

Unleashing Diffusion Transformers for Visual Correspondence by Modulating Massive Activations

Chaofan Gan^{1,2} Yuanpeng Tu³ Xi Chen³ Tieyuan Chen¹

Yuxi Li¹ Mehrtash Harandi² Weiyao Lin^{1*}

¹Shanghai Jiao Tong University ²Monash University

²The University of Hong Kong

{ganchaofan, tieyuanchen, lyxok1, wylin}@sjtu.edu.cn

{chaofan.gan, mehrtash.harandi}@monash.edu

{tjtuyuanpeng, xichen.csai}@gmail.com

Abstract

Pre-trained stable diffusion models (SD) have shown great advances in visual correspondence. In this paper, we investigate the capabilities of Diffusion Transformers (DiTs) for accurate dense correspondence. Distinct from SD, DiTs exhibit a critical phenomenon in which very few feature activations exhibit significantly larger values than others, known as *massive activations*, leading to uninformative representations and significant performance degradation for DiTs. The massive activations consistently concentrate at very few fixed dimensions across all image patch tokens, holding little local information. We analyze these dimension-concentrated massive activations and uncover that their concentration is inherently linked to the Adaptive Layer Normalization (AdaLN) in DiTs. Building on these findings, we propose the **Diffusion Transformer Feature (DiTF)**, a training-free AdaLN-based framework that extracts semantically discriminative features from DiTs. Specifically, DiTF leverages AdaLN to adaptively localize and normalize massive activations through channel-wise modulation. Furthermore, a channel discard strategy is introduced to mitigate the adverse effects of massive activations. Experimental results demonstrate that our DiTF outperforms both DINO and SD-based models and establishes a new state-of-the-art performance for DiTs in different visual correspondence tasks (e.g., with +9.4% on Spair-71k and +4.4% on AP-10K-C.S.).

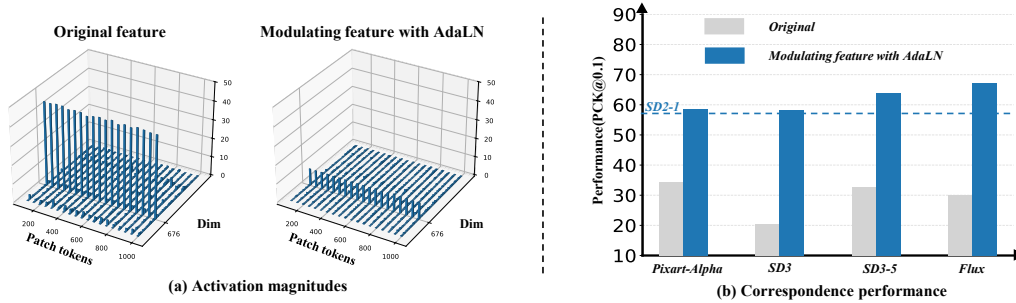


Figure 1: **AdaLN enhances DiT features by mitigating massive activations.** (a) Original DiT features show concentrated massive activations. (b) Semantic correspondence performance using different features. Original DiT features yield poor performance due to the presence of massive activations. By modulating these activations, AdaLN significantly boosts correspondence performance.

*Corresponding Author.

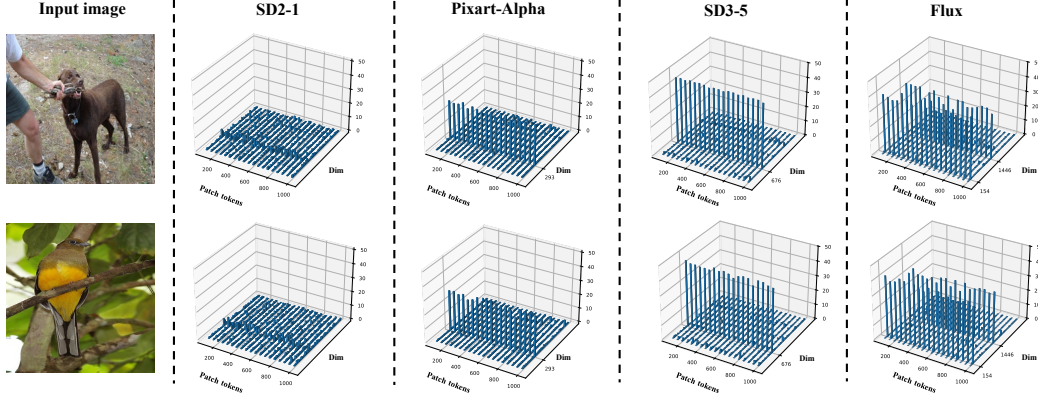


Figure 2: **Massive activations in Diffusion Transformers (DiTs).** We visualize the activation magnitudes (z-axis) of features across various diffusion models. Unlike the Stable Diffusion (SD2-1) model, all DiTs exhibit a distinctive phenomenon where very few feature activations show significantly higher activation values, more than 100 times larger than others. We refer to this phenomenon as *massive activations*, which has also been observed in large language models (LLMs) [20].

1 Introduction

Detecting dense correspondence between images plays an important role in various vision tasks such as segmentation [1, 2, 3, 4], image editing [5, 6, 7], and point tracking [8, 9, 10, 11]. Various works [12, 13, 14] utilize pre-trained self-supervised models like DINOv2 [12] to extract image features, which are then adapted for precise semantic correspondence. With the advent of text-to-image (T2I) generative models [15, 16], recent studies [17, 18, 14] have demonstrated the powerful capability of pre-trained SD [15] as an effective feature extractor [14, 18, 19] for visual correspondence.

Compared with SD, recent DiTs (e.g., Pixart-alpha [21], SD3 [22], and Flux [23]) demonstrate great superiority in scalability, robustness, model efficiency, and generation quality. While pre-trained SD models have been successfully leveraged as feature extractors, the potential of DiTs for visual perception tasks remains largely unexplored. To bridge this gap, we take the first step to investigate the feasibility of using pre-trained DiTs as feature extractors for visual correspondence tasks.

However, when directly extracting the original features for representation learning, as done in SD, DiTs exhibit disappointing performance for visual perception tasks, as shown in Figure 1. To better understand this discrepancy, we analyze the statistical properties of features extracted from both SD and DiTs (see Figure 2). Notably, we observe that DiT feature maps exhibit massive activations, in contrast to the smoother distribution observed in SD2.1 [15]. These massive activations share a few *fixed dominant* dimensions, resulting in a *high degree of directional similarity* for feature vectors in the latent space [24]. This property makes it difficult to distinguish spatial feature vectors using cosine similarity, leading to semantically uninformative and indiscriminate representations.

In this work, we aim to deepen the understanding of massive activations for representation learning, thereby enabling the extraction of semantically discriminative representations from DiTs. We begin by investigating the spatial and dimensional properties of these activations and find that they consistently emerge in a small set of fixed feature dimensions across all image tokens, which carry limited local information that is crucial for our task. Further analysis reveals an intriguing connection between massive activations and the AdaLN in DiTs, where the concentrated dimensions of massive activations align with the residual scaling factors produced by AdaLN. To further understand this phenomenon, we investigate the role of AdaLN in modulating DiT features. Our findings reveal that the built-in AdaLN effectively localizes massive activations and performs semantically meaningful channel-wise modulation to suppress them, thereby enhancing the semantic richness and discriminative capacity of DiT features.

Based on these findings, we propose a training-free AdaLN-based framework: **Diffusion Transformer Feature (DiTF)**, to extract structural and semantic features from DiTs. Specifically, DiTF leverages the built-in AdaLN in DiTs to adaptively normalize massive activations through channel-wise modulation. In addition, a channel discard strategy is introduced to further mitigate the adverse effects of massive activations on representation learning. Our main contributions are summarized as follows.

- We identify and characterize massive activations in DiTs and find them consistently concentrated in a small number of fixed feature dimensions across all spatial tokens.
- We analyze the source of these dimensionally concentrated massive activations and uncover their strong connection to the AdaLN layer in DiTs.
- We propose a training-free AdaLN-based framework, **Diffusion Transformer Feature (DiTF)**, for extracting semantically discriminative representations from DiTs.
- Our framework outperforms both DINO and SD-based models, establishing a new state-of-the-art for Diffusion Transformers on visual correspondence tasks.

2 Related Work

Visual Correspondence. Visual correspondence aims to establish dense correspondence between two different images, which plays a crucial role in various visual tasks such as segmentation [1, 2, 3, 4], image editing [5, 6, 7], and point tracking [8, 9, 10, 11]. Early methods like HOG [25], and SIFT [26] design hand-crafted features with scale-invariant key-points and then establish image correspondences. Various works attempt to learn image correspondence in a supervised learning regime. However, the supervised-based methods [27, 28, 29, 30, 31] depend on ground-truth correspondence annotations, which are challenging to apply to datasets lacking precise correspondence annotations. To handle this challenge, some studies have employed DINO [12, 32] and Generative Adversarial Networks (GAN) [33] to extract visual features for correspondence. More recently, [34, 14, 19, 35] have demonstrated that SD models can serve as effective feature extractors for various perception tasks.

Diffusion Model. Large-scale pre-trained diffusion models [15, 36, 37, 38, 16] are capable of producing high-quality images with reasonable structures and compositional semantics, indicating their robust spatial awareness and semantic understanding capabilities. Consequently, numerous studies have utilized pre-trained SD models as feature extractors, extracting features from images for visual perception tasks [39, 40, 41, 42]. Inspired by these works, [17] and [18] have employed the pre-trained SD model [15], as a representation learner for visual correspondence. In addition, [14] investigated the distinct properties of SD features compared to the DINO [35] feature, and jointly leveraged them for improved performance. Based on this, GeoAware-SC [43] proposes adaptive pose alignment to improve the geometric awareness of the SD features.

Massive Activations. Massive activations have been extensively studied in large language models (LLMs) [20, 44, 45]. Several works [20, 46] have shown that these activations are concentrated in a small number of fixed dimensions, particularly in the start and delimiter tokens. [47] further revealed that the massive values in the query and key vectors originate from the positional encoding mechanism RoPE[48]. In addition to LLMs, similar phenomena have also been observed in Vision Transformers (ViTs)[13, 49, 20]. Specifically, [20] identified attention artifacts in ViT feature maps, predominantly appearing in low-information background regions. DVT [49] further traced these artifacts to the influence of positional embeddings. Recent studies on accelerating DiTs [50, 51] have identified massive activation outliers in DiTs and revealed that these outliers lead to unstable model quantization and excessively high distillation losses during knowledge distillation.

