# VIPAMIN: Visual Prompt Initialization via Embedding Selection and Subspace Expansion

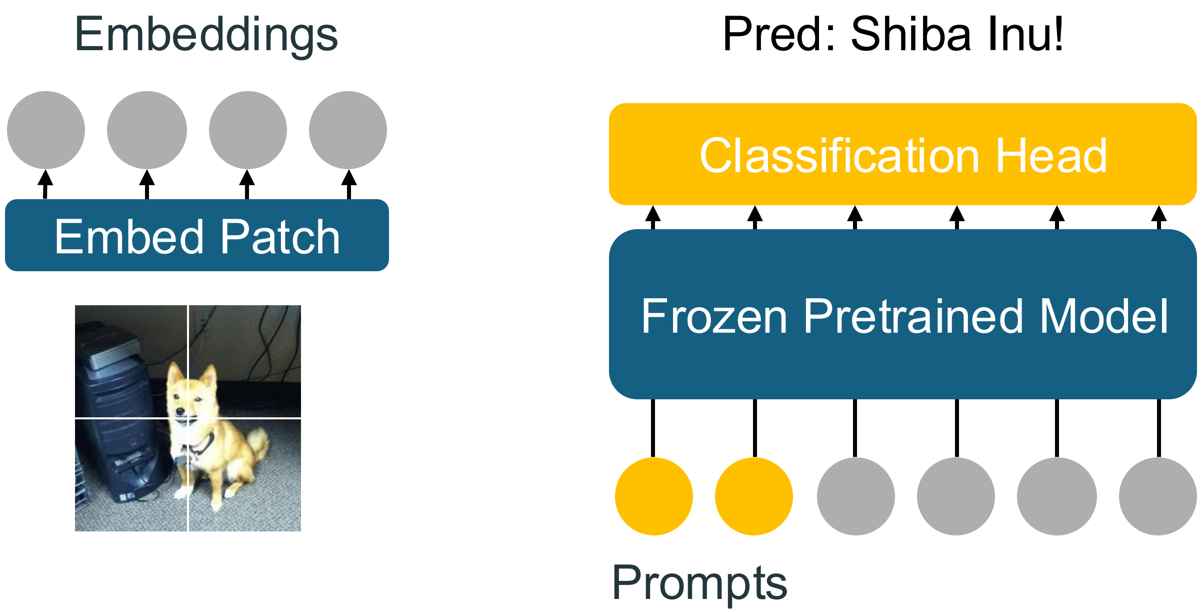
NeurIPS 2025 Poster

Jaekyun Park, Hye Won Chung

KAIST, School of Electrical Engineering

