Splicing ViT Features for Semantic Appearance Transfer

Narek Tumanyan* Omer Bar-Tal* Shai Bagon Tali Dekel
Weizmann AI Center (WAIC), Dept. of Computer Science and Applied Math, The Weizmann Inst. of Science
*Indicates equal contribution.

Project webpage: https://splice-vit.github.io

Figure 1. Given two input images—a source *structure* image and a target *appearance* image–our method generates a new image in which the structure of the source image is preserved, while the visual appearance of the target image is transferred in a *semantically* aware manner. That is, objects in the structure image are "painted" with the visual appearance of semantically related objects in the appearance image. Our method leverages a self-supervised, pre-trained ViT model as an external semantic prior. This allows us to train our generator only on a single input image pair, without any additional information (e.g., segmentation/correspondences), and without adversarial training. Thus, our framework can work across a variety of objects and scenes, and can generate high quality results in high resolution (e.g., HD).

Abstract

We present a method for semantically transferring the visual appearance of one natural image to another. Specifically, our goal is to generate an image in which objects in a source structure image are "painted" with the visual appearance of their semantically related objects in a target appearance image. Our method works by training a generator given only a single structure/appearance image pair as input. To integrate semantic information into our framework—a pivotal component in tackling this task—our key idea is to leverage a pre-trained and fixed Vision Transformer (ViT) model which serves as an external semantic prior. Specifically, we derive novel representations of structure and appearance extracted from deep ViT features, untwisting them from the learned self-attention modules. We then establish an objective function that splices the desired structure and appearance representations, interweaving them together in the space of ViT features. Our framework, which we term "Splice", does not involve adversarial training, nor does it require any additional input information such as semantic segmentation or correspondences, and can generate high resolution results, e.g., work in HD. We demonstrate high quality results on a variety of in-the-wild image pairs, under significant variations in the number of objects, their pose and appearance.

1. Introduction

> *"Rope splicing is the forming of a semi-permanent joint between two ropes by partly untwisting and then interweaving their strands." [2]*

What is required to transfer the visual appearance between two semantically related images? Consider for example the task of transferring the visual appearance of a spotted cow in a flower field to an image of a red cow in a grass field (Fig. 1). Conceptually, we have to associate regions in both images that are semantically related, and transfer the visual appearance between these matching regions. Additionally, the target appearance has to be transferred in a realistic manner, while preserving the structure of the source image – the red cow should be realistically "painted" with black and white spots, and the green grass should be covered with yellowish colors. To achieve it under noticeable pose, appearance and shape differences between the two images, *semantic* information is imperative.

Indeed, with the rise of deep learning and the ability to learn high-level visual representations from data, new vision tasks and methods under the umbrella of "visual appearance transfer" have emerged. For example, the image-to-image translation trend aims at translating a source image from one domain to another target *domain*. To achieve that, most methods use generative adversarial networks (GANs),

given image collections from both domains. Our goal is different – rather than generating *some* image in a target domain, we generate an image that depicts the visual appearance of a *particular* target image, while preserving the structure of the source image. Furthermore, our method is trained using only a single image pair as input, which allows us to deal with scenes and objects for which an image collection from each domain is not handy (e.g., spotted cows and red cows image collections).

With only a pair of images available as input, how can we source semantic information? We draw inspiration from Neural Style Transfer (NST) that represents content and an artistic style in the space of deep features encoded by a pre-trained classification CNN model (e.g., VGG). While NST methods have shown a remarkable ability to *globally* transfer artistic styles, their content/style representations are not suitable for *region-based*, semantic appearance transfer across objects in two natural images [12]. Here, we propose novel deep representations of appearance and structure that are extracted from DINO-ViT – a Vision Transformer model that has been pre-trained in a self-supervised manner [4]. Representing structure and appearance in the space of ViT features allows us to inject powerful semantic information into our method and establish a novel objective function that is used to train a generator using only the single input image pair.

DINO-ViT has been shown to learn powerful and meaningful visual representation, demonstrating impressive results on several downstream tasks including image retrieval, object segmentation, and copy detection [4, 1]. However, the intermediate representations that it learns have not yet been fully explored. We thus first strive to gain a better understanding of the information encoded in different ViT's features across layers. We do so by adopting "feature inversion" visualization techniques previously used in the context of CNN features. Our study provides a couple of key observations: (i) the global token (a.k.a [CLS] token) provides a powerful representation of visual appearance, which captures not only texture information but more global information such as object parts, and (ii) the original image can be reconstructed from these features, yet they provide powerful semantic information at high spatial granularity.

Equipped with the above observations, we derive novel representations of structure and visual appearance extracted from deep ViT features – untwisting them from the learned self-attention modules. Specifically, we represent visual appearance via the global [CLS] token, and represent structure via the self-similarity of keys, all extracted from the last layer. We then train a generator on a single input pair of structure/appearance images, to produce an image that *splices* the desired visual appearance and structure in the space of ViT features. Our framework does not require any additional information such as semantic segmentation and does not involve adversarial training. Furthermore, our model can be trained on high resolution images, producing high quality results in HD. We demonstrate a variety of semantic appearance transfer results across diverse natural image pairs, containing significant variations in the number of objects, pose and appearance.

2. Related Work

The problem we tackle here is *semantic* visual appearance transfer between two *in-the-wild*, *natural* images, without user guidance. To the best of our knowledge, there is no existing method addressing specifically this challenge. We review the most related trends and methods.

