A. Additional implementation details

During training of MambaMatcher in Tab. 1, we use an effective batch size of 80 by distributing 10 batches to 8 RTX 3090 GPUs. For other comparison and ablative experiments, we run the experiments on a 'small' subset of SPair-71k, which is around 20% the size of the original SPair-71k dataset [39], with varying effective sizes across 2 GPUs. The batch sizes vary because different feature and correlation aggregation schemes required different amount of vRAM. For example, when using FastFormer [53], only 3 batches could fit into a single GPU when training.

Details of soft sampler (Sec. 4.3. Given a source keypoint $\mathbf{k}_s = (x_{k_s}, y_{k_s})$, we define a soft sampler $\mathbf{W}^{\mathbf{k_s}} \in \mathbb{R}^{H \times W}$:

$$\mathbf{W}^{\mathbf{k_s}}(i,j) = \frac{\max(0, \tau - \sqrt{(x_{k_s} - j)^2 + (y_{k_s} - i)^2})}{\sum_{i'j'} \max(0, \tau - \sqrt{(x_{k_s} - j')^2 + (y_{k_s} - i')^2})}, \tag{10}$$

where τ is a distance threshold from the keypoint, and $\sum_{ij} \mathbf{W}^{k_s}(i,j) = 1$. The role of the soft sampler is to sample each transferred keypoint $\hat{\mathbf{P}}(i,j)$ by assigning weights which are inversely proportional to the distance to the keypoint $\mathbf{k}_s$. We can obtain sub-pixel accurate keypoint matches as follows:

$$\hat{\mathbf{k}}_t = \sum_{(i,j) \in H \times W} \hat{\mathbf{P}}(i,j) \mathbf{W}^{k_s}(i,j). \tag{11}$$

We use $\tau = 0.1$ for training, and $\tau = 0.05$ for inference.

Experimental environment. All experiments are run on a machine with an Intel(R) Xeon(R) Gold 6242 CPU, with up to 8 GeForce RTX 3090 GPUs.

B. Additional details of baseline methods

We provide the details of each baseline approach (shown in Table 1 of the main manuscript) in Table 6, which was omitted due to spatial constraints.

Table 6. **Additional details of baseline methods.**

Method	Feature backbone	Supervision	Data augmentation
DHPF, CHM, MMNet, PWarpC-NCNet, NeMF, SCorrSAN	ResNet101	kp-pair	x
TransforMatcher, CATs++, HCCNet, UFC	ResNet101	kp-pair	o
DIFT	SD2.1	None	x
DINO + $\text{SD}_{\text{zero-shot}}$	DINOv2, SD1.5	None	x
DINO + $\text{SD}_{\text{supervised}}$	DINOv2, SD1.5	kp-pair	x
Diffusion Hyperfeatures	SD1.5	None	x
Hedlin et al. [15]	SD1.4	None	x
SD4Match	SD2.1	kp-pair	x
MambaMatcher	DINOv2	kp-pair	o

C. Effect of positional encoding

Currently, we do not explicitly encode the potential spatial relationships between correlation elements during the sorting and processing steps. While spatial relationships are important, our primary goal is to resolve ambiguities by focusing on the most significant correspondences first. Prioritizing high-similarity scores enables the model to establish a strong contextual foundation. Nonetheless, we experiment the effect of fixed 4D sinusoidal encoding and learnable positional embedding to further analyze the effect of *explicitly* encoding spatial relationships in Table 7, where it can be seen that there is no clear visible improvement in performance with the integration of sinusoidal or learned positional embeddings. Investigating more sophisticated methods to integrate spatial context could potentially enhance the model's ability to capture inter-image relationships.

D. Feature backbone / Data augmentation comparison

We provide the results of MambaMatcher when using varying backbones, with or without data augmentation, on SPair-71k for a fairer comparison in Table 8. Noting that PCK@0.05/0.10 for TransforMatcher [20] are 32.4/53.7 with data augmentation, these results show that the similarity-aware selective scan shows enhanced efficacy over multiple layers of additive attention (FastFormers [53]).

Table 7. **Effect of using 4D sinusoidal positional encoding.**

Method	PCK @ 0.05	PCK @ 0.10	PCK @ 0.15
Ours	61.6	77.8	84.3
Ours with Sinusoidal P.E.	61.2	77.8	84.2
Ours with Learnable P.E.	61.5	77.6	84.5

Table 8. **PCK of MambaMatcher on SPair-71k when using varying feature backbones and data augmentation.** We follow the data augmentation scheme used in CATs [5] and TransforMatcher [20]

Backbone	Data aug.	PCK@0.05	PCK@0.10	PCK@0.15
ResNet101	x	38.2	53.3	61.3
ResNet101	o	41.0	58.5	67.4
DINOv2	x	57.9	74.6	81.8
DINOv2	o	61.6	77.8	84.3

E. Statistical significance of performance gap in comparison to FastFormers

We conduct 3 repeated experiments with varying seeds to report the mean and variance of PCK results on the 'small' subset of SPair-71k in Table 9. While the performance gain is not dramatic, MambaMatcher offers advantages in terms of computational overhead (memory, latency) as previously shown in Table 10.

Table 9. **PCK results on SPair-71K over multiple runs** We report the results when using FastFormers in comparison to our similarity-aware selective scan as the correlation aggregation. The experiments were conducted 3 times - the mean and standard variation across the runs are reported. It can be seen that our scheme consistently yields better performances across PCK thresholds.

