Supplementary Materials for
Frequency Guidance Matters in Few-Shot Learning

Hao Cheng[1] Siyuan Yang[1] Joey Tianyi Zhou[2,3] Lanqing Guo[1] Bihan Wen[1*]
[1]Nanyang Technological University [2]Centre for Frontier AI Research (CFAR), A*STAR, Singapore
[3]Institute of High Performance Computing (IHPC), A*STAR, Singapore
{hao006,siyuan005,lanqing001,bihan.wen}@ntu.edu.sg zhouty@cfar.a-star.edu.sg

In this supplement, we offer detailed information about four datasets and the implementation of few-shot generalization tasks (Section A). Additionally, we present further frequency analysis of few-shot methods to investigate the necessity and effectiveness of frequency components on the overall performance of the proposed model (Section B). Furthermore, we provide additional experimental results for better clarity (Section C).

A. Implementations

A.1. Datasets

We conduct extensive experiments on four few-shot datasets, *i.e.*, *mini*ImageNet [11], *tiered*ImageNet [9], CUB [12], and FS-DomainNet [3].

- ***mini*ImageNet** [11] is a subset of the ILSVRC-12 challenge [6] proposed for few-shot classification, which contains 100 diverse classes with 600 images of size $84 \times 84 \times 3$ in each category. Following the setting [8] used in previous works, all 100 classes are divided into 64, 16, and 20 classes for training, validation, and testing, respectively.
- ***tiered*ImageNet** [9] is also a subset of ImageNet, which contains more classes that are organized in a hierarchical structure, *i.e.*, 608 classes from 34 top categories. For the general few-shot setting, we follow the setups proposed by [9] and split 608 categories into 351, 97, and 160 for training, validation, and testing, respectively. For the coarse-to-fine annotated setting, we split the dataset with 20, 6, and 8 super classes for training, validation, and testing, respectively.
- **CUB** [12] contains total 11788 images from 200 different birds, and is initially proposed for fine-grained image classification. Following the few-shot split in [2, 5], 200 classes are divided into 100, 50, and 50 for training, validation, and testing, respectively.
- **FS-DomainNet** [3] considers both few-shot and cross-domain scenarios to evaluate the generalizability across different domains. It contains 527156 images with 299 classes from 6 domains (*i.e.*, Sketch, Quickdraw, Real, Painting, Clipart, and Infograph), selected from DomainNet [7]. We follow the setup proposed by [3] which splits 299 categories into 191, 47, and 61 for training, validation, and testing, respectively.

A.2. Few-shot Generalization Settings

A.2.1 Cross-Dataset Generalization

In this setting, we only consider the distribution gaps of classes between training and testing datasets with the same style, *i.e.*, *mini*ImageNet and CUB with natural images.

***mini*ImageNet → CUB.** Unlike the general setting in previous works [2, 14], which trains few-shot models on all 100 classes of *mini*ImageNet, only 64 classes from the training set of *mini*ImageNet are used for meta-training, while the models are evaluated on the testing set of CUB with 50 classes.

A.2.2 Cross-Domain Generalization

Unlike the cross-dataset generalization, we also further consider the generalized performance of few-shot methods from training (source) domains to the testing (target) domain, where the domain represents the image style, *e.g.*, natural or painting. We split the cross-domain scenarios into two specific settings according to the number of the source domains, *i.e.*, cross-domain from single (*e.g.*, *mini*ImageNet) or multiple source domains (*e.g.*, FS-DomainNet). Note that FS-DomainNet contains 6 distinct domains of 345 classes, we select the quickdraw domain as the target domain for evaluation under these two settings and left the other five domains for meta-training.

***mini*ImageNet → Quickdraw.** In this setting, 64 classes from the training set of *mini*ImageNet are used for meta-

[*] Corresponding author.

