Appendix: Unsupervised Embedding Adaptation via Early-Stage Feature Reconstruction for Few-Shot Classification

A. Preprocessing

In this section, we describe the preprocessing including equations in our paper. Assume we are given the embedding support set S_f and embedding query set Q_f. We apply centering and l2-normalization to the embedding samples for reconstruction training as described in (A.2). Preprocessed embeddings $z \in Z_{\text{preprocess}}$ are used as an input to the reconstruction module g_ϕ. The same preprocessing (centering and l2-normalization) is applied at the output of reconstruction module to compute the reconstruction loss $\mathcal{L}_{\text{FR}}$ as in (A.5).

$$\bar{z} = \frac{1}{|S_f \cup Q_f|} \sum_{z \in S_f \cup Q_f} z \tag{A.1}$$

$$S_{\text{preprocessed}} = \left\{ (z', y) | z' = \frac{z - \bar{z}}{\|z - \bar{z}\|_2}, (z, y) \in S_f \right\}, \quad Q_{\text{preprocessed}} = \left\{ z' | z' = \frac{z - \bar{z}}{\|z - \bar{z}\|_2}, z \in Q_f \right\} \tag{A.2}$$

$$Z_{\text{preprocessed}} = S_{\text{preprocessed}} \cup Q_{\text{preprocessed}} \tag{A.3}$$

$$\bar{z}_\phi = \frac{1}{|Z_{\text{preprocessed}}|} \sum_{z \in Z_{\text{preprocessed}}} \mathbb{E}_\mu [g_\phi(z \odot \mu)] \tag{A.4}$$

$$\mathcal{L}_{\text{FR}}(\phi) = -\frac{1}{|Z_{\text{preprocessed}}|} \sum_{z \in Z_{\text{preprocessed}}} \mathbb{E}_\mu \left[z^T \frac{g_\phi(z \odot \mu) - \bar{z}_\phi}{\|g_\phi(z \odot \mu) - \bar{z}_\phi\|_2} \right] \tag{A.5}$$

As new embeddings for few-shot classification, we apply only l2-normalization since it performs the best. The new embedding sets S^{ESFR} and Q^{ESFR} for the few-shot classification task are as follows:

$$S^{\text{ESFR}} = \left\{ (z', y) | z' = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{g_{\phi_*^i}(z)}{\|g_{\phi_*^i}(z)\|_2}, (z, y) \in S_{\text{preprocessed}} \right\} \tag{A.6}$$

$$Q^{\text{ESFR}} = \left\{ z' | z' = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{g_{\phi_*^i}(z)}{\|g_{\phi_*^i}(z)\|_2}, z \in Q_{\text{preprocessed}} \right\} \tag{A.7}$$

For BD-CSPN (Liu et al., 2020) and our method used with BD-CSPN, additional shifting-term is added for query samples before preprocessing. In this case, we define $S_{\text{preprocess}}$ and $Q_{\text{preprocess}}$ as follows:

$$\text{shifting-term: } \triangle = \frac{1}{S_f} \sum_{z_s \in S_f} z_s - \frac{1}{Q_f} \sum_{z_q \in Q_f} z_q \tag{A.8}$$

$$Q_f^{\text{shifted}} = \left\{ z' | z' = z + \triangle, z \in Q_f \right\} \tag{A.9}$$

$$\bar{z} = \frac{1}{|S_f \cup Q_f^{\text{shifted}}|} \sum_{z \in S_f \cup Q_f^{\text{shifted}}} z \tag{A.10}$$

$$S_{\text{preprocessed}} = \left\{ (z', y) | z' = \frac{z - \bar{z}}{\|z - \bar{z}\|_2}, (z, y) \in S_f \right\}, \quad Q_{\text{preprocessed}} = \left\{ z' | z' = \frac{z - \bar{z}}{\|z - \bar{z}\|_2}, z \in Q_f^{\text{shifted}} \right\} \tag{A.11}$$

B. Comparison to TIM

Table B.1. Table describes the performance comparison with TIM (Boudiaf et al., 2020) of 5-way 1- and 5-shot accuracies (in %) on *mini*-ImageNet, *tiered*-ImageNet and CUB. The performance of TIM-GD is from the paper (Boudiaf et al., 2020). We use preprocessing with shifting-term to acquire new embeddings for TIM-GD + ESFR since it performs better.