Method	PCK@0.05	PCK@0.10	PCK@0.15
FastFormers [20] (6 layers)	59.9 ± 0.74	76.9 ± 1.40	83.9 ± 1.27
Mamba + Similarity-aware Selective Scan (Ours)	60.6 ± 0.54	78.2 ± 0.76	85.0 ± 0.86

F. Analysis on efficiency of MambaMatcher

For an intuitive overview, we measure module-wise maximum GPU memory usage and latency in Table 10. The values are cumulative in the order of DINOv2 (feature extraction), feature aggregation, and correlation aggregation. This shows that our design incurs the lowest latency while using less memory and fewer parameters than FastFormer, demonstrating a favorable balance between computational overhead and performance[1].

Table 10. **Memory, Latency and # Params comparison across correlation schemes.** Our scheme strikes the most favorable balance between performance and efficiency.

Module	GPU Memory (GB)	Latency (ms)	# Params
DINOv2 [43]	0.97	10.3	86.6M
Feature aggregation	1.17	12.3	42.5M
Correlation aggregation			
- $\text{Conv4D}_{k=3}$ [38]	1.17	41.0	1.3K
- FastFormers [20] (6 layers)	1.67	28.8	26.0K
- Mamba_{4D} + Similarity-aware Selective Scan (Ours)	1.64	16.4	5.1K

G. FLOPs analysis of MambaMatcher

In the Table 11, we report the FLOPs of MambaMatcher using open-source libraries `ptflops` and `calflops`.

[1] We report the FLOPs of MambaMatcher in Appendix G.

Table 11. **FLOPs of MambaMatcher measured using open-source libraries.**

Module	`ptflops`	`calflops`
DINOv2 [43]	359.32G	358.99G
Feat. agg	2.45T	2.45 T
Conv4D$_{k=3}$ [38]	2.06G	2.06G
FastFormers [20] (6 layers)	43.54G	43.05G
Mamba + Similarity-aware Selective Scan (Ours)	27.54M	3.84G

While FLOPs serve as a standardized measure of computational complexity, we noticed that existing libraries fail to accurately capture the FLOPs of various modules due to technical complexities, *e.g.,* reliance on operations registered as `nn.Modules`. Additionally, certain libraries for measuring FLOPs crash when encountered with hardware-optimized algorithms from `xFormers` [27], which are used in the DINOv2 backbone of our method. Consequently, we believe that this measurement may not be entirely fair or representative of the actual computational overhead and efficiency.

To address this gap, we conduct a theoretical calculation of FLOPs for varying correlation aggregation schemes. We consider an input with dimensions $N \times C = 30^4 \times 16$, consistent with MambaMatcher. We assume the same dimensions for the input and output *i.e.,* $C = C_{\text{in}} = C_{\text{out}}$.

4D convolution, kernel size 3.
$2 \times N \times C_{\text{in}} \times C_{\text{out}} \times k^4 = 33.6$ GFLOPs

Vanilla dot-product attention. Assuming single head, QKV dim = 16.
QKV projection: $3 \times (2 \times N \times C_{\text{in}} \times C_{\text{out}})$
Dot-product: $2 \times (N^2 \times C)$
Softmax: $3 \times (N^2)$
Weighted sum of V: $2 \times (N^2) \times C$
Total = 44.0 TFLOPs

FastFormers (Additive attention). Assuming single head, QKV dim = 16.
QKV projection: $3 \times (2 \times N \times C_{\text{in}} \times C_{\text{out}})$
Softmax and weighted sum: $2 \times (3 \times N + 2 \times N \times C)$
Global vector addition: $2 \times (N \times C)$
Projection: $2 \times N \times C^2$ Total = 1.74 GFLOPs

Mamba: selective state-space machines. Hyperparameters following MambaMatcher.
Input projection: $2 \times 2 \times N \times C_{\text{in}} \times C_{\text{inner}}$
1D convolution: $2 \times C_{\text{inner}} \times k \times C_{\text{inner}}$
Projection to A, B, dt: $2 \times N \times C_{\text{inner}} times (2 \times d_{\text{model}} + 1)$
Selective scan: $9 \times N \times d_{\text{model}} \times d_{\text{state}}$
Element-wise multiplication: $N \times C_{\text{inner}}$
Output projection: $2 \times N \times C_{\text{inner}}\ times C_{\text{in}}$
Total FLOPs = 23.1 GFLOPs

Ours: Selective state-space machines with Similarity-aware Selective Scan. Same as above, but additional sorting overhead. Assuming each comparison and swap operation involves approximately 4 FLOPs:
Sorting: $4 \times (NlogN) = 0.064$GFLOPs
Total FLOPs = 23.2 GFLOPs

Note that the above values ignore many details, including activation, normalization, residual connections, or actual number of aggregation layers used. The above theoretical calculation serve to provide a vague estimate of FLOPs for each scheme. However, we suggest that the number of FLOPs does not directly translate to computational overhead in learning-based methods, as many variables such as parallelism, hardware optimization, and intermediate representations directly impact GPU memory usage and latency.

H. Generalizability of MambaMatcher

Trained on PF-PASCAL, evaluated on PF-WILLOW. We present the results of MambaMatcher on the PF-WILLOW [13] dataset. The PF-WILLOW dataset contains 900 image pairs for testing only and is evaluated using the model trained on the PF-PASCAL dataset. The results are illustrated in Table 12, where it can be seen that while MambaMatcher performs competitively, it does not outperform existing methods. This is unlike our results on the PF-PASCAL and SPair-71k datasets (Table 1), where MambaMatcher outperforms all existing benchmarks. This may be attributed to supervised training, which causes the feature and correlation aggregation layers to be trained

specifically for the training domain. Another possibility is that the Mamba layer lacks generalizability to unseen domains compared to other methods built on convolutional or attention-based layers.