Domain Transfer & Image-to-Image Translation. The goal of these methods is to learn a mapping between source and target *domains*. This is typically done by training a GAN on a *collection* of images from the two domains, either paired [11] or unpaired [40, 19, 14, 37, 24]. Swapping Autoencoder (SA) [25] trains a domain-specific GAN to disentangle structure and texture in images, and swap these representations between two images in the domain. In contrast to SA, our method is not restricted to any particular domain, and it does not require a collection of images for training, nor it involves adversarial training.

Recently, image to image translation methods trained on a single example were proposed [7, 3, 18]. These methods only utilize low-level visual information and lack semantic understanding. Our method is also trained only on a single image pair, but leverages a pretrained ViT model to inject powerful semantic information into the generation process (see Sec. 4 for comparison).

Neural Style Transfer (NST). In its classical setting, NST transfers an *artistic* style from one image to another [9, 12]. STROTSS [16] uses pre-trained VGG features to represent style and their self-similarity to capture structure in an optimization-based framework to perform *artistic* style transfer in a *global* manner. In contrast, our goal is to transfer the appearance between *semantically* related objects and regions in two *natural* images.

Semantic style transfer methods also aim at mapping appearance across semantically related regions between two images [21, 17, 35, 34]. However, these methods are usually restricted to color transformation [36, 34, 38], or depend on additional semantic inputs (e.g., annotations, segmentation, point correspondences, etc.) [9, 13, 5, 16]. Other works tackle the problem for specific controlled domains [29, 30]. In contrast, we aim to work with arbitrary, in-the-wild input pairs.

Vision Transformers (ViT). ViTs [8] have been shown to achieve competitive results to state-of-the-art CNN architectures on image classification tasks, while demonstrating impressive robustness [22]. DINO-ViT [4] is a ViT model that has been trained, without labels, using a self-distillation approach. The effectiveness of the learned representation has been demonstrated on several downstream tasks, including image retrieval and segmentation.

Amir et al. [1] have demonstrated the power of DINO-ViT Features as dense visual descriptors. Their key observation is that deep DINO-ViT features capture rich seman-

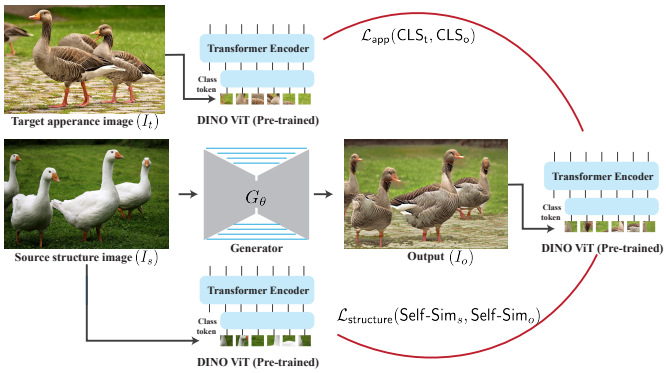


Figure 2. **Pipeline.** Our generator G_θ takes an input structure image I_s and outputs I_o. We establish our training losses using a pre-trained and fixed DINO-ViT model, which serves as an external semantic prior: we represent *structure* via the self-similarity of keys in the deepest attention module (Self-Sim), and *appearance* via the [CLS] token in the deepest layer. Our objective is twofold: (i) $\mathcal{L}_{\text{app}}$ encourages the [CLS] of I_o to match the [CLS] of I_t, and (ii) $\mathcal{L}_{\text{structure}}$ encourages the self-similarity representation of I_o and I_s to be the same. See Sec. 3.3 for details.

tic information at fine spatial granularity, e.g, describing semantic object *parts*. Furthermore, they observed that the representation is shared across different yet related object classes. This power of DINO-ViT features was exemplified by performing "out-of-the-box" unsupervised semantic part co-segmentation and establishing semantic correspondences across different objects categories. Inspired by these observations, we harness the power of DINO-ViT features in a novel generative direction – we derive new perceptual losses capable of splicing structure and semantic appearance across semantically related objects.

3. Method

Given a source structure image I_s and a target appearance image I_t, our goal is to generate an image I_o, in which objects in I_s are "painted" with the visual appearance of their semantically related objects in I_t.

Our framework is illustrated in Fig. 2: for a given pair $\{I_s, I_t\}$, we train a generator $G_\theta(I_s) = I_o$. To establish our training losses, we leverage DINO-ViT – a self-supervised, pre-trained ViT model [4] – which is kept fixed and serves as an external high-level prior. We propose new deep representations for *structure* and *appearance* in DINO-ViT feature space; we train G_θ to output an image, that when fed into DINO-ViT, matches the source structure and target appearance representations. Specifically, our training objective is twofold: (i) $\mathcal{L}_{\text{app}}$ that encourages the deep appearance of I_o and I_t to match, and (ii) $\mathcal{L}_{\text{structure}}$, which encourages the deep structure representation of I_o and I_s to match.

We next briefly review ViT architecture, then provide qualitative analysis of DINO-ViT's features in Sec. 3.2, and describe our framework in Sec. 3.3.

3.1. Vision Transformers – overview

In ViT, an image I is processed as a sequence of n non-overlapping patches as follows: first, *spatial tokens* are formed by linearly embedding each patch to a d-dimensional vector, and adding learned position embeddings. An additional learnable token, a.k.a [CLS] token, serves as a global representation of the image.