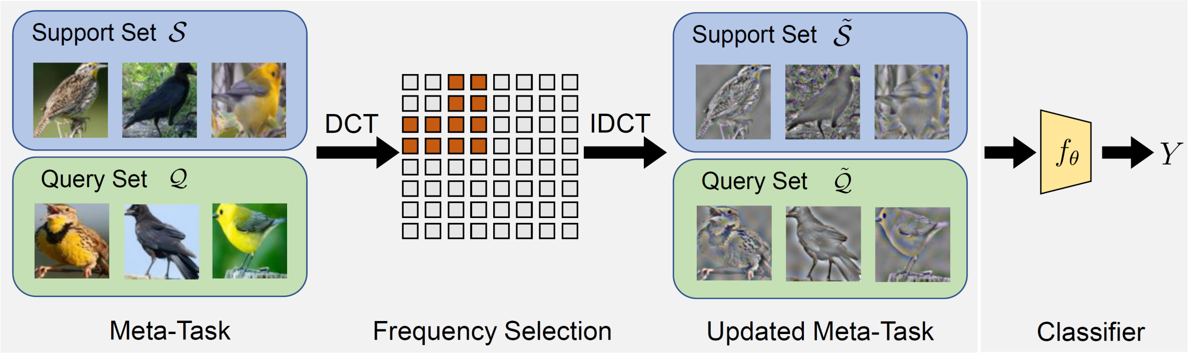


Figure A: Overview of the evaluation pipeline for few-shot learning with frequency selection. For example, given a 3-way 1-shot input task ($\mathcal{S}$, $\mathcal{Q}$), we first convert each image X_i into the frequency domain D_i with DCT. Then we select partial frequency components (e.g., middle FCs here) and convert them back to the spatial domain ($\tilde{\mathcal{S}}$, $\tilde{\mathcal{Q}}$). After that, we employ the normalization and class prediction Y with the trained model f_θ.

training, while the models are evaluated on the testing set of the Quickdraw domain with 61 classes.

FS-DomainNet → Quickdraw. In this setting, 191 classes of 5 domains (Sketch, Real, Painting, Clipart, and Infograph) from the training set of FS-DomainNet are used for meta-training, while the models are evaluated on the testing set of Quickdraw domain with 61 classes. Following the few-shot generalization setting, there is no overlap between training and testing domains or classes.

A.2.3 Coarse-to-Fine Annotated Generalization

In this setting, we consider the annotation difference between training and testing sets for few-shot generalization. To evaluate the generalization performance under this setting, we conduct experiments on *tiered*ImageNet, which provides the hierarchical annotations with 20, 6, and 8 super classes for training, validation, and testing.

***tiered*ImageNet (Coarse) → *tiered*ImageNet (Fine).** 20 superclasses from the training set of coarse-annotated *tiered*ImageNet are used from meta-training, while the models are evaluated on 160 classes from the testing set of fine-annotated *tiered*ImageNet.

B. Frequency Analysis for few-shot learning

B.1. Discrete Cosine Transform

To generate the frequency representation of an input RGB image $X \in \mathbb{R}^{H \times W \times 3}$, we apply the 2D-Discrete Cosine Transform [1] (2D-DCT) function denoted as $DCT(\cdot)$ as the following

$$D \triangleq DCT(X) = TXT', \tag{1}$$

where T denotes the 1D-DCT, making the $DCT(\cdot)$ operator separable. We can then remove or preserve partial frequency components of each input image, followed by an

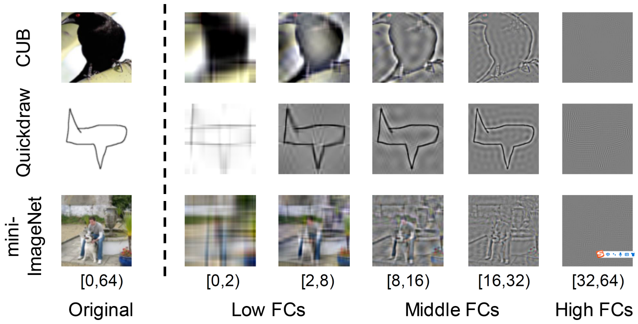


Figure B: Visualization of original images from three few-shot datasets, as well as the images reconstructed with different frequency components dubbed FCs (64 in total) containing the different information, *i.e.*, Low FCs contain the color and style information, Middle FCs contain the image structures, and High FCs contain the finer details, respectively.

Inverse 2D-DCT function denoted as $IDCT(\cdot)$ to transform it back into the spatial domain with the original input size, which is represented as:

$$\tilde{X} = IDCT(\ell(D)) = T'\ell(D)T\,. \tag{2}$$

Here $\ell(\cdot)$ denotes the masking operator in the frequency domain, to select partial frequency components of the image as shown in Figure B.

Note that there are other available tools to convert the spatial image to frequency space, such as Discrete Fourier Transform (DFT). In this paper, we choose DCT as it is a real-valued transform, compared to complex-valued DFT which may double the complexity of the model involving complex coefficients.