Table 12. **Results of MambaMatcher on the PF-WILLOW dataset.** We perform competitively with existing methods, but do not outperform all existing methods unlike on PF-PASCAL or SPair-71k.

Method	PF-WILLOW			
	$@\alpha_{\text{bbox}}$		$@\alpha_{\text{bbox-kp}}$	
	0.05	0.10	0.05	0.10
DHPF [2020]	49.5	77.6	-	71.0
CHM [2021]	52.7	79.4	-	69.6
CATs++ [2022]	56.7	81.2	47.0	72.6
PWarpC-NCNet [2022]	-	-	48.0	76.2
TransforMatcher [2022]	-	76.0	-	65.3
NeMF [2022]	-	-	60.8	75.0
SCorrSAN [2022]	54.1	80.0	-	-
HCCNet [2024]	-	74.5	-	65.5
UFC [2023]	58.6	81.2	50.4	74.2
DIFT [2023]	58.1	81.2	44.8	68.0
DINO+$\text{SD}_{\text{zero-shot}}$ [2024]	-	-	-	-
DINO+SD_{sup} [2024]	-	-	-	-
Diffusion Hyperfeatures [2024]	-	78.0	-	-
Hedlin et al. [15]	53.0	84.3	-	-
SD4Match [2023]	-	-	52.1	80.4
Ours	56.2	81.1	47.4	72.1

Trained on SPair-71k, evaluated on PF-PASCAL. While we provide the generalization performance of MambaMatcher on the PF-WILLOW dataset in Table 12, we report additional generalization results in Table 13. Results on PF-PASCAL were trained on SPair-71k, and vice versa. The results indicate that while the generalizability of MambaMatcher is not state-of-the-art, it generalizes competitively with other state-of-the-art methods in certain cases, such as being trained on PF-PASCAL and tested on SPair-71k. While domain generalization is advantageous, we suggest that a lack of cross-dataset generalization does not diminish the overall significance of our method. If large-scale datasets for semantic correspondence become available, this problem is likely to be alleviated significantly for all semantic matching methods.

Table 13. **PCK on SPair-71k after being trained on PF-PASCAL.**

Model	PCK@0.05	PCK@0.10	PCK@0.15
CATs [5]	13.6	27.0	-
TransforMatcher [20]	-	30.1	-
SD4Match [30]	27.2	40.9	-
MambaMatcher (Ours)	26.5	40.9	49.1

I. Comparison on the DINOv2 layers used

We show the comparative experiments on the layers if DINOv2 used in this work to validate our use of layers 4-11. The experiments were carried out on the 'small' set of SPair-71k. The results in Tab. 14 shows that better features can be obtained across the depths of the DINOv2 backbone, with the 11th layer token features exhibiting the best performance. Tab. 15 aims to choose the best combination of layers to extract the feature maps from. While the PCK performance improves gracefully as more layers are used, we choose to use layers 4-11 as the performance improvement beyond that becomes diminishing, and using layers 4-11 provides us with a favorable compromise between memory usage (around 70% memory usage compared to using all 0-11 layers) and PCK performance.

Table 14. **Comparison between different layers of the DINOv2 backbone.**

Layers used	SPair-71k (s) @α_{img}		
	0.05	0.10	0.15
0	0.9	3.8	8.2
1	1.5	5.3	11.2
2	1.7	6.1	12.2
3	4.2	11.1	18.8
4	7.3	16.1	24.6
5	10.2	20.6	29.8
6	13.1	23.6	31.8
7	17.5	29.8	39.0
8	20.7	35.2	45.7
9	23.9	40.3	51.5
10	25.2	42.5	54.1
11	25.2	43.1	55.7

Table 15. **Comparison between different layers combinations of the DINOv2 backbone**.

Layers used	SPair-71k (s) @α_{img}		
	0.05	0.10	0.15
11	25.2	43.1	55.7
10-11	29.2	46.4	56.8
9-11	28.9	46.7	58.0
8-11	29.6	47.4	58.3
7-11	30.4	48.5	58.8
6-11	30.8	48.4	58.7
5-11	30.9	48.4	58.6
4-11	30.8	48.6	59.0
3-11	31.0	48.7	59.0
2-11	31.2	48.9	58.7
1-11	31.4	48.9	58.8
0-11	31.4	48.9	58.7

J. Comparison on the layers used for sorting

Currently, when performing multi-level correlation aggregation via the Similarity-Aware Selective Scan, we sort the correlation sequence based on the similarity scores from the final correlation map (the $2L$-th level, Sec. 4.2). We compare the results when using a different standard for sorting the correlation sequence in Tab. 16, where we show that our current configuration of using the $2L$-th level demonstrates the best results.

Table 16. **Effect of using different configurations for sorting the correlation sequence.**

Method	PCK @ 0.05	PCK @ 0.10	PCK @ 0.15
Last layer ($2L$-th, Ours)	61.6	77.8	84.3
Penultimate-layer ($2L$-1th)	61.0	77.7	84.2
Mean across layers	60.9	76.6	83.8

K. PCK per image v.s. PCK per point

While it is conventional to calculate the mean PCK per image (sum of image-wise PCK averaged over the number of images) when reporting the PCK results, some methods confuse this concept with PCK per point (sum of pair-wise PCK averaged over the number of point pairs). Tab. 17 shows the results, where it can be seen that MambaMatcher evaluated using PCK-per-point (denoted as MambaMatcher*) yields higher values in comparison.

L. PCK per category

We present the category-wise PCK in Tab. 18, where it can be seen that MambaMatcher yields the best results overall.