3 Preliminaries

Adaptive Layer Norm. AdaLN operation $\text{AdaLN}(z; \gamma, \beta)$ adaptively normalizes feature z with the adaptive scale γ and shift β parameters.

$$\hat{z} = (1 + \gamma) \text{LayerNorm}(z) + \beta \quad (1)$$

where LayerNorm is the layernorm operation [52].

DiT Architecture. We follow the architecture used in [36]. DiT [36] comprises two parts: a VAE [53], which comprises an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$ to project images from pixel space to latent space, and diffusion transformer blocks $\mathcal{A} = \{\mathcal{A}_k\}_{k=1}^N$, where k is the index and N is the number of blocks. Given an input image x_0 , we first encode it to the latent representation $z_0 = \mathcal{E}(x_0)$ with the encoder $\mathcal{E}$ and then add corresponding noise ϵ to the latent to obtain a noisy latent representation $z_t \in \mathbb{R}^{C \times H \times W}$ according to a pre-defined timestep t . The DiT block $\mathcal{A}_k$ encodes the original feature z_t^k as follows:

$$z_t^{k+1} = z_t^k + \alpha_k \mathcal{A}_k(z_t^k) \quad (2)$$

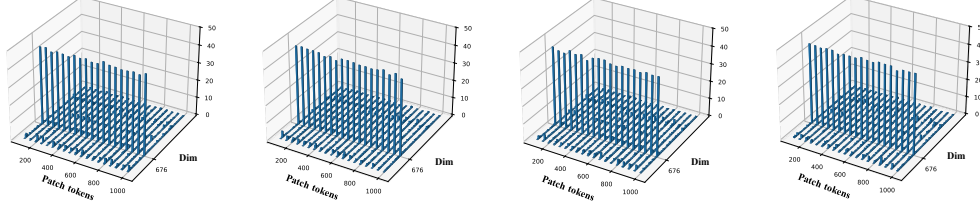


Figure 3: **Massive Activations in SD3-5.** We visualize the activation magnitudes of four different image features extracted using SD3-5. Notably, massive activations consistently distribute in a fixed dimension (676) across all image patch tokens.

where each Diffusion Transformer block employs residual connections [54], with α_k representing the residual scaling factor computed by the AdaLN-zero layer. It is important to note that the original feature throughout this paper is denoted as z_t^k , and our analysis of massive activations is conducted based on this representation.

DiT block $\mathcal{A}_k$. DiT block [36] includes a modulation layer called AdaIN-zero layers [36], which regress two groups of scale and shift parameters based on the given timestep t and additional conditions (e.g., text embedding c).

$$\{(\gamma_k^i, \beta_k^i, \alpha_k^i)\}_{i=1}^2 = \text{MLP}_k(t, c) \quad (3)$$

where MLP is utilized to regress channel-wise scale and shift parameters $\gamma_k^i, \beta_k^i, \alpha_k^i \in \mathbb{R}^C$. γ_k^i, β_k^i served as parameters for i_{th} AdaLN and α_k^i is applied to scale i_{th} residual connections.

Next, the block-input feature z_t^k is passed through the Self-Attention layer to yield intermediate feature $z_t^{(k,2)}$.

$$\hat{z}_t^{(k,1)} = \text{AdaLN}(z_t^{(k,1)}; \gamma_k^1, \beta_k^1) \quad (4)$$

$$z_t^{(k,2)} = \text{Self-Attention}(\hat{z}_t^{(k,1)}) + \alpha_k^1 z_t^{(k,1)} \quad (5)$$

Finally, the intermediate feature $z_t^{(k,2)}$ is fed into the Feedforward layer to yield block-output feature $z_t^{(k+1,1)}$.

$$\hat{z}_t^{(k,2)} = \text{AdaLN}(z_t^{(k,2)}; \gamma_k^2, \beta_k^2) \quad (6)$$

$$z_t^{(k+1,1)} = \text{Feedforward}(\hat{z}_t^{(k,2)}) + \alpha_k^2 z_t^{(k,2)} \quad (7)$$

where $z_t^{(k,1)}$ and $z_t^{(k+1,1)}$ denote the same features as z_t^k and z_t^{k+1} , respectively. For clarity, we refer to $z_t^{(k,1)}$ and $z_t^{(k,2)}$ as the pre-AdaLN intermediate features, and $\hat{z}_t^{(k,1)}$ and $\hat{z}_t^{(k,2)}$ as the post-AdaLN intermediate features in the subsequent sections. The residual scaling factor α_k in Equation (2) corresponds to α_k^2 . For simplicity, we use AdaLN to refer to the AdaLN-zero module in DiTs throughout this paper. See the Appendix A for an illustration of DiTs architecture [36].

4 DiTF: Diffusion Transformer Feature

In this section, we explore how to eliminate the undesirable effects of massive activations and extract semantically meaningful features from DiTs. Specifically, we first analyze the spatial and dimensional characteristics of massive activations in DiTs, then discuss why the AdaLN layer is well-suited to addressing massive activation in the context of representation learning, and finally present our approach for extracting semantic-discriminative features from DiTs.

4.1 Massive Activations in Diffusion Transformers

Massive activations are high-magnitude scalar values. We first introduce a quantitative criterion to characterize massive activations in DiT features. Following the definition proposed for LLMs [20], we adopt a generalized formulation: an activation is considered a massive activation if its magnitude is approximately 100 times greater than the median magnitude within its feature map. It is notable that massive activations are scalar values, which are determined jointly by the patch and dimensions [20].

Table 1: **Massive activations exhibit relatively low variance compared to non-massive activations.** We present the mean and variance of activation values across image patches for selected feature dimensions ranked by average magnitude. The 1st-ranked dimension (massive activations) shows substantially lower variance relative to its mean compared to the 10th and 20th (no-massive activations), indicating limited local information.

Model	1st	2th	10th	20th	Median
SD3-5	-44.51 ± 0.50	2.03 ± 1.39	-0.18 ± 2.36	-0.25 ± 2.07	0.21
Flux	40.66 ± 3.99	-31.62 ± 8.50	-0.41 ± 3.16	0.49 ± 2.92	0.13

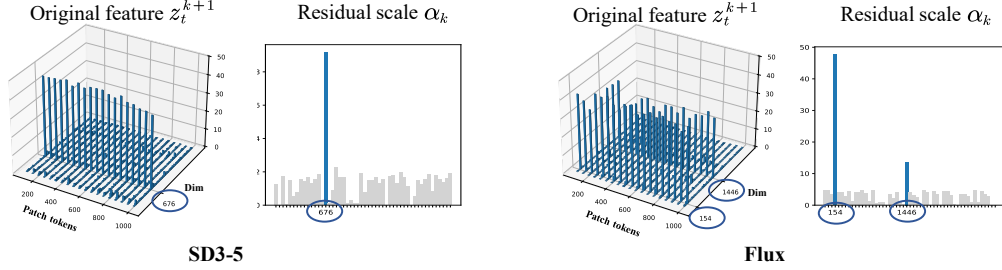


Figure 4: **Massive activations dimensions align with the residual scaling factor α_k .** We visualize the magnitudes for the original feature z_t^{k+1} and residual scaling factor α_k .

Massive activations are present in very few fixed dimensions. We identify the dimensional locations of massive activations within the original DiT feature. As shown in Figure 3, the SD3-5 model exhibits massive activations concentrated in a single feature dimension (676). Similarly, in the Flux model, massive activations appear in two specific dimensions (154 and 1446). Analyses of other DiTs are provided in the Appendix. These observations suggest that massive activations consistently occur in a small number of fixed feature dimensions, which aligns with similar findings in LLMs [20].

Massive activations appear across all spatial tokens. As shown in Figures 2 and 3, massive activations consistently appear across all image patch tokens, regardless of the input image. For instance, in the SD3-5 model, the 676th feature dimension exhibits massive activations across all patch tokens. This phenomenon differs from the observations in LLMs and Vision Transformers (ViTs). Specifically, [20] reported that massive activations in LLMs are primarily concentrated in special tokens such as start and delimiter tokens, while [13] found that outlier activations in ViTs tend to emerge in low-information tokens.

Massive activations hold little local information. To better understand the nature of massive activations, we analyze their spatial variance by measuring the mean and variance of feature values across image patches. Feature dimensions are ranked by their averaged activation magnitude across patches. We then compare the top-ranked dimension (massive activations) with the 10th and 20th highest-ranked (non-massive) dimensions to assess their spatial discriminability. As shown in Table 1, the variance of the massive activations (1st) is substantially lower relative to its mean compared to the non-massive activations (10th and 20th), indicating that massive activations are less spatially discriminative. This suggests that massive activations encode limited local information.

AdaLN enables accurate localization of massive activations. It can be seen that the original features in DiTs are computed as $z_t^{k+1} = z_t^k + \alpha_k \mathcal{A}_k(z_t^k)$ via residual connections, where a scaling factor α_k is applied to weight the block output $\mathcal{A}_k(z_t^k)$ (see Equation (2)). To examine the potential link between massive activations and the residual scaling factor α_k , we analyze their dimensional correspondence in Figure 4 and observe that they consistently co-occur in the same channels. Notably, α_k is regressed by the AdaLN-zero layer, suggesting that AdaLN-zero enables accurate localization of massive activations.

4.2 Channel-wise Modulation with AdaLN

Based on these observations, we hypothesize that AdaLN aggregates the activations in the block output into the dimension corresponding to the highest values in the scaling factor α_k , resulting in dimension-concentrated massive activations. While such concentration may not be inherently

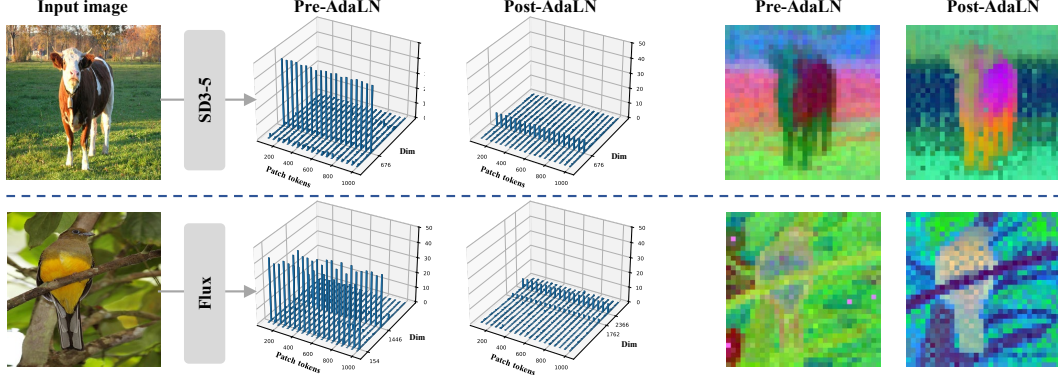


Figure 5: **AdaLN reduces massive activations and enhances feature semantics and discrimination.** We visualize the activation magnitudes and feature maps of both pre-AdaLN and post-AdaLN features in DiT block.

problematic, we argue that it is suboptimal for representation learning. Therefore, we further investigate how AdaLN interacts with the original features and massive activations within them.

