Disentangling Structure and Appearance in ViT Feature Space

NAREK TUMANYAN, OMER BAR-TAL, SHIR AMIR, SHAI BAGON, and TALI DEKEL,
Weizmann Institute of Science, Israel

Fig. 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. We derive novel disentangled appearance and structure representations from our semantic prior, which allows us to train a generator 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).

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. To integrate semantic information into our framework, our key idea is to leverage a pre-trained and fixed Vision Transformer (ViT) model. Specifically, we derive novel disentangled representations of structure and appearance extracted from deep ViT features. We then establish an objective function that splices the desired structure and appearance representations, interweaving them together in the space of ViT features. Based on our objective function, we propose two frameworks of semantic appearance transfer – "Splice", which works by training a generator on a *single and arbitrary* pair of structure-appearance images, and "SpliceNet", a *feed-forward* real-time appearance transfer model trained on a *dataset* of images from a *specific domain*. Our frameworks do not involve adversarial training, nor do they require any additional input information such as semantic segmentation or correspondences. We demonstrate high-resolution results on a variety of in-the-wild image pairs, under significant variations in the number of objects, pose, and appearance. Code and supplementary material are available in our project page: splice-vit.github.io.

CCS Concepts: • **Computing methodologies** → **Shape representations**; **Appearance and texture representations**; **Image-based rendering**; *Image processing*.

Additional Key Words and Phrases: Style Transfer, Real-Time Style Transfer, Feature Inversion, Vision Transformers

Authors' address: Narek Tumanyan, narek.tumanyan@weizmann.ac.il; Omer Bar-Tal, omer-bar.tal@weizmann.ac.il; Shir Amir, shiramiremail@gmail.com; Shai Bagon, shai.bagon@weizmann.ac.il; Tali Dekel, tali.dekel@weizmann.ac.il,
Weizmann Institute of Science, Rehovot, Israel.

1 INTRODUCTION

"Rope splicing is the forming of a semi-permanent joint between two ropes by partly untwisting and then interweaving their strands." [Beech 2005]

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 line of work 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.

Given a pair of structure and appearance images, how can we source semantic information necessary for the task of semantic appearance transfer? 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 [Jing et al. 2020]. 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 [Caron et al. 2021]. 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 for semantic appearance transfer. Based on our objective function, we propose two frameworks of semantic appearance transfer: (i) a generator trained on a *single and in-the-wild* input image pair, (ii) a *feed-forward* generator trained on a dataset of *domain-specific* images.

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 [Amir et al. 2022; Caron et al. 2021; Melas-Kyriazi et al. 2022; Siméoni et al. 2021; Wang et al. 2022]. 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 attention module of last layer. We then design a framework of training 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 single-pair framework, which we term *Splice*, 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. Training on a single pair allows us to deal with arbitrary scenes and objects, without the need to collect a dataset of a specific domain. 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.

While demonstrating exciting results, Splice also suffers from several limitations. First, for every input pair, it requires training a generator from scratch, which usually takes $\sim$ 20 minutes of training until convergence. This makes Splice inapplicable for real-time usage. Second, Splice is limited to observing only a single image pair and is subject to instabilities during its optimization process. Therefore, it may result in poor visual quality and incorrect semantic association in case of challenging, unaligned input pairs. To overcome these limitations, we further extend our approach to training a *feed-forward* generator on a collection of *domain-specific* images. Our feed-forward framework, which we term *SpliceNet*, is trained directly by minimizing our novel structure and appearance ViT perceptual losses, without relying on adversarial training. SpliceNet is orders of magnitude faster than Splice, enabling real-time applications of semantic appearance transfer, and is more stable at test-time. Furthermore, due to being trained on a dataset, SpliceNet acquires better semantic association, demonstrates superior generation quality and is more robust to challenging unaligned input pairs. However, as SpliceNet is trained on a *domain-specific* dataset, it is limited to working with image pairs from that domain. In contrast, Splice works with *arbitrary, in-the-wild* input pairs, without any domain restriction.

We introduce two key components in the design of SpliceNet – (i) injection of appearance information by direct conditioning on the [CLS] token feature space, and (ii) a method for distilling semantically associated structure-appearance image pairs from a diverse collection of images.

A key component in designing a feed-forward appearance transfer model is the way the network is conditioned on the input appearance image. To leverage the readily available disentangled appearance information in the [CLS] token, we design a CNN architecture that directly benefits from the information encoded in the input [CLS] token, yet controls appearance via modulation. Specifically, our model takes as input a structure image and a target [CLS] token; inspired by StyleGAN-based architectures, the content is encoded into spatial features, while the input [CLS] token is directly mapped to modulation parameters. Explicitly conditioning the model on the [CLS] token significantly simplifies the learning task, resulting in better convergence that leads to faster training and higher visual quality.

We train SpliceNet using natural image pairs from a given domain, using our DINO-ViT perceptual losses. In artistic style transfer the training examples consist of randomly sampled content and style pairs. However, in our case, the semantic association between the input images is imperative. Specifically, our training pairs should fulfill region-to-region semantic correspondence, yet differ in appearance. Such pairs cannot be simply achieved by random pairing. Therefore, we propose an approach, leveraging DINO-ViT features, to automatically distill such training examples out of an image collection. This allows us to train our model on diverse datasets, depicting unaligned natural poses. We thoroughly evaluate the importance of our architectural design and structure-appearance distillation.

2 RELATED WORK

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 [Isola et al. 2017] or unpaired [Kim et al. 2017; Liu et al. 2017; Park et al. 2020a; Yi et al. 2017; Zhu et al. 2017]. Swapping Autoencoder (SA) [Park et al. 2020b] and Kim et al. [Kim et al. 2022] train a domain-specific GAN to disentangle structure and texture in images, and swap these representations between two images in the domain. These methods propose different self-supervised losses integrated in a GAN-based framework for learning disentangled latent codes from scratch. In contrast, our method relies on disentangled descriptors derived from a pre-trained ViT feature space, and does not require any adversarial training. This significantly simplifies the learning task, allowing us to: (i) train a generator given only a single pair of images as input, while not being restricted to any particular domain, (ii) train a feed-forward model on challenging unaligned domains, in which the GAN-based methods struggle.

Recently, image-to-image translation methods trained on a single example were proposed [Benaim et al. 2021; Cohen and Wolf 2019; Lin et al. 2020]. These methods only utilize low-level visual information and lack semantic understanding. Our Splice framework is also trained only on a single image pair, but leverages a pre-trained ViT model to inject powerful semantic information into the generation process. Moreover, single-pair methods are based on slow optimization-processes. Our SpliceNet framework extends Splice to a feed-forward model, allowing real-time applications of semantic appearance transfer on a specific domain.

Neural Style Transfer (NST). In its classical setting, NST transfers an *artistic* style from one image to another [Gatys et al. 2017; Jing et al. 2020]. STROTSS [Kolkin et al. 2019] uses pre-trained VGG features to represent style and their self-similarity to capture structure in an optimization-based style transfer framework. To allow real-time use, a surge of feed-forward models have been proposed, trained using the VGG perceptual losses [Chen and Schmidt 2016; Dumoulin et al. 2017; Huang and Belongie 2017; Johnson et al. 2016; Li and Wand 2016b; Li et al. 2017; Ulyanov et al. 2016]. However, using second-order feature statistics results in *global artistic* style transfer, and is not designed for transfering style between *semantically related* regions. In contrast, our goal is to transfer the appearance between *semantically related* objects and regions in two *natural* images, which we achieve by leveraging novel perceptual losses based on a pre-trained ViT.

Semantic style transfer methods also aim at mapping appearance across semantically related regions between two images [Li and Wand 2016a; Mechrez et al. 2018; Wang et al. 2018; Wilmot et al. 2017]. However, these methods are usually restricted to color transformation [Wang et al. 2018; Xu et al. 2020; Yoo et al. 2019], or depend on additional semantic inputs (e.g., annotations, segmentation, point correspondences, etc.) [Champandard 2016; Gatys et al. 2017; Kim et al. 2020; Kolkin et al. 2019]. Other works tackle the problem for specific controlled domains [Shih et al. 2014, 2013]. In contrast, we aim to semantically transfer fine texture details in a fully automatic manner, without requiring any additional user guidance. Moreover, our Splice framework can handle arbitrary, in-the-wild input pairs, without being domain-restricted. While SpliceNet is domain-specific, it enables real-time semantic appearance transfer due to its feed-forward design.