The set of tokens are then passed through L Transformer layers, each consists of normalization layers (LN), Multi-head Self-Attention (MSA) modules, and MLP blocks:

$$\hat{T}^l = \text{MSA}(\text{LN}(T^{l-1})) + T^{l-1}$$
$$T^l = \text{MLP}(\text{LN}(\hat{T}^l)) + \hat{T}^l$$

where $T^l(I) = \left[t^l_{cls}(I), t^l_1(I) \dots t^l_n(I)\right]$ are the output tokens for layer l for image I.

In each MSA block the (normalized) tokens are linearly projected into queries, keys and values:.

$$Q^l = T^{l-1} \cdot W^l_q,\ K^l = T^{l-1} \cdot W^l_k,\ V^l_h = T^{l-1} \cdot W^l_v \quad (1)$$

which are then fused using multihead self-attention to form the output of the MSA block (for full details see [8]).

After the last layer, the [CLS] token is passed through an additional MLP to form the final output, e.g., output distribution over a set of labels [8]. In our framework, we leverage DINO-ViT [4], in which the model has been trained in a self-supervised manner using a self-distillation approach. Generally speaking, the model is trained to produce the same distribution for two different augmented views of the same image. As shown in [4], and in [1], DINO-ViT learns powerful visual representations that are less noisy and more semantically meaningful than the supervised ViT.

3.2. Structure & Appearance in ViT's Feature Space

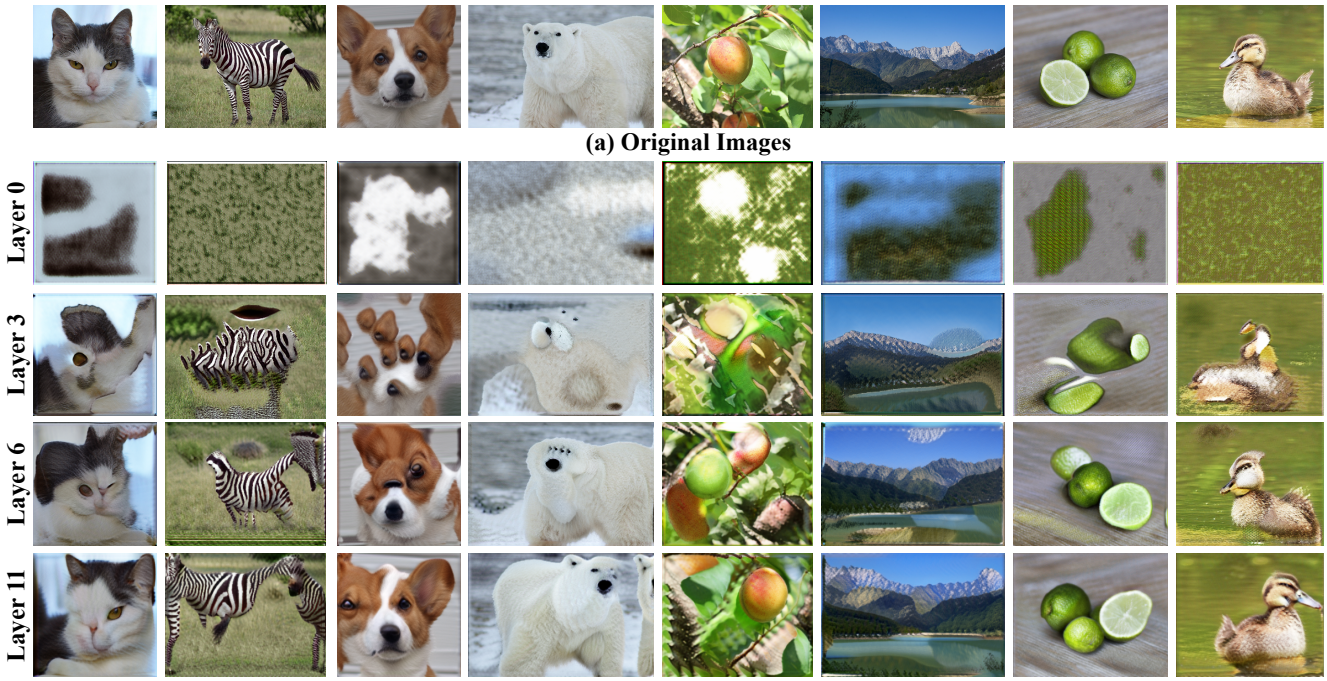
The pillar of our method is the representation of *appearance* and *structure* in the space of DINO-ViT features. For appearance, we want a representation that can be spatially flexible, i.e., discards the exact objects' pose and scene's spatial layout, while capturing global appearance information and style. To this end, we leverage the [CLS] token, which serves as a *global* image representation.

For structure, we want a representation that is robust to local texture patterns, yet preserves the spatial layout, shape and perceived semantics of the objects and their surrounding. To this end, we leverage deep *spatial* features extracted from DINO-ViT, and use their *self-similarity* as structure representation:

$$S^L(I)_{ij} = \text{cos-sim}\left(k^L_i(I), k^L_j(I)\right) \quad (2)$$

cos-sim is the cosine similarity between keys (See Eq. 1). Thus, the dimensionality of our self-similarity descriptor becomes $S^L(I) \in \mathbb{R}^{(n+1)\times(n+1)}$, where n is the number of patches.