AdaLN alleviates massive activations. To better understand how AdaLN interacts with massive activations, we compare the properties of pre-AdaLN and post-AdaLN intermediate features within DiT blocks. As shown in Figure 5, massive activations in feature channels are significantly reduced after AdaLN processing. This suggests that AdaLN can effectively identify such concentrated activations and adaptively normalize them through channel-wise modulation, using the scaling and shifting parameters.

AdaLN enhances feature spatial semantics and discrimination. Furthermore, we visualize different feature maps in Figure 5. It can be observed that the pre-AdaLN feature maps in DiTs exhibit limited semantic coherence, primarily capturing low-level textures such as object color (e.g., the cow and bird), and show weak discrimination between objects and background. In contrast, the post-AdaLN features display markedly improved spatial coherence and clearer semantic boundaries across object parts, suggesting that AdaLN improves spatial semantics and feature discrimination. More visualization results can be found in the Appendix, where the same phenomenon is observed.

4.3 Extracting features from Diffusion Transformers

Based on these observations, we propose to extract DiT features with the channel-wise modulation by AdaLN. Specifically, the feature extraction process in DiTs can be decomposed into two stages: (1) extracting the original feature z_t^k from the DiT block $\mathcal{A}_k$, and (2) modulating it via adaptive channel-wise scaling and shifting using the AdaLN layer.

$$z_t^k = \mathcal{A}_k(z_t^{k-1}), \quad \gamma_k, \beta_k = \text{MLP}_k(t, c) \quad (8)$$

$$\hat{z}_t^k = (1 + \gamma_k) \text{LayerNorm}(z_t^k) + \beta_k \quad (9)$$

where z_t^k is the intermediate pre-AdaLN feature, omitting the index i for clarity. $\hat{z}_t^k$ denotes the final extracted feature. For simplicity, we refer to it as $\hat{z}$ in the following section.

Channel discard. As shown in Figure 5, it can be observed that the post-AdaLN features still exhibit a few weakly massive activations, which will slightly compromise the local semantics of feature maps. Therefore, we propose a simple yet effective channel discard strategy to further suppress their influence. Specifically, we zero out the corresponding dimensions of these weakly massive activations in the post-AdaLN features as they hold little local information.

5 Experiments

5.1 Experimental Setup

Model Variants. We evaluate our strategies on four different pre-trained DiTs: Pixart-alpha [21], SD3 [22], SD3-5 [22], and Flux [23]. The channel-wise modulation with AdaLN operation for DiTs is conditioned on the input timestep t and text embedding c , except for Pixart-alpha [21], which

Table 2: **Performance comparison on dataset SPair-71k.** We present per-class and average PCK@0.10 on the test split. The compared methods are divided into two categories: supervised (S) and unsupervised (U). We report PCK per point results for the (U) methods and PCK per image results for the (S) methods, following [43]. †: Integrating DINOv2 features for correspondence. The highest mAPs are highlighted in bold, while the second highest mAPs are underlined.

Method	Aer	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dog	Horse	Motor	Person	Plant	Sheep	Train	TV	V All
S SCOT [55]	34.9	20.7	63.8	21.1	43.5	27.3	21.3	63.1	20.0	42.9	42.5	31.1	29.8	35.0	27.7	24.4	48.4	40.8	35.6
PMNC [27]	54.1	35.9	74.9	36.5	42.1	48.8	40.0	72.6	21.1	67	58.1	50.5	40.1	54.1	43.3	35.7	74.5	59.9	50.4
SCorSAN [28]	57.1	40.3	78.3	38.1	51.8	57.8	47.1	67.9	25.2	71.3	63.9	49.3	45.3	49.8	48.8	40.3	77.7	69.7	55.3
CATs++ [29]	60.6	46.9	82.5	41.6	56.8	64.9	50.4	72.8	29.2	75.8	65.4	62.5	50.9	56.1	54.8	48.2	80.9	74.9	59.8
DHF [34]	74.0	61.0	87.2	40.7	47.8	70.0	74.4	80.9	38.5	76.1	60.9	66.8	66.6	70.3	58.0	54.3	87.4	60.3	64.9
SD+DINO (S) [14]	81.2	66.9	91.6	61.4	57.4	85.3	83.1	90.8	54.5	88.5	75.1	80.2	71.9	77.9	60.7	68.9	92.4	65.8	74.6
U ASIC [56]	57.9	25.2	68.1	24.7	35.4	28.4	30.9	54.8	21.6	45.0	47.2	39.9	26.2	48.8	14.5	24.5	49.0	24.6	36.9
DINOv2+NN [12]	<u>72.7</u>	<u>62.0</u>	85.2	<u>41.3</u>	40.4	52.3	51.5	71.1	36.2	67.1	<u>64.6</u>	67.6	61.0	68.2	30.7	62.0	54.3	24.2	55.6
DIFT [18]	63.5	54.5	80.8	34.5	46.2	52.7	48.3	77.7	39.0	76.0	54.9	61.3	53.3	46.0	57.8	57.1	71.1	<u>63.4</u>	57.7
DiTF _{sd3-5} (ours)	66.5	56.8	86.3	40.1	<u>51.3</u>	<u>58.6</u>	<u>58.8</u>	81.3	47.8	<u>79.6</u>	60.2	<u>68.5</u>	<u>65.7</u>	<u>73.5</u>	64.4	<u>67.8</u>	<u>69.7</u>	66.5	<u>64.6</u>
DiTF _{f1ux} (ours)	74.3	65.0	88.1	48.1	53.2	60.7	60.7	84.9	<u>42.4</u>	82.8	68.4	72.1	70.9	74.2	<u>62.1</u>	72.6	66.0	60.3	67.1
SD+DINO† [14]	73.0	64.1	86.4	40.7	52.9	55.0	53.8	78.6	45.5	77.3	64.7	69.7	63.3	69.2	58.4	67.6	66.2	53.5	64.0
GeoAware-SC† [43]	<u>78.0</u>	66.4	<u>90.2</u>	44.5	60.1	<u>66.6</u>	60.8	82.7	53.2	<u>82.3</u>	69.5	<u>75.1</u>	66.1	71.7	58.9	71.6	83.8	55.5	<u>69.6</u>
DiTF _{sd3-5} (ours)†	79.1	67.8	90.6	<u>48.3</u>	<u>56.1</u>	69.2	65.0	85.7	59.4	84.6	72.6	76.7	<u>69.9</u>	<u>75.5</u>	<u>62.1</u>	74.4	85.6	59.6	72.2
DiTF _{f1ux} (ours)†	76.3	67.8	88.5	50.8	55.8	60.9	<u>60.8</u>	<u>82.9</u>	47.1	81.3	<u>70.5</u>	72.3	70.0	76.2	62.6	<u>72.5</u>	64.8	<u>56.5</u>	67.6

Table 3: **Performance comparison on datasets SPair-71k, AP-10K, and PF-Pascal datasets at different PCK levels.** We report the performance of the AP-10K intra-species (I.S.), cross-species (C.S.), and cross-family (C.F.) test sets. We report the PCK per image results following [43]. †: Integrating DINOv2 features for correspondence. The highest mAPs are highlighted in bold, while the second highest mAPs are underlined.

Method	SPair-71k			AP-10K-I.S.			AP-10K-C.S.			AP-10K-C.F.			PF-Pascal		
	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.05	0.10	0.15
S SCorSAN [28]	3.6	36.3	55.3	-	-	-	-	-	-	-	-	-	81.5	93.3	96.6
CATs++ [29]	4.3	40.7	59.8	-	-	-	-	-	-	-	-	-	84.9	93.8	96.8
DHF [34]	8.7	50.2	64.9	8.0	45.8	62.7	6.8	42.4	60.0	5.0	32.7	47.8	78.0	90.4	94.1
SD+DINO (S) [14]	9.6	57.7	74.6	9.9	57.0	77.0	8.8	53.9	74.0	6.9	46.2	65.8	80.9	93.6	96.9
U DINOv2+NN [12]	6.3	38.4	53.9	6.4	41.0	60.9	5.3	37.0	57.3	4.4	29.4	47.4	63.0	79.2	85.1
DIFT [18]	7.2	39.7	52.9	6.2	34.8	50.3	5.1	30.8	46.0	3.7	22.4	35.0	66.0	81.1	87.2
DiTF _{sd3-5} (ours)	<u>7.9</u>	<u>48.1</u>	61.2	7.6	50.2	63.3	<u>6.6</u>	<u>48.0</u>	<u>60.2</u>	5.8	<u>33.5</u>	<u>47.9</u>	<u>88.5</u>	<u>95.2</u>	<u>97.3</u>
DiTF _{f1ux} (ours)	8.3	49.3	64.0	7.6	<u>49.7</u>	<u>62.2</u>	6.8	48.2	61.7	<u>5.7</u>	34.2	48.9	89.5	95.8	97.6
SD+DINO† [14]	7.9	44.7	59.9	7.6	43.5	62.9	6.4	39.7	59.3	5.2	30.8	48.3	71.5	85.8	90.6
GeoAware-SC† [43]	9.9	49.1	65.4	11.3	49.8	68.7	9.3	44.9	64.6	7.4	34.9	52.7	74.0	86.2	90.7
DiTF _{sd3-5} (ours)†	11.8	53.1	67.4	15.7	54.4	71.3	13.1	52.0	69.4	9.8	41.4	56.7	89.7	96.2	97.8
DiTF _{f1ux} (ours)†	<u>10.1</u>	<u>50.0</u>	<u>66.0</u>	<u>12.9</u>	<u>52.0</u>	<u>68.8</u>	<u>11.2</u>	<u>49.7</u>	<u>67.1</u>	<u>9.0</u>	<u>39.9</u>	<u>55.6</u>	<u>89.6</u>	<u>95.9</u>	<u>97.6</u>

follows its AdaLN design and is conditioned only on timestep t . Additional experimental details, including DiT configurations and hyperparameter settings, are provided in Appendix C.