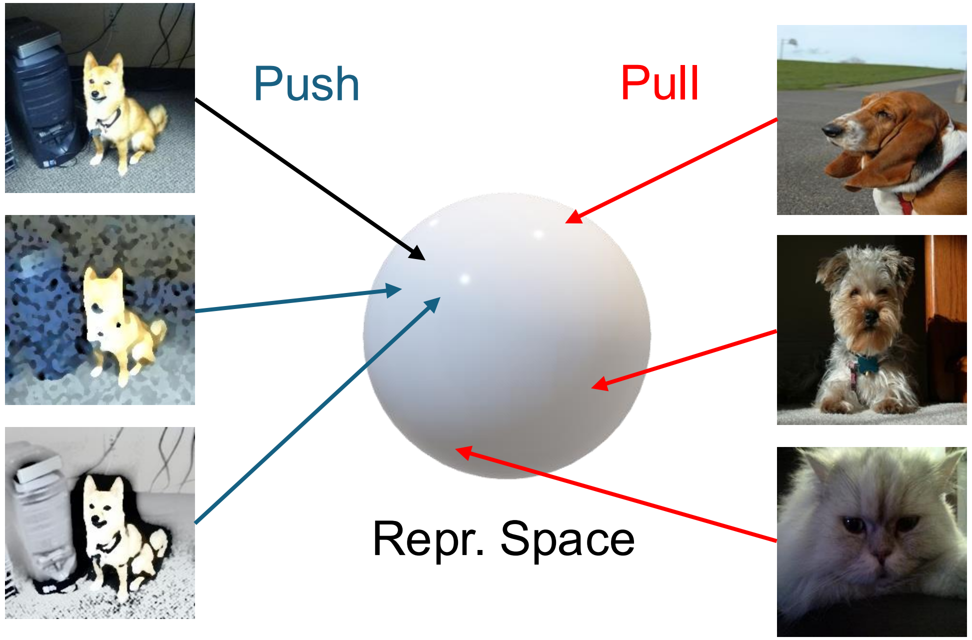
# INTRODUCTION

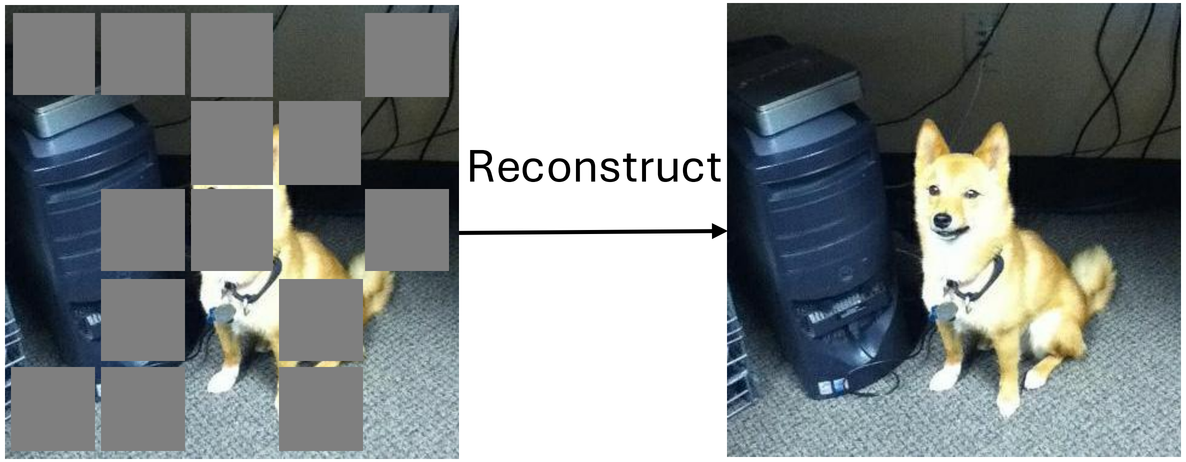
# VISUAL PROMPT TUNING




- Efficient alternative of full fine-tuning
- Introducing a small number of trainable tokens (prompts)

