Appendix for *Instance-based Max-margin for Practical Few-shot Recognition*

In this appendix, we present comprehensive implementation details for experiments conducted in both the proposed pFSL and traditional FSL settings. Additionally, we offer more in-depth analysis and results of our method.

1. Implementation Details

First, we describe the common details in all experiments. The features z extracted by the backbone $\mathcal{M}$ were $L2$-normalized. Whenever a linear classifier was trained, we always used Adam as the optimizer whose learning rate was gradually decreased in a cosine scheduling, and the label smoothed cross-entropy loss [14] was used as the learning objective.

1.1. Setup for pFSL

The pre-trained model $\mathcal{M}$ was trained using various self-supervised learning methods [1, 3–5, 9] on the ImageNet-1K [13] training set. We obtained the pre-trained models from their respective official implementations. To store the features extracted by the backbone $\mathcal{M}$ in advance for conveniently conducting experiments, images were resized to 256 pixels along the shorter side using bicubic resampling, then were center cropped to 224 $\times$ 224 pixels.

For the proposed IbM2, we always sampled 200 virtual examples for every original training instance (*i.e.*, $R = 200$). We initially set the value of accuracy threshold T in Algorithm 1 of the main paper as 0.9. By preliminarily training a linear classifier on the original training set D before performing Algorithm 1, we got the empirical training accuracy upper bound (denoted as ACC$_{up}$), set T as the minimum of its initially predefined value (0.9 here) and ACC$_{up}$ (where $T = \min\{0.9, \text{ACC}_{up}\}$ in this case) to ensure its validity. During the process of searching for the best value of ϵ, as shown in Algorithm 1, we initialized a linear classifier once, and repeatedly trained it with learning rate 1.0 for 20 epochs until the searching range became tight enough.

Next we trained a new randomly initialized linear classifier for IbM2 (based on $D^{\hat{\epsilon}}$, the set of virtual examples) or the baseline method (based on the original training set D) for reporting the evaluation accuracy on test set. In this stage, the learning rate was initialized as 0.05 and scaled as *init_lr* $\times$ *batch_size / 256*. For ViT [7] architectures, the linear classifier was trained with batch size 256 for 100 epochs. For ResNet50 [10], the linear classifier was trained with batch size 512 for 60 epochs.

1.2. Setup for Traditional FSL

In this setup, the pre-trained models were generated by different approaches [6, 11, 12] on the base set. For PMF [11], following their guidance, we resized the input images into 224 $\times$ 224 pixels while keeping them in small resolutions for other two baselines [6, 12] (80 $\times$ 80 pixels in *mini*-ImageNet [15] and 32 $\times$ 32 pixels in CIFAR-FS [2]).

The initial value of T in the proposed IbM2 was set as 0.999. During the search for ϵ, the linear classifier was initialized once and trained for 50 epochs with learning rate 1.0 at each search step. To get the final linear classifier, we trained a new linear classifier with 200 epochs for a 5-way few-shot task on training set of baseline (original training set) and IbM2 (virtually sampled training set after the optimal $\hat{\epsilon}$ was got), respectively.

As suggested in PMF [11], we sampled a small number (20) of extra few-shot episodes sharing very similar semantics with the evaluated ones from the novel split on each scenario, in order to select the learning rate ranged in {0.00001, 0.0001, 0.001, 0.01, 0.1, 1} for the final training stage of IbM2 or the baseline. The selected optimal learning rate was used to train all 500 episodes of that scenario.

For PMF [11], we discarded its original fine-tuning pipeline of updating weights of the whole backbone. We only learned the linear classifier on top of the frozen features as in pFSL setting to make the comparison fairer.

We performed experiments on PMF [11] using ViT [7] and ResNet50 [10], $S2M2_R$ [12] using WRN-28-10 [16] and Meta-Baseline [6] using ResNet12 [10] as the backbone. The other implementation details of traditional FSL are the same with pFSL as aforementioned.