M. Potential when using larger resolutions

In Table 19, we report the GPU memory / latency usage when using different correlation aggregation module at varying image resolutions (thus, varying feature and correlation map resolutions). Note that the memory usage is cumulative i.e., maximum GPU memory usage during the forward run. It can be seen that our similarity-aware selective scan incurs consistently lower GPU memory usage and latency compared to FastFormers. Most notably, the difference in latency is dramatic; the hardware optimizations of Mamba enables the similarity-aware selective scan to be performed with only a small increase in latency even when the image sizes become significantly larger. This further justifies our usage of Mamba, given larger image inputs i.e., consequently, longer correlation sequences.

We report the performance of MambaMatcher on a different set of image resolutions in Table 20. It shows that while using larger image resolutions does result in improved PCK results, there are diminishing returns as the image resolutions becomes larger.

Table 17. **Results of MambaMatcher on PF-PASCAL and SPair-71k datasets.** MambaMatcher outperforms existing baselines on both datasets. MambaMatcher * denotes PCK-per-point metrics, which outperforms MambaMatcher. This shows that PCK-per-point yields higher results in comparison to PCK-per-image.

Method	Image res.	PF-PASCAL @α_{img}			SPair-71k @α_{bbox}			time (*ms*)	memory (GB)
		0.05	0.10	0.15	0.05	0.10	0.15		
DHPF [2020]	240×240	75.7	90.7	95.0	20.9	37.3	47.5	58	1.6
CHM [2021]	240×240	80.1	91.6	94.9	27.2	46.3	57.5	54	1.6
MMNet [2021]	224×320	77.6	89.1	94.3	-	40.9	-	86	-
PWarpC-NCNet [2022]	400×400	79.2	92.1	95.6	31.6	52.0	61.8	-	-
TransforMatcher [2022]	240×240	80.8	91.8	-	32.4	53.7	-	54	1.6
NeMF [2022]	512×512	80.6	93.6	-	34.2	53.6	-	8500	6.3
SCorrSAN [2022]	256×256	81.5	93.3	-	-	55.3	-	28	1.5
HCCNet [2024]	240×240	80.2	92.4	-	35.8	54.8	-	30	2.0
CATs++ [2022]	512×512	84.9	93.8	96.8	40.7	59.8	68.5	-	-
UFC [2023]	512×512	**88.0**	94.8	97.9	48.5	64.4	72.1	-	-
DIFT [2023]	768×768	69.4	84.6	88.1	39.7	52.9	-	-	-
DINO+SD$_{\text{zero-shot}}$ [2024]	840^2 / 512^2	73.0	86.1	91.1	-	64.0	-	-	-
DINO+SD$_{\text{sup}}$ [2024]	840^2 / 512^2	80.9	93.6	96.9	-	74.6	-	-	-
Diffusion Hyperfeatures [2024]	224×224	-	86.7	-	-	64.6	-	6620	-
Hedlin et al. [15]	0.93×ori.	-	-	-	28.9	45.4	-	90k<	-
SD4Match [2023]	768×768	84.4	<u>95.2</u>	<u>97.5</u>	59.5	75.5	-	-	-
MambaMatcher (Ours)	420×420	87.3	<u>95.9</u>	**98.2**	<u>61.6</u>	<u>77.8</u>	<u>84.3</u>	74	2.1
MambaMatcher * (Ours)	420×420	<u>87.6</u>	**96.0**	**98.2**	**63.3**	**79.2**	**85.6**	74	2.1

Table 18. **Category-wise PCK on the SPair-71k dataset**.

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dog	Horse	Motor	Person	Plant	Sheep	Train	TV	All
DINOv2 [2023]	69.9	58.9	86.8	36.9	43.4	42.6	39.3	70.2	37.5	69.0	63.7	68.9	55.1	65.0	33.3	57.8	51.2	31.2	53.9
DIFT [2023]	61.2	53.2	79.5	31.2	45.3	39.8	33.3	77.8	34.7	70.1	51.5	57.2	50.6	41.4	51.9	46.0	67.6	59.5	52.9
SD+DINO [2024]	71.4	59.1	87.3	38.1	51.3	43.3	40.2	77.2	42.3	75.4	63.2	68.8	56.0	66.1	52.8	59.4	63.0	55.1	59.3
NCNet [2018]	17.9	12.2	32.1	11.7	29.0	19.9	16.1	39.2	9.9	23.9	18.8	15.7	17.4	15.9	14.8	9.6	24.2	31.1	20.1
PMNC [2021]	54.1	35.9	74.9	36.5	42.1	48.8	40.0	72.6	21.1	67.6	58.1	50.5	40.1	54.1	43.3	35.7	74.5	59.9	50.4
TransforMatcher [2022]	59.2	39.3	73.0	41.2	52.5	66.3	55.4	67.1	26.1	67.1	56.6	53.2	45.0	39.9	42.1	35.3	75.2	68.6	53.7
SCorrSAN [2022]	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
SD4Match [2023]	75.3	67.4	85.7	64.7	62.9	86.6	76.5	82.6	64.8	86.7	73.0	78.9	70.9	78.3	66.8	64.8	91.5	86.6	<u>75.5</u>
MambaMatcher (Ours)	82.9	61.0	91.9	61.0	62.7	89.9	83.8	89.9	60.6	86.7	81.2	81.6	73.7	79.5	70.0	71.5	93.0	86.4	**77.8**

Table 19. **Efficiency comparison when using larger image resolutions.**

Image res.	Feature res.	Correlation agg.	GPU memory (GB)	latency (ms)
420	30	Ours	1.64	16.4
420	30	FastFormer	1.67	28.8
560	40	Ours	3.25	16.6
560	40	FastFormer	3.16	28.9
700	50	Ours	6.47	17.7
700	50	FastFormer	6.27	55.6

