

# **Supplementary Material: Vision Transformers with Self-Distilled Registers**

## **Sections**

1. Implementation details for Self-Distillation (section [1](#))
2. Implementation details for Quantitative Evaluation (section [2](#))
3. Additional Qualitative Examples for Segmentation (section [3](#))
4. Additional Qualitative Heatmaps for PH-Reg Zero-Shot (section [4](#))
5. Proof of Test Time Augmentation (section [5](#))

# 1 Implementation details for Self-Distillation

2 In this section, we provide a detailed overview of how we implement self-distillation in PH-Reg.  
3 Our self-distillation framework consists of one teacher model and one student model. While both  
4 the teacher and student model are initialized from the same weights, the teacher is frozen, while  
5 additional register parameters are added to the student network.

## 6 1.1 Model Architectures

7 **Teacher Model Architecture.** For CLIP based models, since we focus on zero-shot open-vocabulary  
8 segmentation, we utilize the NACLIP modification to the final layer. This modification does not  
9 introduce any additional weights to the teacher network, and is training-free. Our empirical analysis  
10 in Table shows that NACLIP’s neighborhood attention mechanism improves feature consistency. For  
11 DINOv2, we directly use the final output layer, without any modification to the teacher network.

12 **Student Model Architecture.** Based on the results from the ablation studies we integrate 16 register  
13 tokens into the student model. For the CLIP based student, to ensure representational alignment we  
14 directly take the  $v$  head from the output layer (the MaskCLIP output). For DINO based students, we  
15 do not apply such modifications. We bicubically upsample the positional embedding so it matches the  
16 input image. Unless otherwise specified, this modification is applied consistently, while all other  
17 layers remain unchanged.

## 18 1.2 Model Implementation

Table 1: **Model implementation libraries and weights.** We compare models trained using different datasets and objectives.

Model	Library	Weight
CLIP	clip (OpenAI)	ViT-B-16
OpenCLIP	open_clip	hf-hub:laion/CLIP-ViT-B-16-laion2B-s34B-b88K
DFN-CLIP	open_clip	hf-hub:apple/DFN2B-CLIP-ViT-B-16
DINOv2	transformers (Hugging Face)	facebook/dinov2-base

19 We provide the model weights and the corresponding implementation libraries in Table 1.

## 20 1.3 Optimization

21 In the distillation process, the shorter side of each input image is resized using bicubic interpolation  
22 to 448 for CLIP-based models and 518 for DINOv2. The resized image is then randomly cropped  
23 into a square of size (448, 448) or (518, 518), respectively. For each input image, we generate  
24  $N = 10$  augmentations using random shifts and horizontal flips. Assuming an image length of 1, we  
25 uniformly sample the shift for both the horizontal and vertical axes from  $[-0.15, 0.15]$ . While the  
26 horizontal flip is sampled with probability 0.5. To ensure each patch is covered, we do not apply any  
27 augmentation to the first image of the 10. All shifted images are concatenated and fed into the teacher  
28 model, while the original (unshifted) images are used as input to the student model. The target feature  
29 is computed as the average of these 10 augmentations. To accommodate the resized input images for  
30 both the teacher and student models, we consistently resize the positional embeddings using bicubic  
31 interpolation. During training, the weights of the teacher model are frozen. In the student model, we  
32 allow updates to registers, the positional embeddings, the convolutional patch embedding layer, and  
33 the final transformer layer containing the self-attention mechanism.

34 The distillation framework is implemented in PyTorch, with distributed training managed via PyTorch  
35 Accelerate. Training is conducted on 4 NVIDIA Ada 6000 GPUs, with mixed-precision optimization  
36 to balance computational efficiency and numerical stability. Detailed training configurations are  
37 provided in Table 2 and Table 3.

Table 2: Configs for CLIP-based models.

Config	Value
optimizer	AdamW
initial learning rate	3e-4
final learning rate	1e-5
weight decay	1e-2
optimizer momentum	$\beta_1=0.9, \beta_2=0.999$
learning rate scheduler	Exponential Scheduler
batch size	16
training epochs	100
augmentation	RandomSquareCrop

Table 3: Configs for DINOv2.