# SELF-SUPERVISED LEARNING



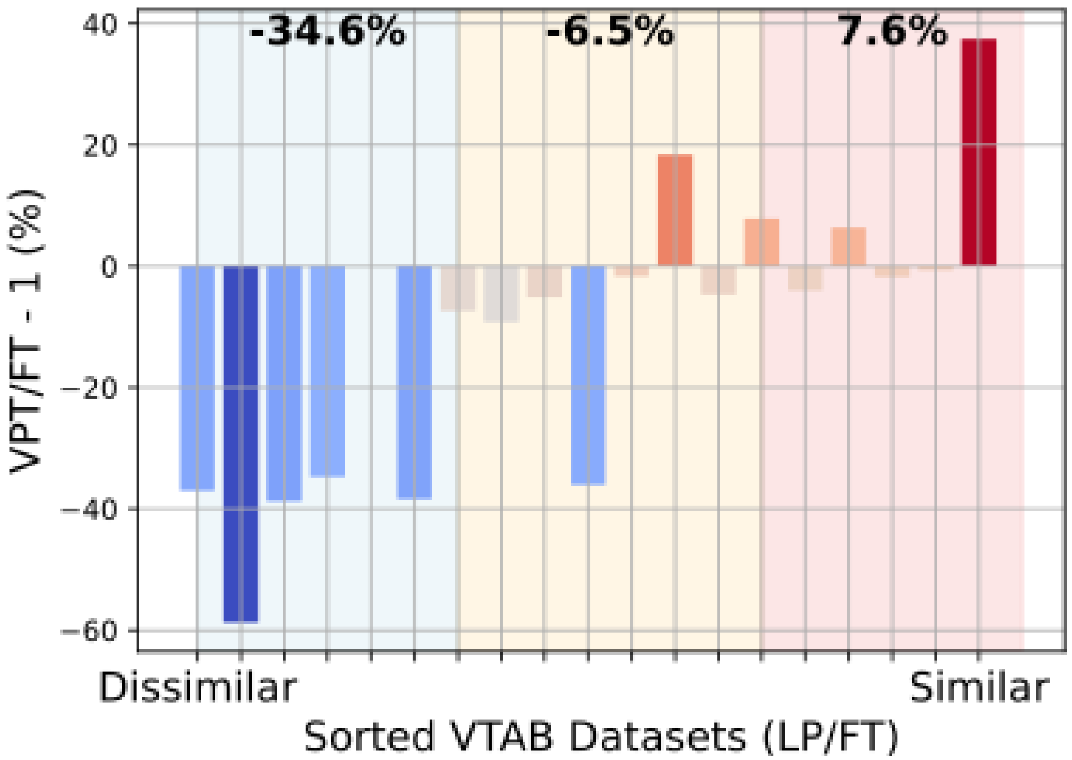

- Pretraining from large unlabeled datasets
- Contrastive learning (e.g., MoCo-v3), Masked image modeling (e.g., MAE)

# MOTIVATION

# MOTIVATION

Observation 1

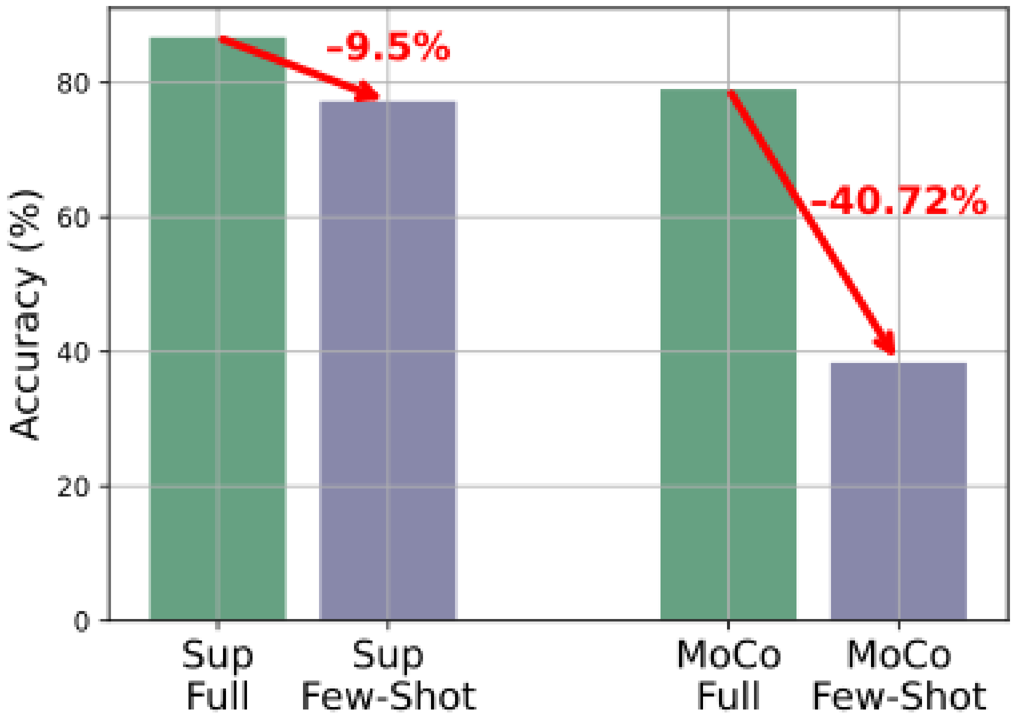
Large performance gap between VPT and fine-tuning on dissimilar tasks