N. Additional visualizations of refined correlation map

We provide additional visualizations of refined correlations in Fig. 7 and Fig. 8. Fig. 3 demonstrates that our refined correlation map can better localize keypoints - Fig. 7 and Fig. 8 aim to provide a deeper insight into this phenomenon. In Fig. 7 and Fig. 8, the top-

Table 20. **PCK results on SPair-71k when using varying image resolutions.**

Image res.	Feature res.	Correlation res.	PCK @ 0.05	PCK @ 0.10	PCK @ 0.15
238	17^2	17^4	26.4	39.7	46.5
420	30^2	30^4	61.6	77.8	84.3
840	60^2	60^4	64.2	78.4	85.2

left images represent an image pair with a ground truth correspondence. The top-right image visualizes the output correlation map from MambaMatcher. This visualization helps illustrate that during the final prediction of $\hat{\mathbf{C}}$ using linear projection, the wrong maps are effectively disregarded, and the accurate maps are primarily weighted for aggregation, resulting in our final accurate correlation map.

O. Correlation robustness analysis

We aim to provide insights into how the model's correlation refinement process depends on high-confidence correspondences, by replacing the top-$k\%$ of the correlation scores with zeros in Fig. 9. It shows that the higher the k, *i.e.*, more high-confidence correlation values are removed, the localization performance degrades more. This evidences that the correlation refinement process relies heavily on the strongest initial correspondences to produce accurate final predictions.

P. Cumulative contribution analysis of each correlation state

In Fig. 10, we visualize the cumulative contribution analysis of each correlation state, *i.e.*, each entry of the correlation sequence, depending on how we order the correlation sequence. We use integrated gradients to compute attribution weights for each correlation token, in order to reveal how quickly the model accumulates useful evidence when tokens are ordered by different criteria. The cumulative curves show the proportion of total attribution mass accumulated as a function of token fraction, with Area Under Curve (AUC) metrics quantifying concentration efficiency. Higher AUC values indicate that important evidence is concentrated in fewer tokens, while lower values suggest more distributed attribution. This visualization illustrates that traversing the tokens in the descending order of correlation scores achieves higher concentration of useful evidence compared to ascending or random orderings, demonstrating that our refinement process effectively leverages high-confidence initial correspondences.

Q. Additional qualitative results

We provide additional qualitative results in Fig. 11.

