

NLA enables consistent video editing. First, each atlas is discretized into an image. A user can then manually apply edits using an editing program. These edits are mapped back to the input video using the computed (u, v)-mapping. To get the reconstructed color c^p for pixel p, the color of the predicted foreground color c_f^p, background color c_b^p and predicted opacity value α^p of the edited atlas are blended as follows:

$$c^p = (1 - \alpha^p)c_b^p + \alpha^p c_f^p$$

3.3   Our Stylization Pipeline

Instead of having to manually apply edits to the discretized atlases as in [14], we wish to apply edits automatically using a target text prompt. Specifically, we are interested in modifying the RGB values such that they conform with a target text. We focus on the Atlas MLP A, since A makes the RGB predictions. Instead of using a single atlas MLP A for all atlases, we create two copies of the atlas MLP after pre-training the NLA network, one editing atlas MLP for the foreground object we want to stylize (A) and one for all other atlases (A_b). We then freeze the weights of M_b, M_f, M_α, A_b and finetune the weights of A. While the goal in NLA is to optimize the model to produce the best reconstruction, we instead want to create a stylization that conforms with an input target text.

Our stylization pipeline is illustrated in Fig. 2. The input is a set of raw frames Q_{Raw}. Since we know the position of the object in the video from the α-map, we can compute a bounding box containing the whole object. Once we have the bounding box, we crop each frame in Q_{Raw} such that it only contains the content within the bounding box plus a small margin. All the cropped frames $Q_{Cropped}$ are passed through a pre-trained NLA model, where all MLP weights have been frozen, except for the weights of the A MLP. The NLA method produces a set of stylized frames Q_{Style}.

To fine-tune the weights of A, we sample training batches in both time and space. Our sampling method is illustrated in Fig. 3. First, we sample a set Q_{Sample} uniformly at random among all frames in the input video and pass them through the stylization pipeline (Fig. 2) to create a set Q_{Style}.

For each frame in Q_{Style} we sample n_{Global} views I^{Global} and a set of n_{Local} views I^{Local}. Each of the n_{Global} views is produced by sampling a crop with a size in the range $[0.9, 1.0]$ of the original frame size. Each of the n_{Local} views is produced by sampling a crop with a size in the range $[0.1, 0.5]$ of the original frame size. To ensure the local views contain the object we want to stylize, we use the α-map of the frame to determine the position of the object in the frame. We then sample until we get n_{Local} views where at least $\frac{1}{3}$ of the sampled view is part of the object.

Once the local and global views have been sampled, we apply a random perspective transformation and a random background removal which with some probability p removes the background (details in supplementary) Additionally, each augmented frame is normalized with the same mean and standard deviation as used to train the CLIP model [23].

Our objective function is composed of three main losses defined in CLIP's feature space: (1) L_{Local} which focuses on local semantics, (2) L_{Global} which focuses on global semantics, and (3) L_{Temp} which encourages temporal consistency. Additionally, we use the regularization term L_{sparsity} introduced in NLA [14], which encourages sparse representations. In all loss terms, when we compute the similarity between a text and each sampled view, we use the average embedding across all views:

$$I_{emb}^{Local} = \frac{1}{n_{Local}} \sum_{i=1}^{n_{Local}} E_{img}\left(I_i^{\text{Local}}\right), I_{emb}^{Global} = \frac{1}{n_{Global}} \sum_{i=1}^{n_{Global}} E_{img}\left(I_i^{\text{Global}}\right) \quad (1)$$

Local loss L_{Local} is applied to all views in I^{Local}. The goal is to modify the image, such that the local details conform with a target text T_{Local}:

$$L_{\text{Local}} = 1 - \text{sim}\left(I_{emb}^{Local}, E_{txt}\left(T_{\text{Local}}\right)\right) \tag{2}$$

where $\text{sim}(a, b) = \frac{a \cdot b}{|a||b|}$ is the cosine similarity, and E_{txt} denote CLIP's pre-trained text encoder. The local views have a more zoomed-in view of the object we are stylizing. Additionally, the local target text T_{Local} contains local specific semantics, e.g. "*rough* cactus texture." Hereby, the local loss can focus on the texture and fine-grained details of the stylization we apply to the input video.