The effectiveness of self-similarly-based descriptors in capturing *structure* while ignoring *appearance* information have been previously demonstrated by both classical methods [28], and recently also using deep CNN features for



(a) Original Images

(b) CLS token inversion results across layers

Figure 3. **Inverting the [CLS] token across layers.** Each input image (a) is fed to DINO-ViT to compute its global [CLS] token at different layers. (b) Inversion results: starting from a noise image, we optimize for an image that would match the original [CLS] token at a specific layer. While earlier layers capture local texture, higher level information such as object parts emerges at the deeper layers (see Sec. 3.2).

artistic style transfer [16]. We opt to use the self similarities of *keys*, rather than other facets of ViT, based on [1].

Understanding and visualizing DINO-ViT's features. To better understand our ViT-based representations, we take a *feature inversion* approach – given an image, we extract target features, and optimize for an image that has the same features. Feature inversion has been widely explored in the context of CNNs (e.g., [31, 20]), however has not been attempted for understanding ViT features yet. For CNNs, it is well-known that solely optimizing the image pixels is insufficient for converging into a meaningful result [23]. We observed a similar phenomenon when inverting ViT features (see Supplementary Materials on our website – SM). Hence, we incorporate "Deep Image Prior" [33], i.e., we optimize for the weights of a CNN F_θ that translates a fixed random noise z to an output image:

$$\arg\min_{\theta} ||\phi(F_\theta(z)) - \phi(I)||_F, \tag{3}$$

where $\phi(I)$ denotes the target features, and $||\cdot||_F$ denotes Frobenius norm. First, we consider inverting the [CLS] token: $\phi(I) = t^l_{cls}(I)$. Figure 3 shows our inversion results across layers, which illustrate the following observations:

1. From shallow to deep layers, the [CLS] token gradually accumulates appearance information. Earlier layers mostly capture local texture patterns, while in deeper layers, more global information such as object parts emerges.

2. The [CLS] token encodes appearance information in a *spatially flexible manner*, i.e., different object parts can stretch, deform or be flipped. Figure 4 shows multiple runs of our inversions per image; in all runs, we can notice similar global information, but the diversity across runs demonstrates the spatial flexibility of the representation.

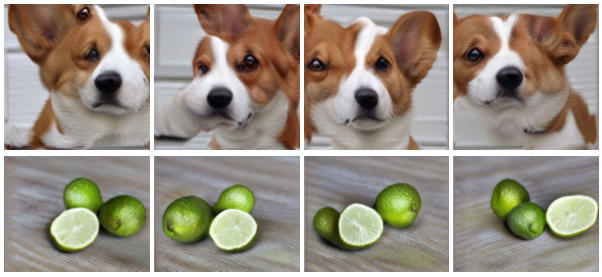
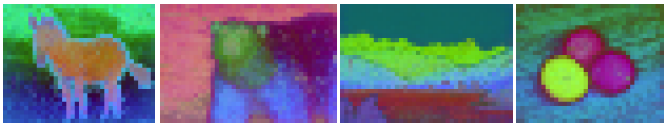


Figure 4. **[CLS] token inversion over multiple runs.** The variations in structure in multiple inversion runs of the same image demonstrates the spatial flexibility of the [CLS] token.

Next, in Fig. 5(a), we show the inversion of the spatial keys extracted from the last layer, i.e., $\phi(I) = K^L(I)$. These features have been shown to encode high level information [4, 1]. Surprisingly, we observe that the original image can still be reconstructed from this representation.

To discard appearance information encoded in the keys, we consider the self-similarity of the keys (see Sec. 3.2). This is demonstrated in the PCA visualization of the keys' self-similarity in Fig. 5(b). As seen, the self-similarity mostly captures the structure of objects, as well as their distinct semantic components. For example, the legs and the

(a) Keys inversion results (layer 11)



(b) PCA of keys' self-similarity (3 leading components, layer 11)

Figure 5. **Visualization of DINO-ViT keys.** (a) Inverting keys from the deepest layer surprisingly reveals that the image can be reconstructed. (b) PCA visualization of the keys' self-similarity: the leading components mostly capture semantic scene/objects parts, while discarding appearance information (e.g., zebra stripes).

body of the polar bear that have the same texture, are distinctive.

3.3. Splicing ViT Features

Based on our understanding of DINO-ViT's internal representation, we turn to the task of training our generator.

Our objective function takes the following form:

$$\mathcal{L}_{\text{splice}} = \mathcal{L}_{\text{app}} + \alpha\mathcal{L}_{\text{structure}} + \beta\mathcal{L}_{\text{id}}, \tag{4}$$

where α and β set the relative weights between the terms. The driving loss of our objective function is $\mathcal{L}_{\text{app}}$, and we set $\alpha = 0.1, \beta = 0.1$ for all experiments.

Appearance loss. The term $\mathcal{L}_{\text{app}}$ encourages the output image to match the appearance of I_t, and is defined as the difference in [CLS] token between the generated and texture image:

$$\mathcal{L}_{\text{app}} = \left\| t^L_{[\text{CLS}]}(I_t) - t^L_{[\text{CLS}]}(I_o) \right\|_2, \tag{5}$$

where $t^L_{[\text{CLS}]}(\cdot) = t^L_{cls}$ is the [CLS] token extracted from the deepest layer (see Sec. 3.1).

Structure loss. The term $\mathcal{L}_{\text{structure}}$ encourages the output image to match the structure of I_s, and is defined by the difference in self-similarity of the keys extracted from the attention module at deepest transformer layer:

$$\mathcal{L}_{\text{structure}} = \left\| S^L(I_s) - S^L(I_o) \right\|_F, \tag{6}$$

where $S^L(I)$ is defined in Eq. (2).

