SCAPE: A Simple and Strong Category-Agnostic Pose Estimator*

Yujia Liang, Zixuan Ye, Wenze Liu, and Hao Lu

National Key Laboratory of Multispectral Information Intelligent Processing Technology; School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China
{yjl,hlu}@hust.edu.cn

Abstract. Category-Agnostic Pose Estimation (CAPE) aims to localize keypoints on an object of any category given few exemplars in an in-context manner. Prior arts involve sophisticated designs, *e.g.*, sundry modules for similarity calculation and a two-stage framework, or takes in extra heatmap generation and supervision. We notice that CAPE is essentially a task about feature matching, which can be solved within the attention process. Therefore we first streamline the architecture into a simple baseline consisting of several pure self-attention layers and an MLP regression head—this simplification means that one only needs to consider the attention quality to boost the performance of CAPE. Towards an effective attention process for CAPE, we further introduce two key modules: i) a global keypoint feature perceptor to inject global semantic information into support keypoints, and ii) a keypoint attention refiner to enhance inter-node correlation between keypoints. They jointly form a Simple and strong Category-Agnostic Pose Estimator (SCAPE). Experimental results show that SCAPE outperforms prior arts by 2.2 and 1.3 PCK under 1-shot and 5-shot settings with faster inference speed and lighter model capacity, excelling in both accuracy and efficiency. Code and models are available at github.com/tiny-smart/SCAPE.

Keywords: 2D pose estimation · class-agnostic · few-shot

1 Introduction

Category-Agnostic Pose Estimation (CAPE), introduced by Xu *et al.* [34], is a recent emerging topic in pose estimation. It extends conventional category-specific pose estimation [14,24,31,33] and multi-category pose estimation [36] to unseen categories given few image-annotation examples. In their preliminary solution POMNet [34], CAPE is regarded as a similarity matching problem, solved by generating a similarity map for keypoint prediction. Under a similar vein, CapeFormer [29] adopts an additional transformer decoder [18] to iteratively refine matched keypoints in a two-stage manner. Besides applying self-attention blocks to extract image features, their models further involve a series of sophisticated

* Yujia Liang and Zixuan Ye contributed equally. Hao Lu is the corresponding author.

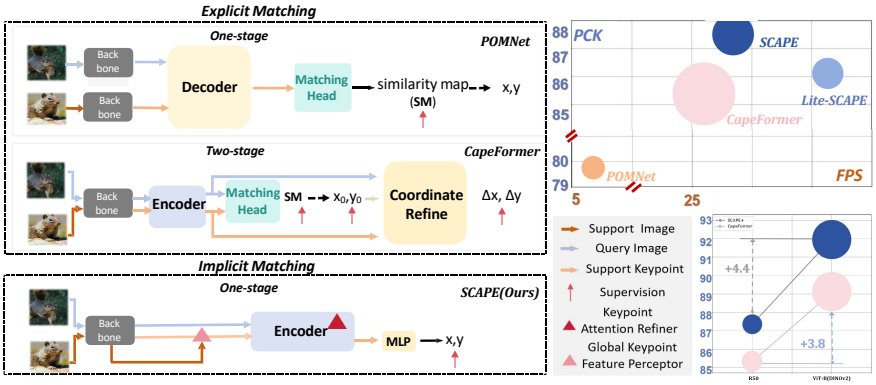

Fig. 1: Comparison with prior arts. (a) POMNet [34] relies on similarity matching to obtain similarity maps and infer keypoint coordinates. (b) CapeFormer [29] presents a two-stage framework, iteratively refining unreliable initial predictions. (c) Our SCAPE employs self-attention for feature interaction and directly regresses keypoints without explicit matching. For better similarity matching, we introduce two modules. The circle size indicates the model parameters (excluding backbone). The y-axis represents the accuracy (PCK), and the x-axis indicates the inference speed (FPS). Our model facilitates the seamless integration of state-of-the-art self-supervised learning techniques for scaling Vision Transformers (ViTs).

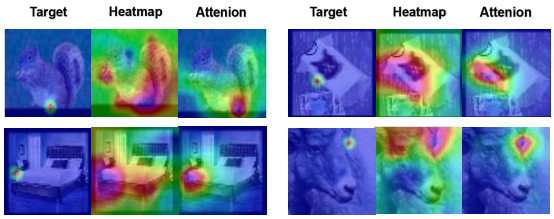

Fig. 2: Implicit attention map is closer to ground truth than explicit similarity map. The second column represents the similarity map obtained from CapeFormer, and the third column denotes the final layer of attention map between support keypoints and query image in the first stage of CapeFormer.

modules for feature matching. Their pipelines are shown in Fig. 1. Actually, there is a common sense that transformer attention is already a strong operator for similarity calculation and matching, suggesting redundancy in previous models and matching policies. Per Fig. 2, an initial exploration of CapeFormer [29] shows that the attention maps exhibit even closer responses to ground truth before extra stage-two post-processing. This reveals that extra complex blocks or additional supervision may be redundant or suboptimal.