Global loss The global loss is applied to views in I^{Global} that all include the entire object being stylized. The intended goal is that the global loss will preserve the overall context. In the target text T_{Global}, we include words that describe the global context of the object we are trying to stylize, e.g. "A *swan* made of cactus." The global loss formulation is then given by:

$$L_{\text{Global}} = 1 - \text{sim}\left(I_{emb}^{Global}, E_{txt}\left(T_{\text{Global}}\right)\right) \tag{3}$$

Temporal loss We use a triplet loss to include a temporal aspect and enforce that consecutive frames should be more similar in CLIP's embedding space than frames that are further apart. To compute the temporal loss, we sample three frames t_1, t_2, t_3, where we have that $t_1 < t_2, < t_3$ w.r.t. the order of the frames in the input video. We then enforce that the similarity between the sampled global views of t_1 and t_2 in the CLIP embedding space should be greater than the similarity between t_1 and t_3: Let $I_{emb(t_1)}^{Global}, I_{emb(t_2)}^{Global}, I_{emb(t_3)}^{Global}$ denote the average embedded global views (computed in Eq. 1) for each of the three frames. Then we compute the triplet loss L_{Temp} as follows:

$$\text{Sim}_{t_1 t_3 - t_1 t_2} = \text{sim}\left(I_{emb(t_1)}^{Global}, I_{emb(t_3)}^{Global}\right) - \text{sim}\left(I_{emb(t_1)}^{Global}, I_{emb(t_2)}^{Global}\right)$$
$$L_{Temp} = \lambda_{Temp} \cdot \max\left(0, \text{Sim}_{t_1 t_3 - t_1 t_2}\right) \tag{4}$$

where λ_{Temp} is a weighting of the temporal loss. If the frames are further away from each other, the temporal loss should contribute less to the overall loss. For this reason, we weigh the contribution of the temporal loss by a Gaussian probability density function g with a mean equal to zero and a standard deviation of five. We compute the weight of a triplet by applying g to the difference between t_1 and t_3:

$$\lambda_{Temp} = g(t_3 - t_1)$$

Sparsity loss We use the same sparsity loss as in NLA [14]. Its intended function is to encourage points that are mapped to the background atlas to have a zero value in the foreground atlas, e.g. if a point p is mapped to the background atlases it should not contain information about the foreground atlas.

$$L_{\text{sparsity}} = \left\|\left(1 - \alpha^P\right) c_f^P\right\| \tag{5}$$

where c_f^P is the predicted color at p for the foreground layer and α^P is the opacity value at location p.

Full objective The full loss term that is minimized is represented as:

$$L = \lambda_{Sparsity}L_{\text{Sparsity}} + \lambda_{Temp}L_{\text{Temp}} + \lambda_{Local}L_{\text{Local}} + \lambda_{Global}L_{\text{Global}} \qquad (6)$$

where λ_{Local},λ_{Global}, λ_{Temp}, $\lambda_{Sparsity}$ are hyperparameters used to control the weighting of each loss term. As default $\lambda_{Local} = \lambda_{Global} = 1$, while λ_{Temp} and and $\lambda_{Sparsity}$ vary depending on the input video.

	Q1 (Realism)	Q2 (Matching Text)
Blended Diffusion [1]	2.22 ($\pm$1.08)	2.10 ($\pm$1.00)
Ours	**3.47** ($\pm$1.10)	**3.94** ($\pm$0.99)

Table 1: Mean opinion scores (1-5) and standard deviation for Q1 and Q2.

4 Experiments

We evaluate our method on a set of videos from the DAVIS dataset [22] across a diverse set of target text prompts. Our goal is to perform consistent and natural video editing. For this purpose, we present both a quantitative and qualitative evaluation of our results and perform a careful ablation study of each loss term in our objective function.

In Sec. 4.1 we demonstrate the capabilities of our method by showing various stylizations for different videos. We also present a quantitative evaluation, where we compare our method to an image baseline method applied to each frame in the video. In Sec. 4.2 we show that the specificity of text prompts influences the details of the results. In Sec. 4.3 we demonstrate how text augmentation affects the results of the stylizations. In Sec. 4.4 we conduct a series of ablations on our loss terms and demonstrate how the local and global losses focus on different semantics. Finally, in Sec. 4.5 we illustrate some limitations of our method.