Identity Loss. The term $\mathcal{L}_{\text{id}}$ is used as a regularization. Specifically, when we feed I_t to the generator, this loss encourages G_θ to preserve the keys representation of I_t:

$$\mathcal{L}_{\text{id}} = \left\| K^L(I_t) - K^L(G_\theta(I_t)) \right\|_F \tag{7}$$

Similar loss terms, defined in RGB space, have been used as a regularization in training GAN-based generators for image-to-image translation [24, 32, 40]. Here, we apply the identity loss with respect to the *keys* in the deepest ViT layer, a semantic yet invertible representation of the input image (as discussed in section 3.2).

Data augmentations and training. Since we only have a single input pair $\{I_s, I_t\}$, we create additional training examples, $\{I^i_s, I^i_t\}_{i=1}^N$, by applying augmentations such as crops and color jittering (see Appendix A.4 for implementation details). G_θ is now trained on multiple *internal examples*. Thus, it has to learn a good mapping function for a *dataset* containing N examples, rather than solving a test-time optimization problem for a single instance. Specifically, for each example, the objective is to generate $I^i_o = G_\theta(I^i_s)$, that matches the structure of I^i_s and the appearance of I^i_t.

4. Results

Datasets. We tested our method on a variety of image pairs gathered from Animal Faces HQ (AFHQ) dataset [6], and images crawled from Flickr Mountain. In addition, we collected our own dataset, named *Wild-Pairs*, which includes a set of 25 high resolution image pairs taken from Pixabay, each pair depicts semantically related objects from different categories including animals, fruits, and other objects. The number of objects, pose and appearance may significantly change between the images in each pair. The image resolution ranges from 512px to 2000px.

Sample pairs from our dataset along with our results can be seen in Fig. 1 and Fig. 6, and the full set of pairs and results is included in the SM. As can be seen, in all examples, our method successfully transfers the visual appearance in a semantically meaningful manner at several levels: (i) *across objects:* the target visual appearance of objects is being transferred to to their semantically related objects in the source structure image, under significant variations in pose, number of objects, and appearance between the input images. (ii) *within objects:* visual appearance is transferred between corresponding body parts or object elements. For example, in Fig. 6 top row, we can see the appearance of a single duck is semantically transferred to each of the 5 ducks in the source image, and that the appearance of each body part is mapped to its corresponding part in the output image. This can be consistently observed in all our results.

The results demonstrate that our method is capable of performing semantic appearance transfer across diverse image pairs, unlike GAN-based methods which are restricted to the dataset they have been trained on.

4.1. Comparisons to Prior Work

There are no existing methods that are tailored for solving our task: semantic appearance transfer between two natural images (not restricted to a specific domain), without explicit user-guided inputs. We thus compare to prior works in which the problem setting is most similar to ours in some aspects (see discussion in these methods in Sec. 2): (i) *Swapping Autoencoders (SA)* [25] – a domain-specific, GAN-based method which has been trained to "swap" the texture and structure of two images in a realistic manner; (ii) *STROTSS* [16], the style transfer method that also uses self-similarity of a pre-trained CNN features as the content

Figure 6. **Sample results on in-the-wild image pairs.** For each example, shown left-to-right: the target appearance image, the source structure image and our result. The full set of results is included in the SM. Notice the variability in number of objects, pose, and the significant appearance changes between the images in each pair.

descriptor, (ii) *WCT*^{2} [38], a photorealistic NST method.

Since SA requires a dataset of images from two domains to train, we can only compare our results to their trained models on AHFQ and Flicker Mountain datasets. For the rest of the methods, we also later compare to image pairs from our *Wild-Pairs* examples. We evaluate our performance across a variety of image pairs both qualitatively, quantitatively and via an AMT user study.

4.1.1 Qualitative comparison

Figure 7 shows sample results for all methods (additional results are included in the SM). In all examples, our method correctly relates semantically matching regions between the input images, and successfully transfer the visual appearance between them. In the landscapes results (first 3 columns), it can be seen that SA outputs high quality images but sometimes struggles to maintain high fidelity to the structure and appearance image: elements for the appearance image are often missing e.g., the fog in the left most example, or the trees in the second from left example. These visual elements are captured well in our results. For AHFQ, we noticed that SA often outputs a result that is nearly identical to the structure image. A possible cause to such behavior might be the adversarial loss, which ensures that the swapping result is a realistic image according to the the distribution of the training data. However, in some cases, this requirement does not hold (e.g. a German Shepherd with leopard's texture), and by outputting the structure image the adversarial loss can be trivially satisfied.[1].

NST frameworks such as STROTSS and WCT^2 well preserve the structure of the source image, but their results often depict visual artifacts: STROTSS's results often suffer from color bleeding artifacts, while WCT^2 results in global color artifacts, demonstrating that transferring color is insufficient for tackling our task.

Our method demonstrates better fidelity to the input structure and appearance images than GAN-based SA, while training only on the single input pair, without requiring a large collection of examples from each domain. With respect to style transfer, our method better transfers the appearance across semantically related regions in the input images, such as matching facial regions (e.g., eyes-to-eyes, nose-to-nose), while persevering the source structure.