We thus streamline the pipeline, and build a simple baseline including several self-attention layers and an MLP head, where the output coordinates are directly supervised. This simplification makes it easier to improve the localization accu-

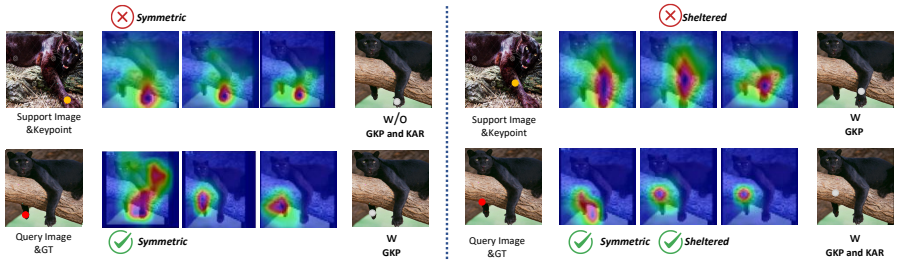

Fig. 3: Visualization of the last three attention maps between keypoints and query image and result. (left) The query target is the right foot; however, the attention easily focuses on the left foot with a similar appearance, leading to inaccurate estimation. GKP injects global information for the support keypoints, equipping it with relative positional information to distinguish left and right keypoints. (right) However, GKP struggles when facing shelter. By modeling the correlation between keypoints, we enable the inference from visible to invisible. Despite the right knee is sheltered, the model can infer its correct position by locating the right foot.

racy of CAPE, by only concerning about the attention quality in self-attention. On this basis, we then share insights to fit the internal attention maps right for CAPE. We begin with two representative hard cases in CAPE: symmetric keypoints and sheltered keypoints. i) Symmetric keypoints, due to their similar appearance, are difficult to be distinguished by the model. As shown in Fig. 3, the attention mechanism struggles to capture correct keypoint positions and turns to other confusing keypoint choices. To enhance the discrimination ability for symmetric keypoints, we propose a Global Keypoint feature Perceptor (GKP) module to fuse the global information from the support image into support keypoint. By leveraging the surrounding information, the attention map can easily identify keypoints that have symmetric counterparts or similar neighbor keypoints. ii) Sheltered keypoints are another difficult case, where attention fails to find the invisible target keypoint. A natural solution is to infer its position with the assistance of other visible correlated keypoints. Therefore, we attempt to address this issue by establishing the intrinsic correlation among all keypoints, whose necessity and benefit have also been verified in category-specific pose estimation (CSPE) [16, 17, 22]. We thus introduce the Keypoint Attention Refiner (KAR) module, which infuses the intrinsic relationship among keypoints as a refinement for the attention map. KAR effectively enables the reasoning power from visible to sheltered keypoints, as shown in Fig. 3.

The simplified transformer architecture, the direct MLP-style regression, and the GKP and KAR modules jointly constitute our Simple and strong Category-Agnostic Pose Estimator (SCAPE). The results on the MP-100 dataset [34] indicate that, SCAPE invites 2.2 and 1.3 PCK improvements under the 1-shot and the 5-shot setting, respectively, with ResNet-50 backbone. With ViT backbones and strong pretraining, we observe significantly improved CAPE metrics, where SCAPE further reaches 91.95 and 93.98 PCK metric under the 1-shot and

the 5-shot setting, with more pronounced relative improvement compared with the state of the art. Excluding the backbone, SCAPE uses only 51% parameters compared with CapeFormer [29]. Ablation studies are also conducted to justify our propositions. In a nutshell, sufficient evidence has been provided to show that SCAPE is a better baseline for CAPE.

2 Related Work

2.1 Category-Agnostic Pose Estimation

Category-Specific Pose Estimation (CSPE) has been the focus of the field for decades [14, 15, 24, 26, 30, 31, 33, 35]. Yet, estimating pose on new categories often means model redesign and retraining. Recently, the task of CAPE is introduced [34], and two baseline approaches POMNet [34] and Capeformer [29] are also proposed, enabling pose estimation on new categories with few exemplar images. The difference against CSPE is that the two approaches both focus on similarity metrics: keypoints are obtained by comparing the similarity between the query and support features. POMNet concatenates keypoint tokens with the query image, measuring similarity using a window to generate a similarity map, indirectly predicting coordinates from the map. Capeformer extends this approach into a two-stage framework, introducing a second stage to iteratively refine the unreliable matches from the first stage. This two-stage framework significantly enhances accuracy, but it is computationally expensive and adds 6.4× more parameters than POMNet (excluding the backbone). Our first goal is to reduce the computational burden in both model capacity and computation, without hurting accuracy.

2.2 Similarity Matching

Similarity matching is widely used in vision tasks like object detection [3, 4] and tracking [1], formulated as a problem of template matching. CAPE can also be viewed as executing similarity matching by aligning sparse support keypoints to the most similar keypoints in the target image. Semantic correspondence have typically explored point-level similarity matching using matching heads. However, challenges arise with models like SuperGlue [27] and SCOT [19], indicating that relying solely on matching may lead to multiple responses in similarity maps, thus requiring additional post-processing modules like optimal transport for refinement. In the sparse setting of CAPE, CapeFormer [29] uses a two-stage process to optimize matching results. With the rise of transformer, self-attention shows strong implicit matching capabilities. Recent studies in semantic correspondence has shifted towards transformer-based implicit matching, as seen in COTR [11] and ACTR [32], which uses transformers for feature matching and fusion, without post-processing. However, sparse point semantic matching in the context of CAPE remains underexplored. This work focuses on leveraging implicit matching abilities of transformer for accurate CAPE.

