Emergent Correspondence from Image Diffusion

Luming Tang* **Menglin Jia*** **Qianqian Wang***
Cheng Perng Phoo **Bharath Hariharan**
Cornell University

Abstract

Finding correspondences between images is a fundamental problem in computer vision. In this paper, we show that correspondence emerges in image diffusion models *without any explicit supervision*. We propose a simple strategy to extract this implicit knowledge out of diffusion networks as image features, namely DIffusion FeaTures (DIFT), and use them to establish correspondences between real images. Without any additional fine-tuning or supervision on the task-specific data or annotations, DIFT is able to outperform both weakly-supervised methods and competitive off-the-shelf features in identifying semantic, geometric, and temporal correspondences. Particularly for semantic correspondence, DIFT from Stable Diffusion is able to outperform DINO and OpenCLIP by 19 and 14 accuracy points respectively on the challenging SPair-71k benchmark. It even outperforms the state-of-the-art supervised methods on 9 out of 18 categories while remaining on par for the overall performance. Project page: https://diffusionfeatures.github.io.

1 Introduction

Drawing correspondences between images is a critical primitive in 3D reconstruction [73], object tracking [22, 90], video segmentation [92], image and video editing [102, 58, 98]. This problem of drawing correspondence is easy for humans: we can match object parts not only across different viewpoints, articulations and lighting changes, but even across drastically different categories (e.g., between cats and horses) or different modalities (e.g., between photos and cartoons). Yet, we rarely if ever get explicit correspondence labels for training. The question is, can computer vision systems similarly learn accurate correspondences without any labeled data at all?

There is indeed some evidence that contrastive self-supervised learning techniques produce good correspondences as a side product of learning on unlabeled data [10, 28]. However, in this paper, we look to a new class of self-supervised models that has been attracting attention: diffusion-based generative models [32, 79]. While diffusion models are primarily models for image synthesis, a key observation is that these models produce good results for image-to-image translation [53, 85] and image editing [8, 80]. For instance, they can convert a dog to a cat without changing its pose or context [61]. It would appear that to perform such editing, the model must implicitly reason about correspondence between the two categories (e.g., the model needs to know where the dog's eye is in order to replace it with the cat's eye). We therefore ask, do image diffusion models learn correspondences?

We answer the question in the affirmative by construction: we provide a simple way of extracting correspondences on real images using pre-trained diffusion models. These diffusion models [41] have at the core a U-Net [71, 17, 70] that takes noisy images as input and produces clean images as output. As such they already extract features from the input image that can be used for correspondence. Unfortunately, the U-Net is trained to *de-noise*, and so has been trained on *noisy* images. Our strategy

*Equal contribution.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

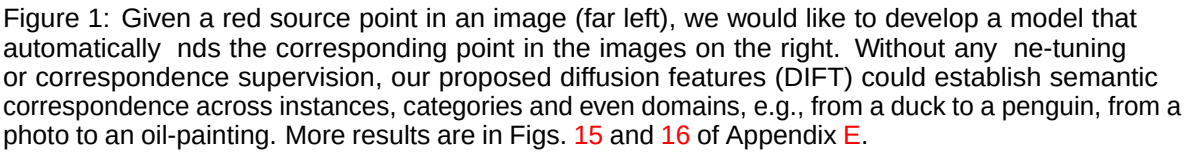

Figure 1: Given a red source point in an image (far left), we would like to develop a model that automatically finds the corresponding point in the images on the right. Without any fine-tuning or correspondence supervision, our proposed diffusion features (DIFT) could establish semantic correspondence across instances, categories and even domains, e.g., from a duck to a penguin, from a photo to an oil-painting. More results are in Figs. 15 and 16 of Appendix E.

for handling this issue is simple but effective: we add noise to the input image (thus simulating the forward diffusion process) before passing it into the U-Net to extract feature maps. We call these feature maps (and through a slight abuse of notation, our approach) Diffusion Features (DIFT). DIFT can then be used to find matching pixel locations in the two images by doing simple nearest neighbor lookup using cosine distance. We find the resulting correspondences are surprisingly robust and accurate (Fig. 1), even across multiple categories and image modalities.

We evaluate DIFT with two different types of diffusion models, on three groups of visual correspondence tasks including semantic correspondence, geometric correspondence, and temporal correspondence. We compare DIFT with other baselines, including task-specific methods, and other self-supervised models trained with similar datasets and similar amount of supervision (DINO [10] and OpenCLIP [36]). Although simple, DIFT demonstrates strong performance on all tasks without any additional fine-tuning or supervision, outperforms both weakly-supervised methods and other self-supervised features, and even remains on par with the state-of-the-art supervised methods on semantic correspondence.