Finally, we also include a comparison to SinCUT [24], a recent GAN-based image translation method. As demonstrated in Fig. 8, SinCUT performs well for the landscape example, but since it can only utilize low-level visual infor-

[1] We verified these results with the authors [25]

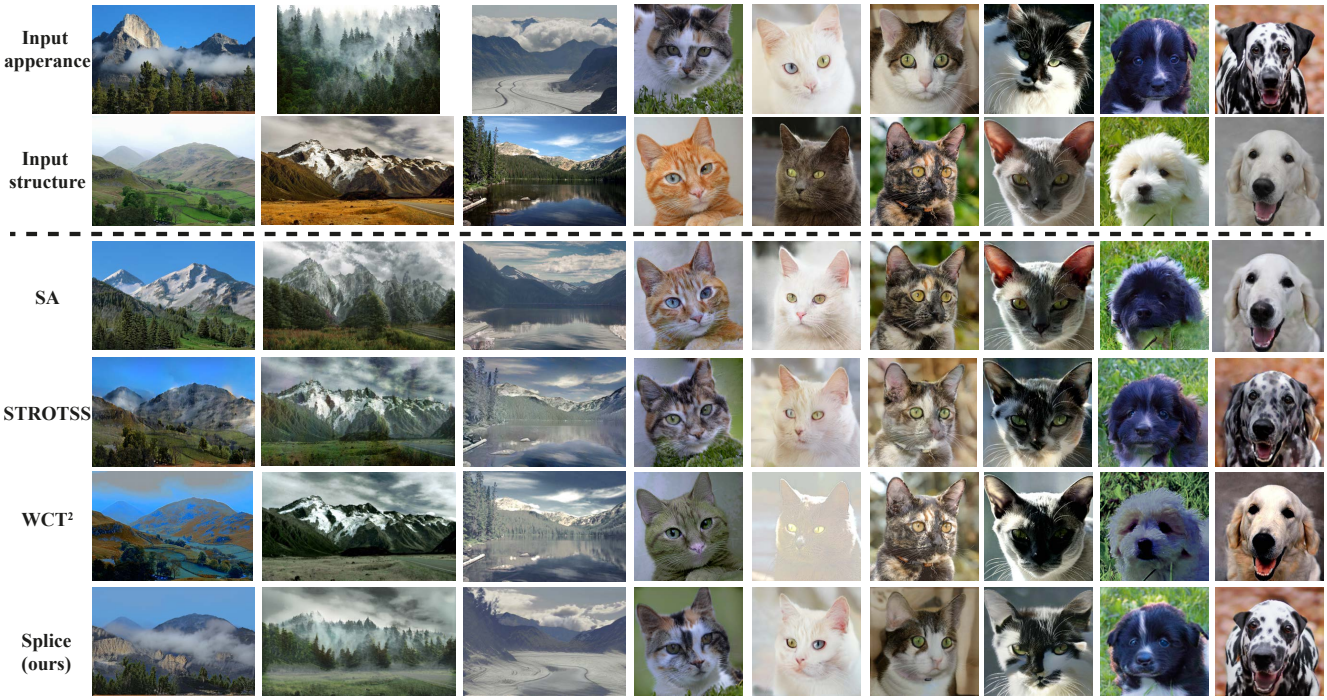


Figure 7. **Comparisons with style transfer and swapping autoencoders.** First two rows: input appearance and structure images taken from the AFHQ and Flickr Mountains. The following rows, from top to bottom, show the results of: swapping autoencoders (SA) [25], STROTSS [16], and WCT^2 [38]. See SM for additional comparisons.

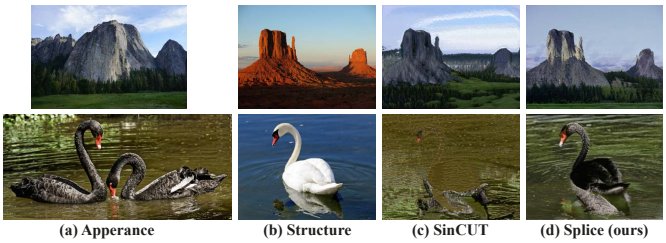


Figure 8. **Comparison to SinCUT [24].** SinCUT results (c), when trained on each input pair (a-b), demonstrates it works well when the translation is mostly based on low-level information (top), but fails when higher level reasoning is required (bottom). (d) Our method successfully transfers the appearance across semantic regions, and generates high quality results w/o adversarial training.

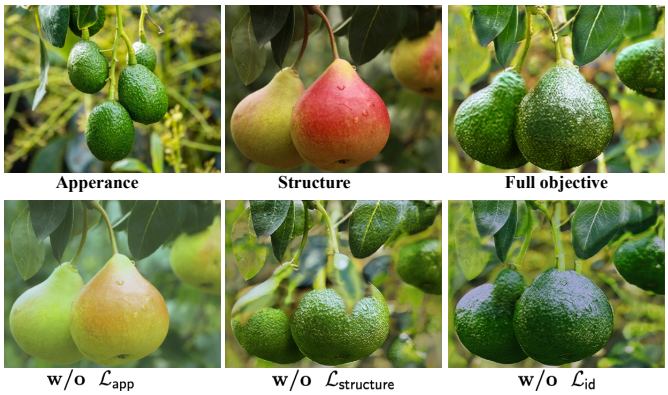


Figure 9. **Loss ablation.** Our results when training without specific loss terms. When one of our loss terms is removed, the model fails to map the target appearance, preserve the input structure, or maintain fine details. See Sec. 4.2 for more details.

mation, it fails to transfer the appearance of the swan in the second example. Our method successfully transfers the appearance across semantically realted regions, and generates high quality results w/o adversarial loss.