Datasets and Evaluation Metric. We evaluate our DiTF on semantic correspondence, geometric correspondence, and temporal correspondence. For semantic correspondence, we conduct experiments on three popular benchmarks: SPair-71k [57], PF-Pascal [58], AP-10K benchmark [43]. The AP-10K benchmark [43] is a new large-scale semantic correspondence benchmark built on AP-10K [59]. It comprises 2.61 million/36,000 data pairs, and spans three settings: the main intra-species set, the cross-species set, and the cross-family set. Following [18, 14], we adopt the percentage of correct key-points (PCK) metric. See Appendix for the results of two additional correspondence tasks.

5.2 Main results on semantic correspondence

To evaluate our model DiTF, we conducted two types of semantic correspondence experiments: in-category semantic correspondence on SPair-71k [57] and PF-Pascal [58], and cross-category semantic correspondence on AP-10K [43].

In-category Semantic Correspondence We conducted comprehensive experiments across different datasets where the results can be found in Tables 2 and 3. From these results, we could obtain the following observations: **1) State-of-the-art performance.** Overall, our models achieve state-of-the-art performance across different datasets. Specifically, our model DiTF_{sd3-5}(ours)† achieves 2.6% ↑ on SPair-71k and 2.6% ↑ on AP-10K-I.S. **2) Superior feature extraction capabilities.** It is readily apparent that our feature extraction models for DiTs, such as DiTF_{sd3-5}(ours) and DiTF_{f1ux}(ours), outperform traditional feature extractors like SD2-1 (DIFT [18]) and DINOv2 [12] across various datasets, demonstrating their strong and robust feature extraction abilities. Specifically, our DiTF_{f1ux}(ours) achieves a performance of 67.1% (9.4% ↑ over DIFT) on the SPair-71k dataset, and obtains a performance of 62.2% (1.3% ↑ over DINOv2) on dataset AP-10K-I.S. **3) Feature**

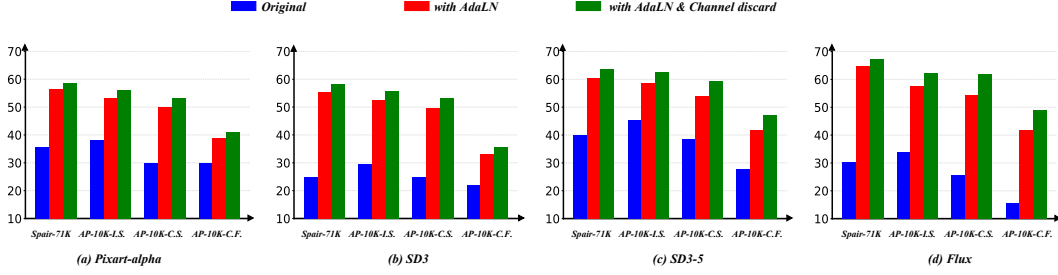


Figure 6: Investigation of the impact of channel modulation with AdaLN and channel discard strategy across different datasets.

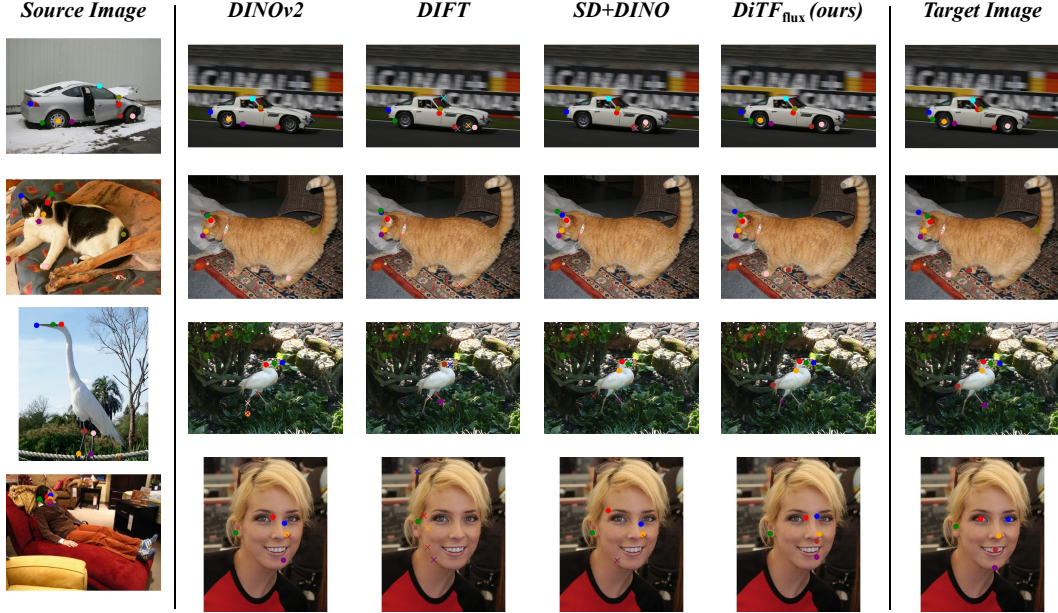


Figure 7: Visualization of semantic correspondence prediction on SPair-71k using different features. Different colors represent different key points where circles denote correctly predicted points under the threshold $\alpha_{bbox} = 0.1$ and crosses denote incorrect matches.

redundancy across DiTs and DINOv2: As shown in Table 2, when integrating DINOv2, our $\text{DiTF}_{\text{sd3-5}}$ achieves a notable performance increase (from 64.6% to 72.2%). In comparison, our $\text{DiTF}_{\text{flux}}$ shows a limited improvement, only a 0.5% gain (from 67.1% to 67.6%). We interpret this performance discrepancy from feature alignment. FLUX aligns closely with DINOv2, causing redundancy, while SD3.5 benefits more from DINOv2’s complementary distribution. This alignment may arise from two factors: Improved alignment between diffusion models and DINOv2 with larger models and longer training [60]; Overlap in training data.

Cross-category Semantic Correspondence In addition, we conducted cross-category semantic correspondence experiments on the AP-10K dataset, as shown in Table 3. The results indicate that our models enable the extraction of accurate semantic features for cross-category image pairs and the drawing of correct correspondences, outperforming the previous best method, GeoAware-SC [43], by nearly 4.8% (64.6% to 69.4%) on AP-10K-C.S. Furthermore, the results show that previous methods, such as DINOv2 and DIFT, experience significant performance degradation on the cross-category task. In contrast, our model demonstrates stronger robustness for the cross-category correspondence.

5.3 Further analysis

Evaluation of Channel Modulation with AdaLN. We conducted comprehensive semantic correspondence experiments across different datasets on various DiTs. The results, as depicted in Figure 6 and Table 4, reveal several key observations: **1) Effectiveness.** When integrated with the channel modulation by AdaLN, all the DiT models experience an absolute performance boost of more than 20%, demonstrating that AdaLN is capable of effectively mitigating massive activations and

Table 4: **Ablation study of our model DiTF_{flux}** on the dataset SPair-71k and AP-10K. We report the PCK per point for SPair-71k and PCK per image for AP-10K-I.S.

Model Variants	SPair-71k			AP-10K-I.S.		
	0.01	0.05	0.10	0.01	0.05	0.10
Original	2.3	14.6	29.9	2.4	16.5	33.9
+ AdaLN	7.8	52.3	65.3	6.4	46.8	60.3
+ Channel discard	8.3	56.3	67.1	7.6	49.7	62.2

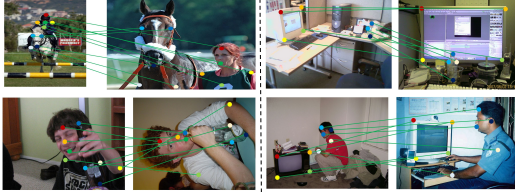


Figure 8: **Visualization of multi-to-multi correspondences with our model.** Green lines indicate correct matches and red incorrect

enhancing feature semantics and discrimination. **2) Robustness.** These results indicate that channel modulation with AdaLN enables the accurate normalization of features across different categories, leading to significant improvements on cross-category datasets.

Evaluation of Channel Discard strategy. As shown in Figure 6 and Table 4. We can observe that our channel discard strategy can effectively locate and eliminate the weakly massive activations in the DiT feature across different DiTs, leading to smoother feature maps and higher correspondence matching accuracy.

Impact of conditions in AdaLN for Channel-wise Modulation. From the results in Table 5, we can observe that the timestep t is essential for modulating massive activations, bringing a significant performance boost. In contrast, attempting to modulate features solely based on c proves ineffective. This phenomenon arises from the inherent nature of DiTs. During training, DiTs introduce noise to images at varying levels of t , making the timestep an essential condition for feature modulation. It serves as a key factor in locating the channel positions of massive activations and enhancing feature quality, enabling their transformations into a clean and semantically meaningful representation.

Qualitative Results. We visualize some semantic correspondence predictions from different models in Figures 7 and 8. The visualization reveals that our DiT-based model yields robust and accurate correspondences across various complex scenes, including changes in viewpoint, multiple objects, and dense matching. These results demonstrate the superiority and robustness of our models.

5.4 Other Task: Semantic Segmentation

To assess the generalizability of DiTF, we conduct semantic segmentation experiments on ADE20K [61]. We extract activated DiT features using DiTF without applying the channel discard strategy and train a segmentation head following DINOv2 [12]. As shown in Table 6, DiTF_{flux} outperforms DINOv2, with channel modulation of AdaLN significantly enhancing feature semantics and discriminability, leading to a substantial mIoU improvement (from 44.8 to 54.8). These results highlight the effectiveness and broad applicability of our approach.

6 Conclusion

In this paper, we identify and characterize massive activations in Diffusion Transformers (DiTs). We observe that these activations consistently emerge in a few fixed dimensions across all image patch tokens and carry limited local information. We further demonstrate that the built-in AdaLN mechanism in DiTs effectively suppresses massive activations while enhancing feature semantics and discriminability. Building on these insights, we propose a training-free AdaLN-based framework, Diffusion Transformer Feature (DiTF), which extracts semantically discriminative features from DiTs through channel-wise modulation using AdaLN. Extensive experiments on visual correspondence tasks validate the robustness and effectiveness of our approach.

Table 5: **Conditions ablation study of AdaLN for feature channel-wise modulation** on SPair-71k. We report the PCK@0.10 per point.

Condition	Pixart-Alpha	SD3	SD3-5	Flux
<i>original</i>	35.6	24.8	39.8	29.9
c	-	17.9	36.8	20.3
t	55.3	54.6	58.2	60.9
t & c	-	54.9	58.3	63.6

Table 6: **Semantic segmentation on ADE20K.**

Method	mIoU ^{ss}	mIoU ^{ms}
<i>self-supervised pre-training</i>		
DINOv2 [12]	47.7	53.1
<i>SD-based pre-training</i>		
VPD [42]	53.7	54.4
<i>DiTs-based pre-training</i>		
DiTF _{flux} (w/o AdaLN)	43.8	44.8
DiTF _{flux}	53.6	54.8

7 Limitations and Future Work.

In this work, we primarily focus on understanding and mitigating massive activations in Diffusion Transformers (DiTs) from the perspective of representation learning. However, the emergence and potential roles of massive activations in the generative process of DiTs remain underexplored. We believe that investigating these activations from a generative viewpoint could provide deeper insights into their underlying mechanisms and potentially contribute to improving generative performance.