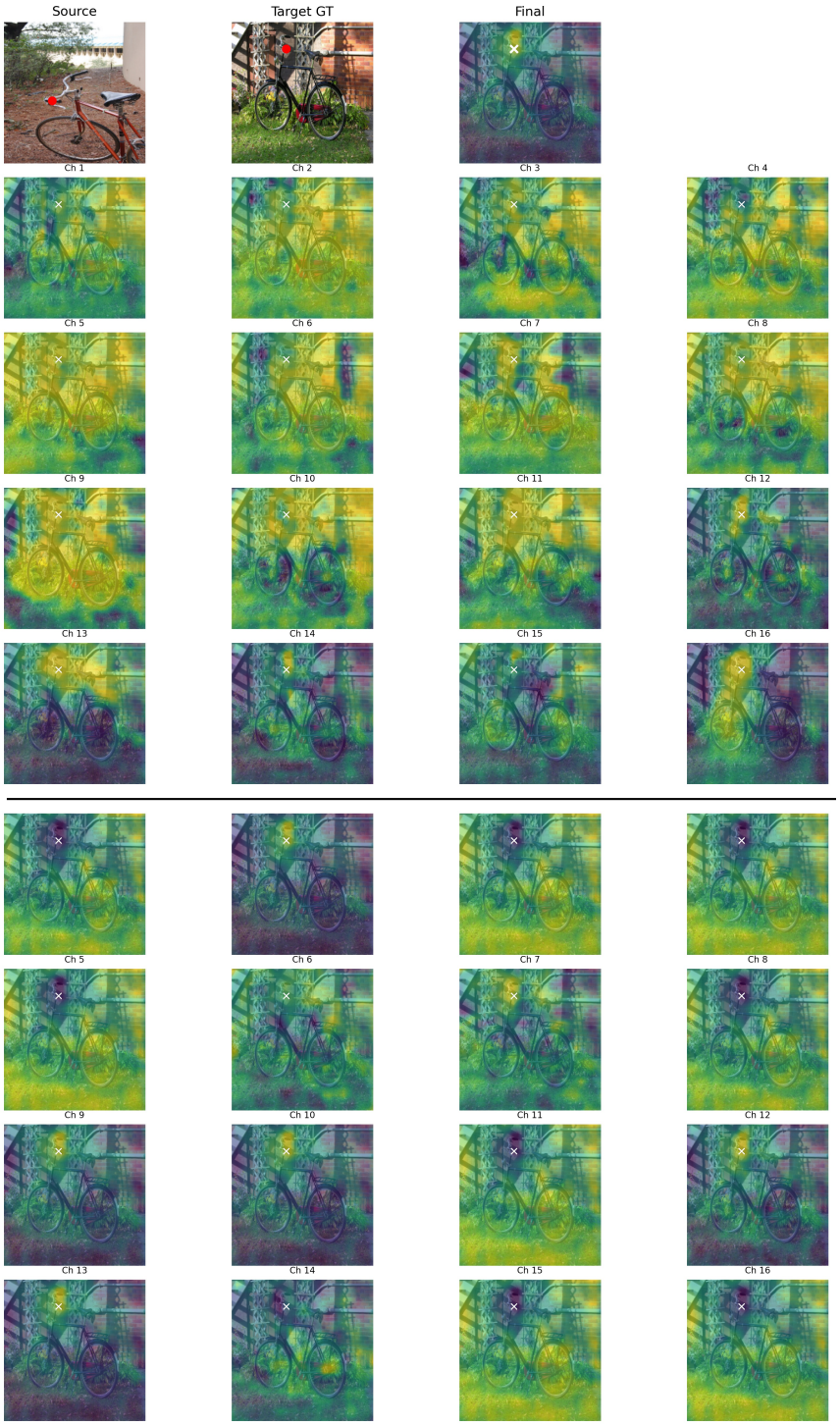

Figure 7. **Additional visualization of similarity-aware selective scan of MambaMatcher**. (First row) Source and target images with GT keypoints shown in red, and our final correlation tensor. (First block of 4x4) The pre-refinement correlation maps, per channel. (Second block of 4x4) The post-refinement correlation maps, per channel. As observed, after refinement, each correlation map is refined to be either *perfectly* wrong, *i.e.*, high attention everywhere other than the GT position, or accurately reflecting the keypoint position. This visualization helps illustrate that during the final prediction of $\hat{\mathbf{C}}$ using linear projection, the wrong maps are effectively disregarded, and the accurate maps are primarily weighted for aggregation, resulting in our final accurate correlation map.

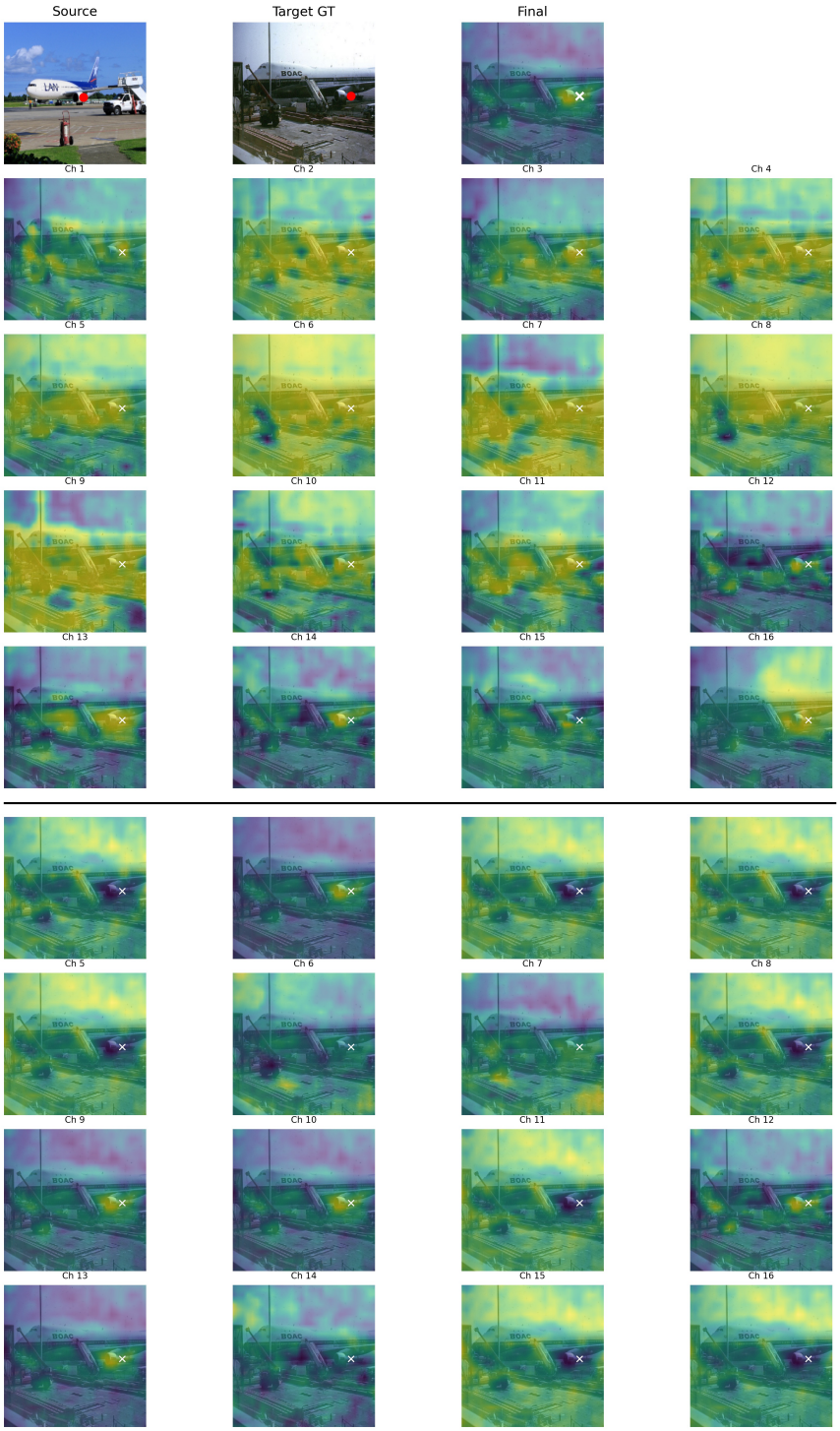

Figure 8. **Additional visualization of similarity-aware selective scan of MambaMatcher**. (First row) Source and target images with GT keypoints shown in red, and our final correlation tensor. (First block of 4x4) The pre-refinement correlation maps, per channel. (Second block of 4x4) The post-refinement correlation maps, per channel. As observed, after refinement, each correlation map is refined to be either *perfectly* wrong, *i.e.*, high attention everywhere other than the GT position, or accurately reflecting the keypoint position. This visualization helps illustrate that during the final prediction of $\hat{\mathbf{C}}$ using linear projection, the wrong maps are effectively disregarded, and the accurate maps are primarily weighted for aggregation, resulting in our final accurate correlation map.