4.1.2 Quantitative comparison

There is no existing automatic metric suitable for evaluating semantic appearance transfer across two natural images. We follow existing style/appearance transfer methods, which mostly rely on human perceptual evaluation (e.g., [12, 21, 13, 25]), and perform an extensive user study on Amazon Mechanical Turk (AMT).

Human Perceptual Evaluation We design a user survey suitable for evaluating the task of appearance transfer across semantically related scenes. We adopt the Two-alternative Forced Choice (2AFC) protocol suggested in [25, 16]. Participants are shown with 2 reference images: the input structure image (A), shown in grayscale, and the input appearance image (B), along with 2 alternatives: our result and another baseline result. The participants are asked: *"Which image best shows the shape/structure of image A combined with the appearance/style of image B?"*.

We perform the survey using a collection of 65 images in total, gathered from AFHQ, Mountains, and Wild-Pairs. We collected 7000 user judgments w.r.t. existing baselines. Table 4.1.2 reports the percentage of votes in our favor. As

(a) A and B are semantically related, B is non-realistic

(b) A and B are from different, unrelated object categories

Figure 10. **Semantic appearance transfer across different domains.** (a) Objects in the input images (A-B) are semantically related, yet B is non-realistic. (b) Objects are from unrelated object categories. See Sec. 4.3 for discussion.

	SA	STROTSS	WCT^2
Wild-Pairs	-	**79.0** $\pm$ 13.0	**83.1** $\pm$ 14.9
mountains	**56.3** $\pm$ 10.0	**58.8** $\pm$ 14.2	**60.3** $\pm$ 12.1
AFHQ	**71.8** $\pm$ 7.7	**59.7** $\pm$ 15.3	**61.0** $\pm$ 18.3

Table 1. **AMT perceptual evaluation**. We report results on AMT surveys evaluating the task of appearance transfer across semantically related scenes/objects (see Sec. 4.1.2). For each dataset and a baseline, we report the percentage of judgments in our favor (mean, std). Our method outperforms all baselines: GAN-based, SA [25], and style transfer methods, STROTSS [16], and WCT^2 [38].

	SA	STROTSS	WCT^2	Splice (Ours)
Wild-Pairs	-	0.83±0.11	**0.89±0.06**	0.88±0.06
mountains	0.91±0.07	0.94±0.12	**0.96±0.82**	0.95±0.10

Table 2. **mean IoU of output images with respect to the input structure images**. We extract semantic segmentation maps using Mask-RCNN [10] for the Wild-Pairs collection, and [39] for the mountains collection.

seen, our method outperforms all baselines across all image collections, especially for in the Wild-Pairs, which highlights our performance in challenging settings. Note that SA was trained on 500K mountain images, yet our method perform competitively.

Semantic layout preservation. A key property of our method is the ability to preserve the semantic layout of the scene (while significantly changing the appearance of objects). We demonstrate this through the following evaluation. We run semantic segmentation off-the-shelf model (e.g., MaskRCNN [10]) to compute object masks for the input structure images and our results. Table 2 reports IoU for our method and the baselines. Our method better preserves the scene layout than SA and STROTSS, and is the closet competitor to WCT^2 which only modifies colors, and as expected, achieves the highest IoU.

4.2. Ablation

We ablate the different loss terms in our objective by qualitatively comparing the results for our method when trained with our full objective (Eq. 4), and with a specific loss removed. The results are shown in Fig. 9. As can be seen, without the **appearance loss** (w/o $\mathcal{L}_{\text{app}}$), the model fails to map the target appearance, but only slightly modifies the colors of the input structure image due to the identity loss. That is, the identity loss encourages the model to learn an identity when it is fed with the target appearance image, and therefore even without the appearance loss some appearance supervision is available. Without the **structure loss** (w/o $\mathcal{L}_{\text{structure}}$), the model outputs an image with the desired appearance, but fails to fully preserve the structure of the input image, as can be seen by the distorted shape of the pears. Lastly, we observe that the **identity loss** encourages the model to pay more attention to fine details both in terms of appearance and structure, e.g., the fine texture details of the avocado are refined.

4.3. Limitations

Our performance depends on the internal representation learned by DINO-ViT. Therefore, in cases where the representation does not capture well the semantic association across objects in both images, our method would fail to accomplish that too. Figure 10 shows a few such cases: (a) objects are semantically related but one image is highly non-realistic (and thus out of distribution for DINO-ViT). For some regions, our methods successfully transfer the appearance but for some others it fails. In the cat example, we can see that in B-to-A result, the face and the body of the cat are nicely mapped, yet our method fails to find a semantic correspondence for the rings, and we get a wrong mapping of the ear from image A. In (b), our method does not manage to semantically relate a bird to an airplane.

5. Conclusions

We tackled a new problem setting in the context of style/appearance transfer: semantically transferring appearance across related objects in two in-the-wild natural images, without any user guidance. Our approach demonstrates the power of DINO-ViT as an external semantic prior, and the effectiveness of utilizing it to establish or training losses – we show how structure and appearance information can be disentangled from an input image, and then spliced together in a semantically meaningful way in the space of ViT features, through a generation process. We demonstrated that our method can be applied on a variety of challenging input pairs across domains, in diverse poses and multiplicity of objects, and can produce high-quality result without any adversarial training. Our work unveils the potential of self-supervised representation learning not only for discriminative tasks such as image classification, but also for learning more powerful generative models.