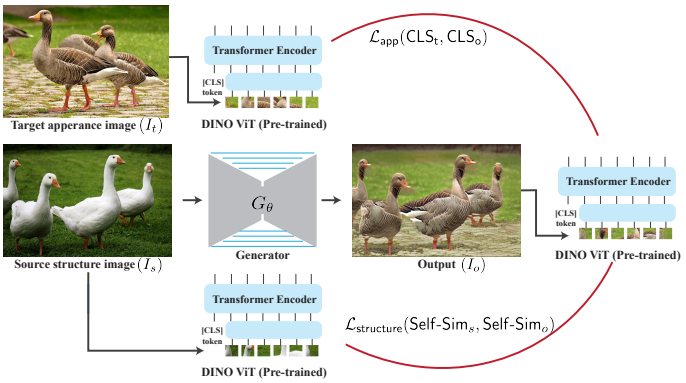


Fig. 2. **Splice pipeline.** Our generator G_θ takes an input structure image I_s and outputs I_o. We establish our training losses using a pre-trained and fixed DINO-ViT model, which serves as an external semantic prior: we represent *structure* via the self-similarity of keys in the deepest attention 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.

Vision Transformers (ViT). ViTs [Dosovitskiy et al. 2021] have been shown to achieve competitive results to state-of-the-art CNN architectures on image classification tasks, while demonstrating impressive robustness to occlusions, perturbations and domain shifts [Naseer et al. 2021]. DINO-ViT [Caron et al. 2021] 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. [Amir et al. 2022] have demonstrated the power of DINO-ViT Features as dense visual descriptors. Their key observation is that deep DINO-ViT features capture rich semantic 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. To this end, we propose Splice – a semantic appearance transfer framework trained on a *single pair* of structure and appearance images. In addition, we extend Splice to a feed-forward model trained on a dataset of images, which we term SpliceNet. While Splice can work with in-the-wild image pairs from arbitrary domains, SpliceNet is trained on a collection of images from a specific

domain, and enables real-time applications due to its feed-forward design.

Our Splice 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 [Caron et al. 2021] – 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.

Additionally, based on our structure and appearance losses, we design SpliceNet – a feed-forward semantic appearance transfer framework, which is illustrated in Fig. 6. The design of SpliceNet consists of two stages: a data-distillation stage, where semantically related pairs are created out of a noisy dataset, and a training stage, where we train a feed-forward generator directly conditioned on ViT feature space.

We next briefly review the ViT architecture in Sec. 3.1, provide qualitative analysis of DINO-ViT's features in Sec. 3.2, describe the Splice framework in Sec. 3.3, and we describe SpliceNet in Sec. 3.4.

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), Multihead Self-Attention (MSA) modules, and MLP blocks:

$$\begin{aligned} \hat{T}^l &= \text{MSA}(\text{LN}(T^{l-1})) + T^{l-1}, \\ T^l &= \text{MLP}(\text{LN}(\hat{T}^l)) + \hat{T}^l, \end{aligned}$$

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:

$$Q^l = T^{l-1} \cdot W^l_q, \;\; K^l = T^{l-1} \cdot W^l_k, \;\; V^l = T^{l-1} \cdot W^l_v, \tag{1}$$

which are then fused using multihead self-attention to form the output of the MSA block (for full details see [Dosovitskiy et al. 2021]).

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 [Dosovitskiy et al. 2021]. In our framework, we leverage DINO-ViT [Caron et al. 2021], 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 [Caron et al. 2021], and in [Amir et al. 2022], 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:

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

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 [Shechtman and Irani 2007], and recently also using deep CNN features for artistic style transfer [Kolkin et al. 2019]. We opt to use the self similarities of *keys*, rather than other facets of ViT, based on [Amir et al. 2022].

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., [Mahendran and Vedaldi 2014; Simonyan et al. 2014]), 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 [Olah et al. 2017]. We observed a similar phenomenon when inverting ViT features (see Supplementary Materials (SM)). Hence, we incorporate "Deep Image Prior" [Ulyanov et al. 2018], 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.

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 [Amir et al. 2022;

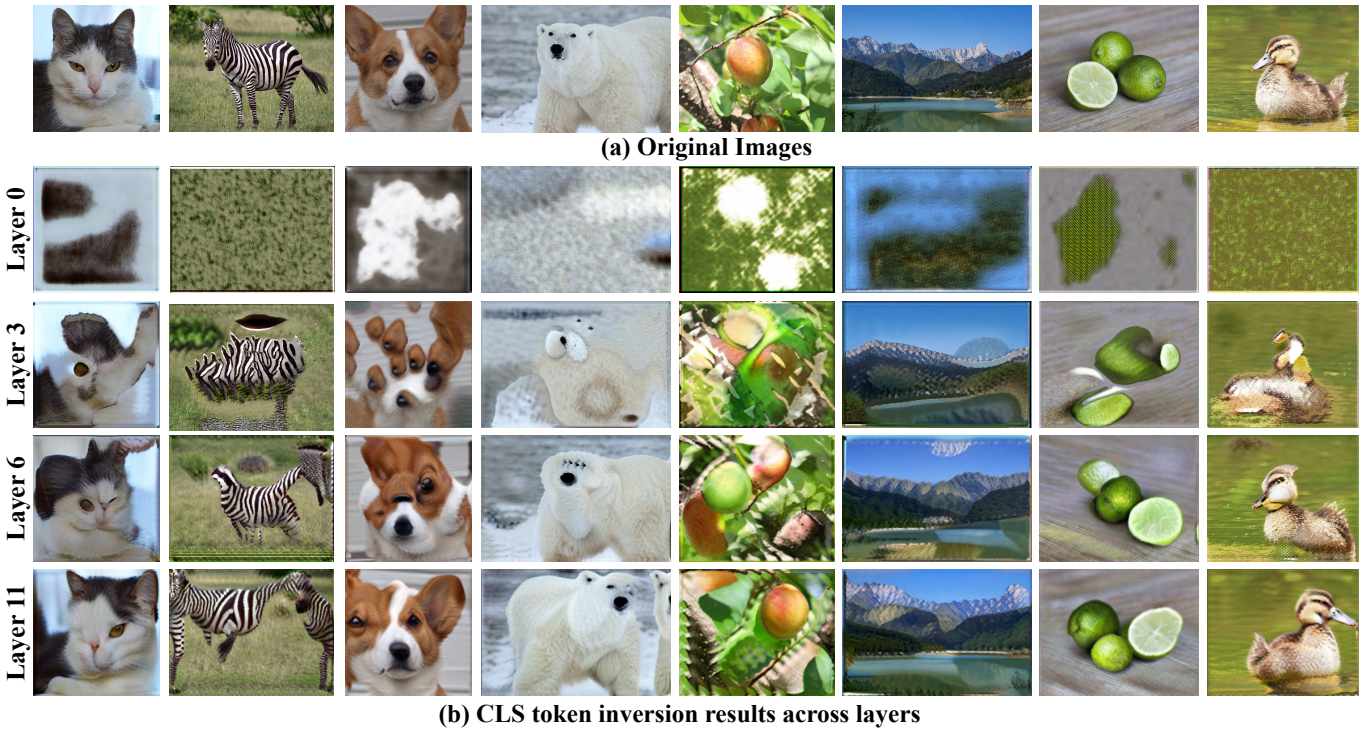


Fig. 3. **Inverting the [CLS] token across layers.** Each input image (a) is fed to DINO-ViT to compute its global [CLS] token at different layers. (b) Inversion results: starting from a noise image, we optimize for an image that would match the original [CLS] token at a specific layer. While earlier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

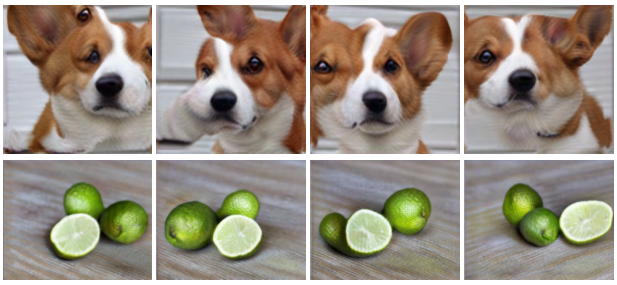


Fig. 4. **[CLS] token inversion over multiple runs.** The variations in structure in multiple inversion runs of the same image demonstrates the spatial flexibility of the [CLS] token.