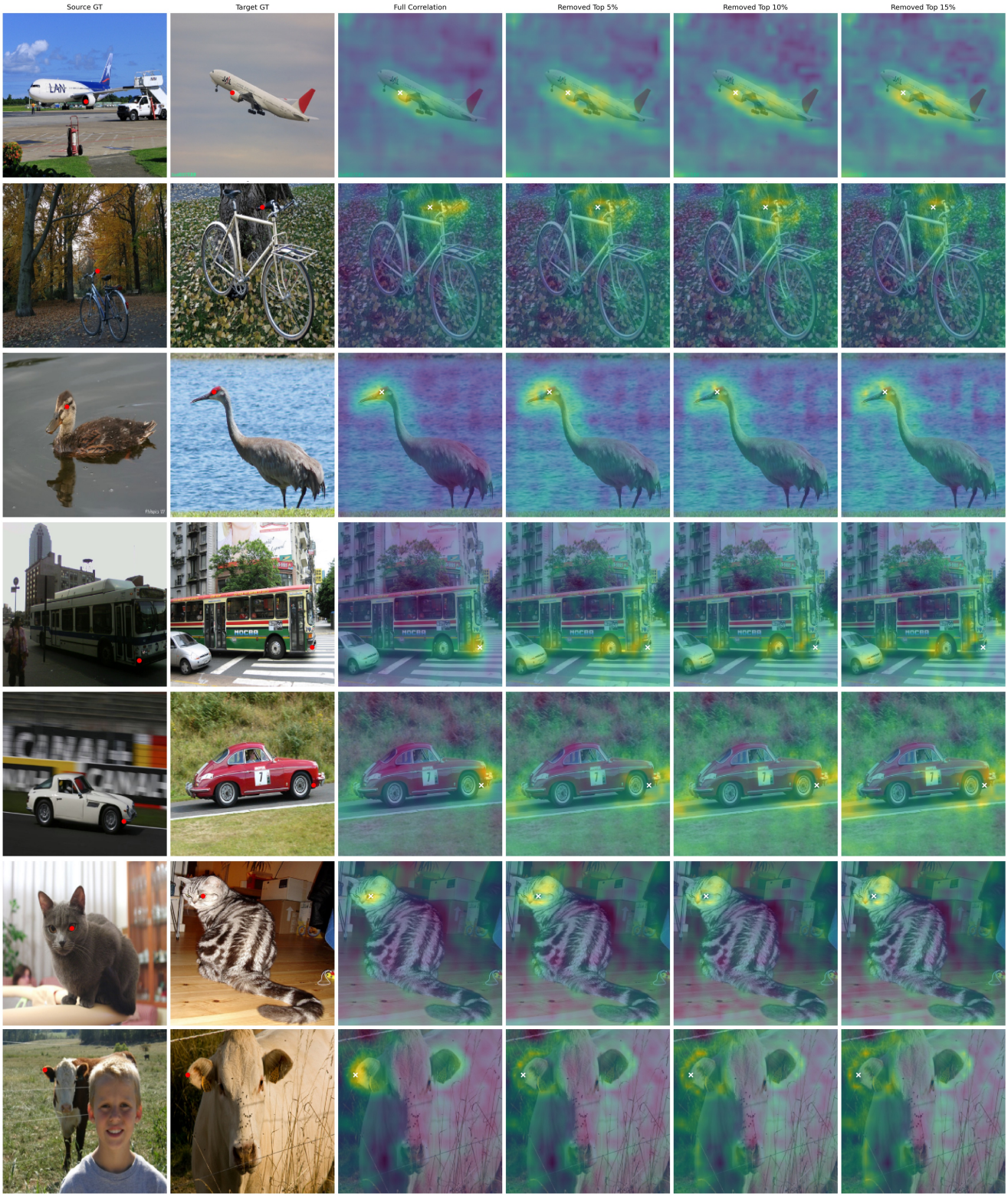

Figure 9. **Correlation robustness analysis**. The first two columns show the source and target images with a GT keypoint pair, followed by the results of our refinement after removing 0% (ours), 5%, 10%, and 15% of the entries with the highest correlation scores. It can be seen that the localization of target keypoint degrades when the top-k% of the correlation scores are replaced with zeros, showing progressively worse localization with higher k. This evidences that establishing a strong and reliable contextual foundation based on accurate matches (*i.e.*, highest correlation score) yields strong benefits to the correlation refinement process.

Figure 10. **Cumulative attribution analysis**. The visualization presents three panels for each keypoint: Source - Target - Cumulative Curves. We use integrated gradients to compute attribution weights for each correlation token, in order to reveal how quickly the model accumulates useful evidence when tokens are ordered by different criteria. The cumulative curves show the proportion of total attribution mass accumulated as a function of token fraction, with Area Under Curve (AUC) metrics quantifying concentration efficiency. Higher AUC values indicate that important evidence is concentrated in fewer tokens, while lower values suggest more distributed attribution. This visualization illustrates that traversing the tokens in the descending order of correlation scores achieves higher concentration of useful evidence compared to ascending or random orderings, demonstrating that the model's refinement process effectively leverages high-confidence initial correspondences.

Figure 11. **Additional qualitative results of MambaMatcher**. Best viewed on electronics, when zoomed-in. The left two columns visualize the ground-truth correspondences. The third column visualizes the predicted target keypoints, and its deviation from the GT keypoints.

Acknowledgement. Seungwook Kim was supported by the Hyundai-Motor Chung Mong-koo Foundation. This work was also supported by the NRF grant (RS-2021-NR059830 (50%)) and the IITP grants (RS-2022-II220113: Developing a Sustainable Collaborative Multi-modal Lifelong Learning Framework (45%), RS-2019-II191906: AI Graduate School Program at POSTECH (5%)) funded by the Korea government (MSIT).

