Appendix

The appendix is organized as follows: we introduce the ten FSC algorithms adopted in the paper in Appendix A. Then, we give the details of the evaluation metrics used in the main paper in Appendix B. In Appendix C, we show statistics of the datasets used in this paper along with detailed training settings. In Appendix D, we analyze different methods for constructing the support and query sets in a FewSTAB task (Appendix D.1), show the scatter plots of wAcc-A versus Acc from all the training settings (Appendix D.2), present more results on the effectiveness of FewSTAB (Appendix D.3), and demonstrate the robustness of FewSTAB with different VLMs (Appendix D.4). Finally, we give more examples of the tasks constructed by FewSTAB in Appendix E.

A Few-Shot Classification Algorithms

ANIL (Almost No Inner Loop) [39]: ANIL is an optimization-based meta-learning method and follows a similar optimization procedure to MAML [9] whose few-shot adaptation algorithm $\mathcal{O}$ is to update the whole model using gradient descent with a few learning samples. ANIL does not update the whole model and instead only updates the classifier in the last layer.
BOIL (Body Only update in Inner Loop) [34]: BOIL is another optimization-based meta-learning method. Its adaptation algorithm $\mathcal{O}$ freezes the update of the classifier and only updates the embedding backbone.
LEO (Latent Embedding Optimization) [42]: LEO is similar to MAML. But instead of directly optimizing high-dimensional model parameters, its adaptation algorithm $\mathcal{O}$ learns a generative distribution of model parameters and optimizes the model parameters in a low-dimensional latent space.
ProtoNet (Prototypical Networks) [46]: ProtoNet is a metric-based meta-learning method. Its adaptation algorithm $\mathcal{O}$ first calculates a prototype representation for each class as the mean vector of each support class, and then uses a nearest-neighbor classifier created with the class prototypes and the Euclidean distance function to predict a query image.
DN4 (Deep Nearest Neighbor Neural Network) [24]: DN4 is a metric-based meta-learning method, which does not use attributes after pooling for classification. Instead, DN4 uses the local attributes before pooling and employs a local descriptor based image-to-class measure for classification.
R2D2 (Ridge Regression Differentiable Discriminator) [5]: R2D2 is a metric-based meta-learning method and adopts ridge regression as the few-shot adaptation algorithm $\mathcal{O}$. The advantage of R2D2 is that ridge regression enjoys a closed-form solution and can learn efficiently with a few training samples.
CAN (Cross Attention Network) [16]: CAN is a metric-based meta-learning method and calculates the cross attention between each pair of class and query features so as to exploit and learn discriminative features for predictions.
RENet (Relational Embedding Network) [19]: RENet is a metric-based meta-learning method. It uses a self-correlational representation module and

a cross-correlational attention module to learn relational patterns within and between images, respectively.

RFS (Rethinking Few-Shot) [50]: RFS is a transfer learning method. It first trains an embedding network using base classes. Then, instead of fine-tuning the last fully-connected classification layer, it learns a new logistic regression classifier with L2-normalized feature vectors from a few samples of novel classes.

Baseline++ [6]: Baseline++ is a transfer learning method. It first pretrains an embedding network using samples from base classes. Then, it fine-tunes the last fully-connected layer with a few samples of novel classes but replaces the standard inner product with a cosine distance between input features and the weight vectors of the layer.

B Evaluation Metrics

Standard accuracy (Acc): Acc measures on average how a few-shot classifier generalizes to different tasks with novel classes not seen before. We define Acc as follows,

$$\text{Acc} = \frac{1}{N_T} \sum_{t=1}^{N_T} \sum_{c=1}^{C} M_c(\mathcal{T}_t; f_\theta, \mathcal{O}), \tag{7}$$

where N_T is the number of test tasks, C is the number of classes per task, $\mathcal{T}_t$ is the t-th C-way N_S-shot task with N_Q query samples per class, f_θ is a few-shot classifier, $\mathcal{O}$ is the few-shot adaptation algorithm associated with f_θ, M_c denotes the classification accuracy on the *query samples* of the class c. This metric is used in Fig. 5.

Class-wise worst classification accuracy (wAcc): wAcc characterizes the performance limit of f_θ in learning novel classes, and we calculate wAcc as the average of the smallest per-class classification accuracy on query samples over N_T tasks, i.e.,

$$\text{wAcc} = \frac{1}{N_T} \sum_{t=1}^{N_T} \min_{c=1,\ldots,C} M_c(\mathcal{T}_t; f_\theta, \mathcal{O}). \tag{8}$$

Depending on what kinds of tasks are used for evaluation, we have the following two types of wAcc:

- **wAcc-R:** If the test tasks are randomly sampled in Eq. (8), then we get wAcc-R on N_T randomly sampled tasks. This metric is used in Tab. 3 as a baseline for highlighting the effectiveness of our FewSTAB in revealing the spurious bias in few-shot classifiers.
- **wAcc-A:** If the N_T test tasks in Eq. (8) are constructed by our FewSTAB, then we get wAcc-A, which characterizes the robustness of a few-shot classifier to spurious bias. This metric is the main metric used in the experiments.