Config	Value
optimizer	AdamW
initial learning rate	1e-4
final learning rate	5e-6
weight decay	1e-2
optimizer momentum	$\beta_1=0.9, \beta_2=0.999$
learning rate scheduler	Exponential Scheduler
batch size	8
training epochs	100
augmentation	RandomSquareCrop

Table 4: **Open-vocabulary semantic segmentation quantitative comparison on 7 datasets.** We report the Pearson correlation coefficient for the zero-shot query against the one-hot ground truth labels. The results are averaged within each image, then averaged across images. Compared to mIoU, pearson does not require knowledge of all of the categories present an image (via softmax). The value ranges from -1 to 1, where 1 = perfect positive correlation, -1 = perfect negative correlation, and 0 = no linear correlation. The best result for each dataset is highlighted in **bolded**.

Method	VOC21	PC60	VOC20	PC59	Stuff	City	ADE20k	Avg.
SCLIP	-0.005	0.349	0.409	0.443	0.323	0.291	0.308	0.303
ClearCLIP	0.012	0.428	0.489	0.543	0.393	0.336	0.418	0.374
NACLIP	0.011	0.422	0.470	0.543	0.392	0.363	0.425	0.375
NACLIP+DVT	0.003	0.438	0.487	0.551	0.395	0.367	0.427	0.381
Ours (PH-Reg)	<b>0.013</b>	<b>0.468</b>	<b>0.494</b>	<b>0.590</b>	<b>0.424</b>	<b>0.381</b>	<b>0.461</b>	<b>0.404</b>

#### 1.4 Pearson Analysis of Zero-Shot Segmentation

In this section we present additional evaluation results on zero-shot open-vocabulary semantic segmentation via the pearson metric. Results are illustrated in Table 4. Overall, PH-Reg CLIP significantly outperforms the baseline models on 7 datasets. Even in the absence of prior category knowledge, PH-Reg CLIP achieves an average performance of 0.404, representing a clear improvement over the second-best method, DVT enhanced NACLIP, with an average performance of 0.381. These results highlight that our approach improves the consistency of dense feature representations by reducing artifact tokens, thereby offering a robust and generalizable enhancement over existing methods.

We further observe that both ClearCLIP and NACLIP achieve competitive results; however, NACLIP significantly outperforms ClearCLIP on ADE20K and Cityscapes. The former requires the model to handle a large number of categories, while the latter demands fine-grained localization of small objects. Based on this observation, we choose NACLIP as our primary teacher model, leveraging its neighbor attention mechanism to enhance the student model’s performance on these challenging tasks.

Table 5: **Dataset specific details for zero-shot open-vocabulary semantic segmentation.** We list the per-dataset resolution, crop size, and stride used for each dataset. We maintain the same settings for all methods within a given dataset.

Dataset	VOC21	PC 60	Object	VOC20	PC 59	Stuff	City	ADE
Resize resolution	448	448	336	336	448	448	560	448
Crop size	336	336	336	336	336	336	224	336
Stride	112	112	112	112	112	112	112	112

## 2 Implementation details for Quantitative Evaluation

In this section, we provide detailed implementation information for our quantitative evaluation experiments. In section 2.1, we present the evaluation details for zero-shot open-vocabulary semantic segmentation (OVSS). In section 2.2, we describe the evaluation details for linear probe based semantic segmentation and monocular depth estimation.

### 2.1 Implementation details of zero-shot open-vocabulary semantic segmentation.

We follow SCLIP and NACLIP in the setup for the open-vocabulary semantic segmentation evaluation. For fairness, we utilize the same parameters for all models. We resize input images such that the shorter side is scaled to a specific resolution, while maintaining the original aspect ratio for the longer side. Additionally, we set fixed crop sizes and strides during evaluation. All evaluation parameters are summarized in Table 5, while all other settings follow their default configurations.

### 2.2 Implementation details of linear probe based evaluation.

Our linear probe evaluation follows prior work (Vision Transformers Need Registers, Denoising Vision Transformers), where a linear layer is trained as a decoding head to predict pixel-wise segmentation or depth logits.

**Semantic Segmentation.** We extract the final output features from the frozen backbone and, if applicable, pass them through the denoiser (for the DVT baseline). A single learnable linear layer is then trained to predict the segmentation logits. For CLIP-based models, both training and testing images are resized to (448, 448), while for DINOv2, the images are resized to (518, 518).

