Demystifying Unsupervised Semantic Correspondence Estimation - Supplementary Material

Mehmet Aygün Oisin Mac Aodha

University of Edinburgh
https://mehmetaygun.github.io/demistfy

A Additional Experiments and Implementation Details

Here we evaluate some of the implementation choices made in the main paper and provide additional implementation details.

A.1 Implementation Details

We perform experiments with two different types of backbones models for our feature encoder Ψ. For the CNN, unless otherwise specified, we extract features from images resized to 384×384, and use the 1024 dimensional features from the conv3 layer of a ResNet-50 [5]. We use a ResNet-50 trained on Imagenet [12] as our supervised baseline, and MoCov3 [3] as our unsupervised CNN. For the Transformer, 8×8 patches from 224×224 images with stride 8 are used as input (similar to [1]) and we extract 736 dimensional features from 9th layer. We also investigate supervised and self-supervised trained backbones. The supervised and self-supervised CNNs are from [5] and [3] and the Transformer models are from [9] and [2] respectively. During training, we upsample feature maps to 64×64 via bilinear interpolation. For our projection head ρ, a single 1×1 2D convolution is trained and the dimension of the features is reduced to 256. During training, as in [4], we freeze the feature encoder Ψ. The projection head is trained for 50 epochs using Adam [7] optimizer with learning rate of 0.001. Unless stated otherwise, we report results using the standard PCK metric with $\alpha = 0.1$ for direct comparison to other methods. For EQ, DVE and LEAD we set the temperature τ to 0.05 and 0.14 for CL as in described in their papers, and set τ_1 to 0.2 and τ_2 to 0.4 for ASYM. We provide an evaluation of different temperature values in the supplementary material.

A.2 Impact of the Temperature Value

In Table A1, we explore the impact of the temperature for the different unsupervised losses. While the performance of LEAD, ASYM, and DVE do not change significantly with different temperature choices, the performance of CL is impacted drastically, i.e. when using the recommended value of 0.14 from their paper, we obtain a PCK of 30.8 for Spair-71K in Table 2 in the main paper. As

Table A1: Temperature ablation experiment for unsupervised losses on Spair-71K. Here we use the 'Sup. pre-trained - CNN' encoder from main paper. With the exception of ASYM, all methods use τ_1 as their τ and do not use τ_2 at all.

Metric	τ_1	τ_2	DVE	CL	LEAD	ASYM
PCK	0.02	0.04	16.5	9.2	31.9	31.7
	0.05	0.1	16.3	8.2	31.7	32.1
	0.1	0.2	16.0	17.2	31.9	33.0
	0.2	0.4	15.7	26.6	31.4	34.0
	0.4	0.8	9.2	15.8	30.1	29.5
PCK†	0.02	0.04	12.9	7.5	25.5	25.4
	0.05	0.1	12.4	6.6	25.4	25.8
	0.1	0.2	12.4	13.8	25.4	26.6
	0.2	0.4	12.1	20.0	25.1	27.2
	0.4	0.8	6.9	11.2	23.8	23.1

noted in the main paper, for EQ, DVE, and LEAD we set the temperate τ to 0.05 and use 0.14 for CL based on the recommendations in the original papers. We use the same temperature values for all datasets.

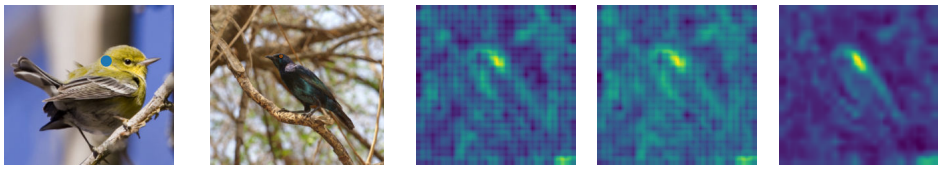
A.3 Impact of Design Choices for ASYM

As our new proposed ASYM loss is an adaptation of LEAD, here we present experiments ablating our design choices. ASYM differs from LEAD in two respects: (i) ASYM uses different temperature values for the correlation maps for the original features and the projected features, and (ii) ASYM uses a mean square error (MSE), as opposed to cross entropy (CE) which is used in LEAD. As can be seen in Table A2, the MSE loss performs worse for LEAD while it improves performance of ASYM. However, the main difference in overall performance is not a result of the choice of penalty function (i.e. MSE versus CE), but the usage of different temperature parameters. In Table A2 we can see that changing the temperature for LEAD has no significant impact on the final performance.