Accuracy gap between wAcc-R and wAcc-A: We obtain the wAcc-R and wAcc-A of a model by testing it with tasks randomly sampled and with tasks constructed by FewSTAB, respectively. The accuracy gap is calculated as the wAcc-R minus the wAcc-A. A large gap indicates the effectiveness of FewSTAB in revealing the robustness of a few-shot classifier to spurious bias. This metric is used in Fig. 4 and Tab. 5.
Accuracy gap between wAcc-A of models trained with different shots: We train a few-shot classifier with C-way (e.g. 5-way) 5-shot and 1-shot training tasks from $\mathcal{D}_{train}$, respectively. Then, we test the obtained two classifiers with the *same* tasks created by FewSTAB and calculate the accuracy gap as the wAcc-A of the model trained with 5-shot tasks minus the wAcc-A of the model trained with 1-shot tasks. A large accuracy gap indicates that increasing training shots can improve a few-shot classifier's robustness to spurious bias. This metric is used in Fig. 6.

Table S1: Numbers of classes along with numbers of samples (in parentheses) in each split of the three datasets.

Split	miniImageNet	tieredImageNet	CUB-200
$\mathcal{D}_{train}$	64 (38.4k)	351 (448.7k)	130 (7.6k)
$\mathcal{D}_{val}$	16 (9.6k)	97 (124.3k)	20 (1.2k)
$\mathcal{D}_{test}$	20 (12k)	160 (206.2k)	50 (3.0k)

C Experimental Settings

We conducted experiments using three datasets: miniImageNet, tieredImageNet, and CUB-200. Each of these datasets has training ($\mathcal{D}_{train}$), validation ($\mathcal{D}_{val}$), and test ($\mathcal{D}_{test}$) sets. Numbers of classes and samples in the three sets of the three datasets are shown in Tab. S1.

We trained eight meta-learning based FSC methods with the ResNet-12 backbone using 5-way 1-shot or 5-way 5-shot tasks from each $\mathcal{D}_{train}$ of the three datasets, resulting in a total of 48 models. For the two transfer learning based methods, RFS and Baseline++, we trained them on each $\mathcal{D}_{train}$ of the three datasets using mini-batch stochastic gradient descent. As a result, we trained a total of 54 models.

To facilitate reproducibility and further research, the training configurations and hyperparameters are provided in Tabs. S2 to S4 for training on the miniImageNet, tieredImageNet, and CUB-200 datasets, respectively. We closely followed the settings in [25] to train these models. In the "Mode" column of these tables, "T(5w1s)" denotes that we trained the corresponding model using 5-way 1-shot tasks, "T(5w5s)" denotes that we trained the corresponding model using 5-way 5-shot tasks, and "B (128)" denotes that we trained the corresponding model using mini-batch stochastic gradient descent with a batch size of 128. In the

Table S2: Training configurations and hyperparameters for training on the miniImageNet dataset. "-" denotes not applicable.

Method	Mode	Learning rate	LR scheduler	Optimizer	Epochs	Training episodes	Episode size
ANIL	T (5w1s)	0.001	-	Adam	100	2000	4
	T (5w5s)	0.001	-	Adam	100	2000	4
LEO	T (5w1s)	0.0005	CosineAnnealingLR	Adam	100	2000	1
	T (5w5s)	0.001	CosineAnnealingLR	Adam	100	2000	1
BOIL	T (5w1s)	0.0006	-	Adam	100	2000	4
	T (5w5s)	0.0006	-	Adam	100	2000	4
ProtoNet	T (5w1s)	0.001	StepLR(20, 0.5)	Adam	100	200	1
	T (5w5s)	0.001	StepLR(20, 0.5)	Adam	100	2000	1
DN4	T (5w1s)	0.001	StepLR(50, 0.5)	Adam	100	2000	1
	T (5w5s)	0.001	StepLR(50, 0.5)	Adam	100	2000	1
R2D2	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	2000	4
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	2000	4
CAN	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	2000	8
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	2000	4
RENet	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	300	1
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	300	1
Baseline++	B (128)	0.01	CosineAnnealingLR	SGD	100	-	-
RFS	B (64)	0.05	MultiStepLR([60, 80], 0.1)	SGD	100	-	-

"LR scheduler" column, "CosineAnnealingLR" denotes a cosine annealing learning rate scheduler, "StepLR(20, 0.5)" denotes a learning rate scheduler which decreases the learning rate after every 20 epochs by multiplying it with 0.5, and "MultiStepLR([60, 80], 0.1)" denotes a learning rate scheduler which decreases the learning rate after 60 epochs and 80 epochs by multiplying it with 0.1 each time. The "Training episodes" column in these tables denotes the number of tasks used in each epoch. The "Episode size" column of these tables denotes the number of tasks jointly used to do a model update.

D Additional Experimental Results

D.1 Ablation Studies

Support set construction methods: To construct the support set in an FSC test task, FewSTAB randomly selects samples that have *mutually exclusive* spurious attributes across the randomly selected classes, which is illustrated in Fig. 2(a) and formally described in Sec. 4.3 in the main paper. To further show the effectiveness of this construction method, we keep the techniques for constructing the query set in an FSC test task, and report in Tab. S5 the results of two alternatives for constructing the support set: randomly selecting samples of the selected classes (**SC1**) and randomly selecting samples with targeted attributes for selected classes with no further constraints on the selected samples (**SC2**). We also include the results of the proposed one: randomly selecting samples with mutually exclusive targeted attributes across the selected classes (**SC3**) in Tab. S5. A larger average drop in Tab. S5 indicates that the corresponding support set construction method is more effective in revealing robustness of few-shot classifiers to spurious bias. We observe that the third technique SC3,

Table S3: Training configurations and hyperparameters for training on the tieredImageNet dataset. "-" denotes not applicable.