2.3 Keypoint Correlation in Pose Estimation

In CSPE, such as in human pose estimation ,the correlation among keypoints has been proven to be a useful cue. Previous CNN-based models like ContextPose [22] have developed modules to model the keypoint correlation. Transformer-based approaches [9, 16, 28] also reveal the implicit expression of keypoint correlation within them. These approaches treat keypoints as learnable tokens, akin to DETR [2], allowing them to learn a universal representation for specific keypoints. The prior correlation among keypoints is also captured. In this way, these transformer models can easily capture correlations through attention maps between keypoints. However, keypoint correlation has yet to be exploited in CAPE. Inspired by the good practices in CSPE, we attempt to find a way to model and exploit keypoint correlation in CAPE.

3 Simple and Strong Category-Agnostic Pose Estimator

Here we present our SCAPE. We first review the framework and then reveal our explorations and understanding of the model and task. These explorations further motivate us to devise two enhancements specifically designed for CAPE.

3.1 Overall Framework

Previous methods [29, 34] interact features and measure the similarity between the query feature $F_q \in \mathbb{R}^{n\times c}$ and support keypoints $F_s \in \mathbb{R}^{k\times c}$ by concatenating them and feed into the convolutional layer to obtain a similarity map $S \in \mathbb{R}^{k\times n}$, then indirectly identifying the target keypoints $P \in \mathbb{R}^{k\times 2}$. POMNet [34] uses a decoder θ_M to infer similarity maps from combined support keypoints and query features, *i.e.*, $S = \theta_M\ (F_s, F_q)$. The keypoint coordinates are obtained by searching the peaks such that $P = \arg\max(S)$. This matching head accounts for 70% of the overall training memory consumption. CapeFormer [29] simplifies this by taking the inner product of F_q and F_s after mapping. Both methods compare F_s and F_q to obtain similarity maps with supervision, categorized as similarity-explicit matching heads. CapeFormer uses a two-stage paradigm, the inferred coordinates P_0 are considered the initial coordinates. Then P_0 is refined at the second stage with offsets by $P^{l+1} = P^l + \texttt{MLP}(F_s^{l+1})$. Our SCAPE adheres to the one-stage paradigm, but discards explicit similarity matching, and uses pure self-attention for implicit similarity matching and MLP direct coordinate regression. To enhance implicit matching in the transformer, we use GKP to enrich the semantics of F_s, reducing subsequent matching stress. We employ KAR to establish strong relations between keypoints, facilitating mutual assistance in point prediction. The technical pipeline of SCAPE model is shown in Fig. 4. We visualize the cosine similarity between the left foot keypoint and the support image (values less than 0.6 are masked) before and after the Global Keypoint Feature Perceptor (GKP) operation. After GKP, the similarity map of support

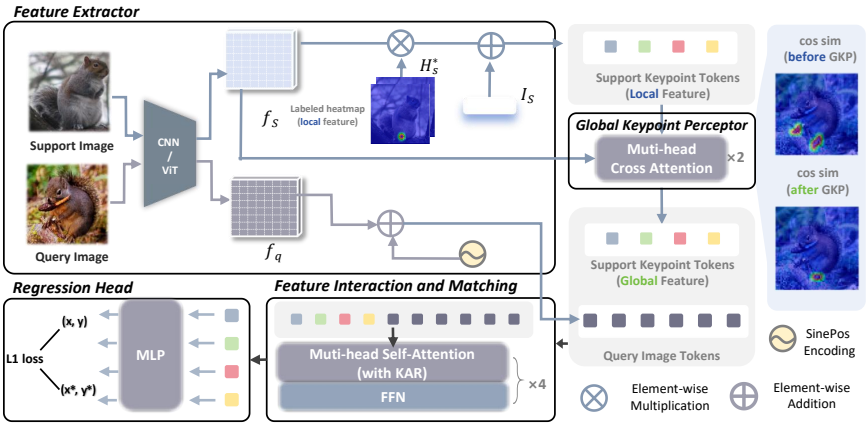

Fig. 4: Technical pipeline of SCAPE. SCAPE consists of four modules: feature extractor, global keypoint feature perceptor, feature interactor, and regression head. The feature extractor is identical to the poir method. After processing support and query features through the backbone, support keypoint tokens are created by a weighted sum of the labeled keypoint heatmap and support features. Support keypoints (local) are then combined with a support keypoint identifier (form Capeformer), while query image tokens are formed from the query features with positional embedding. Support keypoints feed in Global Keypoint Perceptor to cross-attend the support image to obtain global support keypoints. Next, an interaction module composed of multi-head refined self-attention (MHRSA) conduct the interaction between the keypoint and query features. And the Keypoint Attention Refiner (KAR) is inserted into each self-attention stage to refine the attention maps among keypoints. Finally, a simple MLP head are used for support keypoint to regress keypoint coordinates.

keypoints (left foot), fused with global information, exhibits reduced responses compared to symmetric points. This reduction can be considered as alleviating subsequent matching pressure.

