Supplementary Material on Is user feedback always informative? Retrieval Latent Defending for Semi-Supervised Domain Adaptation without Source Data

Junha Song[1,2], Tae Soo Kim[2], Junha Kim[2], Gunhee Nam[2], Thijs Kooi[2], and Jaegul Choo[1]

[1] KAIST {sb020518, jchoo}@kaist.ac.kr
[2] Lunit Inc. {junha.kim, taesoo.kim, ghnam, tkooi}@lunit.io

In this supplementary material, we provide:

A. Comparison with Related Work
- A.1. Active Domain Adaptation
- A.2. Class-Imbalanced Semi-Supervised Learning
- A.3. Test-time Adaptation
- A.4. Learning with User Feedback

B. Further understanding with Simulation Study

C. Additional Ablation Study

D. Additional Experimental Details

E. Additional Discussion
- E.1. Technique novelty
- E.2. Computational overhead
- E.3. Limitations

F. Results of All Domain Shifts.

A Comparison with Related Work

A.1 Active Domain Adaptation

Active domain adaptation (ActiveDA) aims to select the most informative samples being labeled by annotators, given a limited annotating budget. As shown in Figure 7, the machine selects some samples using ActiveDA methods and instructs annotators to label the selected samples. Several ActiveDA methods have been proposed, such as CLUE [55], which employs an entropy-based clustering algorithm to preserve the uncertainty and diversity of labeled data. SDM-AG [86] and DiaNa [25] utilize margin functions between the source and target domains to identify informative samples. In contrast to this ActiveDA scenario, we present an NBF scenario where there is no machine-instructed sample selection, and instead, users *directly* provide feedback as a response to the prediction result. It may lead to more flexible applications since (1) users have the freedom

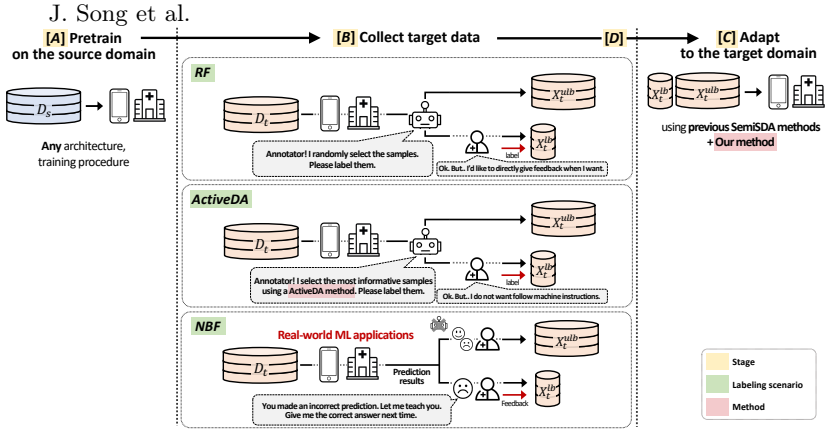

Fig. 7: Comparison between labeling scenarios: random feedback (RF), active domain adaptation (ActiveDA), and negatively biased feedback (NBF).

	state B						
feed. amount	378 (3 labeled data per class)					1890	5040
stage C	RF	NBF	Entropy [62]	CLUE [55]	DiaNA [25]	CLUE [55]	CLUE [55]
AdaMatch [7]	67.6	64.5	65.9	68.6	68.1	76.1	80.3
w/ ours	71.1 (+3.5)	72.0 (+7.5)	71.1 (+5.2)	71.5 (+2.9)	71.3 (+3.2)	78.0 (+1.9)	81.4 (+1.1)

Table 9: We evaluate a SemiSDA method [6] and our method under diverse labeling scenarios. The scenarios include our proposed NBF and ActiveDA scenarios [25, 55]. The difference between ActiveDA and our method is illustrated in Figure 7.

to choose samples, and (2) individual users can impose different standards in selecting samples.

We note that ActiveDA methods are for stage B of Figure 7, while our method is for stage C and proposed to alleviate the problem caused by NBF. Although out of our scope, we evaluate our method under ActiveDA labeling scenarios, where CLUE and DiaNA[1] are employed. The results in Table 9 suggest two points. First, our method complements existing ActiveDA methods, consistently improving their performance. This highlights the importance of adapting the model with a balanced supervised signal throughout adaptation (*i.e.*, stage C) using our method, even when ActiveDA methods like CLUE respect the diversity of labeled samples. Second, our method achieves significant performance gains regardless of the labeling scenario, showing that our method can be applied for reliable adaptation even when the distribution of labeled data is unknown.