Method	Mode	Learning rate	LR scheduler	Optimizer	Epochs	Training episodes	Episode size
ANIL	T (5w1s)	0.001	-	Adam	100	5000	4
	T (5w5s)	0.001	-	Adam	100	5000	4
LEO	T (5w1s)	0.0005	CosineAnnealingLR	Adam	100	5000	1
	T (5w5s)	0.001	CosineAnnealingLR	Adam	100	5000	1
BOIL	T (5w1s)	0.0006	-	Adam	100	5000	4
	T (5w5s)	0.0006	-	Adam	100	5000	4
ProtoNet	T (5w1s)	0.001	StepLR(20, 0.5)	Adam	100	5000	1
	T (5w5s)	0.001	StepLR(20, 0.5)	Adam	100	5000	1
DN4	T (5w1s)	0.001	StepLR(50, 0.5)	Adam	200	5000	1
	T (5w5s)	0.001	StepLR(50, 0.5)	Adam	200	5000	1
R2D2	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	5000	4
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	5000	4
CAN	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	2000	4
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	2000	4
RENet	T (5w1s)	0.1	MultiStepLR([60, 80], 0.05)	SGD	100	1752	1
	T (5w5s)	0.1	MultiStepLR([40, 50], 0.05)	SGD	60	1752	1
Baseline++	B (128)	0.01	CosineAnnealingLR	SGD	100	-	-
RFS	B (128)	0.1	CosineAnnealingLR	SGD	100	-	-

Table S4: Training configurations and hyperparameters for training on the CUB-200 dataset. "-" denotes not applicable.

Method	Mode	Learning rate	LR scheduler	Optimizer	Epochs	Training episodes	Episode size
ANIL	T (5w1s)	0.001	-	Adam	100	2000	4
	T (5w5s)	0.001	-	Adam	100	2000	4
LEO	T (5w1s)	0.0005	CosineAnnealingLR	Adam	100	2000	4
	T (5w5s)	0.001	CosineAnnealingLR	Adam	100	2000	1
BOIL	T (5w1s)	0.0006	-	Adam	100	2000	4
	T (5w5s)	0.0006	-	Adam	100	2000	4
ProtoNet	T (5w1s)	0.001	StepLR(20, 0.5)	Adam	100	2000	1
	T (5w5s)	0.001	StepLR(20, 0.5)	Adam	100	2000	1
DN4	T (5w1s)	0.001	StepLR(50, 0.5)	Adam	100	2000	1
	T (5w5s)	0.001	StepLR(50, 0.5)	Adam	100	2000	1
R2D2	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	2000	4
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	2000	4
CAN	T (5w1s)	0.01	-	Adam	100	100	4
	T (5w5s)	0.01	-	Adam	100	100	4
RENet	T (5w1s)	0.1	CosineAnnealingLR	SGD	100	300	1
	T (5w5s)	0.1	CosineAnnealingLR	SGD	100	600	1
Baseline++	B (128)	0.01	CosineAnnealingLR	SGD	100	-	-
RFS	B (64)	0.05	MultiStepLR([60, 80], 0.1)	SGD	100	-	-

Table S5: Comparison between different techniques used by FewSTAB for constructing the support sets in 5-way 5-shot FSC test tasks. Values in the shaded areas are the accuracy gaps defined as wAcc-R minus wAcc-A. Average drop is the average of accuracy gaps over the ten FSC methods. "-" denotes not applicable.

Method	miniImageNet				tieredImageNet				CUB-200			
	wAcc-R	wAcc-A/Acc. gap			wAcc-R	wAcc-A/Acc. gap			wAcc-R	wAcc-A/Acc. gap		
		SC1	SC2	SC3		SC1	SC2	SC3		SC1	SC2	SC3
ANIL	25.37	19.69	15.64	14.83	30.60	19.55	14.53	13.72	45.47	38.73	32.39	31.63
		5.68	9.73	**10.54**		11.05	16.07	**16.88**		6.74	13.08	**13.84**
LEO	41.33	34.33	28.02	26.31	57.22	40.28	30.23	29.49	59.76	48.04	43.07	46.62
		7.00	13.31	**15.02**		16.94	26.99	**27.73**		11.72	**16.69**	13.14
BOIL	15.21	13.88	13.46	13.09	18.55	15.82	15.11	14.90	21.33	18.42	17.84	19.17
		1.33	1.75	**2.12**		2.73	3.44	**3.65**		2.91	**3.49**	2.16
ProtoNet	51.95	43.37	33.40	32.07	62.53	44.23	31.61	30.95	75.68	67.12	59.72	60.06
		8.58	18.55	**19.88**		18.30	30.92	**31.58**		8.56	**15.96**	15.62
DN4	42.68	36.74	28.62	27.60	40.63	24.32	16.50	16.07	73.58	66.07	58.32	59.25
		5.94	14.06	**15.08**		16.31	24.13	**24.56**		7.51	**15.26**	14.33
R2D2	50.84	44.01	36.47	35.37	61.08	43.34	31.79	31.12	75.20	65.12	56.88	58.66
		6.83	14.37	**15.47**		17.74	29.29	**29.96**		10.08	**18.32**	16.54
CAN	54.23	46.66	37.82	36.44	64.19	45.53	32.23	31.17	61.61	53.91	44.91	41.31
		7.57	16.41	**17.79**		18.66	31.96	**33.02**		7.70	16.70	**20.30**
RENet	56.52	47.48	37.60	36.19	63.49	44.23	31.04	30.27	71.82	63.03	53.27	52.93
		9.04	18.92	**20.33**		19.26	32.45	**33.22**		8.79	18.55	**18.89**
Baseline++	44.94	37.70	30.47	29.52	59.06	40.95	30.74	30.01	29.84	24.21	19.55	16.86
		7.24	14.47	**15.42**		18.11	28.32	**29.05**		5.63	10.29	**12.98**
RFS	55.66	48.17	38.33	36.85	62.71	44.35	31.94	31.15	74.33	64.54	54.41	54.98
		7.49	17.33	**18.81**		18.36	30.77	**31.56**		9.79	**19.92**	19.35
Average drop	-	6.67	13.89	**15.05**	-	15.75	25.43	**26.12**	-	7.94	**14.83**	14.72

which is used by FewSTAB, achieves the largest average accuracy drop among the techniques compared on the miniImageNet and tieredImageNet datasets and achieves a comparable drop to SC2 on the CUB-200 dataset due to the limited number of detected attributes in this dataset.