4.1 Varied stylizations

In Fig. 4 we illustrate our video stylization method applied to three videos and three texts. All local target texts used for these examples are similar to the global target text, but without containing the information about the underlying content, e.g., for the swan, the local target text is "metal skin." The results show that we can apply temporally consistent stylizations that adhere to the target text specification. The swan example shows fine-grained details of the target texts and also preserves the underlying content of the swan. In the boat example, the stylization captures the details of the target text. The boat has a texture similar to shiny aluminum and also has something that looks like a fishing net at the end of the boat. The dog example shows that our method can apply a realistic and consistent stylization to a video containing occlusions.

"Swan"   "Shiny metal swan"   "Swan with wood bark skin"

"Boat"   "Shiny aluminum fishing boat"   "Boat made of copper"

"Dog"   "Dog with zebra fur"   "Golden dog"

Fig. 4: **Example results**. Two representative frames from each edited video together with the global target text.

We quantify the effectiveness of our method by comparing it to an image baseline applied to each frame in a video input. As a baseline, we use a pretrained Blended-diffusion (BF) [1] model with standard configurations. The model takes as input an image, a ROI mask, and a target text. BF performs local (region-based) edits based on a target text description and the ROI mask. We conduct a user study to evaluate the perceived quality of the stylized outputs generated by both the BF model and our method, and the degree to which the outputs adhere to the global target text. Our user study comprises 50 users and 20 stylized videos, each with a different target text. For each video and target text combination, the users are asked to assign a score (1-5) to two factors: (Q1) "How realistic is the video?," (Q2) "does the {*object*} in the video adhere to the text {*content*}. For Q1 we make it clear to the user that "realistic" refers to the quality of the video content. The results are shown in Table 1.
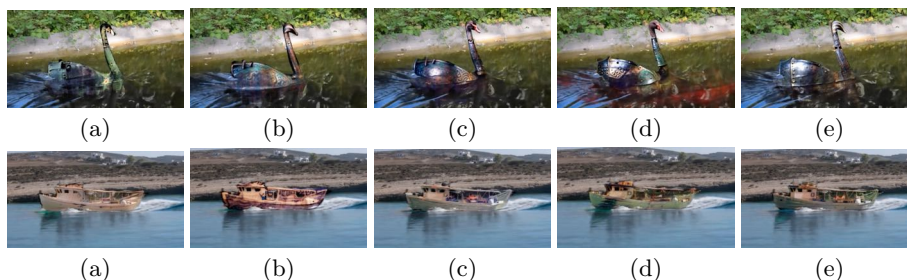

| (a) | (b) | (c) | (d) | (e) |

Fig. 5: **Target text specificity.** Each example shows a representative frame from the video. The experiment shows the specificity of a target affects the stylization. Global target text prompts, *row one:* (a) "Armor," (b) "Iron armor," (c) "Medieval iron armor," (d) "Suit of shiny medieval iron armor," (e) "Full plate shiny medieval iron armor," *row two:* (a) "Boat made of wood," (b) "Boat made of dark walnut wood (c) "Fishing boat made of wood," (d) "Old fishing boat made of wood," (e) "Fishing boat made of wood planks."

4.2 Prompt specificity

We demonstrate that varying the specificity of the target text prompt affects the level of detail in the stylization. Our experiment is motivated by recent work on prompt engineering [32] that shows how slight changes in the target text can have a big impact on the CLIP similarity between a text and an image. Fig. 5 shows an increasing level of detail for two videos and two target texts. The target text specificity not only influences the level of detail, it also makes it easier for CLIP to navigate in its embedding space. In the swan example in column (a), we have "Swan with an armor," which is a more ambiguous target compared to the other swan examples with more detailed target texts and stylizations. We hypothesize that this is because several stylizations can satisfy the more simple target text, while a more specific target text narrows down the set of possible