Caron et al. 2021]. 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 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 representations, we turn to the task of training a generator given a single pair of structure-appearance images. Our framework, which we term Splice, is illustrated in Fig. 2.

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. We set $\alpha = 0.1, \beta = 0.1$ for all experiments of Splice.

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 appearance 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:

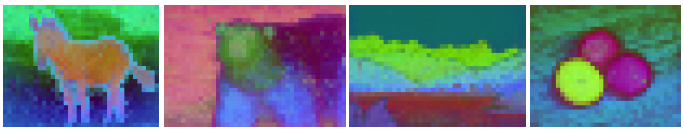
$$\mathcal{L}_{\text{structure}} = \left\| S^L(I_s) - S^L(I_o) \right\|_F, \tag{6}$$

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:

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

(a) Keys inversion results (layer 11)



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

Fig. 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).

Similar loss terms, defined in RGB space, have been used as a regularization in training GAN-based generators for image-to-image translation [Park et al. 2020a; Taigman et al. 2017; Zhu et al. 2017]. 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, $\{I_s^i, I_t^i\}_{i=1}^N$, by applying augmentations such as crops and color jittering (see Appendix D 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 $I_o^i = G_\theta(I_s^i)$, that matches the structure of I_s^i and the appearance of I_t^i.

3.4 SpliceNet: A Feed-forward Model for Semantic Appearance Transfer

While Splice demonstrates exciting results on in-the-wild image pairs as shown in Sec. 4 and Figures 1,8,10, it requires training a generator from scratch for each structure-appearance image pair. This costly optimization process makes the framework infeasible for real-time applications. To this end, we propose SpliceNet – a feed-forward appearance transfer framework. SpliceNet is a feed-forward generator trained on a dataset of images with diverse alignment and appearance, and its objective function is based on the perceptual losses described in Sec. 3.3. While being domain-specific, SpliceNet is orders of magnitude faster than Splice at inference time, allowing real-time applications of semantic appearance transfer.

Given a source structure image I_s and a target appearance image I_t in domain $\mathcal{X}$, we seek a feed-forward model F_θ that outputs a stylized image I_o. A straightforward approach is to directly condition F_θ on the input source-target images themselves, i.e., $I_o = F_\theta(I_s; I_t)$. However, the model would have to implicitly learn to extract appearance information from I_t, while discarding irrelevant spatial information – a challenging task by itself. Instead, our key observation is that such a representation is readily available in DINO-ViT's [CLS] token, which can serve as an input to the model, i.e., $I_o = F_\theta\left(I_s; t^L_{[\text{CLS}]}(I_t)\right)$. Directly conditioning the model on the [CLS] token significantly simplifies the learning task, resulting in better convergence that leads to faster training and higher visual quality. We thoroughly analyze the effectiveness of this design in Sec. 4.4.

Specifically, our framework, illustrated in Fig. 6, consists of a U-Net architecture [Ronneberger et al. 2015], which takes as input the structure image I_s, and a [CLS] token $t^L_{[\text{CLS}]}(I_t)$. The structure image is encoded and then decoded to the output image, while the [CLS] token is used to modulate the decoder's feature. This is done by feeding $t^L_{[\text{CLS}]}(I_t)$ to a 2-layer MLP (M) followed by learnable affine transformations [Karras et al. 2020]. See more details in Appendix A.2.

3.5 Structure-Appearance Pairs Distillation & Training

An important aspect in training our model to transfer appearance across natural images is *data*. While diverse natural image collections are available, randomly sampling structure-appearance image pairs and using them as training examples is insufficient. Such random pairs often cannot be semantically associated (e.g., a zoomed-in face of a dog vs. a full-body, as seen Fig. 7 top row). Thus, training a model with high prevalence of such pairs prevents it from learning meaningful semantic association between the structure and appearance images. We tackle this challenge by *automatically* distilling image pairs (I_s, I_t) that satisfy the following criteria: (i) depict semantic region-to-region correspondence, and (ii) substantially differ in appearance, to encourage the network to utilize the rich information encoded by [CLS] token, and learn to synthesize complex textures.

To meet the above criteria, we need an image descriptor $\mathcal{F}(I)$, invariant to appearance, that can capture the rough semantic layout of the scene. To this end, we leverage the DINO-ViT representation, and use a spatially-coarse version of keys' self-similarity as image descriptor. That is,

$$\mathcal{F}(I) = S_{\text{coarse}}(I) \in \mathbb{R}^{d \times d}, \tag{8}$$

where $S_{\text{coarse}}(I)$ is the self-similarity matrix computed by average pooling the grid of spatial keys, and then plugging the pooled keys, $\bar{K}^L(I)$, to Eq. 2; here $d = \sqrt{n}/w$, where n is the number of spatial features, and w is the pooling window size.

Figure 7 shows top-4 nearest-neighbors retrieved using different descriptors for a given query image. As seen in Fig. 7(c), directly comparing the features results in similar semantic layout, yet all the images depict very similar appearance. Using coarse self-similarity, we obtain a set of images spanning diverse appearances. Furthermore, using *coarse* feature map allows for more variability in the pose of the dogs, which further increases the diversity of our pairs (Fig. 7(d)).

A simple structure-appearance pairing could be achieved by pairing each image $I \in \mathcal{X}$ with its K-nearest-neighbors (KNN) according to the similarity in $\mathcal{F}(I)$. However, such approach does not account for outlier images, which often appear in Internet datasets. To this end, we use a robust similarity metric based on the *Best-Buddies Similarity* (BBS) [Dekel et al. 2015], in which an image pair (I_i, I_j) is considered as inlier if the two images are mutual nearest neighbours.

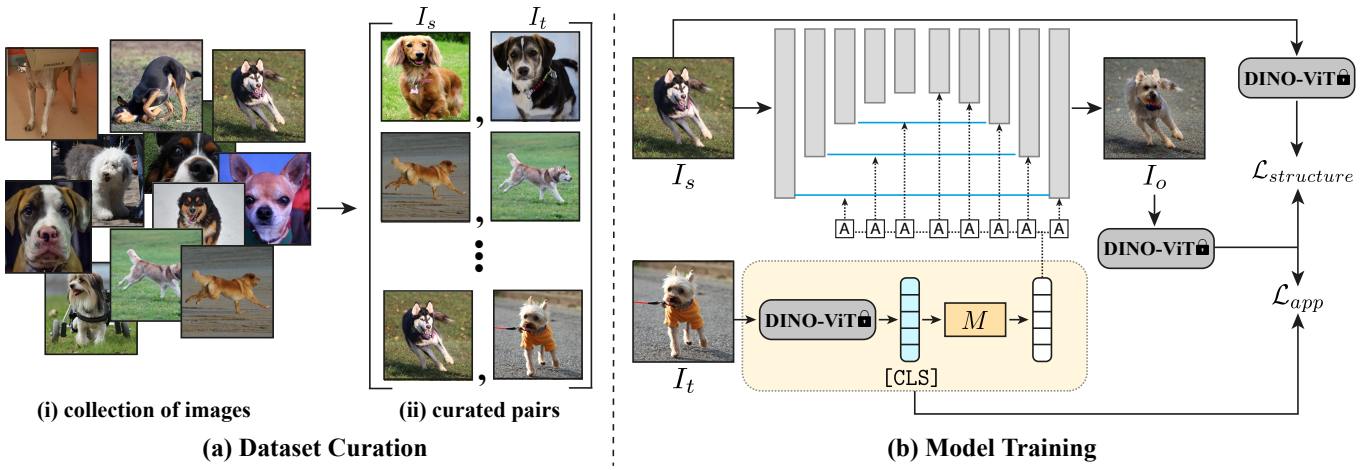


Fig. 6. SpliceNet Pipeline. (a) A diverse image collection is automatically curated for distilling image pairs used for training, each depicts region-to-region semantic correspondences as well as significant variation in appearance. (b) SpliceNet comprises of a UNet architecture, which takes as input: a structure image (I_s), and the [CLS] token extracted from a pre-trained DINO-ViT when fed with the target appearance image (I_t). The structure image is encoded into spatial features, while the [CLS] token is used to adaptively normalize the decoded features. This is done via a mapping network (M) followed by learnable affine transformations [Karras et al. 2020]. Skip connections are used to allow the model to retain fine content details. Our model is trained using DINO-ViT perceptual losses: (1) $\mathcal{L}_{\text{app}}$ that encourages the appearance of I_o and I_t to match, and (2) $\mathcal{L}_{\text{structure}}$, which encourages the structure and perceived semantics of I_o and I_s to match. See Sec. 3.3 for details.