Query set construction methods: There are three techniques used by FewSTAB to construct the query set in a task: the intra-class attribute-based sample selection (**QC1**), the inter-class attribute-based sample selection (**QC2**), which is a special case of the intra-class attribute-based sample selection, and the query sample selection (**QC3**). We have done an ablation study on the effectiveness of the three techniques in Tab. 5 in the main paper using the miniImageNet dataset. Here, we include the results on all the three datasets in Tab. S6. We observe that all the three proposed techniques in FewSTAB are effective with positive accuracy drops for all the ten FSC methods on the three datasets. Moreover, using the inter-class attribute-based sample selection significantly improves the average drops of the intra-class attribute-based sample selection, with 8.17%, 13.26%, and 8.52% absolute gains on the miniImageNet, tieredImageNet, and CUB-200 datasets, respectively.

Table S6: Comparison between different techniques used by FewSTAB for constructing the query sets in 5-way 5-shot FSC test tasks. Values in the shaded areas are the accuracy gaps defined as wAcc-R minus wAcc-A. Average drop is the average of accuracy gaps over the ten FSC methods. "-" denotes not applicable.

Method	miniImageNet				tieredImageNet				CUB-200			
	wAcc-R	wAcc-A/Acc. gap			wAcc-R	wAcc-A/Acc. gap			wAcc-R	wAcc-A/Acc. gap		
		QC1	QC2	QC3		QC1	QC2	QC3		QC1	QC2	QC3
ANIL	25.37	21.61	16.37	14.83	30.60	21.95	14.00	13.72	45.47	39.77	32.58	31.63
		3.76	9.00	**10.54**		8.65	16.60	**16.88**		5.70	12.89	**13.84**
LEO	41.33	36.04	28.36	26.31	57.22	46.94	31.93	29.49	59.76	56.73	48.21	46.62
		5.29	12.97	**15.02**		10.28	25.29	**27.73**		3.03	11.55	**13.14**
BOIL	15.21	14.57	13.70	13.09	18.55	17.61	15.37	14.90	21.33	20.49	19.85	19.17
		0.64	1.51	**2.12**		0.94	3.18	**3.65**		0.84	1.48	**2.16**
ProtoNet	51.95	44.28	34.17	32.07	62.53	49.58	33.56	30.95	75.68	70.33	61.81	60.06
		7.67	17.78	**19.88**		12.95	28.97	**31.58**		5.35	13.87	**15.62**
DN4	42.68	39.25	28.91	27.60	40.63	28.28	17.63	16.07	73.58	70.37	60.61	59.25
		3.43	13.77	**15.08**		12.35	23.00	**24.56**		3.21	12.97	**14.33**
R2D2	50.84	45.68	36.99	35.37	61.08	48.96	33.83	31.12	75.20	69.78	60.34	58.66
		5.16	13.85	**15.47**		12.12	27.25	**29.96**		5.42	14.86	**16.54**
CAN	54.23	47.83	38.16	36.44	64.19	50.58	33.63	31.17	61.61	54.32	42.88	41.31
		6.40	16.07	**17.79**		13.61	30.56	**33.02**		7.29	18.73	**20.30**
RENet	56.52	49.80	38.31	36.19	63.49	49.86	32.76	30.27	71.82	64.26	54.48	52.93
		6.72	18.21	**20.33**		13.63	30.73	**33.22**		7.56	17.34	**18.89**
Baseline++	44.94	39.51	31.26	29.52	59.06	47.08	31.86	30.01	29.84	27.47	18.62	16.86
		5.43	13.68	**15.42**		11.98	27.20	**29.05**		2.37	11.22	**12.98**
RFS	55.66	48.87	39.48	36.85	62.71	49.99	33.64	31.15	74.33	67.60	56.60	54.98
		6.79	16.18	**18.81**		12.72	29.07	**31.56**		6.73	17.73	**19.35**
Average drop	-	5.13	13.30	**15.05**	-	10.92	24.19	**26.12**	-	4.75	13.27	**14.72**

D.2 Scatter Plots of wAcc-A versus Acc

We show the scatter plots of wAcc-A versus Acc (standard accuracy) of the ten FSC methods when they are tested with FewSTAB and randomly constructed FSC test tasks, respectively, on the three datasets in Fig. S1 (exact values are shown in Tab. S7). We observe that an FSC method having a higher Acc does not necessarily have a higher wAcc-A. For example, in Fig. S1(a), BOIL has a higher Acc but a lower wAcc-A than ProtoNet, LEO, and Baseline++. Moreover, we observe that in Fig. S1(b) and (d), for methods that achieve high standard accuracies, e.g., for the top-5 methods in terms of Acc, their relative increments in wAcc-A are small (with differences smaller than 1%) compared with their relative increments in Acc. In other words, methods with higher standard accuracies do not necessarily learn more robust decision rules, since their wAcc-A values remain comparable to those with lower Acc values.