Due to changes in the formulation, the objectives that ASYM and LEAD optimize also differ. For a given pair of points and their similarity score, LEAD reduces the dimensionality of the embeddings for these points while maintaining the same similarity scores as the input feature space. This is achieved by capturing both what is common and not common between the pair of points. Using higher or lower temperature values does not change the feature distances in the LEAD. However, in our ASYM objective, for a point pair which has a high similarity score, the projection needs to make these points even closer in order to match with the same similarity score from the input features as the projected embeddings use a higher temperature value. A visualization of the result of this can be observed in Fig. A1. As expected, for a given keypoint and a target image LEAD produces a very similar similarity map compared to the one calculated

Table A2: Loss and temperature ablation for ASYM and LEAD on Spair-71K. For both methods, Mean Square Error (MSE) and Cross-Entropy (CE) losses are used. ASYM using CE with the same temperature value for both τ_1 and τ_2 is equivalent to LEAD.

Method	τ_1	τ_2	MSE	CE
LEAD	0.05	-	31.5	31.7
	0.1	-	30.6	31.9
	0.2	-	29.9	31.4
	0.4	-	27.4	30.3
ASYM	0.05	0.1	32.1	32.0
	0.1	0.2	33.0	32.8
	0.2	0.4	34.0	32.0

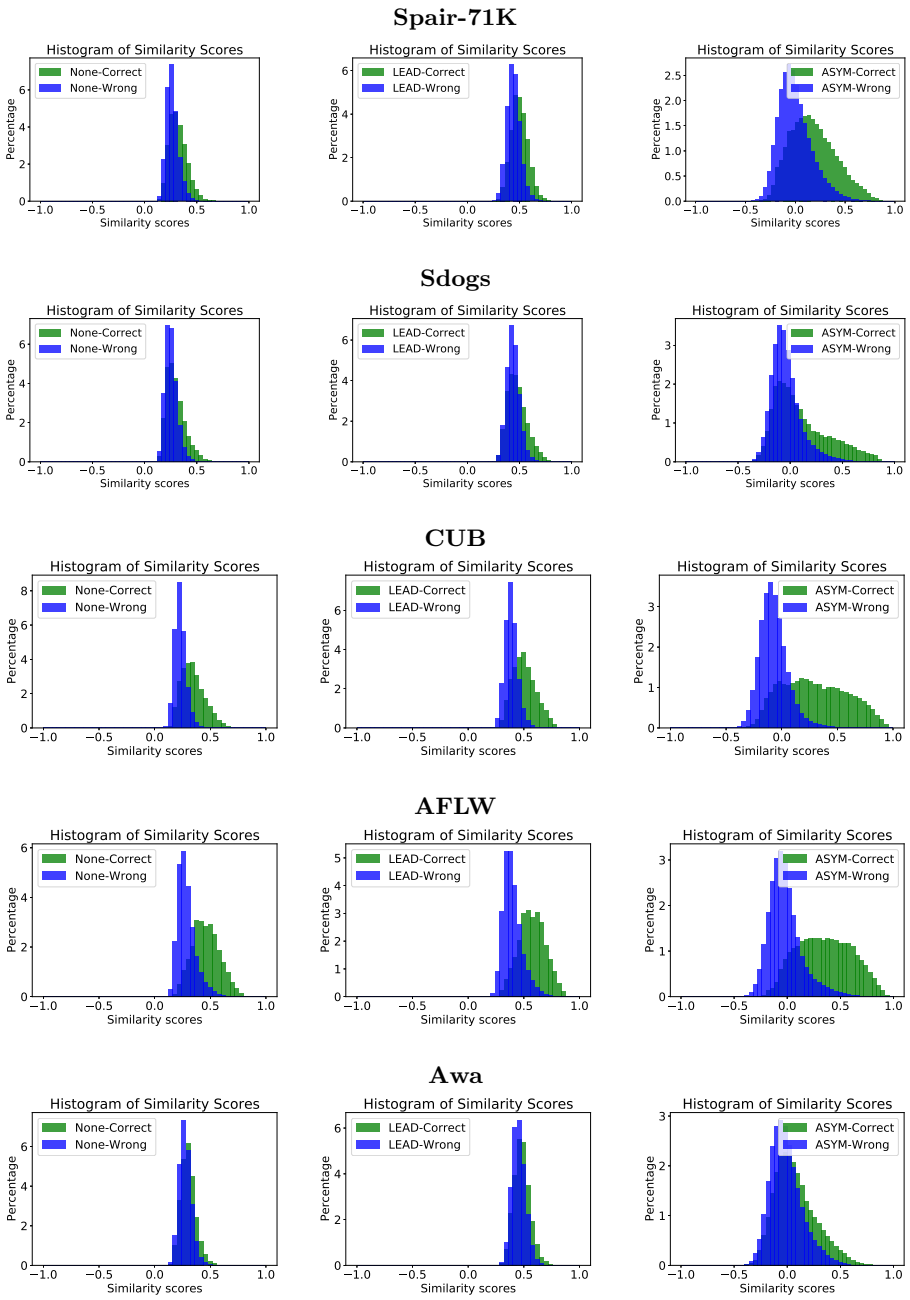


(a) Source (b) Target (c) None (d) LEAD (e) ASYM

Fig. A1: Feature matching scores for different methods for the keypoint on on the birds head (indicated in blue) from the source images in (a) to the target in (b). By design, LEAD matches the distribution from the original feature space shown in (c). We can see that our ASYM method results in a much more sharper distribution around the correct location compared to LEAD.

with the original features. In contrast, ASYM produces a more 'peaked' similarity map, since matching points from original features become closer in the new embedding space.