B.2. Evaluation Pipeline

As shown in Figure B, for a given few-shot meta-task $\mathcal{T}_{\text{test}}^{m} = (\mathcal{S}, \mathcal{Q})$ on the novel testing set, we evaluate the effect of frequency components on the updated meta-task $\tilde{\mathcal{T}}_{\text{test}}^{m} = \{\tilde{\mathcal{S}}, \tilde{\mathcal{Q}}\}$ constructed by removing or preserving partial components (*e.g.*, low-frequency components) of each image in the Discrete Cosine Transform (DCT) frequency domain for class prediction. Apart from the results shown in the main manuscript, we also conduct experiments on all split frequency components under the same settings on the *mini*ImageNet, CUB, *tiered*ImageNet, and FS-DomainNet datasets.

We apply the public implementation of evaluated few-shot methods (ProtoNet [10], FEAT [16], DeepEMD [17], FRN [13], and BML [19]) with the default hyper-parameter settings for model training, and integrate the DCT, IDCT, and frequency selection modules to these methods in the testing stage for evaluation.

B.3. Cross-Dataset Generalization

Table A shows evaluation results of few-shot methods trained on *mini*ImageNet and CUB datasets under the general and cross-dataset settings. We can observe that under the standard setting for the 1-shot task (training and testing on the same dataset, *i.e.*, *mini*ImageNet and CUB), testing without high FCs may decrease the performance on the source dataset, which indicates that models need to capture this information that is not perceivable to humans but essential for classification with less supervision.

Additionally, we also show the correlation matrix of the prototypical feature vector of each class extracted by the backbone in the FEAT [16] in Figure C on the CUB dataset. We observed that the trained model failed to distinguish different species of birds when only preserving high-frequency information in the images. One possible reason is that the high-frequency information encodes more edge details of birds with high similarity scores between images of different classes. Thus, the trained model performs better on images without high FCs. Both t-SNE and correlation matrix show that low and mid-FCs are critical for generalizing the fine-grained task, which encodes more color and shape information of birds.

B.4. Cross-Domain Generalization

We validate the effect of frequency information under two cross-domain few-shot generalization settings and results are shown in Table B.

As observed from the first two columns in Table B, compared with results evaluated on original images with all FCs, all methods perform worse on images only with low FCs for 1-shot and 5-shot settings, while they all achieve better results only with mid or high FCs. CAM [18] results in Figure E also show that the mid and high FCs can capture more class-relevant image regions compared with the original input images. One possible reason is that the target Quickdraw domain contains more patterns that rarely appear in natural images on the source *mini*ImageNet dataset, and the background is entirely irrelevant to the classification, which is mainly captured by low FCs. In contrast, the shape or structure information extracted in the mid and high FCs is essential for this cross-domain classification scenario. Furthermore, we compute the correlation matrix of the image features extracted by the backbone in the FEAT [16] on the testing set of the quickdraw, as shown in Figure D. The results show that high FCs help to increase the intra-class distances while keeping the inter-class distance unchanged, thus improving the classification performance.

B.5. Coarse-to-Fine Generalization

Table C shows the results on the *tiered*ImageNet benchmark with different levels of label annotations. We observe that FEAT performs best across all settings and shot levels compared with the other methods. This is due to the fact that FEAT employs the self-attention module to learn task-specific feature representations, while DeepEMD only utilizes the sample-based earth mover's distance instead of Euclidean/cosine metrics, and BML additionally considers global label information, both of which are vulnerable to class distributions of the entire training set.

The second observation is that the models learned on the coarse-annotated dataset generalize worse than the fine-annotated dataset. A possible explanation is that training between classes with large gaps forces the models to capture the information or features that are relevant to super-class clusters while ignoring the intra-class differences, which can be also seen from the t-SNE visualization in Figure F. We also observe that ProtoNet and BML can perform better than DeepEMD and FEAT when removing the high FCs. We have an assumption that DeepEMD and FEAT both design the channel-wise operation based on the extracted features, which makes them sensitive to the images with different FCs as each channel can capture various frequency components.

B.6. Maximum Mean Discrepancy Analysis

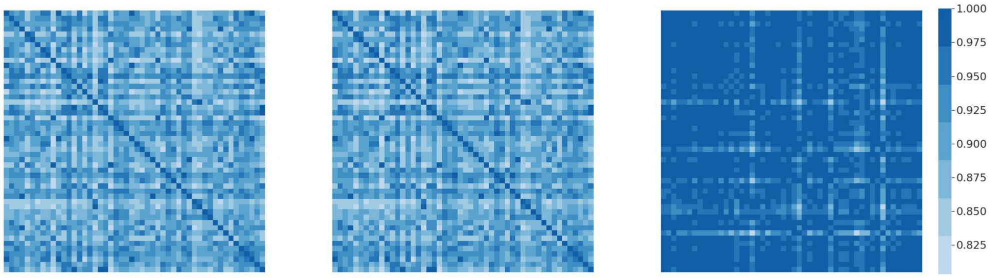
The Maximum Mean Discrepancy (MMD) [4] is a distance measure between two domains based on the embedding of distribution measures in a reproducing kernel Hilbert space $\mathcal{H}$, which has been widely applied in transfer learning problems, and the class-wise MMD could be formulated as follows:

$$\mathrm{MMD}(X, Y) = \left\| \frac{1}{n} \sum_{i=1}^{n} \varphi\left(x_i\right) - \frac{1}{m} \sum_{i=1}^{m} \varphi\left(y_i\right) \right\|_{\mathcal{H}}^{2}, \quad (3)$$

where $X = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^{n \times d}$ and $Y = \{y_1, y_2, ..., y_m\} \in \mathbb{R}^{m \times d}$ are two distributions with n and

Method	*mini*→ *mini*		*mini*→ CUB		CUB→ CUB		CUB→ *mini*		CUB→ *tiered*	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet [10]	**63.56**	**79.86**	**45.22**	**66.29**	**74.20**	**87.38**	**39.56**	**56.76**	37.76	**52.51**
ProtoNet only w/ Low FCs	41.54	57.48	38.10	53.13	54.88	71.99	39.37	55.79	**38.49**	52.02
ProtoNet only w/ Mid FCs	58.83	73.45	37.82	52.50	50.69	68.04	39.36	55.85	35.07	47.55
ProtoNet only w/ High FCs	45.86	56.47	30.74	37.96	31.81	40.03	39.40	55.73	29.78	36.38
ProtoNet w/o Low FCs	60.34	74.83	38.54	53.60	51.82	69.17	39.42	55.87	34.85	46.62
ProtoNet w/o Mid FCs	56.84	71.83	**41.42**	59.02	63.30	79.66	39.41	55.77	37.94	52.07
ProtoNet w/o High FCs	**63.51**	**77.75**	39.97	**67.09**	**74.22**	**87.44**	**39.96**	**56.95**	**38.81**	**53.27**
DeepEMD [17]	**64.93**	**81.73**	**51.72**	**77.34**	**76.34**	**88.52**	**39.41**	**54.04**	**41.94**	**56.58**
DeepEMD only w/ Low FCs	41.33	60.86	42.76	64.04	62.10	79.40	36.55	49.86	39.78	52.67
DeepEMD only w/ Mid FCs	47.96	69.49	39.80	60.40	43.47	62.74	36.12	51.31	34.54	48.65
DeepEMD only w/ High FCs	42.61	59.89	32.45	44.10	31.34	40.74	31.92	45.00	28.71	38.53
DeepEMD w/o Low FCs	49.34	71.17	39.97	61.22	43.88	63.26	36.35	51.56	34.63	48.77
DeepEMD w/o Mid FCs	56.24	74.06	45.95	69.17	68.66	84.38	37.52	52.07	41.13	53.04
DeepEMD w/o High FCs	**64.26**	**80.74**	**52.50**	**78.19**	**75.95**	**88.96**	**39.10**	**53.77**	**41.90**	**56.57**
FEAT [16]	**66.52**	**81.46**	**45.33**	62.28	**76.68**	**87.91**	**41.83**	**55.71**	**38.06**	51.07
FEAT only w/ Low FCs	39.59	54.90	37.64	52.94	58.00	74.10	38.88	53.80	38.00	**51.16**
FEAT only w/ Mid FCs	**65.57**	75.81	39.01	54.78	50.52	67.05	39.77	55.05	34.82	45.83
FEAT only w/ High FCs	42.20	56.28	27.93	36.76	29.68	34.93	32.96	41.97	24.82	25.70
FEAT w/o Low FCs	62.15	77.29	39.38	56.13	51.80	68.53	39.89	55.35	34.37	44.52
FEAT w/o Mid FCs	58.30	74.38	42.28	**62.66**	65.96	81.45	40.19	55.45	36.55	48.59
FEAT w/o High FCs	65.40	**81.62**	**46.63**	**66.99**	**75.88**	**88.65**	**41.93**	**58.31**	**38.93**	**52.21**
BML [19]	**65.41**	**82.17**	**49.85**	**71.22**	**74.36**	**89.75**	**42.56**	**60.59**	**40.41**	**57.20**
BML only w/ Low FCs	39.79	47.86	40.98	58.26	40.60	57.89	35.86	50.97	36.30	50.92
BML only w/ Mid FCs	54.53	68.18	40.66	57.58	50.52	67.05	39.77	55.05	34.82	45.83
BML only w/ High FCs	42.04	58.52	31.77	42.17	29.68	34.93	32.96	41.97	24.82	25.70
BML w/o Low FCs	56.02	74.23	40.98	57.58	51.80	68.53	39.89	55.35	34.37	44.52
BML w/o Mid FCs	54.00	72.88	44.72	65.12	65.96	81.45	40.19	55.45	36.55	48.59
BML w/o High FCs	**65.14**	**82.18**	**49.89**	**71.38**	**74.19**	**89.68**	**42.57**	**61.02**	**40.87**	**57.80**