The values of Acc and wAcc-A on the fine-grained dataset CUB-200 in Fig. S1(e) and (f) show a different pattern from those in Fig. S1(c) and (d). More specifically, methods that achieve high Acc values, e.g., R2D2, ProtoNet, DN4, RENet, and RFS, tend to have comparable relative increments in wAcc-A compared with their relative increments in Acc. This indicates that on a fine-grained dataset, which does not have many spurious attributes, an FSC method

with a higher Acc also tends to have a higher wAcc-A or improved robustness to spurious bias.

In summary, our framework, FewSTAB, reveals new robustness patterns of FSC methods in different evaluation settings.

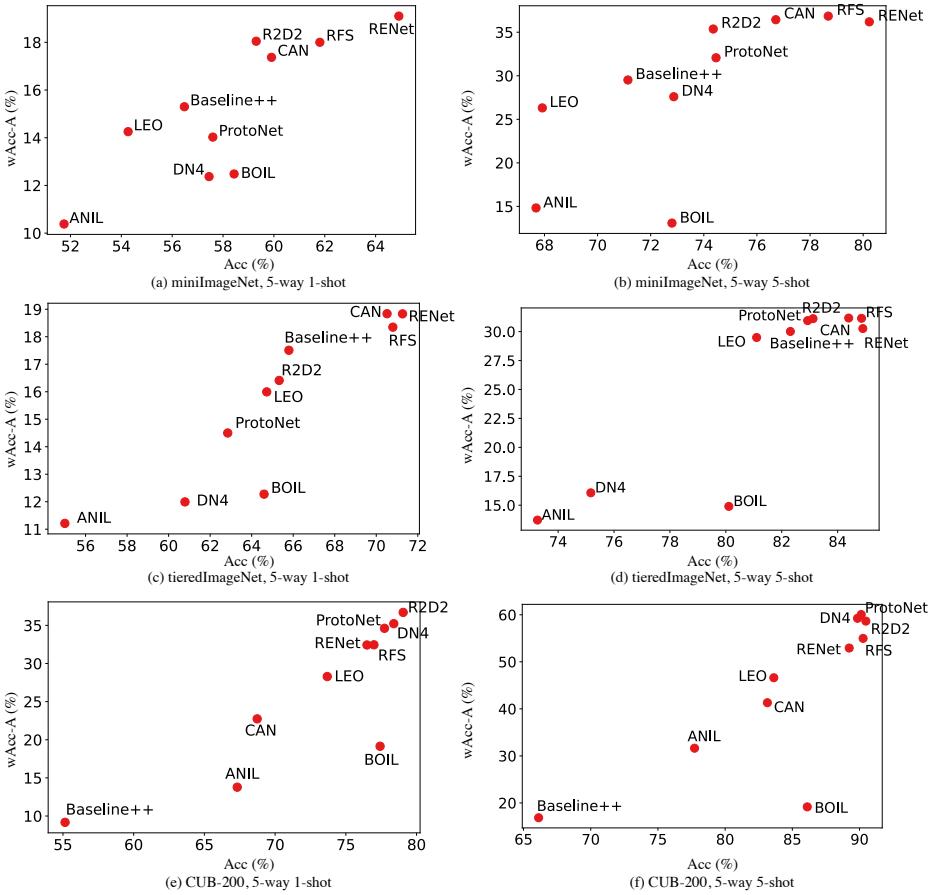

Fig. S1: Scatter plots of wAcc-A versus Acc of the ten FSC methods tested with 5-way 1/5-shot FewSTAB and randomly constructed tasks from the miniImageNet, tieredImageNet, and CUB-200 datasets, respectively. All methods are trained and tested with the same shot number.

D.3 Effectiveness of FewSTAB: More Results

Results on more recent methods. Note that our method selection in Tab. 3 aims to cover *diverse* methods and allow for *rigorous* comparison in the *same*

Table S7: Standard accuracies (Acc) and class-wise worst accuracies obtained with FewSTAB (wAcc-A) with 95% confidence intervals of the ten FSC methods on miniImageNet, tieredImageNet, and CUB datasets. Numbers in the Shot column indicate that the models are both trained (if applicable) and tested on 5-way 1- or 5-shot tasks. Darker colors indicate higher values.