Acknowledgements

The paper is supported in part by the National Natural Science Foundation of China (No. U21B2013, 62325109), and in part by the Shanghai 'The Belt and Road' Young Scholar Exchange Grant (24510742000). Mehrtash Harandi is supported by the Australian Research Council (ARC) Discovery Program DP250100262.

References

- [1] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Unsupervised part segmentation through disentangling appearance and shape. In *CVPR*, 2021.
- [2] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv:2203.08414*, 2022.
- [3] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. *arXiv:2305.13310*, 2023.
- [4] Aliasghar Khani, Saeid Asgari Taghanaki, Aditya Sanghi, Ali Mahdavi Amiri, and Ghassan Hamarneh. Slime: Segment like me. *arXiv:2309.03179*, 2023.
- [5] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *SIGGRAPH*, 2024.
- [6] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *CVPR*, 2020.
- [7] Sooyeon Go, Kyungmook Choi, Minjung Shin, and Youngjung Uh. Eye-for-an-eye: Appearance transfer with semantic correspondence in diffusion models. *arXiv:2406.07008*, 2024.
- [8] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022.
- [9] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv:2307.07635*, 2023.
- [10] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *ICCV*, 2023.
- [11] Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. Dino-tracker: Taming dino for self-supervised point tracking in a single video. In *ECCV*, 2024.
- [12] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023.
- [13] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv:2309.16588*, 2023.
- [14] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *NeurIPS*, 2024.
- [15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.
- [16] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022.
- [17] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *NeurIPS*, 2024.
- [18] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023.
- [19] Xinghui Li, Jingyi Lu, Kai Han, and Victor Adrian Prisacariu. Sd4match: Learning to prompt stable diffusion model for semantic matching. In *CVPR*, 2024.

- [20] Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.
- [21] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-*alpha*: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv:2310.00426*, 2023.
- [22] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.
- [23] black-forest labs. Flux. <https://blackforestlabs.ai/announcing-black-forest-labs/>, 2024.
- [24] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Clearclip: Decomposing clip representations for dense vision-language inference. In *European Conference on Computer Vision*, pages 143–160. Springer, 2024.
- [25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [26] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
- [27] Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N Sinha. Patchmatch-based neighborhood consensus for semantic correspondence. In *CVPR*, 2021.
- [28] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *ECCV*, 2022.
- [29] Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *TPAMI*, 2022.
- [30] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *NeurIPS*, 2021.
- [31] Seungwook Kim, Juhong Min, and Minsu Cho. Transmatcher: Match-to-match attention for semantic correspondence. In *CVPR*, 2022.
- [32] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.
- [33] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020.
- [34] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *NeurIPS*, 2024.
- [35] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv:2112.05814*, 2021.
- [36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.
- [37] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.
- [38] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv:2112.10741*, 2021.
- [39] Tomer Amit, Tal Shaharbandy, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv:2112.00390*, 2021.
- [40] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv:2112.03126*, 2021.

- [41] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023.
- [42] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023.
- [43] Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. Telling left from right: Identifying geometry-aware semantic correspondence. In *CVPR*, 2024.
- [44] Jun Zhao, Zhihao Zhang, Yide Ma, Qi Zhang, Tao Gui, Luhui Gao, and Xuanjing Huang. Unveiling a core linguistic region in large language models. *arXiv preprint arXiv:2310.14928*, 2023.
- [45] Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. SLMRec: Distilling large language models into small for sequential recommendation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [46] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.
- [47] Mingyu Jin, Kai Mei, Wujiang Xu, Mingjie Sun, Ruixiang Tang, Mengnan Du, Zirui Liu, and Yongfeng Zhang. Massive values in self-attention modules are the key to contextual knowledge understanding. *arXiv preprint arXiv:2502.01563*, 2025.
- [48] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [49] Jiawei Yang, Katie Z Luo, Jiefeng Li, Congyue Deng, Leonidas Guibas, Dilip Krishnan, Kilian Q Weinberger, Yonglong Tian, and Yue Wang. Denoising vision transformers. In *European Conference on Computer Vision*, pages 453–469. Springer, 2024.
- [50] Gongfan Fang, Kunjun Li, Xinyin Ma, and Xinchao Wang. Tinyfusion: Diffusion transformers learned shallow. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18144–18154, 2025.
- [51] Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Rui Wan, Widyadewi Soedarmadji, Enshu Liu, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [52] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [53] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.
- [54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [55] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *CVPR*, 2020.
- [56] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. Asic: Aligning sparse in-the-wild image collections. In *ICCV*, 2023.
- [57] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv:1908.10543*, 2019.
- [58] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *CVPR*, 2016.

- [59] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv:2108.12617*, 2021.
- [60] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.
- [61] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.
- [62] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *NeurIPS*, 2018.
- [63] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*, 2018.
- [64] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv:1905.03561*, 2019.
- [65] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *NeurIPS*, 2020.
- [66] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, 2019.
- [67] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *NeurIPS*, 2019.
- [68] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020.
- [69] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.
- [70] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.
- [71] Yingdong Hu, Renhao Wang, Kaifeng Zhang, and Yang Gao. Semantic-aware fine-grained correspondence. In *European Conference on Computer Vision*, pages 97–115. Springer, 2022.
- [72] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [73] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [74] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [75] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [76] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

- [77] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
- [78] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5173–5182, 2017.
- [79] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.
- [80] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. *Advances in Neural Information Processing Systems*, 32, 2019.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main contributions and scope are summarized in the introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Limitations and future works are discussed in the main body.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We have already provided the proof of theoretical results in the experiment section.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Datasets and codes are available for double-blind review in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Datasets and codes are available for double-blind review in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Implementation details are fully provided in the main manuscript and the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the average results of open-source models under three random seeds (2024, 2025, 2026). Besides, the average performance is taken from three experimental runs for all proprietary models.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We extract representation from DiTs on one L40s GPU for about 10 minutes, max GPU memory is about 16 GB.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We conducted the research in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The scope of our research does not involve data or models with a high risk for misuse. We utilize publicly available, non-sensitive datasets and models that do not require special safeguards for responsible release.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We strictly adhered to the license terms for all the assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This question does not apply to our research as no new assets were introduced. Our study solely relies on existing, publicly available resources, and thus, no additional documentation for new assets is necessary.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This question does not apply to our research as it does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This question is not applicable to our research as no human subjects were involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A Diffusion Transformer Architecture

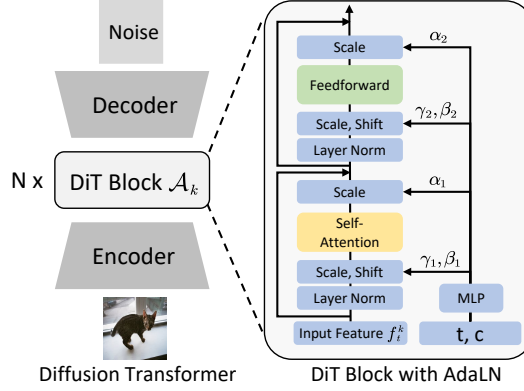


Figure 9: DiT architecture illustration.

We strictly follow the architecture used in DiT [36]. We provide an illustration of the DiT architecture as shown in Figure 9.

B More Visualization Results

B.1 Massive Activations in DiTs

To further illustrate the emergence of massive activations in DiTs, we provide additional visualization results in Figure 13. We show the LayerNorm-normalized activation magnitudes of original features from SD2-1 and various DiTs. While SD2-1 exhibits smooth activations, DiTs consistently show spikes concentrated in a few fixed dimensions across all patch tokens, revealing a fundamental difference that contributes to their degraded performance.

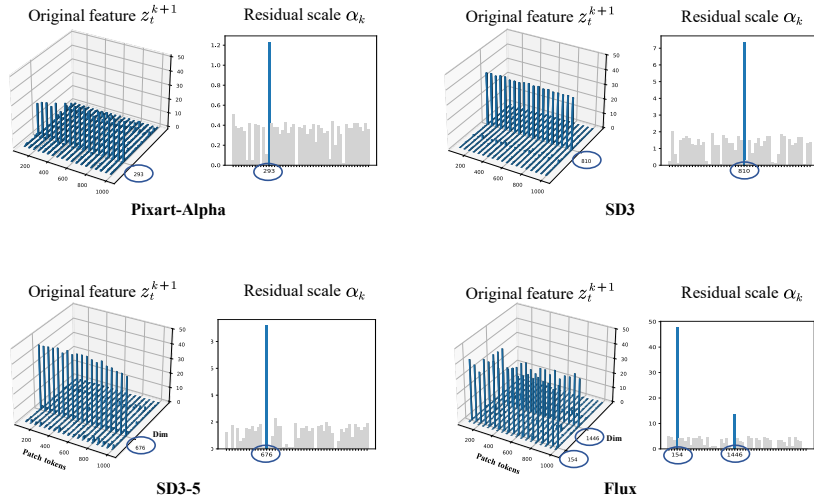


Figure 10: Massive activations dimensions align with the residual scaling factor α_k . We visualize the magnitudes for the original feature z_k^{t+1} and residual scaling factor α_k .

B.2 Massive Activations Dimensions Align with Residual Scaling Factor

In this section, we provide additional visualizations to examine the dimensional alignment between massive activations and the residual scaling factor α_k from the AdaLN layer. As illustrated in Fig-

ure 10, massive activations consistently co-occur with large values of α_k in the same dimensions across all DiTs.

B.3 Channel-wise Modulation with AdaLN

To comprehensively demonstrate the impact of the built-in AdaLN in DiTs, we provide additional visualizations comparing pre-AdaLN and post-AdaLN features across various models, including SD3-5 (Figure 11), Pixart-Alpha (Figure 14), SD3 (Figure 15), and Flux (Figure 16). These results consistently show that AdaLN accurately localizes and normalizes massive activations, while enhancing feature semantics and discrimination through effective channel-wise modulation.