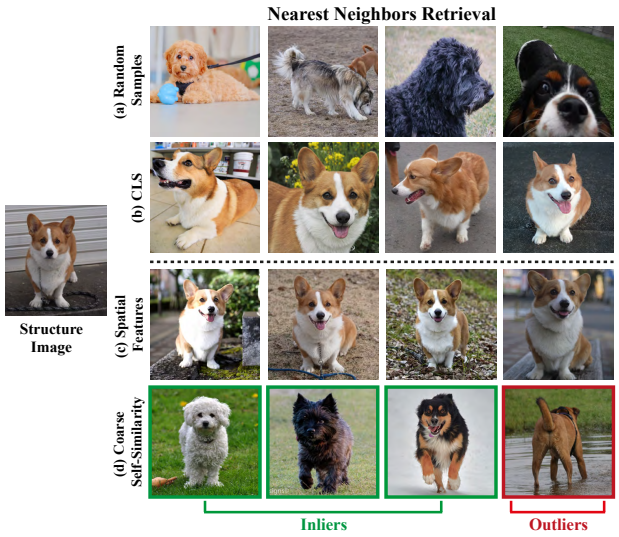


Fig. 7. We retrieve K-nearest-neighbors of a structure image (left) using similarity in (b) [CLS] token. (c) spatial features. (d) Coarse Self-Similarity. Inlier are marked in green, whereas outliers in red. See Sec. 3.5 for details.

Here, we extend this definition to mutual K- nearest-neighbors, and pair each query image I_q with a set of images $\{I_j\}$ that satisfy:

$$I_j \in KNN(I_q, \mathcal{X}) \ \wedge \ I_q \in KNN(I_j, \mathcal{X}). \tag{9}$$

Fig. 7 (bottom row) shows an example of automatically detected inliers/outliers. More examples are included in the SM.

Training. At each training step, we sample an image pair (I_s, I_t) from our distilled paired-dataset and apply various augmentations, such as cropping and flipping (see Appendix D for full details). We then feed I_t to DINO-ViT and extract the [CLS] token $t^L_{[\text{CLS}]}(I_t)$, which is fed to our model $I_o = F_\theta\left(I_s; t^L_{[\text{CLS}]}(I_t)\right)$; our training objective is given in Eq. 4. $\mathcal{L}_{\text{structure}}$ and $\mathcal{L}_{\text{app}}$ have the same definitions as in Splice. $\mathcal{L}_{\text{id}}$ is used as a reconstruction regularization term in case $I_s = I_t = I$, i.e., $\mathcal{L}_{\text{identity}} = \mathcal{D}(I, I_o)$, where $\mathcal{D}(\cdot,\cdot)$ is a chosen distance function. Empirically we found LPIPS [Zhang et al. 2018] to be more stable compared to the keys loss described in Sec. 3.3.

4 RESULTS

4.1 Splice

Datasets. We tested Splice on a variety of image pairs gathered from Animal Faces HQ (AFHQ) dataset [Choi et al. 2020], 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. 8, 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. 8 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

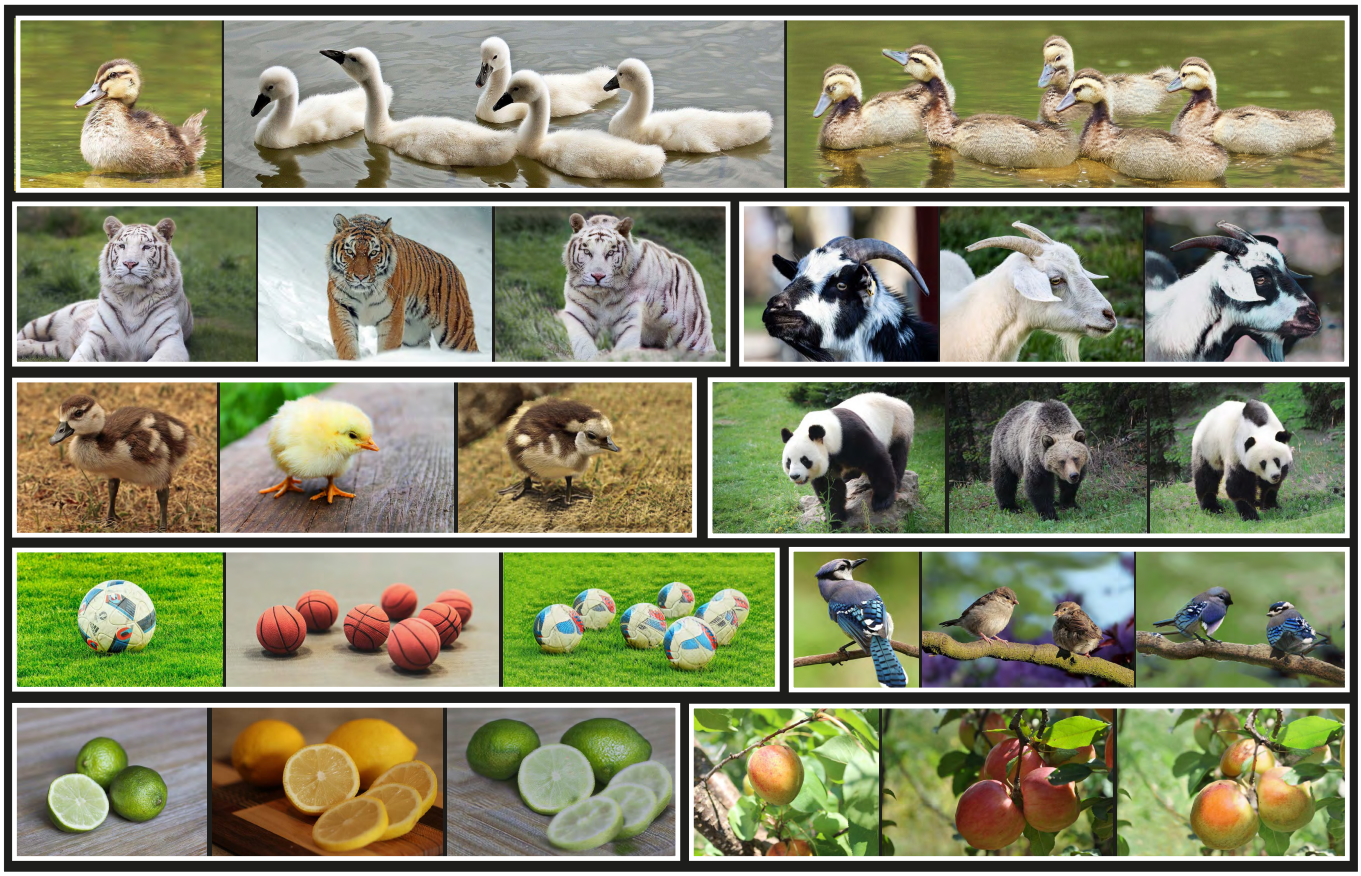


Fig. 8. **Sample results of Splice on in-the-wild image pairs.** For each example, shown left-to-right: the target appearance image, the source structure image and our result. The full set of results is included in the SM. Notice the variability in number of objects, pose, and the significant appearance changes between the images in each pair.

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.

4.2 SpliceNet

Datasets. We trained SpliceNet on the training set of each of the following (separately): Animal Faces HQ (*AFHQ*) [Choi et al. 2020], *Oxford-102* [Nilsback and Zisserman 2008], and two Internet datasets – *SD-Dogs*, and *SD-Horses* – each containing a wide range of poses, and appearance variations [Mokady et al. 2022].

Fig. 8 shows sample results of our method on diverse structure-appearance pairs. SpliceNet consistently transfers appearance between semantically-corresponding regions, while synthesizing high-quality textures. Notably, although the appearance may dramatically change, the structure and perceived semantics of the content image are well preserved across all datasets. Many more results are included in SM.