		***mini*-ImageNet**		***tiered*-ImageNet**		**CUB**	
Method	**Backbone**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
TIM-GD (Boudiaf et al., 2020)	ResNet-18	73.9	**85.0**	79.9	**88.5**	82.2	**90.8**
TIM-GD + ESFR	ResNet-18	**76.02**	84.42	**82.03**	88.11	**84.69**	90.43
TIM-GD (Boudiaf et al., 2020)	WRN	77.8	**87.4**	82.1	**89.8**	-	-
TIM-GD + ESFR	WRN	**79.25**	86.38	**83.58**	89.44	-	-

We separately compare the performance of our method with TIM (Boudiaf et al., 2020) since we believe TIM uses a strong prior that query samples per class are balanced in the standard few-shot classification benchmarks (e.g., 15-query samples per class); while our method combined with the baseline methods (NN, Linear, BD-CSPN (Liu et al., 2020)) does not utilize any query statistics. We find that TIM's proposed regularization term with conditional entropy and label-marginal entropy forces balancing among the predicted number of query samples per class. To be specific, the conditional entropy minimization term encourages the classification model to output confident prediction and the label-marginal entropy maximization term encourages marginal predicted label distribution to be uniform. When both conditional and label-marginal entropy terms are used simultaneously, the predicted labels close to one-hot from uniform class distribution, resulting in the same number of predicted samples for each class. This seems helpful in the balanced query class distribution setting where all query samples per class are the same, but its possible use case will be limited. We find that query class imbalance setting can easily ruin TIM's performance in Section D.

To investigate our method when using query statistics, we experiment with TIM-GD + ESFR. Table B.1 shows the results on standard mini-ImageNet, tiered-ImageNet, and CUB with 5-way 1- and 5-shot settings. For 1-shot settings, our method improves the performance of TIM by 1.5%$\sim$2.5% across all datasets and backbones. As mentioned before (in Section 5.3), this indicates that our method can offer a complementary improvement to semi-supervised learning techniques such as TIM for 1-shot. For 5-shot settings, our method decreases the performance by 0.4%$\sim$1.0%. The decrease in 5-shot performance encourages further research about the simultaneous use of pseudo-label information and unsupervised information, which we leave as future work.

C. Ablation: noise level of dropout

Table C.1. The table shows the influence of drop-rate applied to our method. We experimented with ResNet-18 backbone on *mini*-ImageNet and *tiered*-ImageNet.

	***mini*-ImageNet**		***tiered*-ImageNet**	
rate	**1shot**	**5shot**	**1shot**	**5shot**
0.	68.90	81.53	75.39	85.31
0.1	69.41	81.59	75.90	85.50
0.2	69.90	81.70	76.39	85.63
0.3	70.39	**81.71**	76.78	85.71
0.4	70.63	**81.71**	77.23	85.77
0.5	**70.94**	81.61	**77.44**	**85.84**

We further investigate the effect of the dropout noise level. In the main text, we argued that multiplicative noise by dropout seems well suited for our method. Experiments in Table C.1 with various drop-rate show that the dropout can be used in our method without careful tuning.

D. Few-shot classification with imbalance query class distribution

To verify our method's robustness on various query settings, we experiment with the setting when the numbers of query samples per class are imbalanced. We set the number of query samples per class as (11, 13, 15, 17, 19) and (7, 11, 15, 19, 23). Table D.1 shows that our method consistently improves the performance of few-shot classification regardless of the query imbalance setting. To be more specific, the improvement by our method in different query settings varies within $< 1.5\%$; thus, our method is robust to different query settings. In contrast, TIM (Boudiaf et al., 2020) that uses the strong prior about

Table D.1. This table shows few-shot classification performance when the numbers of query samples per class are imbalanced. For standard settings, the number of query samples per class is equally 15, given as (15, 15, 15, 15, 15). For the imbalance case, we set the number of query samples per class as (11, 13, 15, 17, 19) and (7, 11, 15, 19, 23). The ± describes 95% confidence interval. For these results, we use our implementation version of TIM-GD (Boudiaf et al., 2020), which matches the original paper's performance. For BD-CSPN (Liu et al., 2020) with an imbalance number of query samples, we do not use shift-term since it worsens the performance.