We also compare how the similarity scores change after unsupervised projection. For a source keypoint, we calculate the cosine similarity scores for all pixel embeddings in the target image. If a point is within the threshold area of a target keypoint we refer to these points as 'correct' matches, otherwise they are classed as 'wrong' matches. We visualize the histogram of these scores for all datasets in Fig. A2. As can be seen from the distributions, LEAD results in histograms that are very similar to original input features (i.e. None). However, ASYM reduces the overlap between the correct and wrong distributions. As expected, if the similarity scores for correct matches are not larger than wrong matches, ASYM cannot improve the embeddings significantly, as seen in the Awa dataset.



(a) None (b) LEAD (c) ASYM

Fig. A2: Histograms for cosine similarity scores of embeddings for (a) None, (b) LEAD, and (c) ASYM. Each row is a different dataset.

A.4 Impact of Encoder Feature Layer

In Table A3 we experiment with using features from different feature layers from a CNN (Resnet50 [5]) trained using supervision on Imagenet. The third convolution layer performs best on all datasets, and so we use features from it in all of our experiments for CNNs. For Transformer backbones [9, 2], we used the 9th layer as the initial features, as they were shown to perform best in [1].

Table A3: Evaluation of using pre-trained features from different layers for the Resnet50 trained with Imagenet. The results here for conv_3 correspond to the no projection model (i.e. 'None) from Table 2 (a) in the main paper.

Layer	Spair-71K	SDogs	CUB	AFLW	Awa
conv_1	7.3	5.1	7.9	11.6	5.6
conv_2	12.9	8.6	13.3	27.2	9.1
conv_3	31.8	34.9	51.3	57.4	28.8
conv_4	15.8	10.3	14.0	31.3	9.3

A.5 Impact of Input Image Resolution

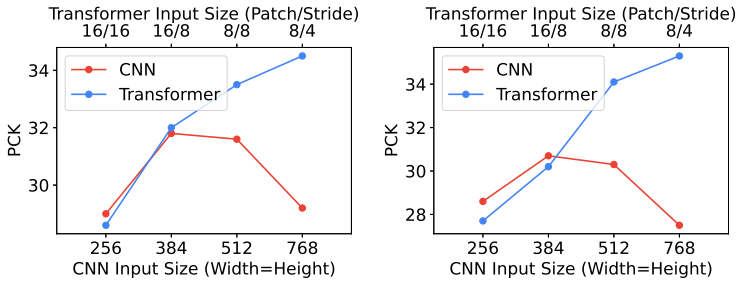
In Fig. A3, we explore the impact of different input image resolutions, using pre-trained embeddings without any projection (i.e. None), for CNN and Transformer backbones. We used CNNs are from [5] and [3] as the supervised and unsupervised CNN, [9] and [2] as the supervised and unsupervised Transformer. Transformers scale well as the number of tokens increases, while the performance of the CNNs saturates as the image resolution is increased. We argue that this is due to not-adaptive nature of the receptive field sizes of CNNs which may overfit to the trained image resolution. As CNNs best performed using an input resolution of 384x384, we use that resolution for in our experiments. While 8x8 patches with stride 4 is the best performing version for transformers, due to computational constraints, we used 8x8 patches with stride 8 as the transformer input in our experiments.

B Additional Results and Analysis

Here we present additional results and more detailed analysis for each of the datasets of interest.

B.1 Detailed Error Analysis for Additional Datasets

We present the detailed error analysis and report scores using our $\text{PCK}^{\dagger}$ metric in Table A4 for each dataset not shown in the main paper. Similar to the



(a) Supervised Pre-training (b) Unsupervised Pre-training

Fig. A3: Semantic correspondence performance of CNNs and Transformers with different input sizes on Spair-71K with no projection. Pre-trained features from models trained on Imagenet with (a) supervised or (b) unsupervised losses are used. Image resolution is fixed to 224x224 for the Transformers. Note that the effective resolution of feature maps from CNNs and Transformers are not comparable for each vertical position in the plots.

Spair-71k results from the main paper, the most common error type is 'miss' among all datasets. Our ASYM approach generally reduces misses compared to other unsupervised losses. With the exception of the AFLW dataset, there is a noticeable difference between PCK† and PCK scores. For AFLW, the keypoints that correspond to each other are well defined and far apart from each other as the faces are large. As a result, there are far fewer swaps, and so PCK† scores are close to their PCK counterparts. In contrast, for CUB, most of the points are distributed close to the head region of the birds which leads to a lot of swaps and a drop in scores for our new proposed metric. This highlights the importance of using a proper metric for evaluating the semantic correspondence task. Matching a keypoint from the beak of a bird to the eye of another bird is not a correct semantic match, but with the current PCK metric it would be labeled as correct if it was within the distance threshold.