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006. 2

[2] Hilton Bristow, Jack Valmadre, and Simon Lucey. Dense semantic correspondence where every pixel is a classifier. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4024–4031, 2015. 2

[3] Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024. 2

[4] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1201–1210, 2015. 1, 2

[5] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *Advances in Neural Information Processing Systems*, 34:9011–9023, 2021. 1, 2, 13, 15

[6] Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 5, 15, 17

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pages 886–893. Ieee, 2005. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6, 8

[9] Yuki Endo, Satoshi Iizuka, Yoshihiro Kanamori, and Jun Mitani. Deepprop: Extracting deep features from a single image for edit propagation. In *Computer Graphics Forum*, pages 189–201. Wiley Online Library, 2016. 1

[10] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 5, 6

[11] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020. 2

[12] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2

[13] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1711–1725, 2017. 2, 6, 14

[14] 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. 2

[15] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 5, 12, 15, 17

[16] Sunghwan Hong, Jisu Nam, Seokju Cho, Susung Hong, Sangryul Jeon, Dongbo Min, and Seungryong Kim. Neural matching fields: Implicit representation of matching fields for visual correspondence. *Advances in Neural Information Processing Systems*, 35:13512–13526, 2022. 5, 15, 17

[17] Sunghwan Hong, Seokju Cho, Seungryong Kim, and Stephen Lin. Unifying feature and cost aggregation with transformers for dense correspondence. In *The Twelfth International Conference on Learning Representations*, 2023. 5, 15, 17

[18] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022. 2, 5, 15, 17

[19] Jeongho Kim, Gyojung Gu, Minho Park, Sunghyun Park, and Jaegul Choo. Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on, 2023. 1

[20] Seungwook Kim, Juhong Min, and Minsu Cho. Transformatcher: Match-to-match attention for semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8707, 2022. 1, 2, 5, 6, 8, 12, 13, 14, 15, 17

[21] Seungwook Kim, Kejie Li, Xueqing Deng, Yichun Shi, Minsu Cho, and Peng Wang. Enhancing 3d fidelity of text-to-3d using cross-view correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10649–10658, 2024. 1

[22] Seungwook Kim, Juhong Min, and Minsu Cho. Efficient semantic matching with hypercolumn correlation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 139–148, 2024. 2, 5, 15, 17

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Com-*

puter Vision and Pattern Recognition, pages 2278–2287, 2019. 4

[25] Jongmin Lee, Yoonwoo Jeong, Seungwook Kim, Juhong Min, and Minsu Cho. Learning to distill convolutional features into compact local descriptors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 898–908, 2021. 1, 2

[26] Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N Sinha. Patchmatch-based neighborhood consensus for semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13153–13163, 2021. 2, 17

[27] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 14

[28] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024. 2

[29] Shuda Li, Kai Han, Theo W Costain, Henry Howard-Jenkins, and Victor Prisacariu. Correspondence networks with adaptive neighbourhood consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10196–10205, 2020. 2

[30] Xinghui Li, Jingyi Lu, Kai Han, and Victor Prisacariu. Sd4match: Learning to prompt stable diffusion model for semantic matching. *arXiv preprint arXiv:2310.17569*, 2023. 1, 2, 5, 15, 17

[31] Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024. 8

[32] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 2

[33] Jiuming Liu, Ruiji Yu, Yian Wang, Yu Zheng, Tianchen Deng, Weicai Ye, and Hesheng Wang. Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy. *arXiv preprint arXiv:2403.06467*, 2024. 2

[34] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024. 2

[35] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 2

[36] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 5, 15, 17

[37] Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. *arXiv preprint arXiv:1709.04057*, 2017. 3

[38] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2940–2950, 2021. 1, 13, 14, 15, 17

[39] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. 1, 6, 12

[40] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 346–363. Springer, 2020. 2, 15, 17

[41] Juhong Min, Seungwook Kim, and Minsu Cho. Convolutional hough matching networks for robust and efficient visual correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8159–8175, 2023. 1

[42] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. 1966. 6, 8

[43] 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 preprint arXiv:2304.07193*, 2023. 2, 3, 4, 13, 14, 17

[44] Chunghyun Park, Seungwook Kim, Jaesik Park, and Minsu Cho. Learning so (3)-invariant semantic correspondence via local shape transform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22978–22987, 2024. 1

[45] William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei A Efros, and Eli Shechtman. Gan-supervised dense visual alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13470–13481, 2022. 1

[46] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *Advances in neural information processing systems*, 31, 2018. 1, 2, 17

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[48] Jiacheng Ruan and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*, 2024. 2

[49] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–364, 2018. 1, 2

[50] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022. 3

[51] 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. 1, 2, 5, 15, 17

[52] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Probabilistic warp consistency for weakly-supervised semantic correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8708–8718, 2022. 1, 5, 15, 17

[53] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. Fastformer: Additive attention can be all you need. *arXiv preprint arXiv:2108.09084*, 2021. 6, 8, 12

[54] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. *arXiv preprint arXiv:2312.10035*, 2023. 8

[55] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*, 2024. 2, 6

[56] Yubiao Yue and Zhenzhang Li. Medmamba: Vision mamba for medical image classification. *arXiv preprint arXiv:2403.03849*, 2024. 2

[57] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 5, 15, 17

[58] Dongyang Zhao, Ziyang Song, Zhenghao Ji, Gangming Zhao, Weifeng Ge, and Yizhou Yu. Multi-scale matching networks for semantic correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3354–3364, 2021. 17

[59] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 2