Shot	Method	miniImageNet		tieredImageNet		CUB-200	
		Acc	wAcc-A	Acc	wAcc-A	Acc	wAcc-A
1	ANIL	51.75±0.39	10.38±0.30	55.00±0.45	11.21±0.30	67.32±0.45	13.78±0.40
	LEO	54.27±0.38	14.26±0.46	64.73±0.46	16.00±0.55	73.68±0.42	28.29±0.80
	BOIL	58.43±0.39	12.48±0.23	64.60±0.43	12.27±0.21	77.42±0.39	19.15±0.29
	ProtoNet	57.60±0.38	14.03±0.49	62.85±0.44	14.50±0.50	77.73±0.39	34.62±0.85
	DN4	57.45±0.36	12.37±0.46	60.79±0.42	11.99±0.47	78.39±0.38	35.22±0.86
	R2D2	59.30±0.39	18.05±0.53	65.33±0.44	16.41±0.54	79.05±0.38	36.70±0.90
	CAN	59.91±0.38	17.37±0.53	70.52±0.43	18.84±0.60	68.73±0.41	22.74±0.72
	RENet	64.91±0.38	19.10±0.57	71.27±0.42	18.83±0.61	76.49±0.36	32.43±0.81
	Baseline++	56.48±0.37	15.30±0.48	65.79±0.42	17.51±0.54	55.15±0.44	9.17±0.47
	RFS	61.81±0.35	18.00±0.53	70.80±0.42	18.35±0.60	76.99±0.35	32.45±0.80
5	ANIL	67.68±0.33	14.83±0.40	73.26±0.35	13.72±0.39	77.72±0.34	31.63±0.55
	LEO	67.92±0.32	26.31±0.59	81.10±0.34	29.49±0.72	83.62±0.30	46.62±0.82
	BOIL	72.80±0.29	13.09±0.22	80.11±0.32	14.90±0.22	86.11±0.26	19.17±0.28
	ProtoNet	74.46±0.28	32.07±0.58	82.93±0.31	30.95±0.70	90.13±0.21	60.06±0.74
	DN4	72.87±0.29	27.60±0.58	75.17±0.36	16.07±0.62	89.85±0.21	59.25±0.77
	R2D2	74.36±0.29	35.37±0.59	83.12±0.30	31.12±0.72	90.47±0.21	58.66±0.82
	CAN	76.71±0.28	36.44±0.65	84.40±0.29	31.17±0.76	83.14±0.27	41.31±0.74
	RENet	80.23±0.26	36.19±0.63	84.90±0.28	30.27±0.76	89.23±0.21	52.93±0.82
	Baseline++	71.14±0.30	29.52±0.57	82.31±0.31	30.01±0.72	66.12±0.35	16.86±0.52
	RFS	78.69±0.26	36.85±0.64	84.86±0.29	31.15±0.76	90.26±0.20	54.98±0.81

setting. Importantly, our method is general and can continue to evaluate emerging methods. To demonstrate, we provide results on recent methods, namely UniSiam [28], PsCo [18], and BECLR [36]. FewSTAB uncovers that, even the state-of-the-art methods still suffer from spurious bias as we observe large gaps between wAcc-R and wAcc-A (Table S9), when we explicitly construct the test tasks to have spurious correlations. This also shows that FewSTAB is effective for various FSC methods.

Results on IFSL. Interventional few-shot learning (IFSL) [63] is a method that specifically addresses spurious correlations in few-shot classification. We follow the settings in [63] and report the results of MAML [9], MN [53], SIB [17], and MTL [48] in Table S10, where "Base" refers to one of the four methods, "+IFSL" denotes using IFSL on top of "Base", and the better performance between the two is in bold. Overall, IFSL is effective in mitigating spurious bias in few-shot classifiers except for some methods, e.g. SIB. This shows that FewSTAB can reveal the improvement made to mitigate spurious bias.

D.4 Robustness of FewSTAB with Different VLMs

We instantiated our FewSTAB with a pre-trained ViT-GPT2 and a pre-trained BLIP, respectively. We calculated the wAcc-A on FSC test tasks constructed

Table S8: Comparison between wAcc-A calculated over 5-way 1/5-shot tasks obtained using Vit-GPT2 and using BLIP. We calculated wAcc-A for ten FSC methods on mini-ImageNet, tieredImageNet, and CUB datasets. Numbers in the Shot column indicate that the models are both trained (if applicable) and tested on 1- or 5-shot tasks. Darker colors indicate higher values.

Shot	Method	miniImageNet		tieredImageNet		CUB-200	
		ViT-GPT2	BLIP	ViT-GPT2	BLIP	ViT-GPT2	BLIP
1	ANIL	10.38±0.30	10.39±0.29	11.21±0.30	10.76±0.29	13.78±0.40	14.74±0.41
	LEO	14.26±0.46	14.38±0.45	16.00±0.55	14.34±0.52	28.29±0.80	31.06±0.80
	BOIL	12.48±0.23	12.51±0.22	12.27±0.21	11.65±0.21	19.15±0.29	20.35±0.29
	ProtoNet	14.03±0.49	13.50±0.46	14.50±0.50	13.25±0.50	34.62±0.85	38.63±0.81
	DN4	12.37±0.46	12.86±0.46	11.99±0.47	11.21±0.46	35.22±0.86	39.51±0.82
	R2D2	18.05±0.53	17.66±0.51	16.41±0.54	15.01±0.53	36.70±0.90	40.61±0.84
	CAN	17.37±0.53	16.89±0.51	18.84±0.60	17.43±0.61	22.74±0.72	24.23±0.71
	RENet	19.10±0.57	18.80±0.54	18.83±0.61	17.29±0.60	32.43±0.81	36.12±0.82
	Baseline++	15.30±0.48	15.06±0.46	17.51±0.54	15.60±0.52	9.17±0.47	10.42±0.50
	RFS	18.00±0.53	17.43±0.50	18.35±0.60	16.81±0.57	32.45±0.80	35.43±0.79
5	ANIL	14.83±0.40	13.67±0.38	13.72±0.39	12.57±0.37	31.63±0.55	33.01±0.56
	LEO	26.31±0.59	24.79±0.57	29.49±0.72	27.92±0.70	46.62±0.82	49.97±0.81
	BOIL	13.09±0.22	12.79±0.22	14.90±0.22	14.63±0.22	19.17±0.28	20.03±0.27
	ProtoNet	32.07±0.58	29.28±0.57	30.95±0.70	28.51±0.68	60.06±0.74	64.67±0.64
	DN4	27.60±0.58	25.28±0.57	16.07±0.62	14.98±0.58	59.25±0.77	65.61±0.67
	R2D2	35.37±0.59	31.81±0.59	31.12±0.72	29.50±0.68	58.66±0.82	64.02±0.77
	CAN	36.44±0.65	33.81±0.62	31.17±0.76	29.28±0.72	41.31±0.74	43.10±0.73
	RENet	36.19±0.63	33.76±0.63	30.27±0.76	28.71±0.72	52.93±0.82	60.29±0.74
	Baseline++	29.52±0.57	27.17±0.55	30.01±0.72	28.20±0.70	16.86±0.52	17.25±0.53
	RFS	36.85±0.64	34.72±0.62	31.15±0.76	29.29±0.72	54.98±0.81	62.33±0.69

by FewSTAB with the two VLMs on the miniImageNet, tieredImageNet, and CUB-200 datasets, respectively.