**Monocular Depth Estimation.** Similar to semantic segmentation, we extract features from the backbone, and pass them through the denoiser if applicable. Following the method in DVT and DINOv2, we then append the [CLS] token to each patch token to enrich the feature representations for all methods. Since our method also learns register tokens, we also append these to the features for the linear probe. We found appending registers to be only helpful for CLIP based models. For DINOv2, we observe better results when only the [CLS] token is appended, and do not use registers for the linear probe. A linear layer is trained using SigLoss and gradient loss (scaled by a factor of 0.5) to predict depth values into 256 uniformly distributed bins. We adopt DVT’s learning rate of  $5e-3$  for all experiments.

80 **3 Additional Qualitative Examples**

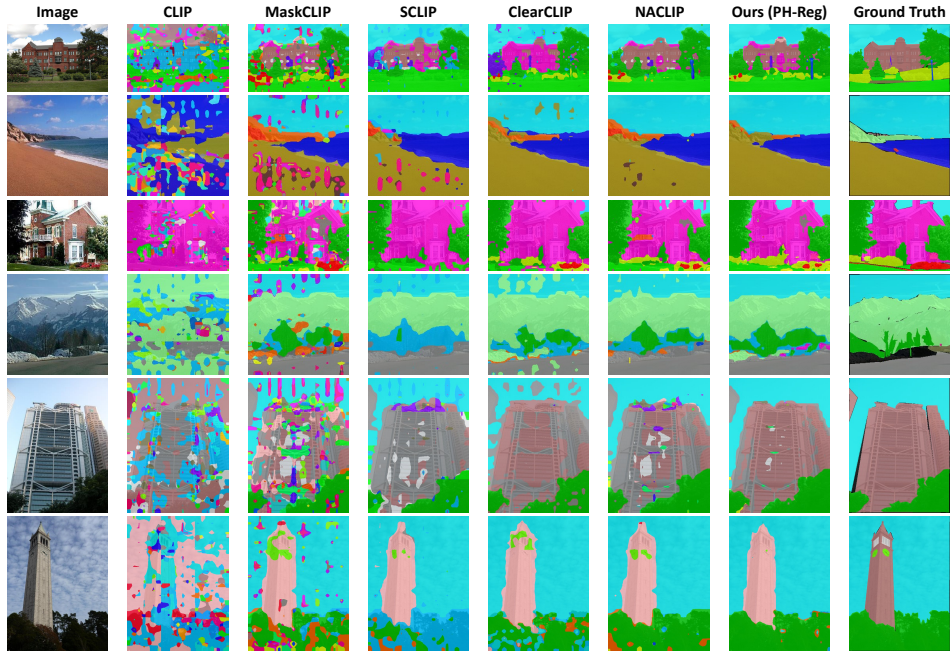


Figure 1: Open-vocabulary semantic segmentation qualitative comparison between different baseline models on ADE20K.

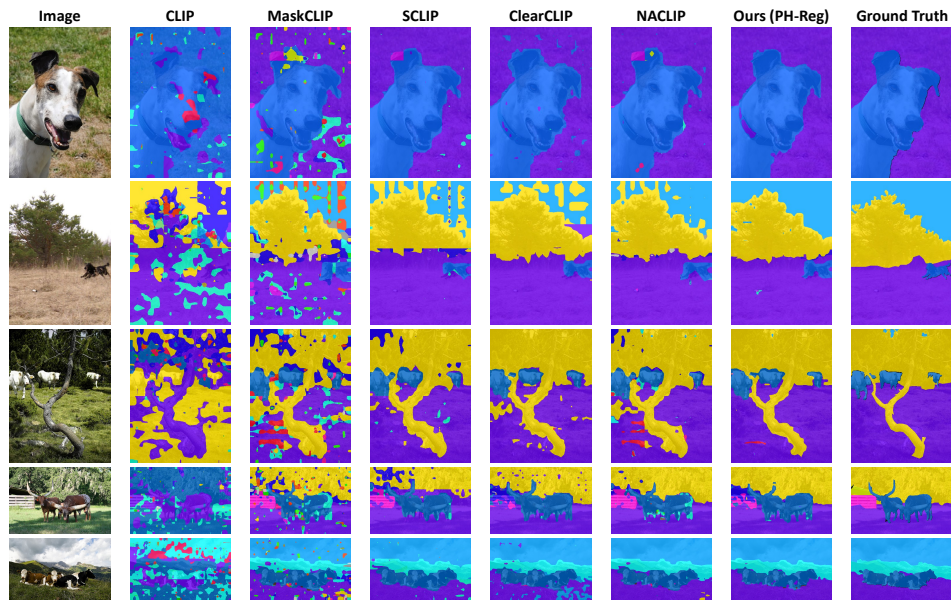


Figure 2: Open-vocabulary semantic segmentation qualitative comparison between different baseline models on Pascal Context59.