4.3 Comparisons

For **Splice**, there are no existing methods that are tailored for solving its task: semantic appearance transfer between two natural images (not restricted to a specific domain), without explicit user-guided inputs. We thus compare Splice 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)* [Park et al. 2020b] – 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* [Kolkin et al. 2019], the style transfer method that also uses self-similarity of a pre-trained CNN features as the content descriptor, (ii) *WCT*^{2} [Yoo et al. 2019], 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.

We compare **SpliceNet** to prior works in which the problem setting is most similar: Splice; WCT^2 [Yoo et al. 2019]; and prominent

Fig. 9. Sample results of SpliceNet trained for (a) *AFHQ*, (b) *Oxford-102* (c) *SD-Dogs* and (d) *SD-Horses*. Across rows: different *structure* images, across columns: different *appearances*. The full set of results is included in the SM.

GAN-based methods: Swapping Autoencoder (SA) [Park et al. 2020a] and [Kim et al. 2022]. We used official implementation and pre-trained models of these methods when available. We trained SA and Kim et al. for the datasets for which no model was provided by the authors.

Table 1(left) reports the number of trainable parameters in each of these models, and their average inference run-time.

4.3.1 Qualitative comparison. Figure 10 shows sample results for all methods (additional results are included in the SM) compared to **Splice**. In all examples, Splice correctly relates semantically matching regions between the input images, and successfully transfers 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.

Splice 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, Splice 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 qualitative comparisons to SinCUT [Park et al. 2020a], a GAN-based image translation method, and to Deep-Image-Analogy [Liao et al. 2017]. As demonstrated in Fig. 11, SinCUT and Deep-Image-Analogy perform well for the landscape example,

but fail to transfer the appearance of the swan in the second example, where a higher-level visual understanding is required. Splice successfully transfers the appearance across semantically realted regions, and generates high quality results w/o adversarial loss.

Figure 10 shows comparison between **SpliceNet** and baselines. As seen by WCT^2 results, transferring colors is insufficient for capturing the target appearance. The GAN-based methods (SA and Kim et. al.), which learn structure/appearance representations from scratch, suffer from either bleeding artifacts or low fidelity to the source structure for aligned datasets (*AFHQ*, and *Oxford-102*). For more diverse and unaligned datasets (*SD-Dogs* and *SD-Horses*), these methods struggle to synthesize complex textures or to preserve the original content. Although Splice can successfully establish semantic association, it is subject to instabilities in its test-time optimization process that sometimes leads to failure cases (e.g., topmost flower,

[1]We verified these results with the authors [Park et al. 2020b]

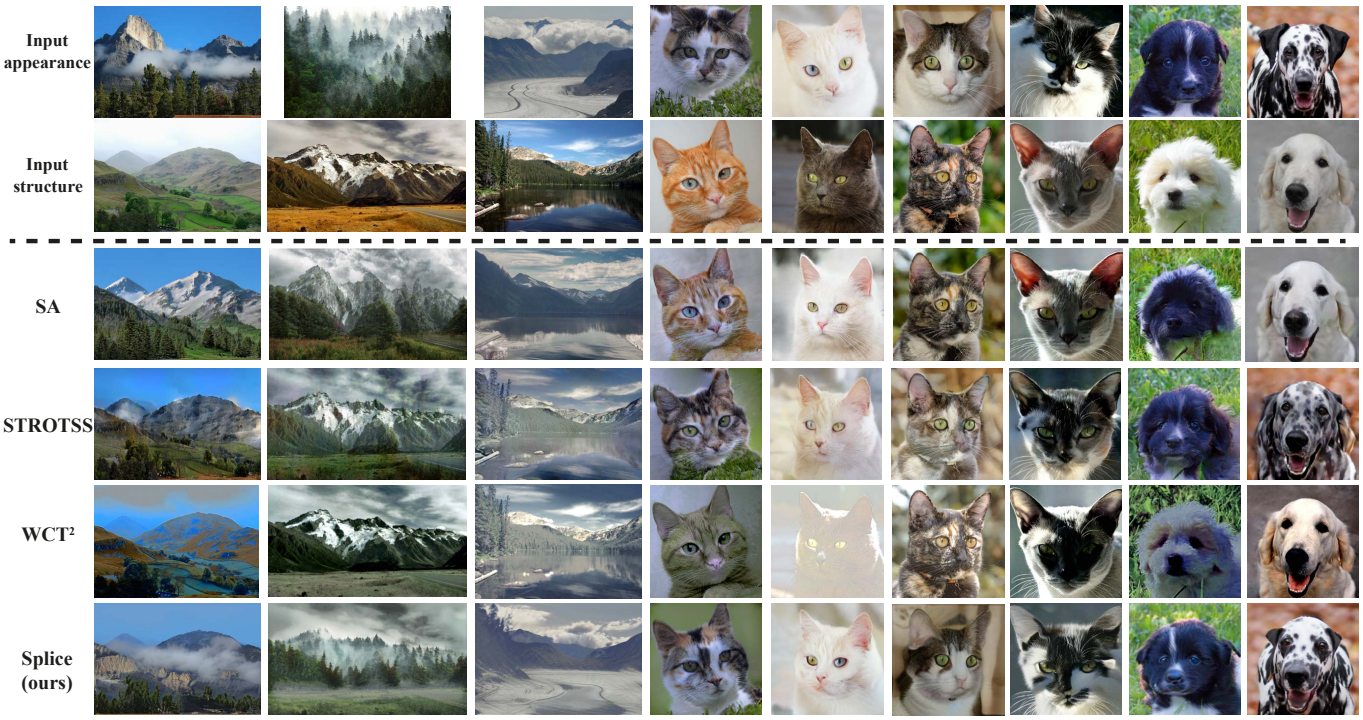


Fig. 10. **Comparisons of Splice with style transfer and swapping autoencoders.** First two rows: input appearance and structure images taken from the AFHQ and Flickr Mountains. The following rows, from top to bottom, show the results of: swapping autoencoders (SA) [Park et al. 2020b], STROTSS [Kolkin et al. 2019], and WCT^2 [Yoo et al. 2019]. See SM for additional comparisons.

second dog), while SpliceNet achieves improved visual quality and stability.

4.3.2 Quantitative comparison. To quantify how well our generated images match the target appearance and preserve the original structure, we use the following metrics: (i) human perceptual evaluation, (ii) semantic layout preservation and (iii) reconstruction.

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 [Kolkin et al. 2019; Park et al. 2020b]. 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?"*.

For evaluating **Splice**, 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 reports the percentage of votes in our favor. As seen, our method outperforms all baselines across all image collections, especially 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.

For evaluating **SpliceNet**, we perform the survey using 80 image-pairs from all datasets. We collected 6500 user judgments w.r.t. existing baselines. Table 1 reports the percentage of votes in our favor. As seen, our method outperforms all baselines across all datasets, especially in the Internet datasets (*SD-Dogs*, *SD-Horses*), which highlights our performance in challenging settings.

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 [He et al. 2017]) to compute object masks for the input structure images and our results.

Table 5 reports IoU for Splice and the baselines. Splice 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.

We perform the same evaluation protocol on SpliceNet and its competitors. We consider the objects relevant to our datasets for which clean and robust segmentation masks could be obtained (cats, dogs, horses). Table 3 reports the average intersection over union (IoU) between the masks computed for the content images and the corresponding stylized results. SpliceNet achieves higher (better) IoU than Kim et al. and Splice and is the closet competitor to WCT^2, which achieves the highest IoU as it only modifies colors.