directions in CLIP's embedding space. The swan examples in columns (d) and (e) indicate that CLIP has some understanding of the different body parts of the swan. The "full plate" target text (d) covers the entire head of the swan while the "suit" of armor in column (e) has a clear cut around the head of the swan.
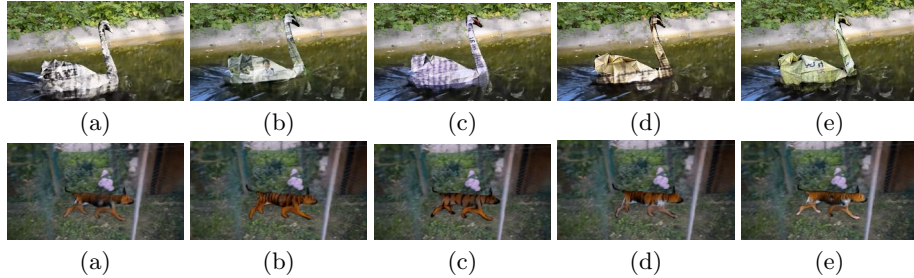

Fig. 6: **Prefix augmentations - varying number of prefixes to sample from in each iteration**. Each figure shows a representative frame from each video and experiment configuration. The examples in row one use the texts: T_{Global}: "Origami swan with white paper skin," T_{Local}: "Origami white paper skin," and in row two: T_{Global}: "Dog with Bengal tiger fur," T_{Local}: "Bengal tiger fur." (a) no prefixes, (b) 4 local, no global, (c) 4 global, no local (d) 4 global & 4 local, (e) 8 global & 8 local. The prefixes are described in the supplementary.
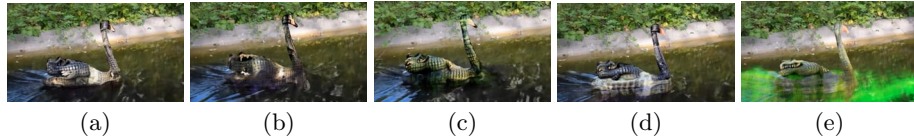

Fig. 7: **Ablation on each loss term** - All experiments were run with the same seed and with the texts: T_{Global}: "A swan with crocodile skin," T_{Local}: "Crocodile skin." Each figure shows a representative frame from a video, where we ablate one of our loss terms in Eq. 6. (a) All losses, (b) w/o local loss, (c) w/o global loss, (d) w/o temporal loss, (e) w/o sparsity loss.

4.3   Text augmentation

We add textual augmentation to our method to address some of the challenges with prompt engineering. Inspired by Zhou et al. [32], we add neutral prefixes to both the local and global target texts, e.g., "a photo of a {}." We then sample a new prefix each iteration for each of the target texts as a form of regularization.
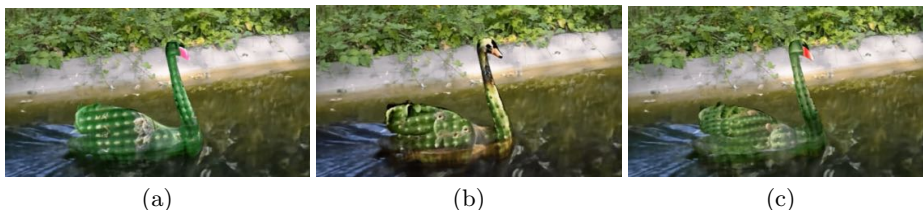
(a)   (b)   (c)

Fig. 8: **Global and local semantics**. A representative frame from each of the edited video frames. The texts used are: (a) T_{Global}: "Swan with cactus skin," T_{Local}: "Cactus skin," (b) T_{Global}: "Swan with cactus skin," T_{Local}: "Rough cactus skin," (c) T_{Global}: "Swan made out of cactus," T_{Local}: "Catus skin." (a) and (b) have the same global target text while (a) and (c) have the same local target texts. In (b) the local details have changed to a more rough texture. In (c) the global swan semantics are better preserved.

Fig. 6 illustrates our text augmentation experiment. We demonstrate that using prefixes increases the quality of the results. In each experiment and each iteration, we sample one prefix among a set of prefixes (details in supplementary) for both the global and local target texts. In this experiment, we vary the number of prefixes to sample from and show that using an equal amount of prefixes for both the local and global target texts produces better quality results than using no prefixes. In both the swan and dog examples for columns (e) and (d) that use multiple prefixes, we see that the stylizations are more detailed than the examples in columns (a-c), e.g., in the dog example (d) and (e) the tiger fur has white pigments around the belly area and also more natural tiger stripes.