B.2 Example Images and Qualitative Results

Random instance pairs from each dataset are depicted in Fig. A4. Spair-71K contains examples of different classes, spanning man-made objects to animal classes. StanfordDogs (SDogs) contains different breeds of dogs in challenging poses with varying appearance. CUB contains bird species. AFLW contains human faces which occupy most of the frame. Unlike CUB and SDogs which only contains images from one species, Awa includes different vertebrate animal categories which enables us to assess inter-category correspondence performance.

We also present some qualitative results for the different unsupervised losses, for all datasets, in Fig A5 and Fig A6. While ASYM generally improves the predictions compared to other unsupervised losses, it still lags behind supervised

Table A4: Evaluation of error types across four different datasets. In addition to PCK, we also report scores for our PCK† metric. Results for Spair-71 are presented in Table 3 in the main paper.

(a) SDogs

Method	Miss↓	Jitter↓	Swap↓	PCK↑	PCK† ↑
EQ	55.9	21.4	25.9	21.2	18.2
DVE	57.7	21.8	24.8	20.5	17.5
CL	40.9	17.9	27.3	37.0	31.9
LEAD	38.0	16.2	31.2	35.1	30.8
ASYM	33.1	16.3	31.4	40.4	35.5
Supervised	23.7	16.7	29.0	53.2	47.3

(b) CUB

Method	Miss↓	Jitter↓	Swap↓	PCK↑	PCK† ↑
EQ	44.0	24.8	35.2	28.1	20.9
DVE	44.3	24.6	35.7	27.7	20.0
CL	24.8	20.1	34.6	54.5	40.7
LEAD	28.1	17.4	31.8	51.5	40.1
ASYM	21.7	16.9	29.8	60.8	48.5
Supervised	14.3	15.2	25.4	72.7	60.2

(c) AFLW

Method	Miss↓	Jitter↓	Swap↓	PCK↑	PCK† ↑
EQ	38.0	26.0	14.2	48.5	47.8
DVE	24.9	21.2	17.3	58.7	57.8
CL	18.0	11.4	15.2	67.3	66.8
LEAD	13.6	10.7	28.8	58.0	57.5
ASYM	11.7	7.9	25.2	63.6	63.1
Supervised	7.0	4.7	12.7	80.8	80.4

(d) AWA

Method	Miss↓	Jitter↓	Swap↓	PCK↑	PCK† ↑
EQ	52.0	19.6	38.7	15.6	10.3
DVE	52.1	19.2	37.8	15.4	10.1
CL	38.4	16.8	41.5	31.7	20.1
LEAD	37.1	16.3	44.0	29.1	18.9
ASYM	32.2	16.7	45.6	34.1	22.1
Supervised	23.4	18.3	46.3	46.1	30.3

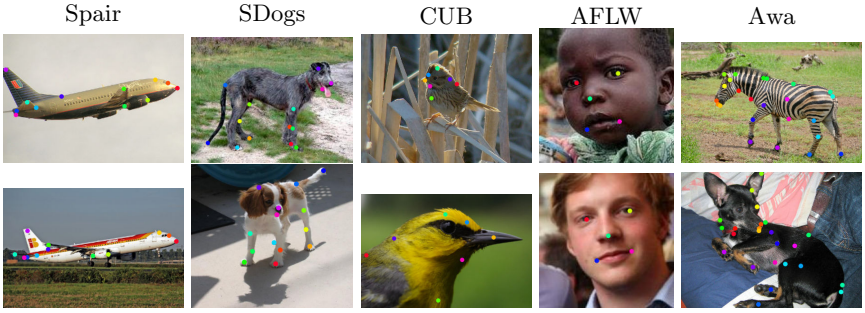
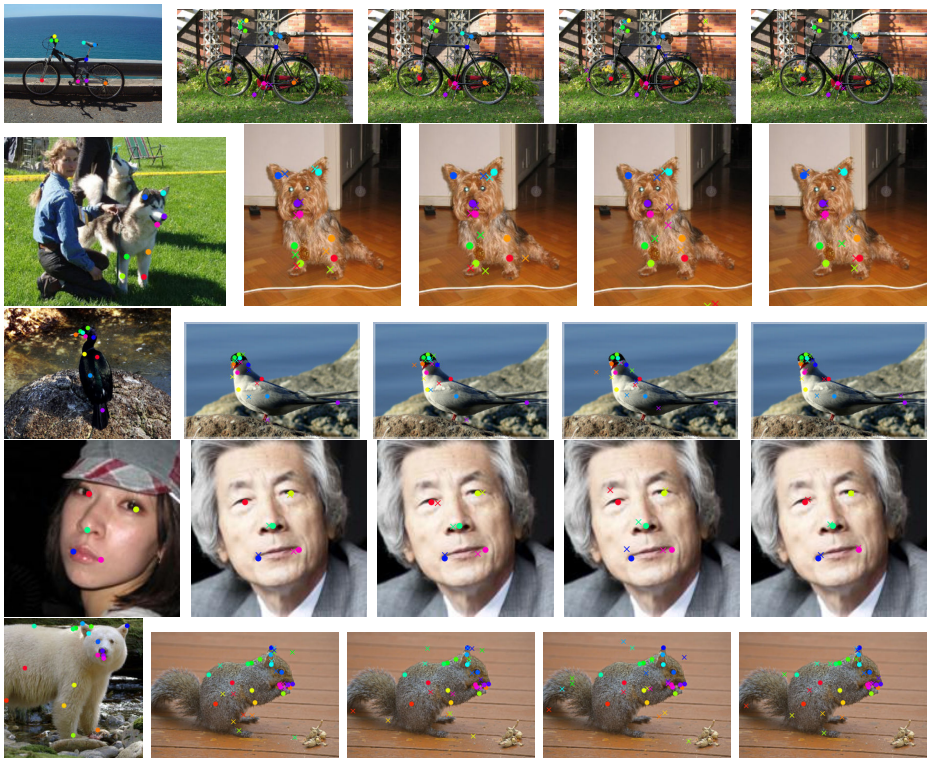