2. Further Comparative Analysis: Traditional FSL vs. pFSL

The core concept of few-shot learning is rapidly enabling a deep learning model to acquire new concepts, leveraging prior knowledge. Building upon this concept, as discussed in Section 3 of the main text, both traditional FSL and the proposed pFSL aim to achieve this objective, but through

Setting	Source of Prior Knowledge	Evaluation Way	Evaluation Run
Traditional FSL	Base Set	≤ 5	≥ 500
pFSL	Unsupervised Large-Scale Pre-trained model	≥ 200	≤ 3

Table 1. Elaborated comparison for traditional FSL and pFSL.

distinct approaches. As illustrated in Table 1, our approach diverges from traditional FSL by leveraging a model unsupervised pre-trained on large-scale datasets, which naturally encodes more general and domain-specific knowledge. This enables our pFSL to emulate the human ability to perform downstream tasks with few-shot capability. Additionally, pFSL exhibits much more simplicity and convenience during evaluation. Traditional FSL typically relies on a small base set as prior knowledge, necessitating multiple individual runs to ensure accuracy due to the inherent instability of one episode accuracy [8]. In contrast, pFSL's many-way characteristics streamline the evaluation process, requiring only three runs to achieve reliable results.

3. More Comparative Results

We present additional results to further highlight IbM2's overall performance in enhancing the pFSL setting across various pre-trained backbones. Table 2 serves as an extension of Table 1 in the main text, with a specific focus on the CUB dataset.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked Siamese Networks for label-efficient learning. In *European Conference on Computer Vision*, page 456–473. Springer, 2022. 1

[2] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, pages 1–15, 2019. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised Vision Transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021. 1

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised Vision Transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9620–9629, 2021. 1

[6] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-Baseline: Exploring simple meta-learning for few-shot learning. In *IEEE/CVF International Conference on Computer Vision*, pages 9042–9051, 2021. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pages 1–21, 2021. 1

[8] Minghao Fu, Yun-Hao Cao, and Jianxin Wu. Worst case matters for few-shot recognition. In *European Conference on Computer Vision*, page 99–115. Springer, 2022. 2

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, 2020. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[11] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9058–9067, 2022. 1

[12] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold Mixup for few-shot learning. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2207–2216, 2020. 1

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015. 1

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 1

[15] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for one shot learning. In *Advances in Neural Information Processing Systems*, page 3637–3645, 2016. 1