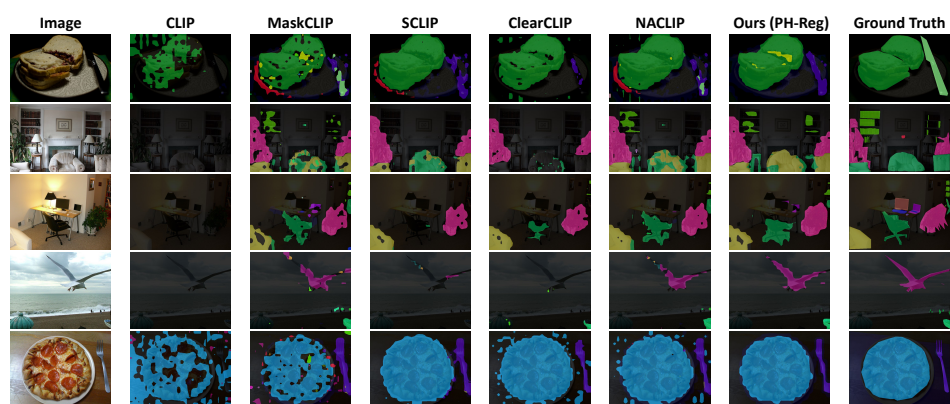


Figure 3: Open-vocabulary semantic segmentation qualitative comparison between different baseline models on COCO Obejct.

81 **4 Additional Qualitative Heatmaps for PH-Reg Zero-Shot**

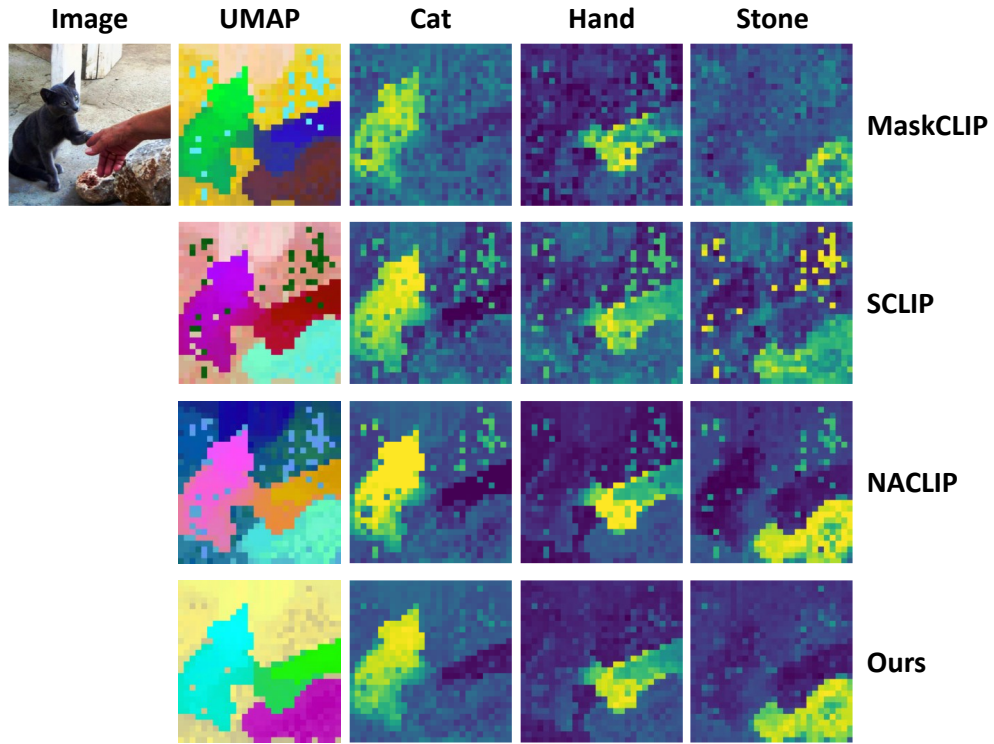


Figure 4: **Zero-shot heatmap results.** Our results have fewer artifacts than other methods.