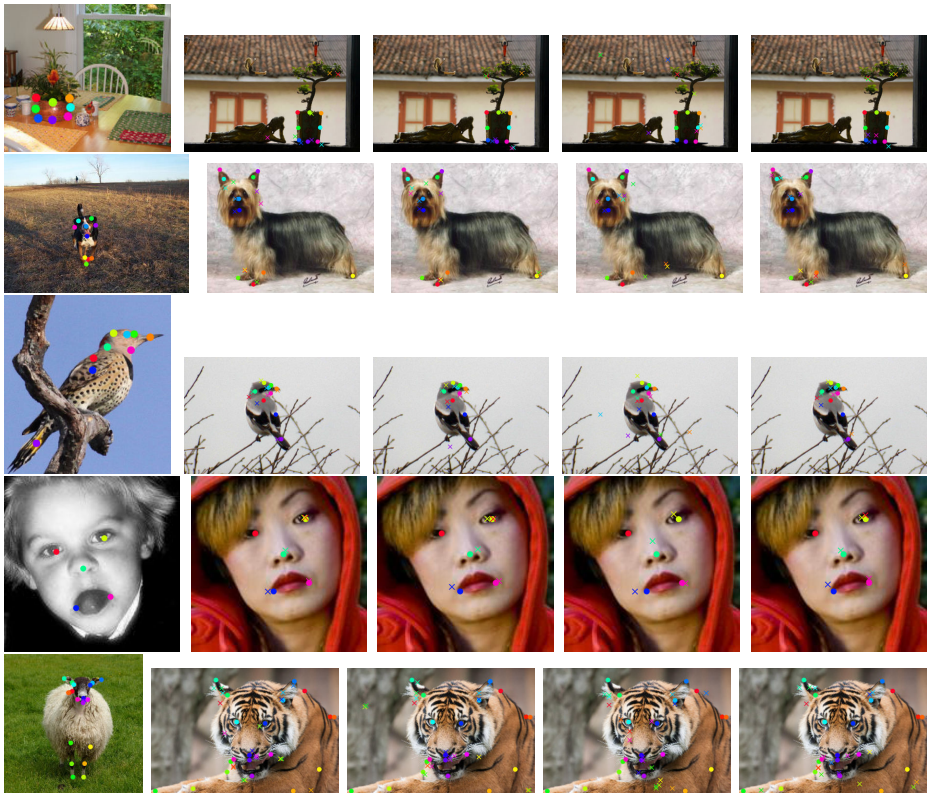
Fig. A4: Examples from each of the datasets with the keypoint annotations that we consider in our paper. The pop row illustrates a source instance and the bottom a target instance.

projection which makes use of ground truth matches for training. AFLW generally contains easy examples with a small percentage of background pixels and only minor changes in pose which makes the task easier. While the PCK scores for AFLW and CUB are close to each other, as can be seen from qualitative results, this can be explained by how PCK evaluates matches which does not necessarily reflect the difficult of the dataset in some cases.