# MOTIVATION

Observation 2

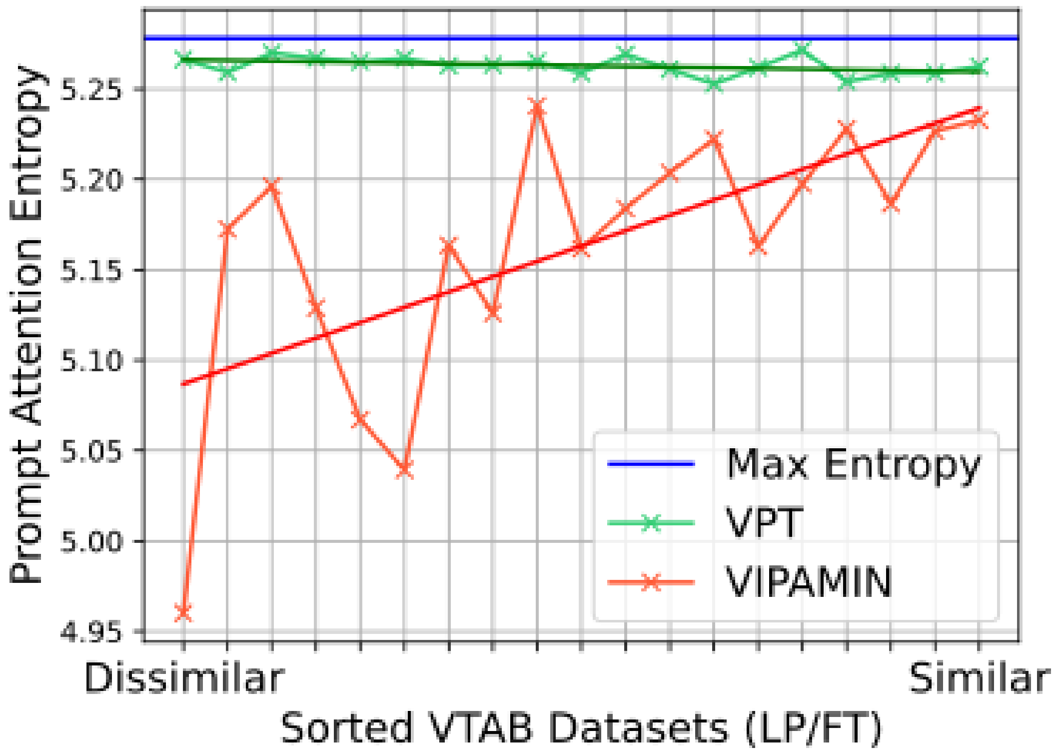
Self-supervised VPT fails to adapt in few-shot regimes



# MOTIVATION – FAILURE MODES OF PROMPT

Uniform Attention

Cross-attention between prompt and embeddings show that prompts do not differentiate image tokens (i.e., spurious background and object are treated equal)

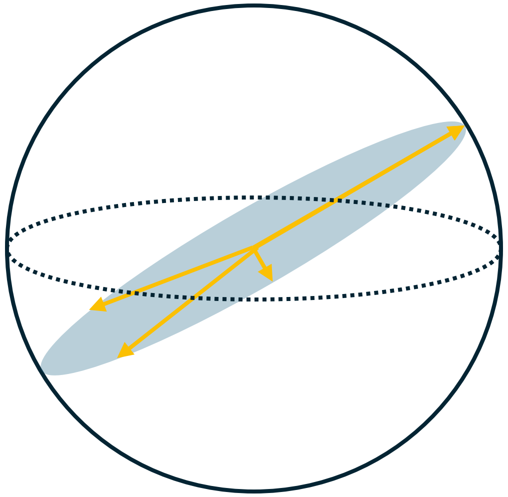


# MOTIVATION – FAILURE MODES OF PROMPT


## Prompt Subspace Collapse

Prompts collapse to pretrained self-attention space, not increasing the rank of new representation (even under the least similar task)