Table A: Full evaluation results of frequency components under standard and cross-dataset few-shot settings. For **A** → **B**, we train few-shot methods on the training set from **A** and evaluate the performance on the testing set from **B**. Note that we train all the models with all FCs and fix the models for different frequency components (FCs) during evaluation. The best and second best results under each setting for each few-shot method are highlighted as **Red** and **Blue**, respectively.



(a) Original Image (b) Image w/o High FCs (c) Image w/ High FCs

Figure C: Correlation matrix between test class prototypes on the CUB dataset. For each class, we compute the mean feature vector as the prototype of this class by considering images with different frequency components, *i.e.*, (a), (b) and (c) represent the original image, image w/o high FCs, and image only w/ high FCs, respectively. The depth of color indicates the similarity score.

m classes, and d is the dimension of prototype feature. $\varphi(\cdot)$ is the mapping function to map the feature into a reproducing kernel Hilbert space.

To validate the effectiveness of different frequency components on various few-shot settings mentioned in the paper, we adopt class-wise MMD to measure the distribution

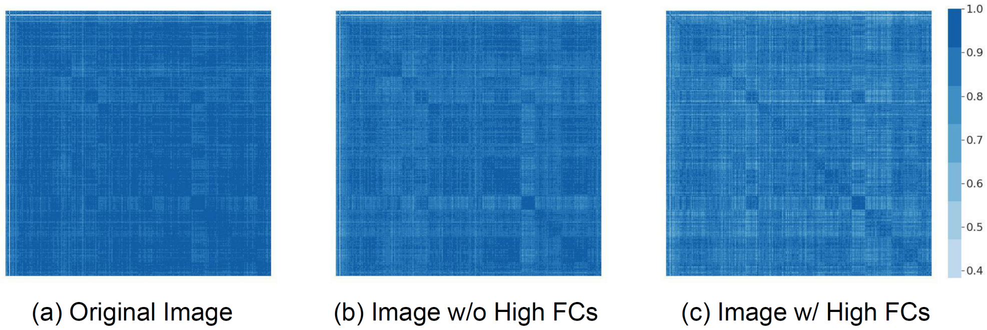


Figure D: Correlation matrix of test sample features for 20 test classes on the Quickdraw domain. The feature extractor is trained on the *mini*ImageNet dataset. For each class, we randomly select 20 images with the same frequency components, *i.e.*, (a), (b), and (c) represent the original image, image w/o high FCs, and image only w/ high FCs, respectively. The depth of color indicates the similarity score.

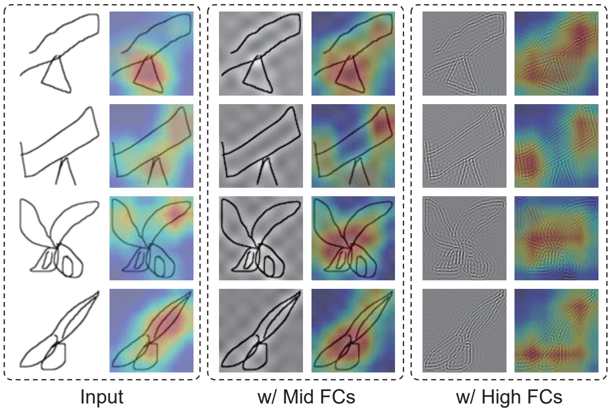


Figure E: CAM [18] visualization of samples from the target Quickdraw domain reconstructed with different frequency components, the first three columns are images with all FCs, only with Mid FCs, and only with high FCs, and the right three columns are the corresponding CAM results, respectively.