4.4   Ablation study

We validate each term in our objective function (Eq. 6) with an ablation study illustrated in Fig. 7. In (a), we see that all losses combined result in a natural stylization that shows clear characteristics of a swan with crocodile skin. In (b) we see that without the **local loss** the results lack fine-grain details. It is still evident what the intended stylization was but the crocodile texture is not as clear as in (a). The results in (c) show that without the **global loss** the global semantics are less clear, e.g., the swan's neck is not stylized as detailed as the rest of the body. In (d), we see that without the **temporal loss**, we get more edits outside the mask of the object. In (e), we see that without the **sparsity loss**, the stylization is very noisy.

In Fig 8 we illustrate how the local loss affects the fine-grained details and the global loss affects the global semantics. We use (a) as a baseline and then vary the local target texts in (b) and the global target texts in (c). In (b) we see, how changing the local target text to include "rough" affects the details of the cactus texture. In (c) we see how the global semantics of the swan's body become more realistic as an effect of changing the global target text.
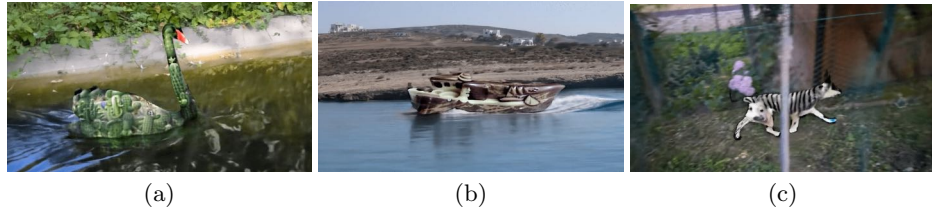
(a)   (b)   (c)

Fig. 9: **Limitations**. A representative frame from each of the edited video frames. The texts used are: (a) T_{Global}: "Swan with cactus skin," T_{Local}: "Cactus skin," (b) T_{Global}: "Boat made out of chocolate," T_{Local}: "Chocolate texture," (c) T_{Global}: "Dog with zebra fur," T_{Local}: "Zebra fur." In (a), the cactus texture is applied like photos of cactus. In (b) the chocolate boat has a sailing ship printed at the end of the boat. In (c), the dog has a face of a dog on its haunches.

4.5   Limitations

Fig. 9 illustrates some of the limitations of our method. During training our method occasionally starts to overfit and produce unintended stylizations. In (a), we see that the body of the swan contains photo-like cacti instead of natural cactus texture. In (b) and (c), we see how our model has used the contexts of the global target text. In (b), a sailing ship has been added to the end of the boat and in (c), a face of a dog has been added. Our method is limited to text-prompts that do not entail features that cross between the decomposed atlas layers. Some changes are best realized through a shape change, e.g, "Swan with long hair.", which is not currently possible in our framework.

5   Conclusion

We considered the problem of developing intuitive and semantic control for consistent editing and styling of objects in videos. This problem poses a challenge in generating consistent content and style changes over time while being able to produce fine-grained details and preserve the global semantics. We proposed a method that uses CLIP [23] and the video object representation [14] to stylize objects in videos by using both a global target text, to control the global semantics of the stylization, and a local target text, to control the fine-grained details. We demonstrated that the specificity and the prefixes of the target texts can have a significant impact on the details produced by our method's stylization. In future work, it would be interesting to investigate the limitations of CLIP. A model that can generate fine-grained stylizations of videos and images could be leveraged to create data augmentations in other learning settings. Another line of future work is to experiment with background stylizations and to extend our model to be able to generate shape changes or even new objects from scratch.