3.2 Explicit Matching Is not Necessary

Previous methods generate similarity maps and supervise them in map level, we refer to this operation as explicit matching. We first study the feature interactor of CapeFormer [29] (all other implementation details stay fixed). As shown in Fig. 2, its similarity map has multiple responses, and the peaks do not converge. Yet, we observe that the final attention map of the feature interactor between support keypoints and query features is an effective similarity map, with clearer peaks. This suggests that the implicit similarity within self-attention seems a better substitution for explicit similarity. We surmise explicit matching has the following drawbacks: (i) Explicit matching methods, such as generating similarity maps and supervising on map level can lead to overfitting and increased matching difficulty, and (ii) it introduces extra blocks or modules that increase the

Table 1: Comparison with prior CAPE methods. 'S-map' is the similarity map, and 'coord' means direct coordinate regression, 'offset' is supervise layer by layer.

	stage	matching form		surpervison			end-to-end
		explicit	implicit	s-map	coord	offset	
POMNet	one	✓		✓			
CapeFormer	two	✓		✓	✓	✓	
SCAPE	one		✓		✓		✓

complexity and bring expensive computation. We thus discard explicit similarity matching in CapeFormer, relying solely on the inherently implicit matching mechanism of attention, and also try direct coordinate regression from F_s,

$$P = \texttt{MLP}(F_s)\,. \tag{1}$$

CapeFormer incorporates multiple loss functions, yet we solely supervise the coordinates P output by the MLP, as indicated in Table 1. Subsequently, we employ ℓ_1 loss to compute loss. Such simple modifications lead us to a promising PCK of 88.6 on the split1 of the MP-100 dataset [34].

In addition, previous methods use DETR-Decoder for feature interaction, which we consider as a rough utilization of the DETR. Unlike the previous DETR-Decoder, which only updated F_s, our approach concatenates F_s and F_q and feeds them into self-attention, enabling simultaneous updates to both F_s and F_q. This interpretation inspires us to replace the previous DETR-Decoder with self-attention layers for feature interaction. Therefore, our feature interaction, as shown in Fig. 4, concatenates F_s and F_q, sending them into the fully self-attention Encoder for simultaneous updates. By doing so, we not only reduce the parameters but also improve accuracy to 89.1 (+0.5).

In previous work, F_s and F_q possess distinct additional encodings. F_s encompasses keypoint identifiers to differentiate between keypoints, while F_q contains positional encoding. Using the same query and key weights for F_s and F_q is inappropriate, we therefore choose to have unshared projection weights for them. Therefore, in self-attention: $K_s = W_{K1}F_s$, $K_q = W_{K2}F_q$, $K = \texttt{cat}(K_s, K_q)$, $Q_s = W_{Q1}F_s$, $Q_q = W_{Q2}F_q$, $Q = \texttt{cat}\ (Q_s, Q_q)$. The PCK further reaches 89.8 (+0.7).

3.3 Global Keypoint Feature Perceptor

Previous approaches initialize the support keypoints F_s with only local features, *i.e.*, extracting feature tokens from the support image centered on support ground-truth keypoints with Gaussian kernels. The extracted F_s lacks global context, which hinders subsequent matching. As shown in Fig. 3, the attention process can not distinguish similar keypoints, leading to predictions shifting onto neighboring keypoints.To address this, we further introduce Global Keypoint Feature Perceptor (GKP) that aims to infuse the global and context information from the support image to enrich the support keypoint representation. A

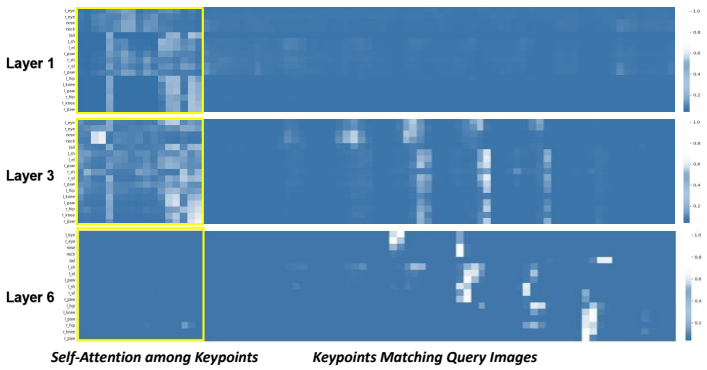

Fig. 5: Visualization of the attention maps of support keypoints. Initially, due to significant disparities between support and query image, accurate matching is challenging, the attention process focuses on self-attention among keypoints (with yellow highlight) to build contextual information. Later attention leans towards implicit matching between keypoints to query images.

cross-attention block where support keypoint features F_s (query) cross-attend the support image enables this fusion process. Note that the introduction of GKP makes the structure more simple. In practice, the first two self-attention blocks are replaced by GKP. This replacement reduces both parameter count and matching complexity, leading to improved performance of 90.3 (+0.5) and accelerated model convergence.

3.4 Keypoint Attenion Refiner

As shown in Fig. 5, in the early attention blocks, due to the different feature distributions, the attention stage concentrates on establishing the correlation among keypoints. As the number of layers increases, attention gradually shifts towards matching the query image. We aim to investigate the effectiveness of early-stage modeling of correlations among keypoints, so we mask the self-attention among keypoints (highlighted in yellow) and observe the PCK drop by -1. We argue that modeling keypoint correlation matters in CAPE.