Acknowledgments: We would like to thank Meirav Galun and Shir Amir for their insightful comments and discussion. This project received funding from the Israeli Science Foundation (grant 2303/20), and the Carolito Stiftung. Dr Bagon is a Robin Chemers Neustein Artificial Intelligence Fellow.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021. 2, 3, 4

[2] Frank Beech. Splicing ropes illustrated. *CCCBR*, 2005. 1

[3] Saguy Benaim, Ron Mokady, Amit Bermano, and Lior Wolf. Structural analogy from a single image pair. *Comput. Graph. Forum*, 2021. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 2, 3, 4, 10

[5] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv*, 2016. 2

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[7] Tomer Cohen and Lior Wolf. Bidirectional one-shot unsupervised domain mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2, 3

[9] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2017. 8

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[12] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE Trans. Vis. Comput. Graph.*, 2020. 2, 7

[13] Sunnie SY Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Deformable style transfer. In *Eur. Conf. Comput. Vis.*, 2020. 2, 7

[14] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning*. PMLR, 2017. 2

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 10

[16] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 4, 5, 7, 8

[17] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[18] Jianxin Lin, Yingxue Pang, Yingce Xia, Zhibo Chen, and Jiebo Luo. Tuigan: Learning versatile image-to-image translation with two unpaired images. In *European Conference on Computer Vision*. Springer, 2020. 2

[19] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. Curran Associates Inc., 2017. 2

[20] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them, 2014. 4

[21] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Eur. Conf. Comput. Vis.*, 2018. 2, 7

[22] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers, 2021. 2

[23] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. 4

[24] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*. Springer, 2020. 2, 5, 6, 7

[25] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *arXiv preprint arXiv:2007.00653*, 2020. 2, 5, 6, 7, 8

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 10

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. 10

[28] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 3

[29] Yi-Chang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Trans. Graph.*, 2014. 2

[30] Yi-Chang Shih, Sylvain Paris, Frédo Durand, and William T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph.*, 2013. 2

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at In-*

ternational Conference on Learning Representations, 2014. 4

[32] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 5

[33] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv:1711.10925*, 2017. 4

[34] Li Wang, Nan Xiang, Xiaosong Yang, and Jianjun Zhang. Fast photographic style transfer based on convolutional neural networks. In *Proceedings of Computer Graphics International 2018*, CGI 2018, New York, NY, USA, 2018. Association for Computing Machinery. 2

[35] Pierre Wilmot, Eric Risser, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *ArXiv*, 2017. 2

[36] Zhongyou Xu, Tingting Wang, Faming Fang, Yun Sheng, and Guixu Zhang. Stylization-based architecture for fast deep exemplar colorization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[37] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[38] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 6, 7, 8

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018. 8

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5

A. Implementation Details

A.1. Generator Network Architecture

We base our generator G_θ network on a `U-Net` architecture [27], with a 5-layer encoder and a symmetrical decoder. All layers comprise 3×3 Convolutions, followed by `BatchNorm`, and `LeakyReLU` activation. The encoder's channels dimensions are $[3 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 128]$ (the decoder follows a reversed order). In each level of the encoder, we add an additional 1×1 Convolution layer and concatenate the output features to the corresponding level of the decoder. Lastly, we add a 1×1 Convolution layer followed by `Sigmoid` activation to get the final RGB output.

A.2. ViT Feature Extractor Architecture

As described in Sec. 3, we leverage a pre-trained ViT model (DINO-ViT [4]) trained in a self-supervised manner as a feature extractor. We use the 12 layer pretrained model in the 8×8 patches configuration (`ViT-B/8`), downloaded from the official implementation at GitHub.

A.3. Training Details

We implement our framework in PyTorch [26] (code will be made available). We optimize our full objective (Eq. 4, Sec. 3.3), with relative weights: $\alpha = 0.1$, $\beta = 0.1$. We use the Adam optimizer [15] with a constant learning rate of $\lambda = 2 \cdot 10^{-3}$. Each batch contains $\{\tilde{I}_s, \tilde{I}_t\}$, the augmented views of the source structure image and the target appearance image respectively. Every 75 iterations, we add $\{I_s, I_t\}$ to the batch (i.e., do not apply augmentations). The resulting images $\{G(\tilde{I}_s), G(\tilde{I}_t)\}$ and $\tilde{I}_t$ are then resized down to 224[pix] (maintaining aspect ratio) using bicubic interpolation, before extracting DINO-ViT features for estimating the losses. Training on an input image pair of size 512×512 takes ~ 20 minutes to train on a single GPU (Nvidia RTX 6000) for a total of 2000 iterations.

A.4. Data Augmentations

We apply data augmentations to the input image pair $\{I_s,\ I_t\}$ to create multiple *internal examples* $\{I_s^i, I_t^i\}_{i=1}^N$. Specifically, at each training step, we apply the following augmentations:

Augmentations to the source structure image I_s:

- random cropping: we uniformly sample a NxN crop such that N is between 95% - 100% of the height of I_s.
- random horizontal-flipping, applied in probability p=0.5.
- random color jittering: in probability p=0.5 we jitter the brightness, contrast, saturation and hue of the image.
- random Gaussian blurring: in probability p=0.5 we apply a Gaussian blurring 3x3 filter (σ is uniformly sampled between 0.1-2.0).

Augmentations to the target appearance image I_t:

- random cropping: we uniformly sample a NxN crop such that N is between 95% - 100% of the height of I_t.
- random horizontal-flipping, applied in probability p=0.5.