2 Related Work

Visual Correspondence. Establishing visual correspondences between different images is crucial for various computer vision tasks such as Structure-from-Motion / 3D reconstruction [2, 78, 60, 74], object tracking [22, 97], image recognition [63, 81, 9] and segmentation [50, 47, 72, 28]. Traditionally, correspondences are established using hand-designed features, such as SIFT [51] and SURF [6]. With the advent of deep learning, methods that learn to find correspondences in a supervised-learning regime have shown promising results [46, 14, 42, 35]. However, these approaches are difficult to scale due to the reliance on ground-truth correspondence annotations. To overcome difficulties in collecting a large number of image pairs with annotated correspondences, recent works have started looking into how to build visual correspondence models with weak supervision [31] or self-supervision [92, 37]. Meanwhile, recent works on self-supervised representation learning has yielded strong per-pixel features that could be used to identify visual correspondence [84, 3, 10, 28]. In particular, recent work has also found that the internal representation of Generative Adversarial Networks (GAN) [23] could be used for identifying visual correspondence [89, 62, 57] within certain image categories. Our work shares similar spirits with these works: we show that diffusion models could generate features that are useful for identifying visual correspondence on general images. In addition, we show that features generated at different timesteps and different layers of the de-noising process

encode different information that could be used for determining correspondences needed for different downstream tasks.

Diffusion Model [78, 32, 79, 41] is a powerful family of generative models. Ablated Diffusion Model [17] first showed that diffusion could surpass GAN's image generation quality on ImageNet [15]. Subsequently, the introduction of classifier-free guidance [33] and latent diffusion model [70] made it scale up to billions of text-image pairs [75], leading to the popular open-sourced text-to-image diffusion model, i.e., Stable Diffusion. With its superior generation ability, recently people also start investigating the internal representation of diffusion models. For example, previous works [85, 31] found that the intermediate-layer features and attention maps of diffusion models are crucial for controllable generations; other works [5, 94, 101] explored adapting pre-trained diffusion models for various downstream visual recognition tasks. Different from these works, we are the first to directly evaluate the efficacy of features inherent to pre-trained diffusion models on various visual correspondence tasks.

3 Problem Setup

Given two images I_1, I_2 and a pixel location p_1 in I_1, we are interested in finding its corresponding pixel location p_2 in I_2. Relationships between p_1 and p_2 could be semantic correspondence (i.e., pixels of different objects that share similar semantic meanings), geometric correspondence (i.e., pixels of the same object captured from different viewpoints), or temporal correspondence (i.e., pixels of the same object in a video that may deform over time).

The most straightforward approach to obtaining pixel correspondences is to first extract dense image features in both images and then match them. Specifically, given a feature map F_i for image I_i, we can extract a feature vector $F_i(p)$ for pixel location p through bilinear interpolation. Then given a pixel p_1 in image I_1, we can obtain the corresponding pixel in image I_2 as:

$$p_2 = \arg\min_p d(F_1(p_1), F_2(p)) \tag{1}$$

where d is a distance metric and we use cosine distance by default in this work.

4 Diffusion Features (DIFT)

In this section, we first review what diffusion models are and then explain how we extract dense features on real images using pre-trained diffusion models.

4.1 Image Diffusion Model

Diffusion models [32, 79] are generative models that aim to transform a Normal distribution to an arbitrary data distribution. In our case, we use image diffusion models, thus the data distribution and the Gaussian prior are both over the space of 2D images.

During training, Gaussian noise of different magnitudes is added to clean data points to obtain noisy data points. This is typically thought of as a "diffusion" process, where the starting point of the diffusion x_0 is a clean image from the training dataset and x_t is a noisy image obtained by "mixing" x_0 with noise:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + (\sqrt{1 - \bar{\alpha}_t}) \epsilon \tag{2}$$

where $\epsilon \sim N(0, I)$ is the randomly-sampled noise, and $t \in [0, T]$ indexes "time" in the diffusion process with larger time steps involving more noise. The amount of noise is determined by $\bar{\alpha}_t$, which is a pre-defined noise schedule. We call this the diffusion forward process.

A neural network f_θ is trained to take x_t and time step t as input and predict the input noise ϵ. For image generation, f_θ is usually parametrized as a U-Net [71, 17, 70]. Once trained, f_θ can be used to "reverse" the diffusion process. Starting from pure noise x_T sampled from a Normal distribution, f_θ can be iteratively used to estimate noise ϵ from the noisy data x_t and remove this noise to get a cleaner data x_{t-1}, eventually leading to a sample x_0 from the original data distribution. We call this the diffusion backward process.

Figure 2: Given a Stable Diffusion generated image, we extract its intermediate layer activations at a certain time step t during its backward process, and use them as the feature map to predict the corresponding points. Although simple, this method produces correct correspondences on generated images already not only within category, but also cross-category, even in cross-domain situations, e.g., from a photo to an oil painting.