However, in Fig. 5, We find attention maps among support keypoints show meaningless patterns, appearing to be random. However, transformer-based human pose estimation [16] suggests that the attention among keypoints tends to be highly regular, showing certain correlation among keypoints. The specific keypoint tokens serve as learnable parameters [9, 16]. As shown in Fig. 6, data-driven learning tokens establish meaningful prior keypoint correlation (initial keypoint similarity) with other keypoints, enabling subsequent transformers to easily model reasonable keypoint correlation (attention map). In CAPE, due to unknown categories, the keypoint correlation in CAPE is difficult to model accurately by the transformer, thus introducing noise into the attention map of

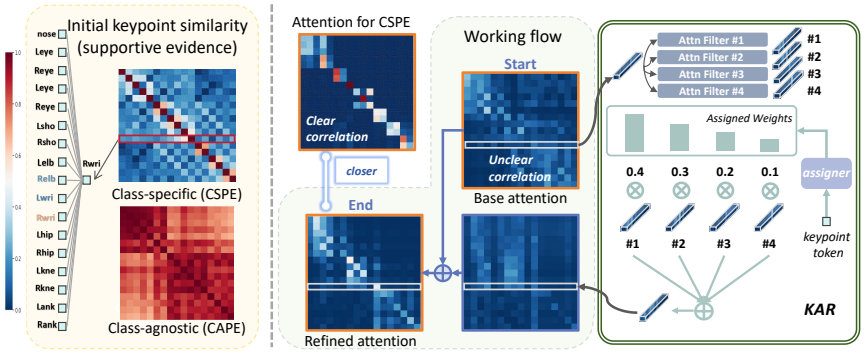

Fig. 6: Attention map modulated by KAR exhibits a clearer correlation. The inner product (use cosine similarity with the range of $[0, 1]$ and no row normalization) among initial keypoint tokens represents the prior keypoint correlation. In CSPE, such prior correlation is used as adjacency and symmetry constraints (strongly correlated with the right wrist are the left wrist and right arm), and transformers can easily model keypoint correlation (attention map) from these priors. And this correlation will aid keypoints in better locating each other. In CAPE, the unclear definition of prior correlation leads to noisy attention maps. KAR uses attention filters to filter out the noise of the base attention map and weights the output through a support keypoint weight assigner. The combined output with base attention yields a refined attention map, closer to the attention in CSPE.

keypoints. We want to strengthen the correlation between keypoints by refining this attention map in self-attention (Encoder) blocks.

Firstly, we introduce the Attention Filter (`AF`) to filter out noise. We set an attention filter, composed of a ReLU layer followed by an MLP, to filter out the undesired parts of the current base attention map ($\mathcal{A}$) and thus make the attention weights more distinct. The ReLU layer mainly charges the filtering, and the MLP re-models these correlations. The positive output signifies important node information, while unimportant details are filtered out. In this way, the PCK further reaches 90.9 (+0.6) on split1. The filter can be written as:

$$\texttt{AF}(\mathcal{A}) = \texttt{MLP}(\texttt{ReLU}(\mathcal{A}))\,. \tag{2}$$

Since keypoints of different categories may benefit from distinct structural data propagation, we further introduce the Keypoint Attention Refiner (KAR). KAR incorporates multiple keypoint filters to enhance the modeling of structural details. It processes support keypoints F_s through a keypoint weight assigner (`Assign`) to determine the weights of different keypoint filters, and one can observe that similar tokens (dog leg and cat leg) possess similar weights (Per supplementary materials). Formally, KAR is defined by:

$$\texttt{KAR}(\mathcal{A}) = \sum_{i=1}^{n(n=4)} \texttt{Assign}_i(F_s)\texttt{AF}_i(\mathcal{A})\,, \tag{3}$$

where $\texttt{Assign}_\texttt{i}$ performs a linear projection on F_s by a weight matrix $W_{\texttt{Assign}}$, with dropout and layer normalization within $\texttt{Assign}_\texttt{i}$ to prevent overfitting. Then, a softmax function is applied for another normalization, defined by

$$\texttt{Assign}(F_s) = \texttt{softmax}(\texttt{layernorm}((\texttt{dropout}(F_s W_{\texttt{assign}}))), \tag{4}$$

The output of KAR is summed with the base attention map $\mathcal{A}$ to yield the refined attention $\mathcal{A}_{\texttt{refined}}$

$$\mathcal{A}_{\texttt{refined}} = \texttt{softmax}(\mathcal{A} + \texttt{KAP}(\mathcal{A})). \tag{5}$$

The refined attention, as shown in Fig. 6, shows clearer keypoint correlation. By collaborating with multiple **AF**, we further advance PCK to 91.6 (+0.7). In addition, previous methods removed [29] support keypoint identifier led to a significant performance decrease. However, adding the KAR module and then removing I_s resulted in a slight decrease.

4 Experiments

4.1 Implementation Details

Dataset. We use the MP-100 dataset provided by Xu *et al*. [34]. Being the first large-scale dataset for CAPE, the MP-100 dataset consists of 100 object categories and contains over 20,000 instances.
Data Pre-processing. Following POMNet [34], we divide the data into 5 splits, each of which includes 100 categories, among which 70 for training, 10 for validation, and 20 for testing, ensuring no overlap between categories. We apply the same data pre-processing as POMNet.
Metric. PCK (Probability of Correct Keypoint) is a widely used metric for evaluating pose estimation algorithms. Like POMNet, we use PCK as the quantitative metric and set the threshold to 0.2 for all categories, and the mean of PCK across 5 splits is also reported. To address the limitation of PCK, we further report AUC [12] and NME [6] to evaluate the localization performance. Please refer to the supplementary material for additional details.
Network Architecture. To validate the adaptation of our approach across different backbones, we choose: 1) ResNet-50 [10], 2) ViT-B [7] and Swin-S [20] pretrained on ImageNet-1K [5], and 3) ViT-S and ViT-B trained with DINOv2 [25]. Like CapeFormer [29], both the support and the query image share the same feature extractor. Then, We first uses 2 Global Keypoint Feature Perceptor (cross-attention), followed by 4 Feature Interactor (self-attention), termed SCAPE. To showcase the effectiveness of our approach, we streamline the model by introducing **Lite-SCAPE**, a baseline with only 3 attention blocks (1 GKP and 2 feature interactors). The parameter for Lite-SCAPE is provided below, and additional performance details can be found in the supplementary material.
Training Details. Following POMNet, we apply the Adam optimizer [13] with a batch size of 16. The model is trained for 180 epochs with an initial learning rate of $2e^{-4}$, while in pervious method [34] [29] the totol epoch is 210. The