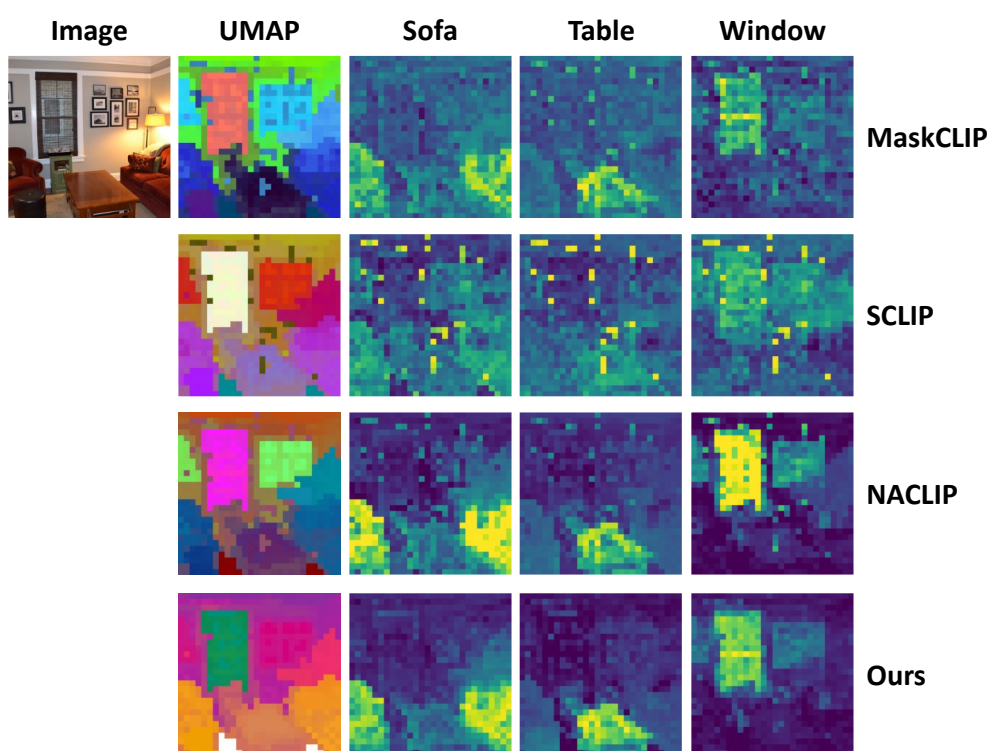


Figure 5: **Zero-shot heatmap results.** Our results have fewer artifacts than other methods.

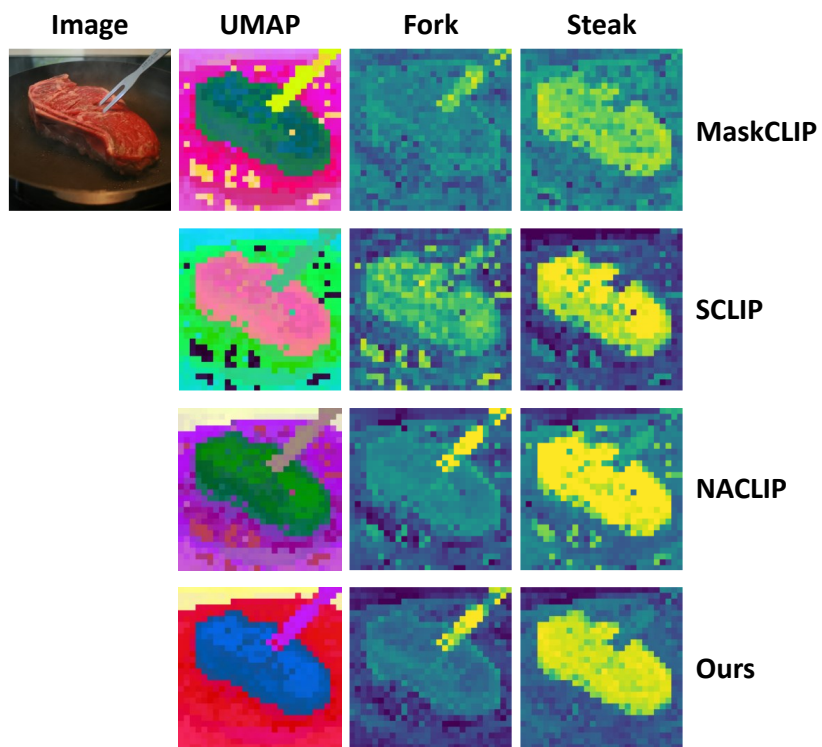


Figure 6: **Zero-shot heatmap results.** Our results have fewer artifacts than other methods.