[16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, pages 1–12, 2016. 1

Dataset	Pre-training Method	Backbone	IbM2	Shot per Class						
				1	2	3	4	5	8	16
ImageNet-1K	DINO	ViT-S/16		39.2 ± 0.3	49.2 ± 0.2	54.1 ± 0.4	56.7 ± 0.2	58.0 ± 0.1	60.4 ± 0.1	62.7 ± 0.1
			✓	39.2 ± 0.3	49.4 ± 0.3	**54.6** ± **0.4**	**57.6** ± **0.1**	**59.3** ± **0.1**	**62.4** ± **0.2**	**65.8** ± **0.1**
	MoCov3	ViT-S/16		32.7 ± 0.6	42.0 ± 0.2	46.9 ± 0.3	49.6 ± 0.4	51.0 ± 0.1	53.8 ± 0.1	56.6 ± 0.2
			✓	**33.9** ± **0.6**	**43.2** ± **0.2**	**48.4** ± **0.3**	**51.3** ± **0.3**	**52.8** ± **0.2**	**56.1** ± **0.1**	**59.8** ± **0.2**
	MSN	ViT-S/16		47.9 ± 0.1	56.2 ± 0.4	59.8 ± 0.3	61.6 ± 0.1	62.4 ± 0.2	64.4 ± 0.3	66.1 ± 0.1
			✓	47.8 ± 0.2	56.4 ± 0.4	**60.5** ± **0.2**	**62.5** ± **0.2**	**63.6** ± **0.2**	**66.0** ± **0.2**	**68.4** ± **0.0**
		ViT-B/4		53.2 ± 0.2	64.5 ± 0.4	68.9 ± 0.2	70.9 ± 0.2	72.0 ± 0.3	73.8 ± 0.1	75.0 ± 0.2
			✓	**54.0** ± **0.1**	64.9 ± 0.5	**69.4** ± **0.2**	**71.4** ± **0.1**	**72.7** ± **0.4**	**74.7** ± **0.0**	**76.4** ± **0.2**
		ViT-L/7		57.3 ± 0.4	66.5 ± 0.4	69.8 ± 0.5	71.6 ± 0.4	72.2 ± 0.2	73.8 ± 0.1	75.1 ± 0.1
			✓	57.7 ± 0.4	66.6 ± 0.5	70.1 ± 0.6	71.8 ± 0.4	72.6 ± 0.2	**74.3** ± **0.1**	**76.0** ± **0.0**
	SimCLR	ResNet50		21.4 ± 0.4	30.3 ± 0.1	36.1 ± 0.3	39.8 ± 0.2	42.0 ± 0.1	46.8 ± 0.1	51.9 ± 0.0
			✓	**23.6** ± **0.4**	**33.4** ± **0.2**	**39.0** ± **0.4**	**42.0** ± **0.3**	**44.2** ± **0.1**	**48.0** ± **0.0**	**52.7** ± **0.0**
	BYOL	ResNet50		26.5 ± 0.3	35.7 ± 0.2	41.5 ± 0.4	45.1 ± 0.2	47.2 ± 0.1	51.8 ± 0.1	57.1 ± 0.1
			✓	**27.5** ± **0.3**	**37.5** ± **0.1**	**43.3** ± **0.4**	**46.8** ± **0.2**	**49.1** ± **0.1**	**53.2** ± **0.1**	**58.0** ± **0.1**
CUB	DINO	ViT-S/16		35.4 ± 1.2	49.0 ± 0.5	56.8 ± 0.8	60.8 ± 0.7	65.2 ± 0.9	70.6 ± 0.9	75.9 ± 0.3
			✓	**36.2** ± **1.4**	**49.6** ± **0.6**	**57.4** ± **1.0**	**62.0** ± **0.6**	**66.4** ± **0.8**	**72.5** ± **0.8**	**79.0** ± **0.2**
	MoCov3	ViT-S/16		18.4± 0.6	27.2 ± 0.2	35.5 ± 1.0	40.0 ± 0.6	45.3 ± 0.9	54.1 ± 0.4	65.3 ± 0.3
			✓	**19.2** ± **0.6**	27.4 ± 0.3	35.6 ± 0.7	40.0 ± 0.3	45.4 ± 0.7	54.2 ± 0.6	**65.8** ± **0.3**
	MSN	ViT-S/16		32.1 ± 1.6	45.0 ± 0.6	53.1 ± 0.6	56.7 ± 0.1	61.4 ± 0.5	67.3 ± 0.0	73.6 ± 0.4
			✓	**33.0** ± **1.4**	**45.8** ± **0.7**	53.2 ± 0.9	57.1 ± 0.4	**62.0** ± **1.0**	**68.4** ± **0.1**	**75.7** ± **0.2**
		ViT-B/4		35.8 ± 1.6	50.0 ± 0.2	58.8 ± 1.0	61.2 ± 1.0	67.2 ± 0.2	73.0 ± 0.7	79.4 ± 0.2
			✓	**38.1** ± **1.5**	**50.7** ± **0.2**	59.1 ± 1.0	**62.4** ± 1.6	67.5 ± 0.3	**73.5** ± **0.3**	**80.2** ± **0.3**
		ViT-L/7		34.9 ± 1.3	49.4 ± 0.4	58.8 ± 0.8	62.7 ± 0.9	67.2 ± 0.3	73.8 ± 0.8	80.4 ± 0.2
			✓	**37.5** ± **1.2**	**50.1** ± **0.5**	59.0 ± 0.8	62.6 ± 0.5	67.5 ± 0.6	73.9 ± 0.4	**81.0** ± **0.1**
	SimCLR	ResNet50		7.8 ± 0.3	11.4 ± 0.4	15.3 ± 0.2	17.3 ± 0.4	19.5 ± 1.2	25.6 ± 0.3	35.9 ± 0.3
			✓	**8.1** ± **0.3**	11.3 ± 0.4	15.2 ± 0.2	**17.8** ± 0.6	**20.1** ± 1.1	25.7 ± 0.5	**37.4** ± **0.8**
	BYOL	ResNet50		14.9 ± 0.8	21.3 ± 0.6	28.4 ± 0.2	32.3 ± 0.2	34.8 ± 1.1	44.0 ± 0.1	55.1 ± 0.5
			✓	**15.5** ± **0.8**	21.1 ± 0.4	28.3 ± 0.4	**32.9** ± **0.3**	**35.4** ± 1.1	**44.5** ± **0.4**	**57.6** ± **0.7**

Table 2. Average of top-1 accuracy (%) with standard deviation across 3 random subsets on ImageNet-1K and CUB.