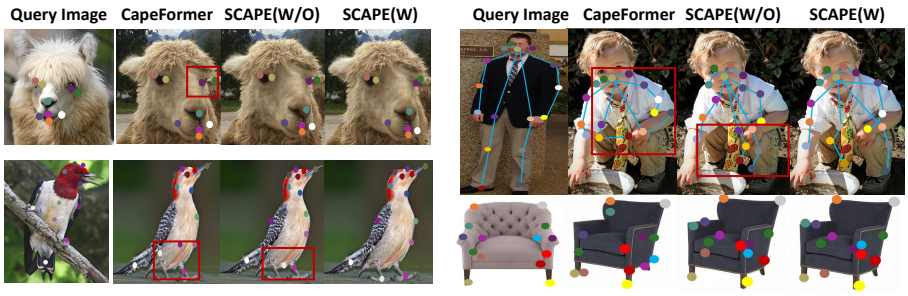

Fig. 7: Visual results of CapeFormer, SCAPE (W/O KAR and GKP) and SCAPE(W KAR and GKP) on the MP-100 dataset. Red boxes indicate incorrect estimates.

Table 2: Comparison with state-of-the-art approaches on the MP-100 dateset under the 1 and 5-shot setting. Best performance is in **boldface**.

1-shot	Backbone	split1	split2	split3	split4	split5	Mean
ProtoNet [21]	R50	46.05	40.84	49.13	43.34	44.54	44.78
MAML [8]	R50	68.14	54.72	64.19	63.24	57.20	61.50
Fine-tune [23]	R50	70.60	57.04	66.06	65.00	59.20	63.58
POMNet [34]	R50	84.23	78.25	78.17	78.68	79.17	79.70
CapeFormer [29]	R50	89.45	84.88	83.59	83.53	85.09	85.31
SCAPE (Ours)	R50	**91.67**	**86.87**	**87.29**	**85.01**	**86.92**	**87.55**

5-shot	Backbone	split1	split2	split3	split4	split5	Mean
ProtoNet [21]	R50	60.31	53.51	49.13	43.34	44.54	44.78
MAML [8]	R50	68.14	54.72	64.19	63.24	57.20	61.50
Fine-tune [23]	R50	70.60	57.84	66.76	66.53	60.24	64.61
POMNet [34]	R50	84.72	79.61	78.00	80.38	80.85	80.71
CapeFormer [29]	R50	91.94	88.92	89.40	88.01	88.25	89.30
SCAPE (Ours)	R50	**93.42**	**89.91**	**90.61**	**89.44**	**89.95**	**90.66**

learning rate is multiplied by 0.1 at the 140-th and the 170-th epoch (cosine annealing can achieve the same effect), respectively. We solely employ ℓ_1 loss to supervise the regression of 2D coordinates.

4.2 Results on the MP-100 Dataset

We first compare the performance to show the effectiveness and efficiency of our approach. Following previous methods, we test our model on MP-100 dataset.
Effectiveness. As shown in Table 2, SCAPE outperforms other approaches on all splits. Compared with CapeFormer, using the same R50 as the backbone, SCAPE achieves an improvement of +2.2 and +1.3 under the 1-shot and the 5-shot setting respectively. Qualitative results are shown in Fig. 7. Additionally, our simple framework is highly compatible with transformer-based backbones. In Table 3, To underscore our high compatibility, we compare SCAPE with CapeFormer using a DINOv2-pretrained VIT-B approach, achieving improvements of +4.4 and +3.4, notably surpassing the performance gap on R50. Note

Table 3: Performance across different transformer backbones. We report the mean across 5 splits on the MP-100 dataset. The improvement relative to R50 as the backbone is highlighted in red.

Method	Backbone	1-shot Mean	5-shot Mean
SCAPE(Ours)	ViT-B	88.19	91.53
SCAPE(Ours)	Swin-S	87.93	92.01
SCAPE(Ours)	ViT-S (DINOv2)	90.74	92.65
CapeFormer	ViT-B (DINOv2)	89.11(+3.8)	91.91(+2.6)
SCAPE(Ours)	ViT-B (DINOv2)	**91.95**(+4.4)	**93.98**(+3.4)

Table 4: Efficiency comparison are tested on $256 \times 256 \times 3$ input on RTX 3090 The experiments are conducted under the 1-shot setting of the MP-100 dataset. R50 is employed as backbone. ΔParams indicates the model parameters (excluding backbone).