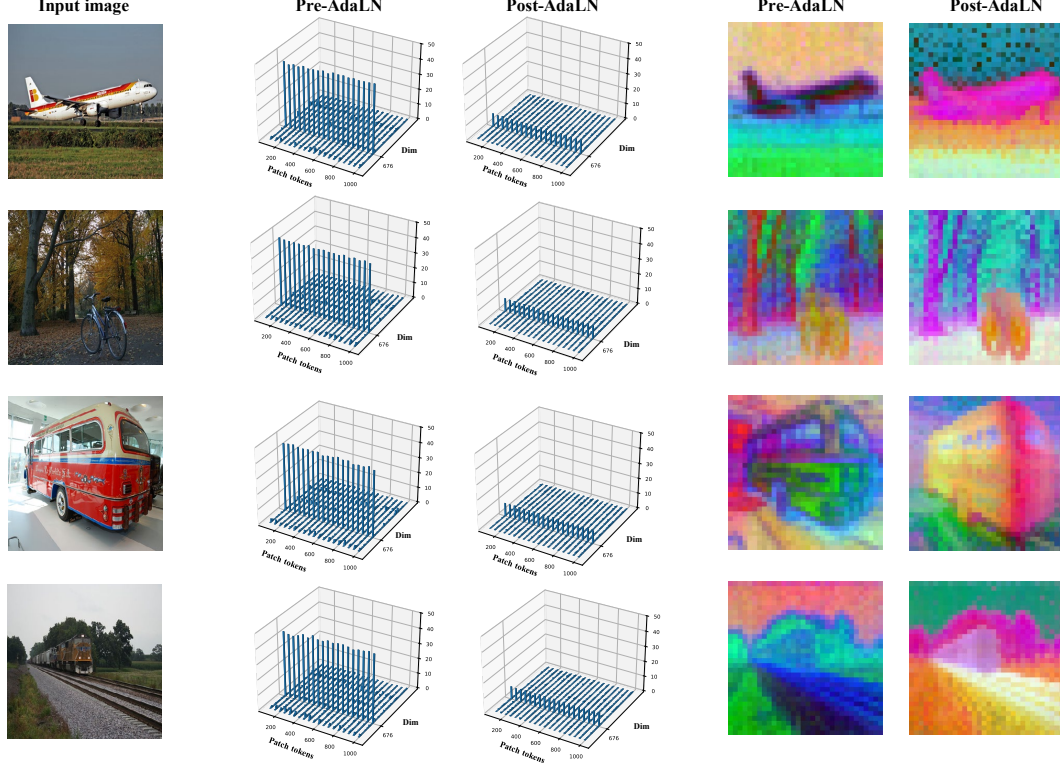


Figure 11: Comparisons of pre-AdaLN and post-AdaLN features in SD3-5.

C Further Implementation Details

Configurations of different DiTs. We employ the pre-trained Diffusion Transformers (DiTs) as a feature extractor for semantic correspondence. Formally, we decompose the universal feature extraction process in DiTs into two stages: (1) extracting the original feature z_t^k from the DiT block $\mathcal{A}_k$, and (2) modulating it via adaptive channel-wise scaling and shifting using the AdaLN layer. For Pixart-alpha [21], SD3 [22], and SD3-5 [22], we first extract the pre-AdaLN feature $z_t^{(k,2)}$ and then activate it as follows.

$$z_t^{(k,2)} = \mathcal{A}_k(z_t^k), \quad \gamma_k^2, \beta_k^2 = \text{MLP}_k(t, c) \quad (10)$$

$$\hat{z}_t^k = (1 + \gamma_k^2) \text{LayerNorm} \left(z_t^{(k,2)} \right) + \beta_k^2 \quad (11)$$

As some of the Flux [23] model’s blocks contain only one group of AdaLN-zero layer, we extract pre-AdaLN feature $z_t^{(k,1)}$ and then activate it as follows.

$$z_t^{(k,1)} = \mathcal{A}_k(z_t^k), \quad \gamma_k^1, \beta_k^1 = \text{MLP}_k(t, c) \quad (12)$$

$$\hat{z}_t^k = (1 + \gamma_k^1) \text{LayerNorm} \left(z_t^{(k,1)} \right) + \beta_k^1 \quad (13)$$

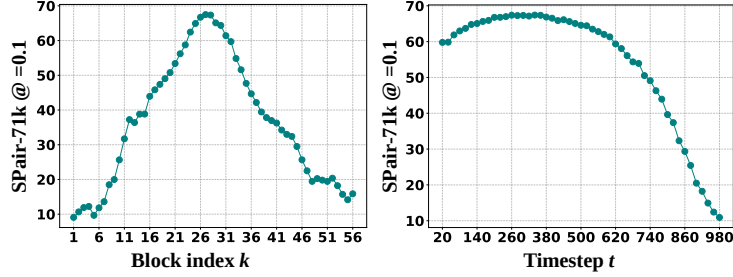


Figure 12: **Investigation of the DiT block index k and timestep t .** We report the results of DiTF_{flux} on dataset Spair-71k.

Table 7: **Configurations of different DiTs for semantic correspondence.**

Method	Layers N	Hidden size d	Timestep t	Block index k
DiTF _{pixart-α}	28	1152	141	14
DiTF _{SD3}	24	1536	340	9
DiTF _{SD3-5}	38	2432	380	23
DiTF _{flux}	57	3072	260	28

The configurations of different DiTs can be found in Table 7, where the total time step T is 1000. We set the input image size as 960x960 for DiTs and 840x840 for the model DINOv2.

Investigation on block index k and timestep t . The layer index k and the timestep t are critical hyperparameters that influence the quality of the extracted features from Diffusion Transformers (DiTs). Previous studies [18, 14] have conducted thorough investigations to identify the optimal layer index k and timestep t for Stable Diffusion. To explore the effects of varying k and t in DiTs, we conducted grid search experiments to identify the optimal hyperparameters. The results are presented in Figure 12. The figure reveals that features extracted from the middle layer achieve optimal performance across different DiTs. Furthermore, feature extraction in DiTs is robust to the timestep for semantic correspondence tasks, as a wide range of t achieves excellent performance.

Integration of DINOv2 feature. We extracted DINOv2 features from the token facet of the 11th layer of the model. We then concatenated the DiT’s features with the DINOv2 features in the channel dimension. To improve the efficiency of correspondence calculation, we computed Principal Component Analysis (PCA) across the pair of images for the features extracted from DiT, as follows:

$$\tilde{z}^s, \tilde{z}^t = \text{PCA}(\hat{z}^s \| \hat{z}^t) \quad (14)$$

where $\hat{z}^s, \hat{z}^t$ are the extracted source and target image DiT features. We only apply the PCA operation to SD3-5 and Flux features due to their high dimension and set the output dimension size as 1280.

D Geometric Correspondence

To comprehensively evaluate our model DiTF, we conduct additional experiments on geometric correspondence.

Datasets. Following [18], we evaluate our model on the HPatches benchmark [78], which comprises 116 sequences: 57 with illumination changes and 59 with viewpoint variations. Adopting the approach from CAPS [68], we detect up to 1,000 key points per image and apply `cv2.findHomography()` to estimate homography using mutual nearest neighbor matches.

Metric. We adopt the corner correctness metric for evaluation where we compute the average error between the four estimated corners of one image and the ground-truth corners with a threshold ϵ pixels, following [68, 18].

Results. To comprehensively evaluate our models, we conducted geometric correspondence experiments on the HPatches benchmark [78], as detailed in Table 8. From the results, it can be observed that our model enables robust feature extraction for image pairs and draws precise geometric correspondence. Specifically, our model DiTF_{flux}(ours) achieve comparable performance 41.9% compared to the state-of-the-art model DiTF (SD-based). These results show that Diffusion Transformers can be employed as an effective feature extractor for geometric correspondence.

Table 8: **Geometric correspondence results on dataset HPatches.** We report the homography estimation accuracy [%] at 1, 3, 5 pixels.

Method	Geometric Supervision	All			Viewpoint Change			Illumination Change		
		$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$
SIFT [26]	None	40.2	68.0	79.3	26.8	55.4	72.1	54.6	81.5	86.9
LF-Net [62]		34.4	62.2	73.7	16.8	43.9	60.7	53.5	81.9	87.7
SuperPoint [63]		36.4	72.7	82.6	22.1	56.1	68.2	51.9	90.8	98.1
D2-Net [64]		16.7	61.0	75.9	3.7	38.0	56.6	30.2	84.9	95.8
DISK [65]	Strong	40.2	70.6	81.5	23.2	51.4	67.9	58.5	91.2	96.2
ContextDesc [66]		40.9	73.0	82.2	29.6	60.7	72.5	53.1	86.2	92.7
R2D2 [67]		40.0	74.4	84.3	26.4	60.4	73.9	54.6	89.6	95.4
<i>w/ SuperPoint kp.</i>										
CAPS [68]	Weak	44.8	76.3	85.2	35.7	62.9	74.3	54.6	90.8	96.9
DINO [12]		38.9	70.0	81.7	21.4	50.7	67.1	57.7	90.8	97.3
OpenCLIP [69]		33.3	67.2	78.0	18.6	45.0	59.6	49.2	91.2	97.7
DIFT [18]	None	45.6	73.9	83.1	30.4	56.8	69.3	61.9	92.3	98.1
DiTF _{flux} (ours)		41.9	70.7	79.5	22.0	50.8	63.4	62.5	91.3	96.2

Table 9: **Temporal correspondence results on DAVIS-2017.** We report the region-based similarity $\mathcal{J}$ and contour-based accuracy $\mathcal{F}$ for DAVIS. Pre-: Pre-trained on videos.

Pre-	Method	Dataset	DAVIS		
			$\mathcal{J} \& \mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
✓	MAST	YT-VOS [70]	65.5	63.3	67.6
	SFC [71]		71.2	68.3	74.0
✗	InstDis [70]	ImageNet [72] w/o labels	66.4	63.9	68.9
	MoCo [73]		65.9	63.4	68.4
	SimCLR [74]		66.9	64.4	69.4
	BYOL [75]		66.5	64.0	69.0
	SimSiam [76]		67.2	64.8	68.8
	DINO [32]		71.4	67.9	74.9
	OpenCLIP [69]	LAION [77]	62.5	60.6	64.4
	DIFT [18]		70.0	67.4	72.5
	DiTF _{flux} (ours)		72.2	69.2	75.1

E Temporal Correspondence

In addition, we conduct experiments to verify the temporal correspondence capability of our DiTF. Specifically, we investigate DiTF’s performance on video object segmentation and pose tracking tasks, employing DiTs as a feature extractor for correspondence.

Datasets. We conduct experiments on the challenge video dataset: DAVIS-2017 video instance segmentation benchmark [79], following [18].