Dimensional Collapse of Trained Prompt

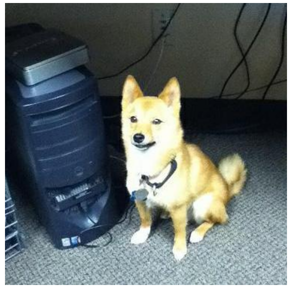


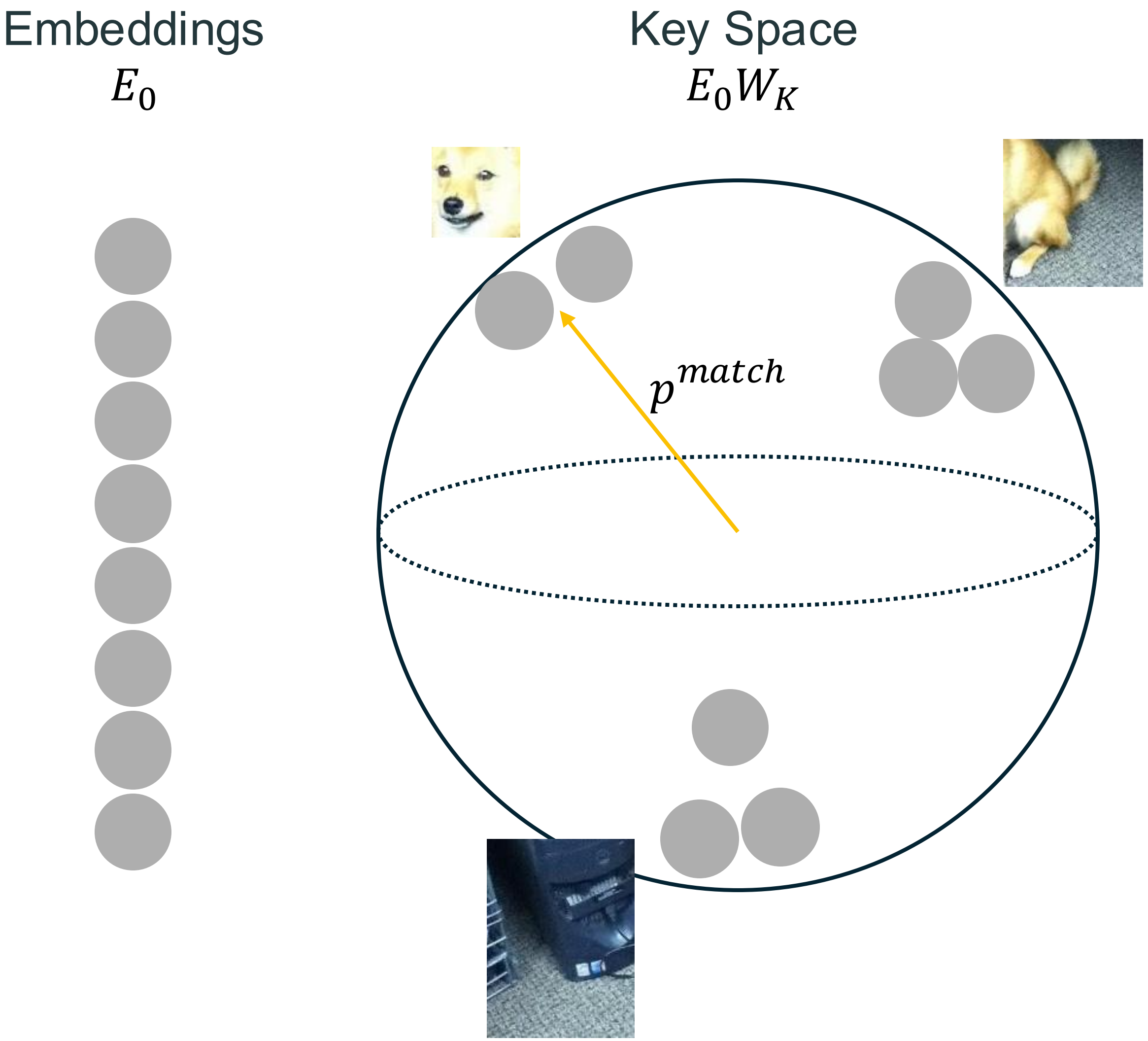
# METHODOLOGY

# METHODOLOGY – VIPAMIN

Specialization via Matching Module

Prompt is initialized with
average of embeddings
that share similar semantics

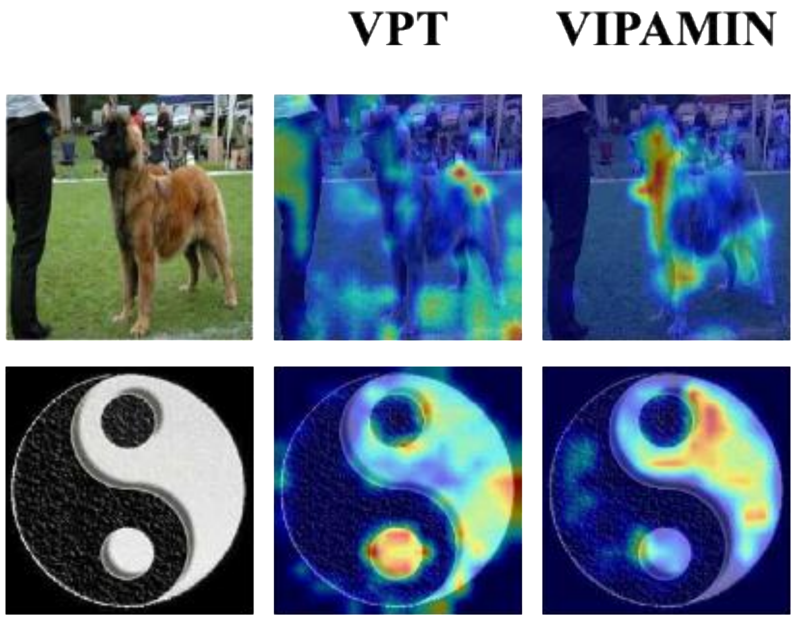




# METHODOLOGY – VIPAMIN

Specialization via Matching Module

Leads to more localized attention

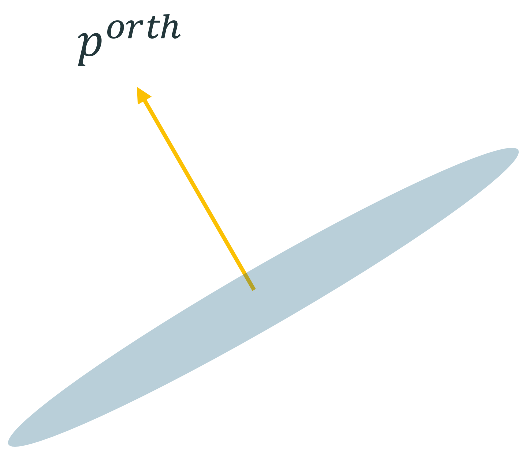



Grad CAM Visualization

# METHODOLOGY – VIPAMIN

## Novelty via Orthogonalizing Module

New directions beyond the pretrained space are injected




Frozen Self-Attention Space

$$p^{init} = (1 - \lambda)p^{match} + \lambda p^{orth}$$

Hyperparameter $\lambda$ tunes the strength of Orth. module

1) Achieving this requires a more nuanced technical approach than just simple orthogonalization. Refer to our paper for more details!