Method	Attn Blocks	GFLOPs	ΔParams (M)	Mem (G)	FPS	PCK Mean
POMNet	6	38.01	+1.19	13.8*2	6.80	79.70
CapeFormer	9	23.68	+7.63	7.8	26.09	85.31
Lite-SCAPE(ours)	3	22.20	+1.95	6.0	36.89	86.13
SCAPE(ours)	6	22.81	+3.88	6.3	29.43	87.33

Table 5: Cross super-category pose estimation under the 1-shot setting.

Method	Human Body	Human face	Vehicle	Furniture
ProtoNet	37.61	57.80	28.35	42.64
MAML	51.93	25.72	17.68	20.09
Fine-tune	52.11	25.53	17.46	20.76
POMNet	73.82	79.63	34.92	47.27
CapeFormer	83.44	80.96	45.40	52.49
SCAPE(ours)	**84.24**	**85.98**	**45.61**	**54.13**

that, SCAPE with ViT-S even outperforms CapeFormer ViT-B. This reveals the benefit of a simple end-to-end architecture for CAPE.

Efficiency. From Table 4, Here we provide more evidence to show that SCAPE achieves a good trade-off between efficiency and performance, further enhancing model performance without much workload. Note that, our light version Lite-SCAPE (with only half blocks of SCAPE and 1/3 of CapeFormer) still outperforms CapeFormer by +0.8 PCK, with reduced additional parameters by 75%, boosting the inference speed by 39%. Our simplicity is not only evident in our design but also reflected in practical efficiency.

4.3 Cross Super-Category Generalization

Previous experiments suggest limited generalization ability, meaning the capacity to predict specific categories heavily relies on knowledge acquired from those

categories. To validate the generalization of our model, we conduct a cross super-category experiment, similar to previous methods. As shown in Table 5. ours exhibits good generalization ability, yielding best performance.

4.4 Ablation Study

Here we verify our design choices: i) the comparison between the regression head and the matching head, along with supervision signal choices; ii) the design choice of KAR; iii) the effectiveness of individual components in SCAPE; iv) the number of attention blocks and the proportion of attention layers used by GKP and feature interaction. All experiments are conducted under the 1-shot setting on split1. And the design in light blue is our choice.

Matching and Regression. Our approach relies on implicit self-attention matching to directly regress coordinates. This is different from the explicit matching used in previous methods that indirectly obtain coordinates from similarity maps. We justify two main differences: 1) the form of matching (implicit vs. explicit), and 2) the form of supervision signal or get keypoints (coordinates vs regressed similarity maps). Results are shown in Table 6. For 1), We use L1 (Line 1) as a baseline which applies the additional matching head and map-level supervision on SCAPE (w/o GKP and KAR). L2 directly regresses similarity maps from the final keypoint tokens and outperforms L1, which shows that implicit matching proves superiority over explicit matching. For 2), we compare L2 and L3, and the results demonstrate that direct regression of coordinates outperforms regressing similarity maps.

Table 6: Validation of matching head and direct regression head.

Paradigm	Result Form	PCK
Matching	similarity map	86.3
Regression	similarity map	88.2
Regression	coordinate	89.1

Table 7: The design of multiple Attention Filters in KAR.

AF	hidden-dim	PCK
1	50	90.5
1	200	90.3
4	50	91.2

Design of Keypoint Attention Refiner (KAR). We introduce to employ multiple(4) Attention Filters to refine the base attention simultaneously. This design can further tackle the varying categories for CAPE (PCK +0.7). To figure out whether the improvement comes from the increased parameters or the multiple aspects modulation of node relationships. We augment a single Attention Filter by increasing its hidden layer dimensions to n times the original, to align with parameters of n Attention Filters for a fair comparison. Specifically, the linear transformation changes from (100-50-100) to (100-200-100), with corresponding results presented in Table 7. Surprisingly, increasing the hidden layer dimensions results in a performance drop of -0.1. On the contrary, increasing the number of Attention Filter can boost the performance, which suggests the improvement comes from better refinement to attention map but not parameters.

Performance Gains of SCAPE Components. In Table 8, the comparison between S1 and S2 indicates that the feature interaction of the full encoder outperforms DETR (used by CapeFormer) by 0.5. By comparing S2 with S3, unsharing query and key weights in self-attention between keypoint tokens and query image leads to a performance gain of 0.7. By contrasting S3 with S5, KAR shows an improvement of +1.4. By comparing S3 with S4, GKP demonstrates an improvement of +1.0, and combining these two modules proposed by us further enhances the performance.

Performance of Different Attention Blocks. As shown in Table 9, employing 6-layer attention blocks, involving 2 layers of GKP to refine initial support keypoint tokens and 4 layers of the self-attention feature interactor, exhibits the best performance. Additionally, increasing the number of attention blocks also improves performance.

Further Exploration. We address some uncertainties in the supplementary materials. (a) Our proposed GKP and KAR are tailored to the CAPE task, and we assess their generalization across other CAPE models. (b) Some existing large models exhibit class-agnostic matching and even multi-task capabilities, it would be interesting to evaluate whether CAPE-specific models are even needed.

Table 8: Performance gains of SCAPE components.

	Interactor				PCK
	Form	Unshare Q/K	KAR	GKP	
S1	DETR	–			88.6
S2	Encoder				89.1
S3	Encoder	✓			89.8
S4	Encoder	✓		✓	90.8
S5	Encoder	✓	✓		91.2
S6	Encoder	✓	✓	✓	91.9

Table 9: Performance of proportion occupied by GKP and feature fusion layers and number of attention blocks.