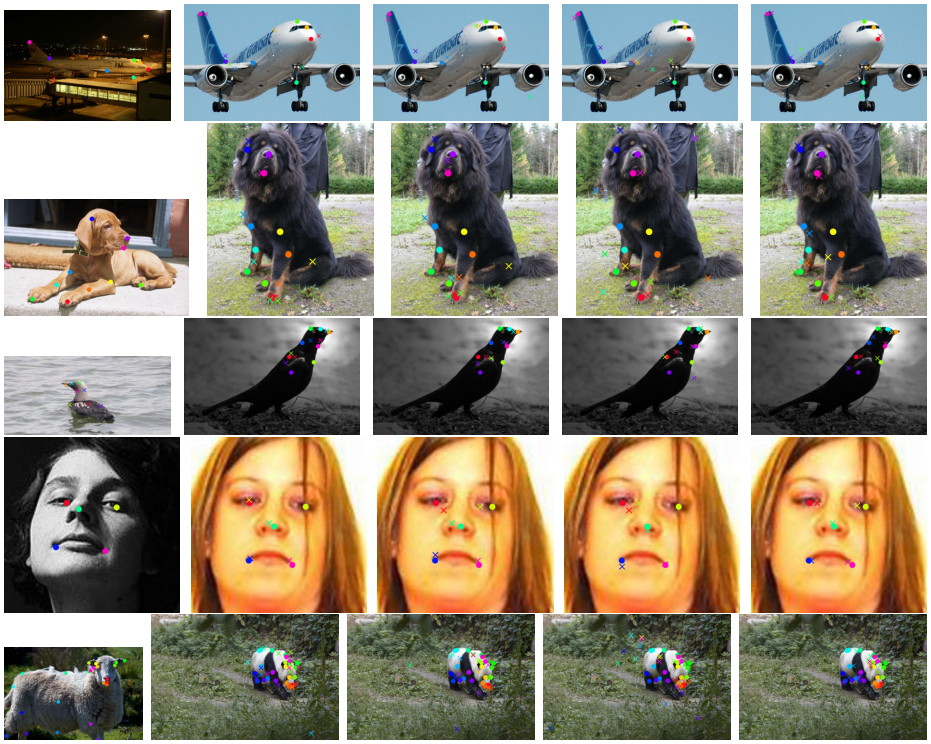
(a) Source (b) ASYM (c) CL (d) DVE (e) Sup

Fig. A5: Qualitative matching results. Each row is a different dataset: Spair, SDogs, CUB, AFLW, and Awa, from top to bottom. The left most image for each row is a source example, and the remaining images visualize matches from different unsupervised methods, where 'o' indicates a ground truth location and 'x' indicates a prediction. Overall, while ASYM cannot match with the performance of Supervised projection, it is better than other unsupervised methods. For instance, in the AFLW example, only our proposed ASYM and supervised baseline able to precisely find correspondences for the all keypoints.



(a) Source (b) ASYM (c) CL (d) DVE (e) Sup

Fig. A6: More qualitative matching results. Each row is a different dataset: Spair, SDogs, CUB, AFLW, and Awa, from top to bottom. The left most image for each row is a source example, and the remaining images visualize matches from different unsupervised methods, where 'o' indicates a ground truth location and 'x' indicates a prediction. For the Awa-Pose dataset example in the bottom row, all of the methods struggle as visual diversity is high between instances and the target example is in a different pose.



(a) Source (b) ASYM (c) CL (d) DVE (e) Sup

Fig. A7: More qualitative matching results. Each row is a different dataset: Spair, SDogs, CUB, AFLW, and Awa, from top to bottom. The left most image for each row is a source example, and the remaining images visualize matches from different unsupervised methods, where 'o' indicates a ground truth location and 'x' indicates a prediction. While most methods perform reasonably good on the AFLW dataset instance, the predictions for the highly articulated objects (e.g. animals), even the supervised baseline cannot obtain satisfactory results.

B.3 Visualizing Learned Feature Embeddings

We present 2d t-SNE [11] visualizations of the keypoint embeddings for the AFLW, CUB, and SDogs datasets in Fig A8. Since Spair contains different classes wherein the keypoints are not semantically consistent across classes, we did not present t-SNE visualization of Spair. Moreover, the Awa dataset contains more than 30 keypoints which makes visualizing them difficult, thus we exclude that as well. To create these plots, we first extracted embeddings from only the keypoint locations. These are 1024 dimensional for the None projection and 256 for other unsupervised methods. We then project these embeddings to 2D using t-SNE, and finally plot them. Each color represents a different keypoint type, which is different depending on the dataset.

LEAD and ASYM look similar to original feature space. One interesting thing is that, CL manages to separate overlapping embeddings when compared to the 'no projection' baseline on the AFLW dataset. This is reflected by their superior PCK scores for this dataset. However, for CUB there are cases where it splits clusters of a keypoints which were a single prominent cluster in the original embeddings space. This perhaps indicates that applying CL can sometimes destroy invariances that were captured in the pre-trained features, thus leading to undesirable changes in the embedding space.