# KEY RESULTS

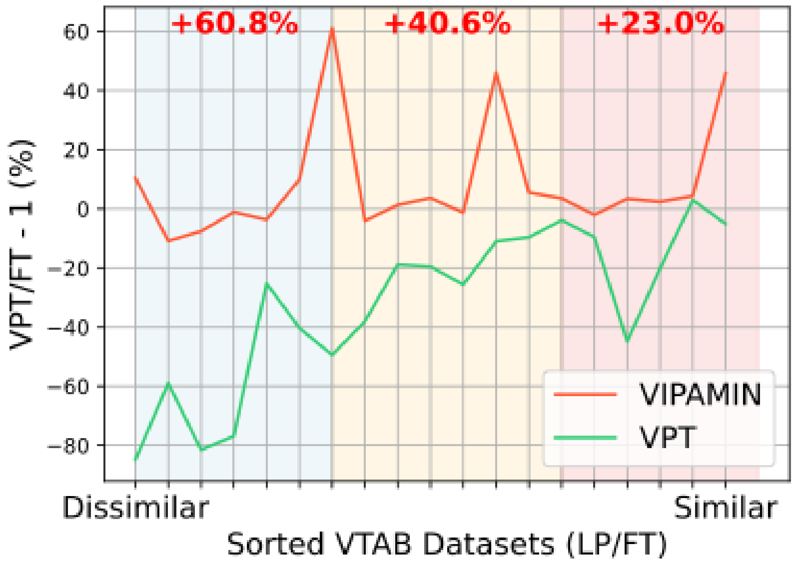
# KEY RESULTS – DISTRIBUTION-SHIFTED TASKS

**MoCo-v3 pretrained ViT-B/16**

| **Method** | *Natural* | *Specialized* | *Structured* | **Mean** |
|---|---|---|---|---|
| Full | 71.95 | **84.72** | 51.98 | 66.23 |
| VPT | 67.34 | 82.26 | 37.55 | 57.94 |
| GateVPT | 74.84 | 83.38 | 49.10 | 65.80 |
| SPT | 74.47 | 83.93 | 55.16 | 68.33 |
| **VIPAMIN** | **76.75** | 84.14 | **56.68** | **69.86** |

**MAE pretrained ViT-B/16**

| **Method** | *Natural* | *Specialized* | *Structured* | **Mean** |
|---|---|---|---|---|
| Full | 59.31 | 79.68 | 53.82 | 61.28 |
| VPT | 39.96 | 69.65 | 27.50 | 40.96 |
| GateVPT | 47.61 | 76.86 | 36.80 | 49.22 |
| SPT | 62.53 | **80.90** | 53.46 | 62.58 |
| **VIPAMIN** | **62.60** | 79.96 | **57.47** | **64.09** |



# KEY RESULTS – FEW-SHOT

| Method | $k=1$ | | | | | | $k=2$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CUB | Birds | Flowers | Dogs | Cars | **Mean** | CUB | Birds | Flowers | Dogs | Cars | **Mean** |
| VPT | 15.7 | 7.7 | 31.4 | 31.2 | 4.7 | 18.1 | 15.6 | 11.7 | 59.0 | 45.4 | 6.4 | 27.6 |
| SPT/rand | 17.2 | 11.7 | 48.9 | 35.5 | 5.3 | 23.7 | 29.8 | 22.5 | 70.4 | 49.0 | 10.9 | 36.5 |
| **VIPAMIN** | **20.1** | **12.6** | **52.8** | **37.5** | **5.7** | **25.8** | **36.0** | **23.1** | **71.6** | **49.4** | **11.1** | **38.2** |

| Method | $k=4$ | | | | | | $k=8$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CUB | Birds | Flowers | Dogs | Cars | **Mean** | CUB | Birds | Flowers | Dogs | Cars | **Mean** |
| VPT | 31.4 | 14.3 | 66.2 | 36.8 | 9.9 | 31.7 | 37.3 | 17.2 | 77.8 | 62.8 | 13.5 | 41.7 |
| SPT/rand | 51.7 | 40.5 | 84.6 | 59.8 | 21.7 | 51.7 | 66.6 | 55.0 | 92.9 | 69.1 | 43.8 | 65.5 |
| **VIPAMIN** | **53.7** | **41.0** | **85.1** | **60.3** | **21.9** | **52.4** | **68.6** | **55.1** | **94.3** | **70.0** | **43.8** | **66.4** |

# CONCLUSION

# CONCLUSION

VIPAMIN is simple, efficient, and effective

- Takes only about 30 seconds
- Solves prompt specialization and representational collapse in self-supervised models
- Improves adaptation to various tasks without adding parameters or complexity

Furthermore…

- Generalizes across modality (language), model scale, and architecture
- Also works well with zero-shot out-of-distribution generalization
- Check out our poster session for more details

THANK YOU FOR LISTENING!