Proportions	GKP	Interactor	PCK
All Interactor	0	6	91.2
1:1	3	3	91.2
1:2	2	4	91.6
Lite-SCAPE	1	2	90.4
SCAPE	2	4	91.6

5 Conclusion and Limitaion

Conclusion We propose a simple, strong, and straightforward CAPE baseline. Initially, we rely solely on implicit attention matching, discarding complex explicit matching heads, and directly regress coordinates with a simple MLP head. Subsequently, to enhance attention quality in CAPE, we introduce GKP to equip support keypoints with global semantics.Then we present KAR to establish correlations among keypoints to enable the inference from related keypoints. SCAPE surpasses the state-of-the-art CAPE models in accuracy and efficiency.

Limitation We test the applicability of SCAPE on multiple instance scenarios across categories and even across domains, as per the supplementary materials. The current CAPE only finds one-to-one correspondence. We believe that future CAPE could explore multi-instance scenarios.

Acknowledgement This work is supported by Hubei Provincial Natural Science Foundation of China under Grant No. 2024AFB566.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. pp. 850–865. Springer (2016)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229. Springer (2020)
3. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: CVPR. pp. 2147–2156 (2016)
4. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. vol. 28 (2015)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
6. Dong, X., Yu, S.I., Weng, X., Wei, S.E., Yang, Y., Sheikh, Y.: Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In: CVPR. pp. 360–368 (2018)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. pp. 1126–1135. PMLR (2017)
9. Geng, Z., Sun, K., Xiao, B., Zhang, Z., Wang, J.: Bottom-up human pose estimation via disentangled keypoint regression. In: CVPR. pp. 14676–14686 (2021)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A., Yi, K.M.: Cotr: Correspondence transformer for matching across images. In: ICCV. pp. 6207–6217 (2021)
12. Khan, M.H., McDonagh, J., Tzimiropoulos, G.: Synergy between face alignment and tracking via discriminative global consensus optimization. In: ICCV. pp. 3811–3819. IEEE (2017)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014)
14. Labuguen, R., Matsumoto, J., Negrete, S.B., Nishimaru, H., Nishijo, H., Takada, M., Go, Y., Inoue, K.i., Shibata, T.: Macaquepose: a novel "in the wild" macaque monkey pose dataset for markerless motion capture. Frontiers in Behavioral Neuroscience **14**, 581154 (2021)
15. Li, S., Li, J., Tang, H., Qian, R., Lin, W.: Atrw: A benchmark for amur tiger re-identification in the wild. arxiv 2019. arXiv preprint arXiv:1906.05586
16. Li, Y., Zhang, S., Wang, Z., Yang, S., Yang, W., Xia, S.T., Zhou, E.: Tokenpose: Learning keypoint tokens for human pose estimation. In: ICCV. pp. 11313–11322 (2021)
17. Liu, H., Chen, Q., Tan, Z., Liu, J.J., Wang, J., Su, X., Li, X., Yao, K., Han, J., Ding, E., et al.: Group pose: A simple baseline for end-to-end multi-person pose estimation. In: CVPR. pp. 15029–15038 (2023)
18. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329 (2022)
19. Liu, Y., Zhu, L., Yamada, M., Yang, Y.: Semantic correspondence as an optimal transport problem. In: CVPR. pp. 4463–4472 (2020)
20. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)

21. Lyu, H., Sha, N., Qin, S., Yan, M., Xie, Y., Wang, R.: Advances in neural information processing systems **32** (2019)
22. Ma, X., Su, J., Wang, C., Ci, H., Wang, Y.: Context modeling in 3d human pose estimation: A unified perspective. In: CVPR. pp. 6238–6247 (2021)
23. Nakamura, A., Harada, T.: Revisiting fine-tuning for few-shot learning. arXiv preprint arXiv:1910.00216 (2019)
24. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV. pp. 483–499. Springer (2016)
25. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
26. Reddy, N.D., Vo, M., Narasimhan, S.G.: Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In: CVPR. pp. 1906–1915 (2018)
27. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR. pp. 4938–4947 (2020)
28. Shi, D., Wei, X., Li, L., Ren, Y., Tan, W.: End-to-end multi-person pose estimation with transformers. In: CVPR. pp. 11069–11078 (2022)
29. Shi, M., Huang, Z., Ma, X., Hu, X., Cao, Z.: Matching is not enough: A two-stage framework for category-agnostic pose estimation. In: CVPR. pp. 7308–7317 (2023)
30. Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y., Su, H., Li, H., Yang, R.: Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In: CVPR. pp. 5452–5462 (2019)
31. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR. pp. 5693–5703 (2019)
32. Sun, Y., Zhao, D., Yin, Z., Huang, Y., Gui, T., Zhang, W., Ge, W.: Correspondence transformers with asymmetric feature learning and matching flow super-resolution. In: CVPR. pp. 17787–17796 (2023)
33. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: ECCV. pp. 466–481 (2018)
34. Xu, L., Jin, S., Zeng, W., Liu, W., Qian, C., Ouyang, W., Luo, P., Wang, X.: Pose for everything: Towards category-agnostic pose estimation. In: ECCV. pp. 398–416 (2022)
35. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: Simple vision transformer baselines for human pose estimation (2022)
36. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose+: Vision transformer foundation model for generic body pose estimation (2022)