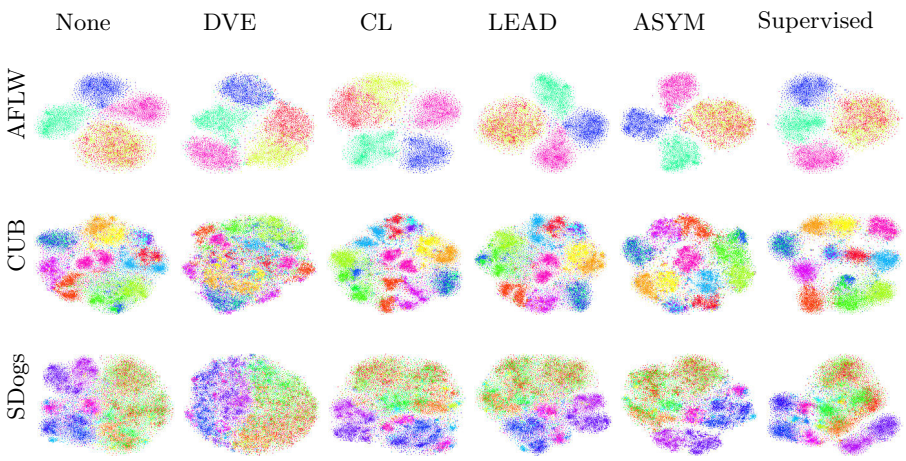


Fig. A8: t-SNE visualization the embeddings learned by different unsupervised losses. Each row is a different dataset, and the colors indicate the ground truth identity of different keypoints.

B.4 Keypoint Regression Evaluation

As noted in the main paper, the two common types of evaluation paradigms for semantic correspondence estimation are: (i) landmark/keypoint regression and

(ii) feature matching. We chose to use feature matching for our results as is does not require additional supervision. However, for completeness here we evaluate embeddings from different unsupervised methods using the regression protocol on two face datasets; MAFL [16] and AFLW$_M$[8]. AFLW$_M$ contains crops from the MTFL[15] dataset, which contains 2,995 examples for testing and 10,122 for training. This is the same dataset that we consider in our main paper as AFLW, as it was referred as AFLW$_M$ in some papers [13, 4, 6] we present here as AFLW$_M$ as well. We report percentage of inter-ocular distance similar to previous work. Please note that lower is better in this metric.

We follow same approach as in [13, 4, 6], i.e. we freeze the embedder Φ and train an additional regression head on top of these features. We use the unsupervised CNN trained on Imagenet for the feature encoder Ψ, and the unsupervised losses are finetuned on the AFLW dataset for both datasets to obtain embeddings which are input to the regression head. The results can be seen in Table A5.

Table A5: Keypoint regression results with percentage of inter-ocular distance. The rows marked as 'Original' are numbers taken from the original papers and differ in the network architecture and in some cases the amount of supervision used. Note that AFLW is referred to as AFLW$_M$ in some of the works below. The numerical scores represent the percentage of inter-ocular distance, where lower scores are better.

Implementation	Method	Feat.dim.	MAFL	AFLW
Original	DVE	64	2.86	7.53
	CL	256	2.64	7.17
	LEAD	256	2.87	6.51
Ours	DVE	256	3.07	8.57
	CL	256	2.96	7.73
	LEAD	256	2.80	7.97
	ASYM	256	2.94	7.98

We also compared the results taken directly from the original papers. While our re-implementation obtains reasonable scores, they are slightly worse than the original reported numbers. This can be explained by the fact that we use a basic encoder which produce dense feature maps in a lower spatial dimension. Compared to CL [4], we use single layer features before projection, as opposed to higher dimensional hypercolumn features. Unlike the original LEAD [6] implementation, our projection operation is a single layer 1x1 2D convolution compared to a fully convolutional decoder which produces higher resolution features used in their paper. Unlike DVE [13], we do not preform end-to-end finetuning. Also, for consistency with our other results the unsupervised losses in our implementations are finetuned on the AFLW dataset instead of CelebA [10], which is

a larger dataset. While one may expect a large drop in performance due to these differences, there is in fact only a one pixel drop. This level of error is likely to be on the order, if not smaller, than human annotation inconsistency. This perhaps highlights the inadequacy of the regression evaluation as the supervision used during training makes the evaluation unfair. Furthermore, it again emphasizes that these types of face datasets are perhaps reaching saturation.

B.5 Pre-training Source and Cross Dataset Evaluation

Here we present the raw numbers for the pre-training data source and cross dataset evaluation experiments from main paper. The results can be found in Table A6 and Table A7, and correspond to the results in Fig. 3 and Fig. 4 in the main paper.

Table A6: Results for using different sources of pre-training dataset. These numbers correspond to those presented in Fig. 3 in the main paper.