Effects on individual and relative measurements. We observe from Tab. S8 that FewSTAB with BLIP produces lower wAcc-A than with ViT-GPT2 on the miniImageNet and tieredImageNet datasets. This indicates that FewSTAB with BLIP is more effective in uncovering the robustness of few-shot classifiers to spurious bias. We reason that BLIP can identify more attributes than ViT-GPT2 (Tab. 2) and therefore more spurious correlations can be formulated by our FewSTAB. However, on the fine-grained CUB-200 dataset, which contains different bird classes, FewSTAB with BLIP is less effective than with ViT-GPT2. Although BLIP can identify more attributes than ViT-GPT2 in this fine-grained dataset, it may also detect more attributes related to classes. To validate this, we first found a set of attributes $\mathcal{U}_{\text{BLIP}}$ unique to BLIP from all the attributes $\mathcal{A}_{\text{BLIP}}$ detected by BLIP, and a set of attributes $\mathcal{U}_{\text{ViT-GPT2}}$ unique to ViT-GPT2 from all the attributes $\mathcal{A}_{\text{ViT-GPT2}}$ detected by ViT-GPT2. Specifically, we have $\mathcal{U}_{\text{BLIP}} = \mathcal{A}_{\text{BLIP}} - \mathcal{A}_{\text{ViT-GPT2}}$, and $\mathcal{U}_{\text{ViT-GPT2}} = \mathcal{A}_{\text{ViT-GPT2}} - \mathcal{A}_{\text{BLIP}}$. Then, we found in $\mathcal{U}_{\text{BLIP}}$ and $\mathcal{U}_{\text{ViT-GPT2}}$ how many attributes contain "bird", "beak", "wing", "breast", "tail", or "mouth", which are all related to the concept of a bird. We found that there are 11 attributes, or 8.5% of total attributes in $\mathcal{U}_{\text{BLIP}}$ that are related to a bird. While there is only 1 attribute (2.4% of total

Table S9: Results on the miniImageNet dataset. V: ViT-GPT2, B: BLIP. All input images are resized to 84×84.

Method	Shot	wAcc-R	wAcc-A (V)	wAcc-A (B)
UniSiam	1	$21.26_{\pm 0.48}$	$13.52_{\pm 0.43}$	$13.49_{\pm 0.42}$
PsCo	1	$21.50_{\pm 0.47}$	$14.30_{\pm 0.40}$	$12.46_{\pm 0.37}$
BECLR	1	$35.57_{\pm 0.80}$	$23.60_{\pm 0.83}$	$22.42_{\pm 0.82}$
UniSiam	5	$45.60_{\pm 0.52}$	$27.76_{\pm 0.57}$	$25.42_{\pm 0.56}$
PsCo	5	$42.15_{\pm 0.52}$	$25.54_{\pm 0.52}$	$22.64_{\pm 0.49}$
BECLR	5	$55.20_{\pm 0.49}$	$37.32_{\pm 0.66}$	$33.42_{\pm 0.68}$

Table S10: wAcc-A comparison (%) on the miniImageNet dataset.

Method	1-shot		5-shot	
	Base	+IFSL	Base	+IFSL
MAML	$\mathbf{13.29}_{\pm 0.55}$	$12.05_{\pm 0.56}$	$28.70_{\pm 0.69}$	$\mathbf{29.82}_{\pm 0.76}$
MN	$17.40_{\pm 0.62}$	$\mathbf{17.72}_{\pm 0.63}$	$30.48_{\pm 0.73}$	$\mathbf{31.51}_{\pm 0.75}$
SIB	$\mathbf{30.09}_{\pm 1.04}$	$27.10_{\pm 0.97}$	$\mathbf{46.73}_{\pm 0.96}$	$46.66_{\pm 0.95}$
MTL	$37.29_{\pm 0.57}$	$\mathbf{40.22}_{\pm 0.57}$	$49.49_{\pm 0.58}$	$\mathbf{52.66}_{\pm 0.58}$

attributes) in $\mathcal{U}_{\text{ViT-GPT2}}$ that is related to a bird. Due to the limited capability of BLIP, these class-related attributes cannot be detected in all the images. Hence, although these attributes are not spurious, they are treated as spurious attributes and used by FewSTAB to construct FSC test tasks. In this case, FewSTAB becomes ineffective in revealing the spurious bias in few-shot classifiers since the classifiers can exploit spurious correlations in the tasks to achieve high accuracies. Nevertheless, from the perspective of comparing the robustness of different FSC methods to spurious bias, the test tasks constructed by FewSTAB using different VLMs can reveal consistent ranks in terms of wAcc-A for different FSC methods (Tab. 6).