***mini*-ImageNet**							
Method	**Backbone**	**(15, 15, 15, 15, 15)**		**(11, 13, 15, 17, 19)**		**(7, 11, 15, 19, 23)**	
		1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
TIM-GD	ResNet-18	73.67±0.33	85.01±0.19	68.93±0.30	79.05±0.17	66.04±0.28	75.60±0.16
NN	ResNet-18	64.04±0.44	79.71±0.32	63.73±0.46	80.01±0.33	63.25±0.47	79.88±0.33
+ESFR	ResNet-18	70.94±0.50	81.61±0.33	70.32±0.52	81.35±0.33	69.74±0.53	81.12±0.34
BD-CSPN	ResNet-18	70.00±0.51	82.36±0.32	68.99±0.51	81.49±0.34	68.26±0.52	81.12±0.34
+ESFR	ResNet-18	73.98±0.55	82.32±0.33	72.39±0.56	81.51±0.34	71.74±0.57	81.17±0.35
TIM	WRN	77.60±0.31	87.31±0.17	72.03±0.28	80.91±0.16	68.86±0.26	77.28±0.15
NN	WRN	66.73±0.44	81.85±0.31	66.64±0.46	82.07±0.31	66.30±0.47	81.98±0.32
+ESFR	WRN	74.01±0.51	83.58±0.31	73.34±0.51	83.27±0.32	72.89±0.52	83.03±0.33
BD-CSPN	WRN	72.74±0.49	84.14±0.30	71.67±0.51	83.34±0.32	71.19±0.51	83.02±0.33
+ESFR	WRN	76.84±0.54	84.36±0.32	75.26±0.55	83.48±0.33	74.66±0.55	83.09±0.34
***tiered*-ImageNet**							
Method	**Backbone**	**(15, 15, 15, 15, 15)**		**(11, 13, 15, 17, 19)**		**(7, 11, 15, 19, 23)**	
		1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
TIM-GD	ResNet-18	79.99±0.33	88.62±0.20	74.21±0.29	81.93±0.18	70.95±0.28	78.36±0.17
NN	ResNet-18	71.60±0.49	84.62±0.36	71.10±0.49	84.59±0.35	70.51±0.49	84.52±0.35
+ESFR	ResNet-18	77.44±0.52	85.84±0.35	76.77±0.53	85.64±0.35	76.21±0.54	85.42±0.36
BD-CSPN	ResNet-18	77.28±0.52	86.55±0.34	76.38±0.52	85.89±0.35	75.63±0.53	85.65±0.36
+ESFR	ResNet-18	80.13±0.56	86.34±0.36	78.72±0.57	85.76±0.36	78.12±0.57	85.50±0.37
TIM	WRN	82.18±0.32	89.87±0.19	76.11±0.28	83.18±0.17	72.72±0.27	79.55±0.16
NN	WRN	72.97±0.49	85.74±0.34	72.17±0.48	85.79±0.34	71.57±0.49	85.70±0.34
+ESFR	WRN	79.13±0.52	87.08±0.34	78.30±0.53	86.90±0.34	77.67±0.53	86.69±0.34
BD-CSPN	WRN	78.89±0.52	87.72±0.32	77.71±0.52	87.11±0.34	77.05±0.53	86.86±0.35
+ESFR	WRN	81.77±0.55	87.61±0.34	80.50±0.55	87.04±0.35	79.67±0.56	86.72±0.35

query statistics, suffers from the change in query setting. The performance of TIM on the 5-shot when the number of query samples per class is (7, 11, 15, 19, 23) shows $-6.2\%\sim-4.2\%$ performance decrease compare to the baseline (NN).