(a) Imagenet

Projection(ρ)	Spair-71K	SDogs	CUB	AFLW	Awa
None	30.7	34.3	47.5	64.3	27.6
NMF	20.6	19.9	44.0	40.8	15.6
PCA	27.4	29.8	50.7	51.0	24.1
Random	26.6	31.5	40.0	60.2	23.3
Supervised	39.5	54.0	73.4	83.8	48.2
EQ[14]	14.3	20.5	26.4	62.8	15.5
DVE[13]	15.0	19.4	28.7	60.6	14.7
CL[4]	29.7	37.9	54.1	77.1	33.4
LEAD[6]	30.5	34.4	48.3	64.9	28.1
ASYM (Ours)	33.2	38.2	54.4	69.7	32.1

(b) iNat2021

Projection(ρ)	Spair-71K	SDogs	CUB	AFLW	Awa
None	21.6	19.3	44.5	42.0	16.1
NMF	18.8	17.9	45.2	33.6	15.6
PCA	21.7	20.2	45.2	42.2	16.7
Random	17.0	14.5	35.8	37.2	12.3
Supervised	28.1	36.4	70.6	58.6	32.6
EQ[14]	10.7	15.4	26.3	40.8	11.7
DVE[13]	10.6	15.4	25.8	38.5	11.2
CL[4]	19.9	19.9	51.9	44.8	18.1
LEAD[6]	21.1	19.4	44.1	41.9	16.0
ASYM (Ours)	21.8	21.8	51.7	44.4	17.7

(c) CelebA

Projection(ρ)	Spair-71K	SDogs	CUB	AFLW	Awa
None	11.6	8.8	13.6	50.3	8.0
NMF	10.0	8.5	12.4	47.6	8.0
PCA	11.7	9.0	13.8	51.2	8.3
Random	10.4	8.1	11.8	43.7	7.5
Supervised	14.3	17.4	28.0	65.2	14.4
EQ[14]	8.6	9.5	12.1	54.0	8.1
DVE[13]	9.0	9.4	12.4	48.3	8.1
CL[4]	10.8	10.1	12.8	62.4	8.0
LEAD[6]	11.5	8.8	13.3	50.0	8.0
ASYM (Ours)	11.5	8.9	13.4	60.7	8.0

Table A7: Cross dataset evaluation results. These results use the 'Sup. pre-trained - CNN' and correspond to the results in Fig. 4 in the main paper.

(a) CL

Test/Train	Spair-71K	SDogs	CUB	AFLW	Awa
Spair-71K	30.8	31.1	31.4	29.1	31.5
SDogs	36.4	37.0	36.8	35.4	36.9
CUB	49.1	47.5	54.5	45.6	48.3
AFLW	62.7	62.1	62.2	67.3	62.7
Awa	30.6	29.9	30.1	27.0	31.7

(b) ASYM

Test/Train	Spair-71K	SDogs	CUB	AFLW	Awa
Spair-71K	34.0	30.9	28.3	25.9	30.2
SDogs	38.4	40.4	31.1	30.9	38.3
CUB	56.2	50.5	60.8	42.8	51.1
AFLW	54.4	58.2	48.6	63.6	56.3
Awa	33.5	33.9	26.6	25.4	34.1

(c) DVE

Test/Train	Spair-71K	SDogs	CUB	AFLW	Awa
Spair-71K	16.3	13.9	15.5	17.3	14.7
SDogs	21.9	20.5	21.3	23.3	20.3
CUB	26.2	24.1	27.7	25.2	23.4
AFLW	41.0	41.2	43.9	58.7	41.4
Awa	16.0	14.2	14.8	17.6	15.4

(d) Supervised

Test/Train	Spair-71K	SDogs	CUB	AFLW	Awa
Spair-71K	38.7	26.7	24.5	17.4	29.2
SDogs	40.1	53.2	29.0	25.1	42.9
CUB	52.5	40.2	72.7	25.4	47.6
AFLW	57.4	56.6	46.5	80.8	58.7
Awa	35.1	34.9	26.9	18.1	46.1

References

1. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. arXiv:2112.05814 (2021)
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
3. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv:2104.02057 (2021)
4. Cheng, Z., Su, J.C., Maji, S.: On equivariant and invariant learning of object landmark representations. In: ICCV (2021)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
6. Karmali, T., Atrishi, A., Harsha, S.S., Agrawal, S., Jampani, V., Babu, R.V.: Lead: Self-supervised landmark estimation by aligning distributions of feature similarity. In: WACV (2022)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
8. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: ICCV workshops (2011)
9. Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., Zhai, X.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
10. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
11. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR (2008)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
13. Thewlis, J., Albanie, S., Bilen, H., Vedaldi, A.: Unsupervised learning of landmarks by descriptor vector exchange. In: ICCV (2019)
14. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised learning of object frames by dense equivariant image labelling. NeurIPS (2017)
15. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: European conference on computer vision. pp. 94–108. Springer (2014)
16. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Learning deep representation for face alignment with auxiliary attributes. PAMI (2015)