Detection accuracies of VLMs. Using different VLMs may generate different sets of attributes. Some sets of attributes may not exactly reflect the data being described, resulting in low detection accuracies. For example, some attributes are not identified by a VLM or the identified attributes do not match with the ground truth attributes. To analyze how the detection accuracy of a VLM affects our framework, we show in Tab. S11 the detection accuracies of the two VLMs that we used in our paper along with the Spearman's rank correlation coefficients between the evaluation results on the ten FSC methods based on the two VLMs. To calculate the detection accuracy of a VLM without the labor-intensive human labeling, we use the outputs of another VLM as the ground truth. Specifically, for the i'th image, we have two detected sets of attributes, $\mathcal{A}^i_{query}$ and $\mathcal{A}^i_{ref}$, representing the attributes from a VLM being evaluated and the ones from another VLM serving as the ground truth attributes. The detection accuracy is

Table S11: Detection accuracies of the ViT-GPT2 and BLIP along with the Spearman's rank correlation coefficients between the results based on the two VLMs.

VLM	Detection accuracy		Spearman's rank correlation coefficient	
	ViT-GPT2	BLIP	1-shot	5-shot
miniImageNet	34.46	31.42	0.98	1.0
tieredImageNet	35.04	32.00	1.0	0.99
CUB-200	70.12	59.28	1.0	0.98

calculated as follows:

$$Acc(VLM_{query}, VLM_{ref}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{i=1}^{N_{test}} \frac{|\mathcal{A}^i_{query} \cap \mathcal{A}^i_{ref}|}{|\mathcal{A}^i_{ref}|}, \tag{9}$$

where $N_{test} = |\mathcal{D}_{test}|$, and $|\cdot|$ denotes the size of a set. For example, to calculate the detection accuracy of ViT-GPT2, we set VLM_{query} =ViT-GPT2 and VLM_{ref} =BLIP. From Tab. S11, we observe that the detection accuracies of the two VLMs are not high, indicating that the attributes identified by the two VLMs are very different. However, the two VLMs are well-established in practice and can identify many attributes from images (Tab. 2). The correlation coefficients in Tab. S11 indicate that for well-established VLMs, the detection accuracies have little impact on the comparison of robustness to spurious bias between different FSC methods.

E Tasks Constructed by FewSTAB

FewSTAB does not construct tasks based on a specific model. Hence, FewSTAB is a fair evaluation framework for different FSC methods, and the tasks constructed by FewSTAB can be used to reveal few-shot classifiers' varied degrees of robustness to spurious bias.

We show a 5-way 1-shot task constructed by FewSTAB using samples from the tieredImageNet and CUB-200 datasets in Fig. S2 and Fig. S3, respectively. Query samples for each class are constructed such that they do not contain the spurious attribute from the support set sample of the same class but contain spurious attributes from support set samples of other classes. For example, in Fig. S2, the class `malamute` has a support set sample with a `rocky` background, but most of its query samples have a `bike` which is the spurious attribute from the support set sample of the `valley` class. Moreover, in Fig. S3, the class `Mallard` has a support set sample with a `sandy` background, but its query samples all have a `water` background similar to that in the support set sample of the `Baltimore Oriole` class. Note that the sample selection may not be ideal due to the limited capacity of VLMs. For example, in Fig. S2, some query images of the class `eggnog` have the spurious attribute `cup` which also appears in the support set image of the class, leading to a high accuracy on these query samples for a model that relies on this spurious attribute. However, this does not affect our

evaluation of different FSC methods on their robustness to spurious bias since the same set of tasks is used to evaluate different FSC methods. Moreover, our metric, wAcc-A, measures the worst per-class classification accuracy over FSC tasks, making our evaluation robust to the sampling noise caused by a VLM.

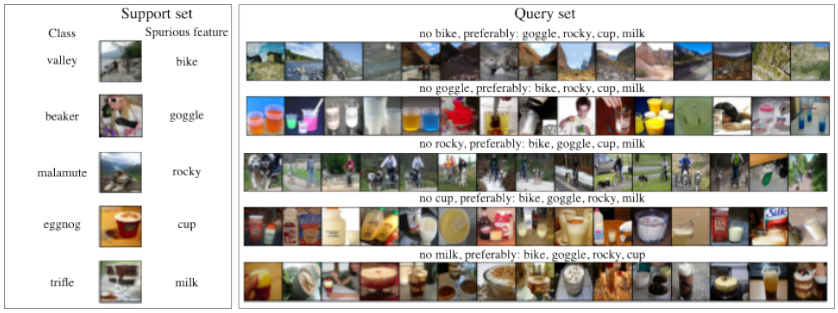

Fig. S2: A 5-way 1-shot task constructed by our FewSTAB using samples from the tieredImageNet dataset. Note that due to the limited capacity of a VLM, the attributes may not well align with human understandings.

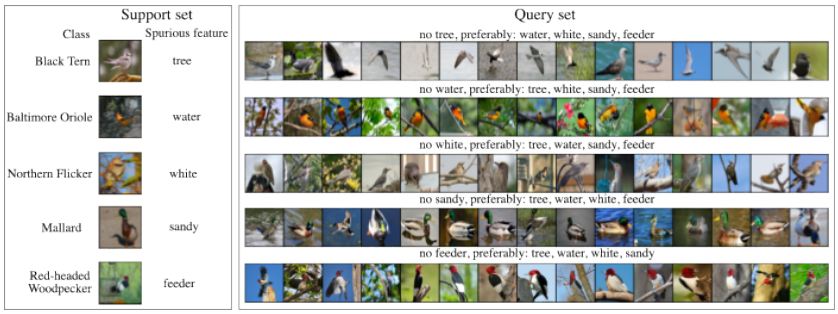

Fig. S3: A 5-way 1-shot task constructed by our FewSTAB using samples from the CUB-200 dataset. Note that due to the limited capacity of a VLM, the attributes may not well align with human understandings.