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. CoRR **abs/2111.14818** (2021), https://arxiv.org/abs/2111.14818

2. Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2live: Text-driven layered image and video editing (2022). https://doi.org/10.48550/ARXIV.2204.02491, https://arxiv.org/abs/2204.02491

3. Bau, D., Andonian, A., Cui, A., Park, Y., Jahanian, A., Oliva, A., Torralba, A.: Paint by word. CoRR **abs/2103.10951** (2021), https://arxiv.org/abs/2103.10951

4. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. CoRR **abs/1809.11096** (2018), http://arxiv.org/abs/1809.11096

5. Chefer, H., Benaim, S., Paiss, R., Wolf, L.: Image-based clip-guided essence transfer. CoRR **abs/2110.12427** (2021), https://arxiv.org/abs/2110.12427

6. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance (2022). https://doi.org/10.48550/ARXIV.2204.08583, https://arxiv.org/abs/2204.08583

7. Frans, K., Soros, L.B., Witkowski, O.: Clipdraw: Exploring text-to-drawing synthesis through language-image encoders (2021). https://doi.org/10.48550/ARXIV.2106.14843, https://arxiv.org/abs/2106.14843

8. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. CoRR **abs/2108.00946** (2021), https://arxiv.org/abs/2108.00946

9. Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 19545–19560. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/e2ef524fbf3d9fe611d5a8e90fefdc9c-Paper.pdf

10. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. CoRR **abs/1612.05478** (2016), http://arxiv.org/abs/1612.05478

11. Jetchev, N.: Clipmatrix: Text-controlled creation of 3d textured meshes. CoRR **abs/2109.12922** (2021), https://arxiv.org/abs/2109.12922

12. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. CoRR **abs/1812.04948** (2018), http://arxiv.org/abs/1812.04948

13. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. CoRR **abs/1912.04958** (2019), http://arxiv.org/abs/1912.04958

14. Kasten, Y., Ofri, D., Wang, O., Dekel, T.: Layered neural atlases for consistent video editing. CoRR **abs/2109.11418** (2021), https://arxiv.org/abs/2109.11418

15. Kim, G., Ye, J.C.: Diffusionclip: Text-guided image manipulation using diffusion models. CoRR **abs/2110.02711** (2021), https://arxiv.org/abs/2110.02711

16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations,

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980

17. Kwon, G., Ye, J.C.: Clipstyler: Image style transfer with a single text condition. CoRR **abs/2112.00374** (2021), https://arxiv.org/abs/2112.00374

18. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. CoRR **abs/2112.03221** (2021), https://arxiv.org/abs/2112.03221

19. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. CoRR **abs/2112.10741** (2021), https://arxiv.org/abs/2112.10741

20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

21. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. CoRR **abs/2103.17249** (2021), https://arxiv.org/abs/2103.17249

22. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbelaez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 DAVIS challenge on video object segmentation. CoRR **abs/1704.00675** (2017), http://arxiv.org/abs/1704.00675

23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. CoRR **abs/2103.00020** (2021), https://arxiv.org/abs/2103.00020

24. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022). https://doi.org/10.48550/ARXIV.2204.06125, https://arxiv.org/abs/2204.06125

25. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. CoRR **abs/2102.12092** (2021), https://arxiv.org/abs/2102.12092

26. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding (2022). https://doi.org/10.48550/ARXIV.2205.11487, https://arxiv.org/abs/2205.11487

27. Sanghi, A., Chu, H., Lambourne, J.G., Wang, Y., Cheng, C.Y., Fumero, M., Malekshan, K.R.: Clip-forge: Towards zero-shot text-to-shape generation (2021). https://doi.org/10.48550/ARXIV.2110.02624, https://arxiv.org/abs/2110.02624

28. Texler, O., Futschik, D., Kucera, M., Jamriska, O., Sochorová, S., Chai, M., Tulyakov, S., Sýkora, D.: Interactive video stylization using few-shot patch-based training. CoRR **abs/2004.14489** (2020), https://arxiv.org/abs/2004.14489

29. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

30. Ye, V., Li, Z., Tucker, R., Kanazawa, A., Snavely, N.: Deformable sprites for unsupervised video decomposition (2022). https://doi.org/10.48550/ARXIV.2204.07151, https://arxiv.org/abs/2204.07151

31. Yin, K., Gao, J., Shugrina, M., Khamis, S., Fidler, S.: 3dstylenet: Creating 3d shapes with geometric and texture style variations. CoRR **abs/2108.12958** (2021), https://arxiv.org/abs/2108.12958

32. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. CoRR **abs/2109.01134** (2021), https://arxiv.org/abs/2109.01134