Reconstruction. When the input appearance and structure images are identical, we expect any appearance transfer method to *reconstruct* the input image. Table 1 reports mean squared error (MSE), and LPIPS [Zhang et al. 2018] computed between the input and the reconstructed image. Naturally, WCT^2 excels in most datasets since

	Params [M]	Runtime [sec]	*AFHQ* MSE	*AFHQ* LPIPS	*AFHQ* Human eval	*Oxford-102* MSE	*Oxford-102* LPIPS	*Oxford-102* Human eval	*SD-Horses* MSE	*SD-Horses* LPIPS	*SD-Horses* Human eval	*SD-Dogs* MSE	*SD-Dogs* LPIPS	*SD-Dogs* Human eval
Kim et al.	56.51	.1251	.0506	.2053	86.79 ± 0.23	.0817	.3658	80 ± 0.17	.0251	.1350	73.84 ± 0.16	.0276	.1707	91.1 ± 0.2
SA	109.03	.0954	.0241	.1452	98.93 ± 0.03	.0355	.1745	90.29 ± 0.21	.0454	.2464	96.53 ± 0.08	.0480	.1442	98.75 ± 0.04
WCT^2	10.11	.3635	**.0001**	**.0019**	88.23 ± 0.21	**.0074**	**.0263**	66.07 ± 0.39	**.0013**	.0270	98.22 ± 0.06	**.0008**	.0147	100 ± 0
Splice	1.04	762	.0167	.0174	93.75 ± 0.11	.0767	**.0263**	69.8 ± 0.38	.0521	.0392	72.23 ± 0.37	.4699	.5365	78.27 ± 0.34
SpliceNet	54.43	**.0892**	.0035	.0078	-	.0135	.0379	-	.0037	**.0144**	-	.0039	**.0107**	-

Table 1. For each baseline, we report: model size and runtime (measures for 512px images on RTX6000 GPU). For each dataset, we report reconstruction error measured by LPIPS↓, MSE↓, and human perceptual evaluation results, measured by the percentage of judgments in our favor (mean, std).

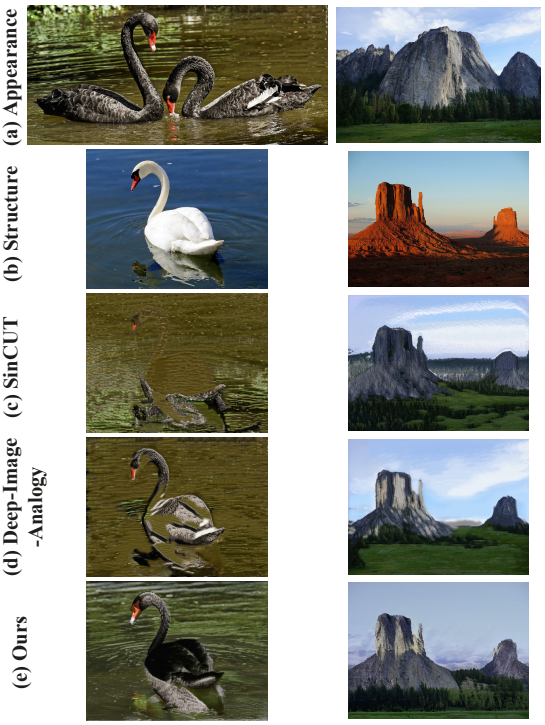


Fig. 11. **Additional Qualitative Comparisons.** SinCUT [Park et al. 2020a] (c) and Deep-Image-Analogy [Liao et al. 2017] (d) results, when trained on each input pair (a-b). These methods work well when the translation is mostly based on low-level information (top), but fail when higher-level reasoning is required (bottom), struggling to make meaningful semantic associations (e.g., the lake is mapped to the swan). (e) Our method successfully transfers the appearance across semantic regions, and generates high-quality results w/o adversarial training.

	Kim et al.	SA	WCT^2	Splice (Ours)	SpliceNet (Ours)
AFHQ	0.826	0.773	0.516	0.677	**0.225**
Oxford-102	0.64	0.933	0.819	0.518	**0.507**
SD-Dogs	0.849	0.612	0.568	0.462	**0.435**
SD-Horses	0.809	**0.568**	0.775	0.869	0.577

Table 2. We report the average SI-FID computed over 100 random pairs from each dataset. Lower is better.

it does not synthesize new textures or modify shapes. SpliceNet surpasses all other methods, including state-of-the-art GAN-based methods, by an order of magnitude.

	Kim et al.	SA	WCT^2	Splice (Ours)	SpliceNet (Ours)
AFHQ	0.953	**0.967**	0.961	0.883	**0.967**
SD-Dogs	0.938	0.801	**0.954**	0.863	**0.954**
SD-Horses	0.928	0.861	**0.948**	0.867	0.940

Table 3. We extract semantic segmentation maps of objects of interest in the content and stylized images. Mean IoU over 100 images are reported for each dataset.

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

Table 4. **Splice AMT perceptual evaluation**. We report results on AMT surveys evaluating the task of appearance transfer across semantically related scenes/objects (see Sec. 4.3.2). 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 [Park et al. 2020b], and style transfer methods, STROTSS [Kolkin et al. 2019], and WCT^2 [Yoo et al. 2019].

	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 5. **Mean IoU of output images with respect to the input structure images**. We extract semantic segmentation maps using Mask-RCNN [He et al. 2017] for the Wild-Pairs collection, and [Zhou et al. 2018] for the mountains collection.

4.4 Ablation

We ablate the loss terms and design choices in our proposed frameworks.

Loss terms. We ablate the different loss terms in our objective function by qualitatively comparing the results when trained with the full objective (Eq. 4), and with a specific loss removed. The results are shown in Fig. 13. As can be seen, without the **appearance loss** (w/o $\mathcal{L}_{\text{app}}$), Splice 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

Fig. 12. **SpliceNet comparisons with baselines.** First two columns depict the input appearance-structure. The following columns show the results of: [Kim et al. 2022], Swapping Autoencoder (SA) [Park et al. 2020b], WCT^2 [Yoo et al. 2019], Splice, and SpliceNet. Top to bottom: *SD-Dogs*, *SD-Horses*, *AFHQ* and *Oxford-102*. See SM for additional comparisons.

in terms of appearance and structure, e.g., the fine texture details of the avocado are refined.

Dataset augmentation. We ablate the usage of dataset augmentation in Splice. In this case, the network solves a test-time optimization problem between two images rather than learning to map between many internal examples. As can be seen in Fig. 13, without data augmentation, the semantic association is largely preserved, however, the realism and visual quality of Splice are significantly decreased.

For SpliceNet, we ablate our key design choices by considering these baselines:

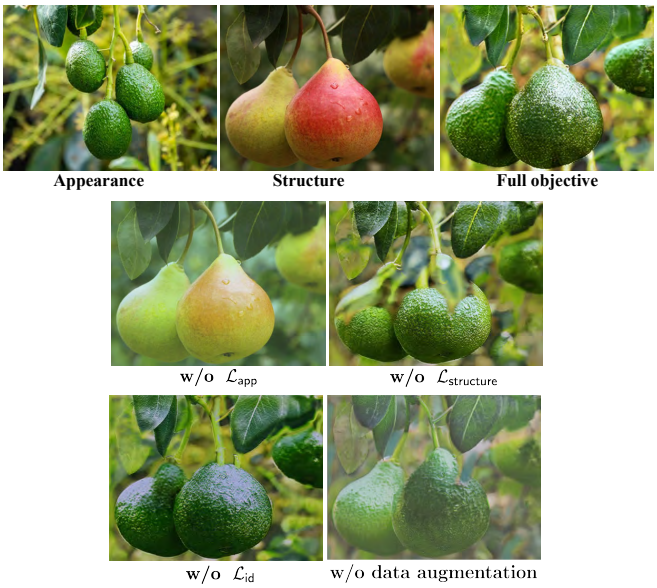


Fig. 13. **Loss and data augmentation ablations.** Splice ablation results of specific loss terms and the data augmentation. When one of our loss terms is removed, the model fails to map the target appearance, preserve the input structure, or maintain fine details. Without dataset augmentation, while the semantic association is largely maintained, the visual quality is significantly decreased. See Sec. 4.4 for more details.

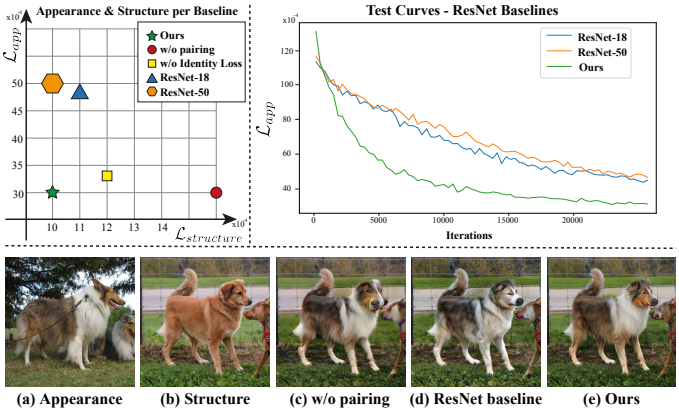


Fig. 14. **SpliceNet ablations.** Top left: we plot style/content losses for several baselines, including our model trained w/o data distillation, and w/o conditioning on the style token (ResNet baseline). Top right: test loss curves computed during training for our framework vs. the ResNet baselines. Bottom: qualitative comparison of a representative pair. See Sec. 4.4 for details.