Metric. Following [18, 80], we adopt the region-based similarity $\mathcal{J}$ and contour-based accuracy $\mathcal{F}$ as the performance metric where we segment the nearest neighbors between the consecutive video frames based on the representation similarity.

Results. The temporal correspondence results can be found in Table 9. From the results, it can be observed that our model exhibits a superior capability to extract video frame features for temporal correspondence. Specifically, our model achieves 72.2% on the dataset DAVIS, which outperforms the previous state-of-the-art DIFT by 2.2%, demonstrating the superior effectiveness of our model.

F Qualitative Results on AP-10K

We show the qualitative comparison of our Diffusion Transformer model DiTF with both Stable Diffusion (SD) and SD+DINO [14] in AP-10K intra-species (Figure 17), cross-species (Figure 18), and cross-family (Figure 19) subset.

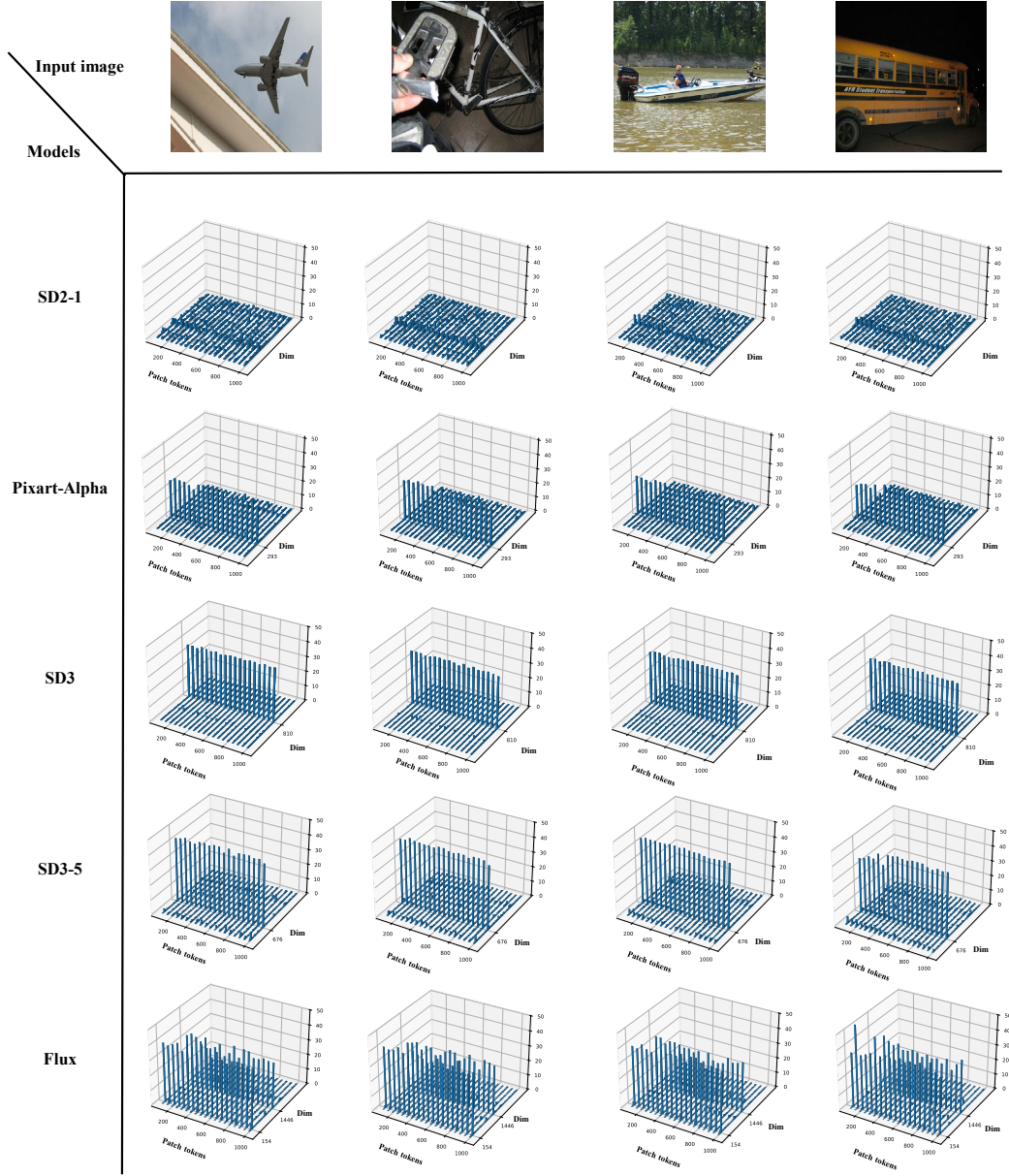


Figure 13: **Massive activations in DiTs.** SDv2-1 does not suffer from massive activations. However, all DiTs exhibit massive activations, which concentrate on very few fixed dimensions across all image patch tokens.

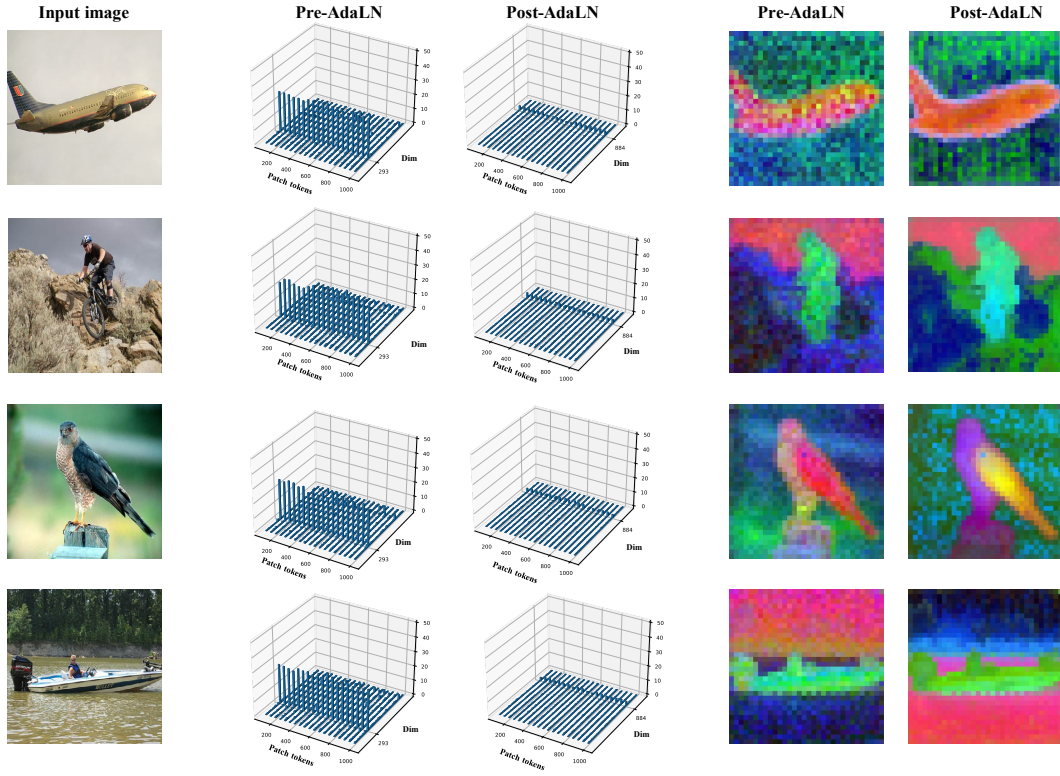


Figure 14: Comparisons of pre-AdaLN and post-AdaLN features in Pixart-Alpha.

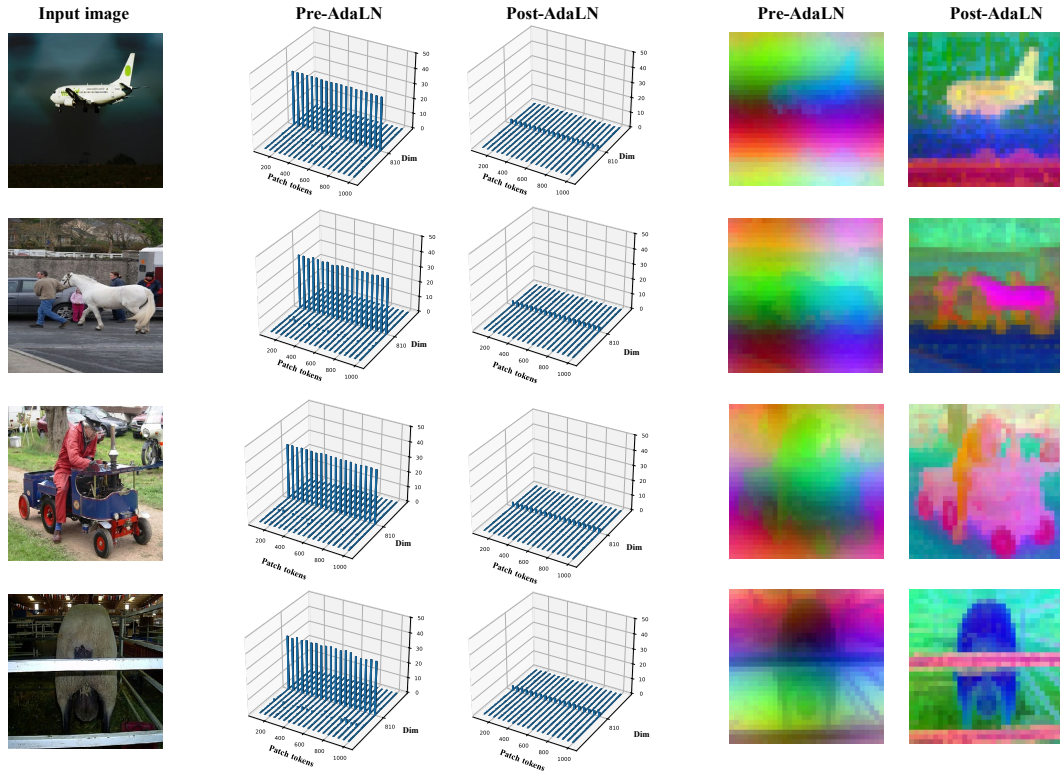


Figure 15: Comparisons of pre-AdaLN and post-AdaLN features in SD3.