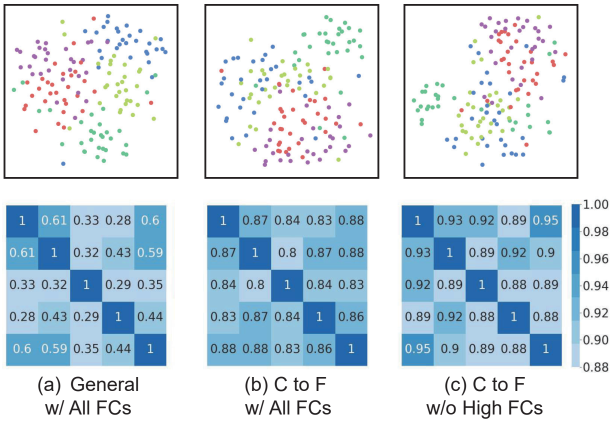


Figure F: T-SNE visualization of the image features (top) and Class Correlation Map visualization (bottom) for the same few-shot task with 40 query samples of 5 fine-annotated classes on the *tiered*ImageNet dataset extract by FEAT trained on the (a) general 351 training classes, (b)(c) coarse-annotated 20 training classes. (a)(b) represents evaluation with all FCs, while (c) represents evaluation without high FCs. Each row and column box in each correlation map denotes one class(way) under the 5-way settings. The numerical value (color depth) of each box indicates the similarity between the two categories corresponding to its row and column. Higher values (that is, darker colors) indicate more similarity between the two categories. C, F denote *tiered*ImageNet with coarse-grained or fine-grained annotations.

gap between the training set and the testing set with images only preserving partial frequency information. All results are based on the ProtoNet [10] method. Specifically, we adopt the trained backbone on the training set with the original images to compute the real prototype of each training class as $X \in \mathbb{R}^{n \times d}$, where n and d are the numbers of training classes and the dimension of prototype for each category, respectively. Similarly, we adopt the trained backbone on each testing set to compute the real prototype of each testing class as $Y \in \mathbb{R}^{m \times d}$, where m and d are the number of testing classes and the dimension of prototype for each category, respectively.

The MMD results are shown in Tables D, E, and F. All tables show a large domain gap between the training and testing set for all three settings, making it challenging for model generalization. We can observe that removing high FCs can narrow the domain gap between CUB and *mini*ImageNet, *i.e.*, from 1.8236 to 1.8011 in Table D, leading to better performance under the cross-dataset few-

Method	*mini* → Q		F→ Q	
	1-shot	5-shot	1-shot	5-shot
ProtoNet	48.65	64.32	59.16	74.43
ProtoNet only w/ Low FCs	43.27	58.06	50.80	66.10
ProtoNet only w/ Mid FCs	49.16	**66.28**	**59.67**	**75.14**
ProtoNet only w/ High FCs	**50.62**	**66.26**	59.43	**75.38**
ProtoNet w/o Low FCs	48.22	63.81	**59.50**	74.64
ProtoNet w/o Mid FCs	48.97	65.53	58.94	74.14
ProtoNet w/o High FCs	**49.26**	65.99	59.30	74.55
DeepEMD	53.65	58.04	**60.55**	78.52
DeepEMD only w/ Low FCs	49.41	66.17	51.66	74.03
DeepEMD only w/ Mid FCs	**53.99**	72.58	59.32	78.65
DeepEMD only w/ High FCs	**56.85**	**75.12**	58.33	73.72
DeepEMD w/o Low FCs	52.66	72.15	59.25	**79.62**
DeepEMD w/o Mid FCs	54.69	**73.94**	57.67	78.42
DeepEMD w/o High FCs	53.63	73.44	**60.73**	**80.17**
FEAT	45.52	55.96	**59.86**	77.23
FEAT only w/ Low FCs	42.46	57.27	32.06	64.28
FEAT only w/ Mid FCs	**47.43**	**62.72**	56.81	77.37
FEAT only w/ High FCs	**47.56**	**63.03**	45.78	75.70
FEAT w/o Low FCs	44.20	53.93	58.39	**77.54**
FEAT w/o Mid FCs	46.56	60.29	55.23	75.99
FEAT w/o High FCs	47.25	60.73	**60.58**	**77.75**
BML	55.29	**75.47**	42.69	65.38
BML only w/ Low FCs	49.29	68.01	25.22	60.04
BML only w/ Mid FCs	54.35	74.27	25.53	65.68
BML only w/ High FCs	**55.77**	**76.48**	42.25	63.11
BML w/o Low FCs	55.09	75.34	**43.18**	**66.86**
BML w/o Mid FCs	**55.43**	75.16	**43.33**	63.00
BML w/o High FCs	54.20	74.14	42.80	**65.59**

Table B: Full Cross-Domain results on the QuickDraw domain. We train the model on the *mini*ImageNet and FS-DomainNet datasets (w/o the QuickDraw domain), respectively, and evaluate on the novel QuickDraw domain. ***mini***, **Q**, and **F** denote *mini*ImageNet, QuickDraw, FS-DomainNet datasets, respectively. The best and second best results under each setting for each few-shot method are highlighted as **Red** and **Blue**, respectively.

shot classification setting. In contrast, only preserving high FCs helps to reduce the domain shift between the source (*mini*ImageNet or FS-DomainNet) and target (Quickdraw) domains in cross-domain generalization settings, *i.e.*, from 2.7887 to 2.4252 and from 4.0125 to 3.2707 in Table F. Table E also shows the gap across different levels of annotations on the *tiered*ImageNet dataset, which explains the necessity of the coarse-to-fine annotated generalization setting. Similar to the cross-dataset generalization setting, we find that considering the low and mid-frequency information helps the model trained on coarse-annotated classes to generalize better to both coarse and fine-annotated unseen classes.