Input [CLS] *token vs. input appearance image.* To demonstrate the effectiveness of directly using DINO-ViT's [CLS] token as input, we consider a baseline architecture that takes as input I_t, the appearance *image*. Specifically, we use an off-the-shelf ResNet backbone [He et al. 2016] to map I_t into a global appearance vector which is mapped to modulation parameters via learnable affine transformations.

No structure/appearance pair distillation. We show the importance of our data curation (Sec. 3.5) by training a model on random image pairs.

Figure 14(bottom) shows a qualitative comparison to the above baselines on a sample pair (see SM for more examples). As seen in Fig. 14(d), without conditioning the model on the [CLS] token, the results suffer from visual artifacts and the model could not deviate much from the original texture. As seen in Fig. 14(c), a model trained without pairs distillation (w/o pairing) can still synthesize textures matching the target appearance, yet fail to preserve the semantic content.

We quantify these results as follows: We randomly sample input pairs from *SD-Dogs* test set, and compute the average structure and appearance losses (Eq. 4). Figure 14(top left) reports the results for all baselines, and validate the expected trends.

Figure 14 (top right) shows the learning curves on *SD-Dogs* test set for the different CNN backbones. As can be seen, directly conditioning on the [CLS] token results in faster convergence, and lower appearance loss. Fig. 14(d).

4.5 Manipulation in [CLS] Token Space

Directly conditioning SpliceNet on the [CLS] token space not only benefits the appearance transfer and training convergence, but also enables applications of appearance transfer by performing manipulations in the [CLS] token space. Specifically, we perform interpolation between the structure and appearance [CLS] tokens to control the stylization extent, and detect appearance modes by performing K-means on the [CLS] tokens of the dataset.

Appearance Interpolation. We can control the extent of stylization by feeding to our model interpolating the style tokens of the style and content images, i.e., $t_i = \alpha_i t^L_{[\text{CLS}]}(I_t) + (1-\alpha_i) t^L_{[\text{CLS}]}(I_s)$. Sample examples are shown in Fig. 16 and more included in SM.

Detecting and Visualizing Appearance Modes. We automatically discover representative appearances, i.e., *appearance modes* in the data. To do so, we extract the [CLS] token for all images in the training set, and apply K-means, where the centroids are used as our *appearance modes*. We visualize the modes by using each as the input [CLS] token to SpliceNet, along with a structure image . Figure 15 shows nine such modes automatically discovered for AFHQ training set, transferred to test set structure images. More examples of appearance modes are in SM.

5 LIMITATIONS

The performance of our frameworks depends on the internal representation learned by DINO-ViT, and is therefore limited in several aspects.

First, our frameworks are limited by the features' expressiveness. For example, our method can fail to make the correct semantic association in case the DINO-ViT representation fails to capture it. Figure 17 shows a few such cases for Splice: (a) objects are semantically related but one image is highly non-realistic (and thus out of distribution for DINO-ViT). For some regions, Splice successfully transfers 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

Fig. 15. **Appearance modes** are discovered by clustering the [CLS] token across all AFHQ training set. (b) We transfer each of the discovered appearance modes to test structure images (a).

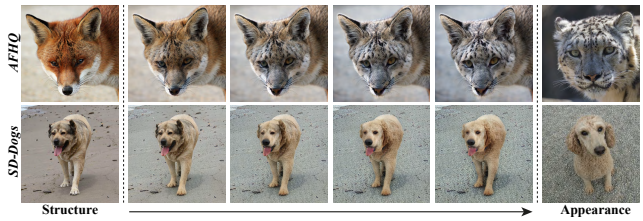


Fig. 16. **Appearance interpolation** Controlling stylization extent via interpolation in [CLS] token space.

cat are nicely mapped, yet Splice fails to find a semantic correspondence for the rings, and we get a wrong mapping of the ear from image A. In (b), Splice does not manage to semantically relate a bird to an airplane. We also found that the [CLS] token cannot faithfully capture distinct appearances of *multiple foreground* objects, but rather captures a joint blended appearance. This can be seen in Fig. 18(top), where SpliceNet transfers the "averaged" appearance to the structure image.

Second, if the structure and appearance test pair contains extreme pose variation, our method may fail to establish correct semantic association, as seen in Fig. 18(bottom).

Third, Splice is restricted to observing only a single image pair and is subject to optimization instabilities, which can lead to incorrect semantic association or poor visual quality, as discussed in Sec. 4.3. SpliceNet overcomes these limitations due to being trained on a dataset, which makes it more robust to challenging inputs and enhances the visual quality.

Finally, DINO-ViT has been trained on ImageNet and thus our models can be trained on domains that are well-represented in DINO-ViT's training data. This can be tackled by re-training or fine-tuning DINO-ViT on other domains.

6 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 our

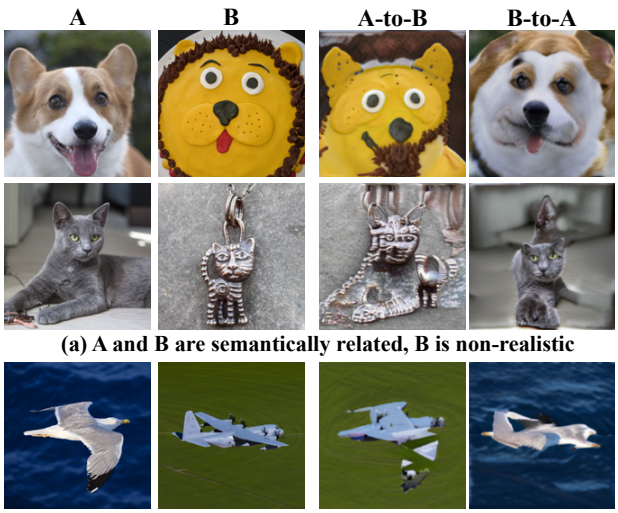


Fig. 17. **Splice limitations.** (a) Objects in the input images (A-B) are semantically related, yet B is non-realistic. (b) Objects are from unrelated object categories. See Sec. 5 for discussion.

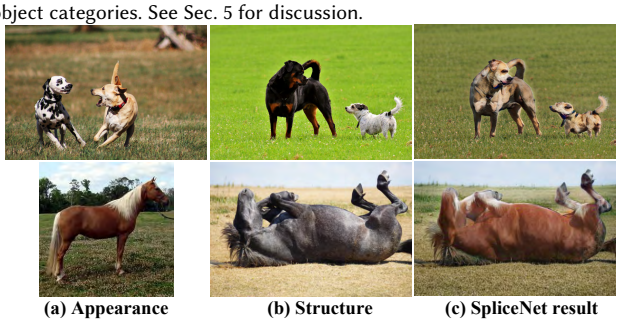


Fig. 18. **SpliceNet limitations.** Top: the target style contains multiple foreground objects, where our result depict a single blended style. Bottom: under extreme pose variations, our method may fail to establish accurate semantic association.

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 propose two frameworks of semantic appearance transfer based on our perceptual losses: (i) Splice, which is a generator trained on a single and arbitrary structure-appearance input pair, and (ii) SpliceNet, a feed-forward generator trained on a domain-specific dataset. Direct conditioning on ViT features boosts the performance of SpliceNet in terms of visual quality and convergence rate. We further showed how to distill suitable training data for SpliceNet from noisy diverse image collections.

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. Through extensive evaluation, we showed that our frameworks, trained with simple perceptual losses, excel state-of-the-art GAN-based methods.

Our evaluations demonstrate that SpliceNet surpasses Splice in terms of visual quality, and is orders of magnitude faster, enabling real-time semantic appearance transfer. Moreover, Splice is limited to observing only a single test-time pair and is subject to instabilities during its optimization process, which may lead to incorrect semantic association and poor visual quality. On the other hand, since SpliceNet is trained on a dataset of semantically related image pairs, it results in a better semantic association and generalization, and is more robust to challenging input pairs. However, SpliceNet is trained on a domain-specific dataset, hence is limited to input images from that domain. In contrast, Splice works on arbitrary, in-the-wild input pairs, without being restricted to a particular domain.