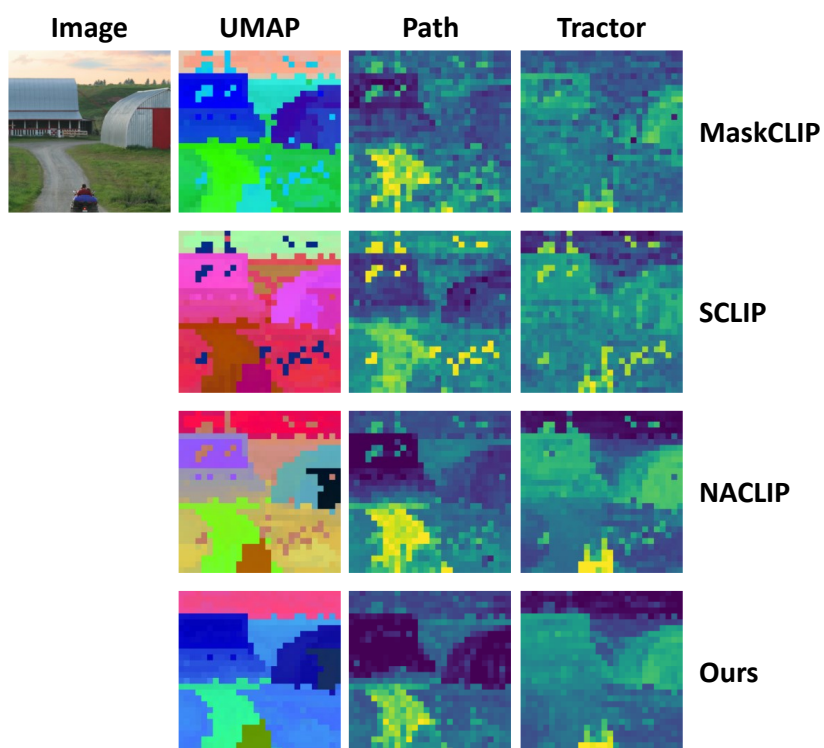


Figure 7: **Zero-shot heatmap results.** Our results have fewer artifacts than other methods.

## 82 5 Optimal Feature Aggregation

83 Let  $\mathbf{f}_1, \dots, \mathbf{f}_n \in \mathbb{R}^d$  be augmented feature vectors from  $n$  different transformations of an input image  
 84  $\mathcal{I}$ . We seek the optimal aggregated feature  $\mathbf{f}^*$  that minimizes the total squared error:

$$\mathbf{f}^* = \arg \min_{\mathbf{f}} \sum_{i=1}^n \|\mathbf{f}_i - \mathbf{f}\|_2^2 \quad (1)$$

85 Expanding the objective:

$$\sum_{i=1}^n (\mathbf{f}_i^\top \mathbf{f}_i - 2\mathbf{f}_i^\top \mathbf{f} + \mathbf{f}^\top \mathbf{f}) \quad (2)$$

86 Dropping constant terms that do not affect the result and simplifying:

$$= n\mathbf{f}^\top \mathbf{f} - 2 \left( \sum_{i=1}^n \mathbf{f}_i \right)^\top \mathbf{f} \quad (3)$$

87 We multiply and divide the right side by  $n$ :

$$= n\mathbf{f}^\top \mathbf{f} - 2n \left( \sum_{i=1}^n \frac{1}{n} \cdot \mathbf{f}_i \right)^\top \mathbf{f} \quad (4)$$

88 Dividing the equation by  $n$  as whole shows us that we need to minimize:

$$\|\mathbf{f} - \frac{1}{n} \sum_{i=1}^n \mathbf{f}_i\|_2^2 \quad (5)$$

89 So it can be derived that the mean of the feature vectors is the minimizer under MSE loss:

$$\mathbf{f}^* = \frac{1}{n} \sum_{i=1}^n \mathbf{f}_i \quad (6)$$