Method	Coarse → Coarse		Coarse → Fine	
	1-shot	5-shot	1-shot	5-shot
ProtoNet	**49.44**	**66.36**	**46.61**	64.08
ProtoNet only w/ Low FCs	30.18	38.64	36.20	47.87
ProtoNet only w/ Mid FCs	41.72	58.25	44.51	**64.35**
ProtoNet only w/ High FCs	31.30	42.99	36.10	47.51
ProtoNet w/o Low FCs	43.61	60.72	43.44	43.22
ProtoNet w/o Mid FCs	44.11	60.04	60.04	59.18
ProtoNet w/o High FCs	**49.39**	**66.36**	**49.66**	**66.37**
DeepEMD	**53.53**	**66.26**	35.27	65.87
DeepEMD only w/ Low FCs	32.21	46.24	40.67	58.50
DeepEMD only w/ Mid FCs	34.94	54.94	48.63	**66.92**
DeepEMD only w/ High FCs	30.69	42.20	37.57	53.20
DeepEMD w/o Low FCs	36.43	57.23	**50.05**	**68.92**
DeepEMD w/o Mid FCs	45.29	60.12	**56.26**	62.26
DeepEMD w/o High FCs	**51.64**	**64.23**	35.14	63.48
FEAT	**55.01**	**69.25**	**47.54**	**64.67**
FEAT only w/ Low FCs	30.77	42.63	28.73	36.70
FEAT only w/ Mid FCs	44.45	58.86	41.12	56.50
FEAT only w/ High FCs	27.00	51.11	33.30	43.06
FEAT w/o Low FCs	46.82	64.70	41.44	56.64
FEAT w/o Mid FCs	46.37	65.19	38.75	54.67
FEAT w/o High FCs	**55.21**	**68.23**	**46.45**	**63.77**
BML	**35.33**	**61.49**	35.14	**52.63**
BML only w/ Low FCs	26.03	34.33	**35.59**	45.59
BML only w/ Mid FCs	27.31	39.61	32.90	50.46
BML only w/ High FCs	26.61	41.96	31.07	42.72
BML w/o Low FCs	27.38	45.03	32.84	48.70
BML w/o Mid FCs	26.88	54.61	34.24	51.33
BML w/o High FCs	**33.65**	**58.79**	**35.77**	**53.42**

Table C: Coarse-to-Fine Annotated few-shot generalization results on *tiered*ImageNet dataset. We train the model on 20 super-classes in the training set and evaluate on the 8 coarse-annotated (Coarse → Coarse) and 160 fine-annotated (Coarse → Fine) testing classes over *tiered*ImageNet dataset, respectively. The best and second best results under each setting for each few-shot method are highlighted as **Red** and **Blue**, respectively.

C. Additional Experimental Results

Comparison with frequency- or spatial-based mask. To investigate the impact of frequency information on mask generation, we consider generating the mask directly based on the gradient in the spatial domain. Specifically, we remove the frequency branch and generate a spatial mask based on backbone network E^{sp}. We adopt the same baseline and achieve 67.24% and 82.82% accuracy under the 5-way 1-shot and 5-way 5-shot settings, respectively. Results show that adding either spatial- or frequency-mask can improve the few-shot performance, while we can obtain a better result with frequency-mask. One possible reason is that directly masking partial regions of spatial images may cause discontinuity, affecting model learning.

Combination with more few-shot methods. We also eval-

	Test Set on *mini*ImageNet			Test Set on CUB		
	All FCs	w/o High FCs	only w/ High FCs	All FCs	w/o High FCs	only w/ High FCs
Train Set on *mini*ImageNet	**0.5450**	0.6223	2.2224	**1.6614**	1.6842	2.9445
Train Set on CUB	1.8236	**1.8011**	4.5190	**0.0806**	0.3803	5.2912

Table D: MMD results for cross-dataset few-shot generalization. The lower MMD value indicates a smaller gap and better performance.