E. Naively applied unsupervised learning methods

Table E.1. Experiment settings

	Candidates
Learning rate	1e-3, **1e-4**
Classifier	Linear, **Cosine**
Additional module	**2-layer FCN**, None
update weights of embedding networks	None, All, **only-the-last-residual-block**
New embeddings	**Backbone output**, Additional module output

We experiment if naively applied unsupervised (or self-supervised) learning can improve few-shot classification in the standard settings. For a fair comparison, we use the pre-trained embeddings of ResNet-18 on *mini*-ImageNet. We test with pretext task-based self-supervised methods of rotation (Gidaris et al., 2018) and jigsaw (Noroozi & Favaro, 2016). For both methods, we use grid search to find the best performing settings; shown in Table E.1. An additional module is inserted between the embedding network and classifier and we use hidden dimensions from Su et al. (2020). For jigsaw tasks, we use 35-permutations from Su et al. (2020). For both methods, the same setting with the bold font on Table E.1 performs the best.

Table-E.2 shows the results with (1) new embeddings are provided when training becomes converged and (2) new embeddings are given via oracle early stopping at the best performing training iteration (for ≥ 1). The result with converged embeddings shows that the naively applied self-supervised learning fails to improve few-shot classification performance. Note that our

Table E.2. Naively applied unsupervised learning results

Method	*mini*-ImageNet 1-shot accuracy	
NN	64.04	
+ESFR	70.94 +6.90	
	(1) Converged	**(2) Oracle early stopping**
Jigsaw	33.1	67.22 +3.18
Rotation	32.2	66.70 +2.66

method achieves 70.94 on the same setting. Our method outperforms both the rotation- and jigsaw-based[1] unsupervised learning methods that even contain oracle early-stopping.

F. Experiments with Conv4

Table F.1. The table shows the experimental results of our method with Conv4-64 backbone on *mini*-ImageNet and *tiered*-ImageNet.

	mini-ImageNet		tiered-ImageNet	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
NN	50.72	67.17	52.18	69.60
+ ESFR	**54.63** +3.91	**68.32** +1.15	**57.56** +5.38	**71.46** +1.86
BD-CSPN	52.73	68.5	54.94	71.53
+ ESFR	**56.24** +3.51	**69.18** +0.68	**60.16** +5.22	**72.37** +0.84

We use pre-trained Conv4-64 backbones following the settings of Wang et al. (2019). We applied the same preprocessing strategy as in the main text. For the reconstruction module, we find that the bottleneck structure (Section 3.1) is helpful for Conv4-64; while reconstructed embeddings still outperform the encoded ones. Thus, we use 800-400-800-1600 as hidden dimensions.

Table-F shows experimental results with Conv4-64. As in the experimental results with ResNet and WideResNet, our method consistently improves the performance of the baseline methods: NN and BD-CSPN. ESFR also offers a complementary improvement to BD-CSPN (Liu et al., 2020) in 1-shot settings. Compare to prior state-of-the-art methods with Conv4-64, our method with BD-CSPN has slightly lower performance. [2]

References

Boudiaf, M., Ziko, I. M., Rony, J., Dolz, J., Piantanida, P., and Ayed, I. B. Information maximization for few-shot learning. In *NeurIPS*, 2020.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Hu, S. X., Moreno, P. G., Xiao, Y., Shen, X., Obozinski, G., Lawrence, N. D., and Damianou, A. C. Empirical bayes transductive meta-learning with synthetic gradients. In *ICLR*, 2020.

Liu, J., Song, L., and Qin, Y. Prototype rectification for few-shot learning. In *ECCV*, 2020.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Su, J.-C., Maji, S., and Hariharan, B. When does self-supervision improve few-shot learning? In *ECCV*, 2020.

Wang, Y., Chao, W., Weinberger, K. Q., and van der Maaten, L. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[1] We improved the performance with the jigsaw task after the rebuttal period. Originally performance with rotation task performs better.

[2] Hu et al. (2020) shows 58.0% and 70.7% accuracies on *mini-ImageNet in 1- and 5-shot settings, respectively.*