We believe that 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 for her 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

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. 2022. Deep ViT Features as Dense Visual Descriptors. *ECCVW What is Motion For?* (2022).

Frank Beech. 2005. Splicing Ropes Illustrated. *CCCBR* (2005).

Saguy Benaim, Ron Mokady, Amit Bermano, and Lior Wolf. 2021. Structural Analogy from a Single Image Pair. *Comput. Graph. Forum* (2021).

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Alex J. Champandard. 2016. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. *arXiv* (2016).

Tian Qi Chen and Mark Schmidt. 2016. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337* (2016).

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T Freeman. 2015. Best-buddies similarity for robust template matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2021–2029.

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. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR* (2021).

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2017. A Learned Representation For Artistic Style. In *International Conference on Learning Representations*.

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling Perceptual Factors in Neural Style Transfer.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*. 1501–1510.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2020. Neural Style Transfer: A Review. *IEEE Trans. Vis. Comput. Graph.* (2020).

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*. Springer, 694–711.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. CVPR*.

Kunhee Kim, Sanghun Park, Eunyeong Jeon, Taehun Kim, and Daijin Kim. 2022. A Style-aware Discriminator for Controllable Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Sunnie SY Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. 2020. Deformable style transfer.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

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

Chuan Li and Michael Wand. 2016a. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis.

Chuan Li and Michael Wand. 2016b. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*. Springer, 702–716.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Diversified texture synthesis with feed-forward networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3920–3928.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual Attribute Transfer Through Deep Image Analogy. *ACM Trans. Graph.* 36, 4, Article 120 (July 2017), 15 pages. https://doi.org/10.1145/3072959.3073683

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

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised Image-to-Image Translation Networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc.

Aravindh Mahendran and Andrea Vedaldi. 2014. Understanding deep image representations by inverting them. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), 5188–5196. https://api.semanticscholar.org/CorpusID:206593185

Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. 2018. The Contextual Loss for Image Transformation with Non-aligned Data.

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. 2022. Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization. In *CVPR*.

Ron Mokady, Omer Tov, Michal Yarom, Oran Lang, Inbar Mosseri, Tali Dekel, Daniel Cohen-Or, and Michal Irani. 2022. Self-Distilled StyleGAN: Towards Generation from Internet Photos. In *ACM SIGGRAPH 2022 Conference Proceedings (SIGGRAPH '22)*. Association for Computing Machinery, Article 50, 9 pages. https://doi.org/10.1145/3528233.3530708

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Khan, and Ming-Hsuan Yang. 2021. Intriguing Properties of Vision Transformers. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=o2mbl-Hmfgd

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature Visualization. *Distill* (2017).

Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. 2020a. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*. Springer.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. 2020b. Swapping Autoencoder for Deep Image Manipulation. In *Advances in Neural Information Processing Systems*.

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. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

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

Eli Shechtman and Michal Irani. 2007. Matching local self-similarities across images and videos. In *CVPR*.

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

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

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. 2021. Localizing Objects with Self-Supervised Transformers and no Labels. *Proceedings of the British Machine Vision Conference (BMVC)*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2017. 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. https://openreview.net/forum?id=Sk2Im59ex

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. 2016. Texture networks: Feed-forward synthesis of textures and stylized images.. In *ICML*, Vol. 1. 4.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2018. Deep Image Prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li Wang, Nan Xiang, Xiaosong Yang, and Jianjun Zhang. 2018. Fast Photographic Style Transfer Based on Convolutional Neural Networks. In *Proceedings of Computer Graphics International 2018 (CGI 2018)*. Association for Computing Machinery, New York, NY, USA.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. 2022. Self-supervised Transformers for Unsupervised Object Discovery using Normalized Cut. In *Conference on Computer Vision and Pattern Recognition*. New Orleans, LA, USA.

Pierre Wilmot, Eric Risser, and Connelly Barnes. 2017. Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses. *ArXiv* (2017).

Zhongyou Xu, Tingting Wang, Faming Fang, Yun Sheng, and Guixu Zhang. 2020. Stylization-Based Architecture for Fast Deep Exemplar Colorization.

Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. In *2017 IEEE International Conference on Computer Vision (ICCV)*.

Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. 2019. Photorealistic Style Transfer via Wavelet Transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

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

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*.

A ARCHITECTURE

A.1 Splice Generator Architecture

We base our generator G_θ network on a `U-Net` architecture [Ronneberger et al. 2015], 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 SpliceNet Generator Architecture

We design our feed-forward model F_θ based on a `U-Net` architecture [Ronneberger et al. 2015]. The input image is first passed through a 1×1 convolutional layer with 32 output channels. The output is then passed through a 5-layer encoder with channel dimensions of $[64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 1024]$, followed by a symmetrical decoder. Each layer of the encoder is a downsampling residual block that is comprised of two consecutive 3×3 convolutions and a 1×1 convolution for establishing the residual connection. The decoder consists of upsampling residual blocks with a similar composition of convolutions and residual connection as in the encoder. In the decoder, the weights of the 3×3 convolutions are modulated with the input [CLS] token. In each layer of the encoder, in order to establish the skip connections to the decoder, the output features are passed through a resolution-preserving residual block, which is concatenated to the input of the decoder layer. The residual blocks in the skip connections have a similar composition of convolutions and modulations as the decoder residual blocks. Finally, the output of the last decoder layer is passed through a modulated 1×1 convolutional layer followed by a `Sigmoid` activation that produces the final RGB output. `LeakyReLU` is used as an activation function in all the convolutional layers of the model.

Our mapping network M is a 2-layer MLP that takes as input the [CLS] token $t_{[\mathrm{CLS}]} \in \mathbb{R}^{768}$ extracted from DINO-ViT, and passes it through one hidden layer and an output layer, both with output dimensions of 768 and with `GELU` activations. Following [Karras et al. 2020], for each modulated convolution in the feed-forward model, an affine transformation is learned that maps the output of the mapping network M to a vector used for modulating the weights.

B VIT FEATURE EXTRACTOR ARCHITECTURE

As described in Sec. 3, we leverage a pre-trained ViT model (DINO-ViT [Caron et al. 2021]) 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.

C TRAINING DETAILS

We implement our framework in PyTorch [Paszke et al. 2019]. We optimize our full objective (Eq. 4, Sec. 3.3), with relative weights: $\alpha = 0.1$, $\beta = 0.1$ for Splice, and $\alpha = 2$, $\beta = 0.1$ for SpliceNet. We use the Adam optimizer [Kingma and Ba 2015] with a constant learning rate of $\lambda = 2 \cdot 10^{-3}$ and with hyper-parameters $\beta_1 = 0$, $\beta_2 = 0.99$. Each batch contains $\{\tilde{I}_s, \tilde{I}_t\}$, the augmented views of the source structure image and the target appearance image respectively. For Splice, every 75 iterations, we add $\{I_s, I_t\}$ to the batch (i.e., do not apply augmentations). All the images (both input and generated) are resized down to 224[pix] (maintaining aspect ratio) using bicubic interpolation, before extracting DINO-ViT features for estimating the losses. The test-time training of Splice 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.

D DATA AUGMENTATIONS

At each training step, given an input pair $\{I_s, I_t\}$, we apply on them the following random augmentations: Augmentations to the source structure image I_s:

- cropping: we uniformly sample a NxN crop; N is between 95% - 100% of the height of I_s (for SpliceNet, we fix N=95%)
- horizontal-flipping, applied in probability p=0.5.
- color jittering: we jitter the brightness, contrast, saturation and hue of the image in probability p, where p=0.5 for Splice and p=0.2 for SpliceNet,
- Gaussian blurring: we apply a Gaussian blurring 3x3 filter (σ is uniformly sampled between 0.1-2.0) in probability p, where p=0.5 for Splice and p=0.1 for SpliceNet,

Augmentations to the target appearance image I_t:

- cropping: we uniformly sample a NxN; N is between 95% - 100% of the height of I_t (for SpliceNet, we fix N=95%).
- horizontal-flipping, applied in probability p=0.5.