	Test Set on *tiered* (C)			Test Set on *tiered* (F)		
	All FCs	w/o High FCs	only w/ High FCs	All FCs	w/o High FCs	only w/ High FCs
Train set on *tiered* (C)	0.6133	**0.5977**	4.1766	1.7659	**1.3447**	4.0631

Table E: MMD results for coarse-to-fine annotated few-shot generalization. *tiered* (C) and (F) represent *tiered*ImageNet datasets with coarse-annotated and fine-annotations, respectively. The lower MMD value indicates a smaller gap and better performance.

	Test Set on Quickdraw		
	All FCs	w/o High FCs	only w/ High FCs
Train Set on *mini*ImageNet	2.7887	2.6104	**2.4252**
Train Set on FS-DomainNet	4.0125	3.4885	**3.2707**

Table F: MMD results for cross-domain few-shot generalization. The lower MMD value indicates a smaller gap and better performance.

Model	***mini*ImageNet**	
	5-way 1-shot	5-way 5-shot
Dynamic [15]	67.76 ± 0.46	82.71 ± 0.31
Dynamic+FGFL	69.40 ± 0.49	84.63 ± 0.36
DeepBDC [14]	67.34 ± 0.43	84.46 ± 0.28
DeepBDC+FGFL	68.62 ± 0.84	85.67 ± 0.61

Table G: Average classification accuracy (%) over *mini*ImageNet with the ResNet-12 as backbone

uate the effectiveness of the proposed FGFL by integrating it with additional few-shot methods, including Dynamic FSL [15] and DeepBDC [14], shown in Table G.

Model Efficiency. We compared the inference time between few-shot methods w/ and w/o our proposed FGFL, shown in Table H. The inference time is calculated for a single meta-task in a 5-way 1-shot scenario on *miniImageNet*, utilizing the ResNet-12 backbone on an NVIDIA RTX A5000 GPU. Results in Table H indicate that FGFL does indeed require additional time to generate masked frequency-domain images using Grad-CAM to assist spatial-domain classification. However, in comparison to the results w/o FGFL (corresponding to the first row in table H), the time cost introduced by FGFL is relatively minor and acceptable. Furthermore, FGFL needs additional parameters (a frequency encoder and classifier) as a trade-off to encode frequency information to improve the few-shot classification, which may potentially constrain the efficiency of FGFL.

Method	ProtoNet [10]	FEAT [16]	Dynamic [15]	DeepBDC [14]
Infer. Time (s)	0.04	0.04	0.24	0.11
w/ FGFL	0.07	0.07	0.26	0.15

Table H: Inference time (second) for a single meta-task under the 5-way 1-shot setting on *miniImageNet*.

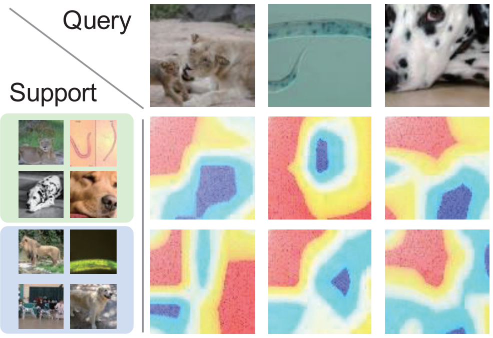


Figure G: Visualizations of frequency masks of query samples generated by different support sets.

Task-adaptive frequency mask. We visualize the frequency mask of query samples generated by different support sets from the same classes, *i.e.*, 5 selected classes in a meta-task, shown in Figure G. Results show that frequency masks can capture more task-specific frequency information as additional knowledge and avoid over-fitting to spatial features, leading to further improvements over spatial masks, aligning with our idea.

References

[1] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974. 2

[2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 1

[3] Hao Cheng, Yufei Wang, Haoliang Li, Alex C Kot, and Bihan Wen. Disentangled feature representation for few-shot image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1

[4] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006. 3

[5] Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018. 1

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[7] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019. 1

[8] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. 1

[9] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. 1

[10] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4080–4090, 2017. 3, 4, 5, 7

[11] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3637–3645, 2016. 1

[12] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1

[13] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8012–8021, 2021. 3

[14] Jiangtao Xie, Fei Long, Jiaming Lv, Qilong Wang, and Peihua Li. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7972–7981, 2022. 1, 7

[15] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5182–5191, 2021. 7

[16] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 3, 4, 7

[17] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020. 3, 4

[18] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition*, 2016. 3, 5

[19] Ziqi Zhou, Xi Qiu, Jiangtao Xie, Jianan Wu, and Chi Zhang. Binocular mutual learning for improving few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8402–8411, 2021. 3, 4