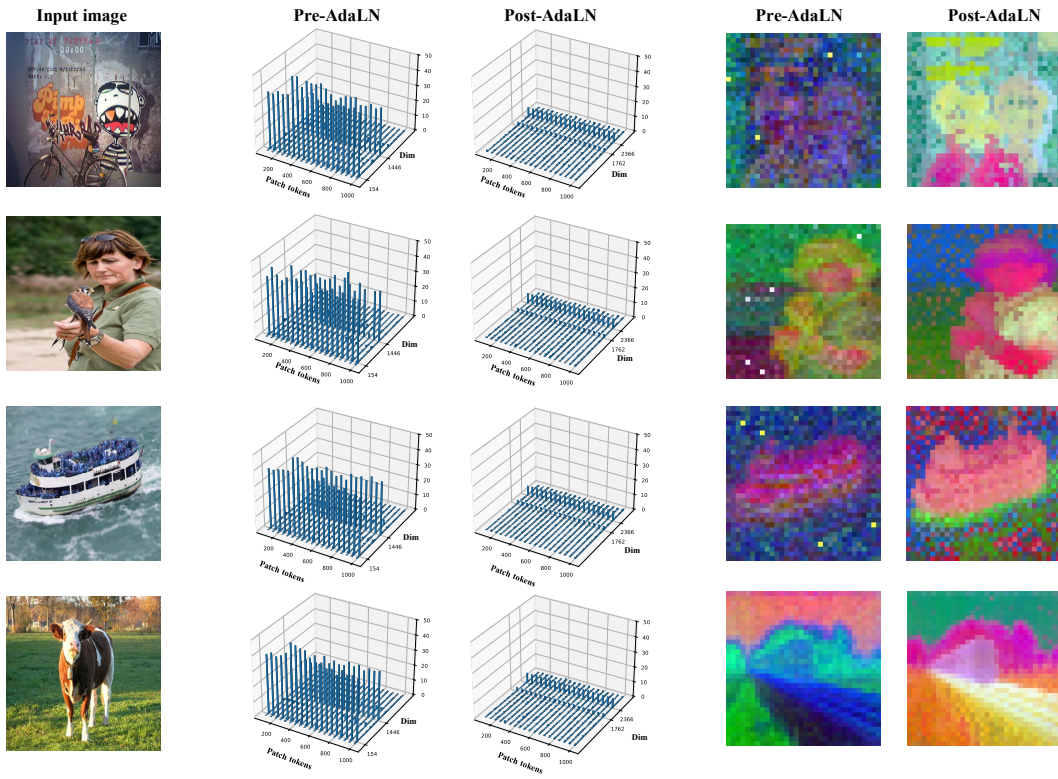


Figure 16: Comparisons of pre-AdaLN and post-AdaLN features in Flux.

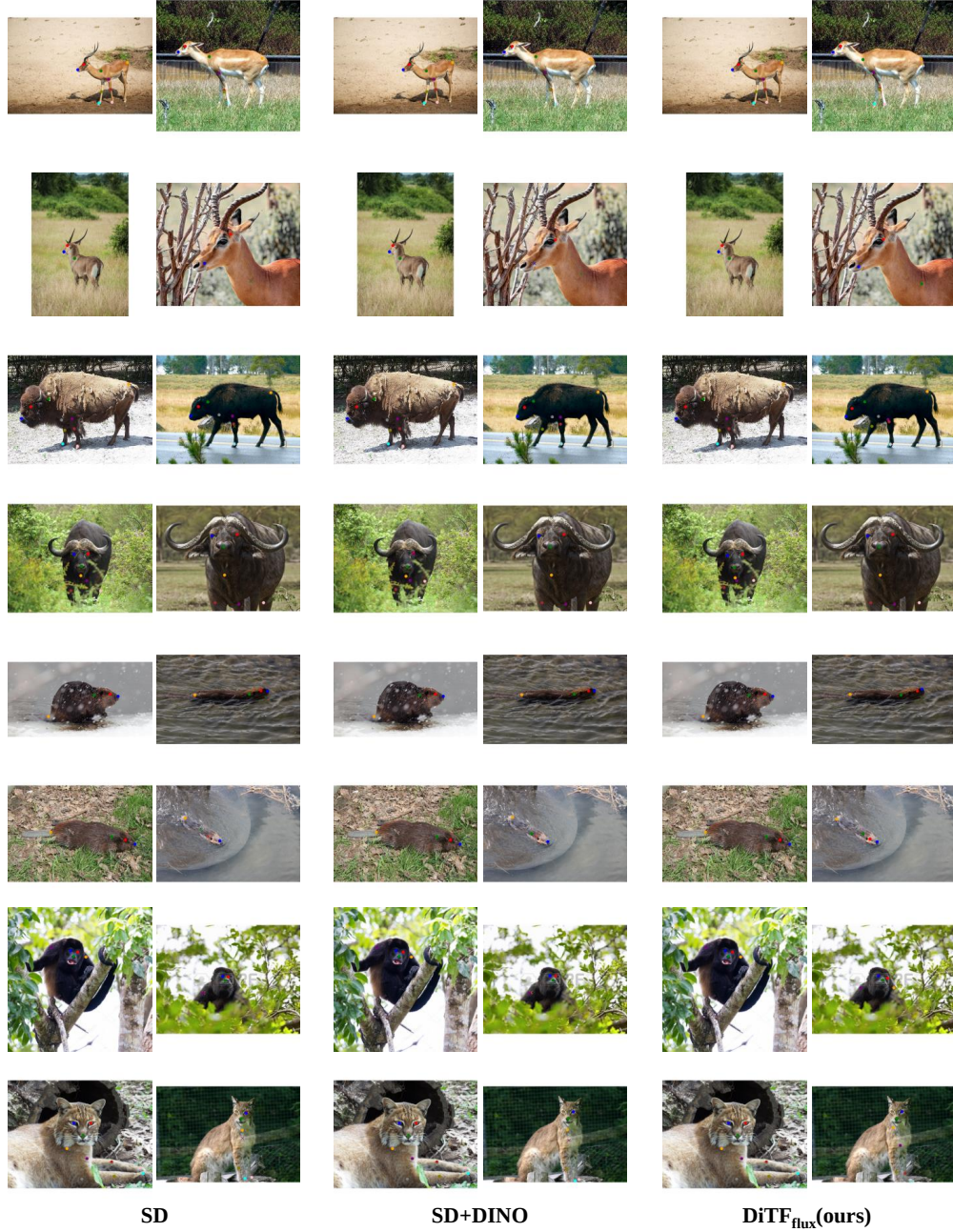


Figure 17: **Qualitative comparison on the AP-10K intra-species set.** Different colors represent different key points where circles denote correctly predicted points and crosses denote for incorrect matches.

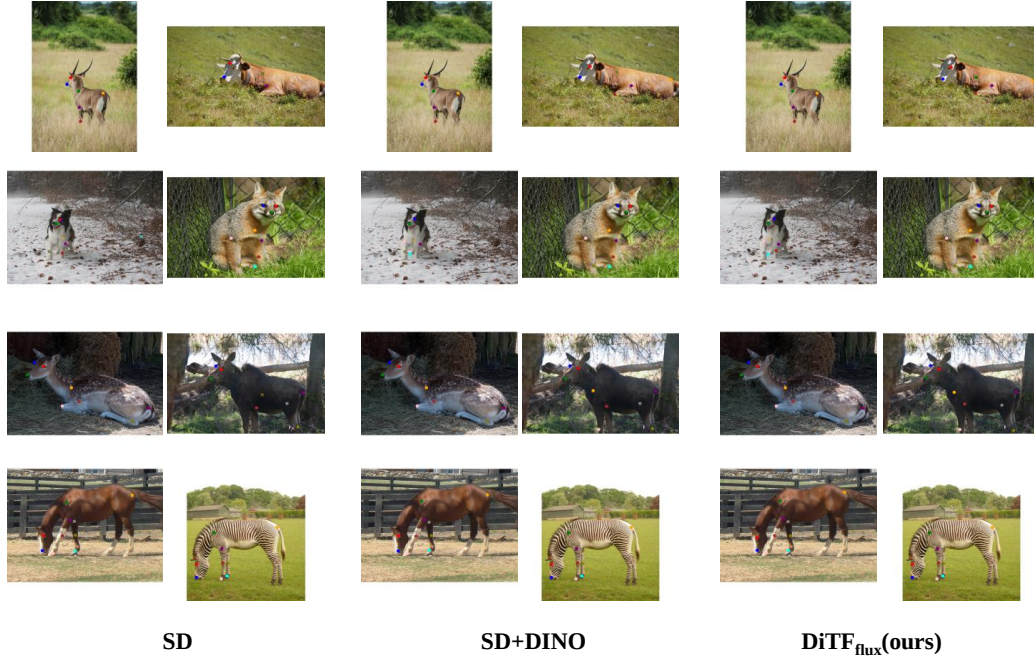


Figure 18: Qualitative comparison on the AP-10K cross-species set.

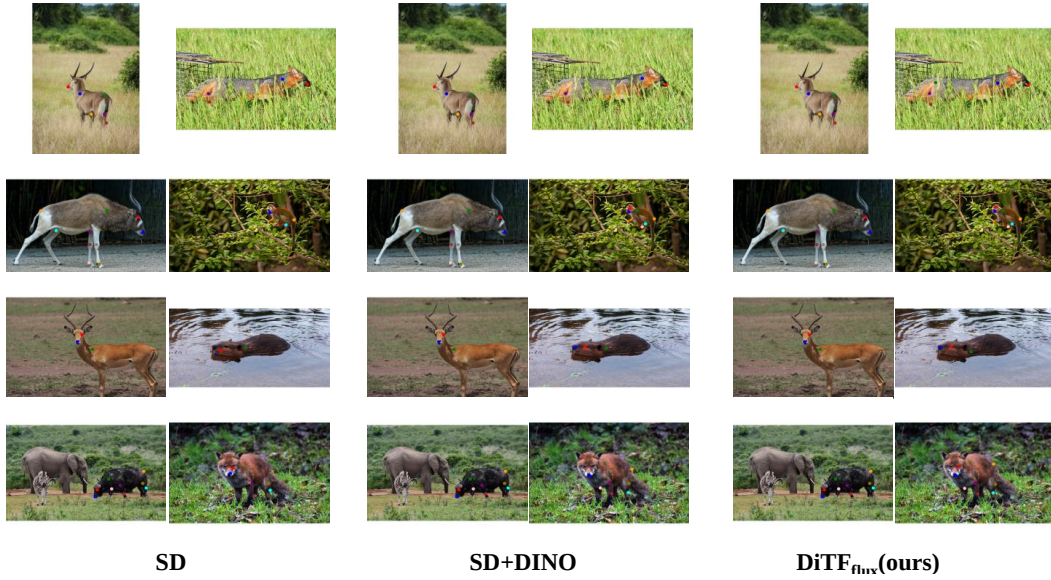


Figure 19: Qualitative comparison on the AP-10K cross-family set.