4.2 Extracting Diffusion Features on Real Images

We hypothesize that diffusion models learn correspondence implicitly [35, 61] in Sec. 1, but how can we extract this correspondence? Consider generated images, where we have access to the complete internal state of the network throughout the entire backward process. Given a generated image from Stable Diffusion [70] , we extract the feature maps of its intermediate layers at a specific time step t during the backward process, which we then utilize to establish correspondences between two different generated images as described in Sec. 3. As illustrated in Fig. 2, this straightforward approach allows us to find correct correspondences between generated images, even when they belong to different categories or domains.

Replicating this approach for real images is challenging because of the fact that the real image itself does not belong to the training distribution of the U-Net (which was trained on noisy images), and we do not have access to the intermediate noisy images that would have been produced during the generation of this image. Fortunately, we found a simple approximation using the forward diffusion process to be effective enough. Specifically, we first add noise of time step t to the real image (Eq. (2)) to move it to the x_t distribution, and then feed it to network together with t to extract the intermediate layer activations as our DIffusion FeaTures, namely DIFT. As shown in Figs. 1 and 3, this approach yields surprisingly good correspondences for real images.

Moving forward, a crucial consideration is the selection of the time step t and the network layer from which we extract features. Intuitively we find that a larger t and an earlier network layer tend to yield more semantically-aware features, while a smaller t and a later layer focus more on low-level details. The optimal choices of t and layer depend on the specific correspondence task at hand, as different tasks may require varying trade-offs between semantic and low-level features. For example, semantic correspondence likely benefits from more semantic-level features, whereas geometric correspondence between two views of the same instance may perform well with low-level features. We therefore use a 2D grid search to determine these two hyper-parameters for each correspondence task. For a comprehensive list of the hyper-parameter values used in this paper, please refer to Appendix C.

Lastly, to enhance the stability of the representation in the presence of random noise added to the input image, we extract features from multiple noisy versions with different samples of noise, and average them to form the final representation.

5 Semantic Correspondence

In this section, we investigate how to use DIFT to identify pixels that share similar semantic meanings across images, e.g., the eyes of two different cats in two different images.

5.1 Model Variants and Baselines

We extract DIFT from two commonly used, open-sourced image diffusion models: Stable Diffusion 2-1 (SD) [70] and Ablated Diffusion Model (ADM) [17]. SD is trained on the LAION [75] whereas ADM is trained on ImageNet [15] without labels. We call these two features DIFT_{sd} and DIFT_{adm} respectively.

To separate the impact of training data on the performance of DIFT, we also evaluate two other commonly used self-supervised features as baselines that share basically the same training data:

Figure 3: Visualization of semantic correspondence prediction on SPair-71k using different features. The leftmost image is the source image with a set of keypoints; the rightmost image contains the ground-truth correspondence for a target image whereas any images in between contain keypoints found using feature matching with various features. Different colors indicate different keypoints. We use circles to indicate correctly-predicted points under the threshold $_{\text{bbox}} = 0:1$ and crosses for incorrect matches. DIFT is able to establish correct correspondences under clustered scenes (row 3), viewpoint changes (row 2 and 4), and occlusions (row 5). See Fig. 17 in Appendix E for more results.

OpenCLIP [36] with ViT-H/14 [18] trained on LAION, as well as DINO [10] with ViT-B/8 trained on ImageNet [15] without labels. Note that for both DIFT and other self-supervised features, we do not ne-tune or re-train the models with any additional data or supervision.

5.2 Benchmark Evaluation

Datasets.We conduct evaluation on three popular benchmarks: SPair-71k [57], PF-WILLOW [27] and CUB-200-2011 [9]. SPair-71k is the most challenging semantic correspondence dataset, containing diverse variations in viewpoint and scale with 12,234 image pairs on 18 categories for testing. PF-Willow is a subset of PASCAL VOC dataset [20] with 900 image pairs for testing. For CUB, following [58], we evaluate 14 different splits of CUB (each containing 25 images) and report the average performance across all splits.

Evaluation Metric. Following prior work, we report the percentage of correct keypoints (PCK). The predicted keypoint is considered to be correct if they lie within $\max(h; w)$ pixels from the ground-truth keypoint for $2 [0; 1]$, where h and w are the height and width of either the image ($_{\text{img}}$) or the bounding box ($_{\text{bbox}}$). To nd a suitable time step and layer feature to use for DIFT and other self-supervised features, we grid search the hyper-parameters using SPair-71k and use the same hyper-parameter settings for PF-WILLOW and CUB.

We observed inconsistencies in PCK measurements across prior literature.[1] Some works [35, 42, 14] use the total number of correctly-predicted points in the whole dataset (or each category split) divided

[1] ScorrSAN [35] and GANgealing [62]'s evaluation code snippets, which calculate PCK per image and PCK per point respectively.