A.2 Class-Imbalanced Semi-Supervised Learning

SemiSDA and SemiSL methods often struggle with the different numbers of labeled data ***between classes***, known as class imbalance [49]. To address this problem, class-imbalanced SemiSL works like CReST [83] propose to balance the quantity of labeled data by using pseudo labels [29,31] in stage D (*i.e.*, generation in CReST) of Figure 7. Recent advancements like DASO [49] further reduce the

[1] For DiaNA [25], we utilize their proposed 'informativeness scoring mechanism' to maintain a pretrained model-agnostic property.

[2] If not specified, we use ResNet-50 and report the average accuracy (%) of seven domain shift scenarios in Table 1 for additional studies.

	method	feedback	FP : FN	average	fracture	pneumothorax
	Source model	-	-	.6768	.6642	.6894
	Pseudo-Label. [1]	RF	-	.7325	.7541	.7109
80 feedback		NBF	40 : 40	.7173 (-.0152)	.7414 (-.0127)	.6931 (-.0178)
80 feedback	with ours	NBF	40 : 40	.7334(+.0162)	.7625 (+.0211)	.7044 (+.0113)
80 feedback		NBF	75 : 5	.7248 (-.0077)	.7494 (-.0047)	.7002 (-.0107)
80 feedback	with ours	NBF	75 : 5	**.7361** (+.0113)	.7653 (+.0159)	.7070 (+.0068)
80 feedback		NBF	5 : 75	.7170 (-.0155)	.7420 (-.0121)	.6921 (-.0188)
80 feedback	with ours	NBF	5 : 75	.7315 (+.0145)	.7679 (+.0260)	.6951 (+.0030)
	Pseudo-Label. [1]	RF	-	.7353	.7565	.7141
160 feedback		NBF	80 : 80	.7162 (-.0192)	.7429 (-.0136)	..6894 (-.0247)
160 feedback	with ours	NBF	80 : 80	.7331 (+.0169)	.7680 (+.0251)	.6983 (+.0088)
160 feedback		NBF	155 : 5	.7237 (-.0117)	.7559 (-.0007)	.6915 (-.0227)
160 feedback	with ours	NBF	155 : 5	**.7358** (+.0121)	.7665 (+.0106)	.7051 (+.0136)
160 feedback		NBF	5 : 155	.7166 (-.0188)	.7438 (-.0128)	.6894 (-.0248)
160 feedback	with ours	NBF	5 : 155	.7300 (+.0134)	.7696 (+.0258)	.6904 (+.0010)
	Fully supervised	-	-	.7744	.8003	.7486

Table 10: Adaptation with different feedback configurations on MIMIC-CXR-V2. We conduct additional experiments to Table 5, where the same pre-trained model is utilized, and the two radiographic findings are considered for simplification. We compare different NBF configurations as we vary the amount of feedback from false positives (FP) and false negatives (FN) errors.

imbalance effect using both a similarity-based and linear classifier. Despite such advances in class-imbalanced SemiSL, the biased (*i.e.*, imbalanced) label distribution ***within the same class*** has been overlooked in the SemiSDA, SemiSL, and class-imbalanced SemiSL works. Therefore, we introduce the new concept of biased labeled data called NBF and demonstrate its unexpected influence on adaptation performance.

Even though our focus in this paper is on the bias within the same class, accounting for the imbalance between classes can still be crucial for reliable domain adaptation. For example, in the medical domain, while radiologists are likely to log the mistakes of the model, the amount of feedback from false negative samples may be small compared to those from false positive samples, given the natural prevalence of disease (e.g., lung cancer is less than 1 in 1000). We simulate this example scenario and evaluate our method in Table 10.

Under different feedback configurations. We take various feedback configurations into account, as depicted in Table 10. Assuming the model acquires 80 or 160 feedback instances for each finding, we alter the feedback quantities from false positive (FP) and false negative (FN) errors, which is similar to the setup of class-imbalanced SemiSL [29, 83]. We only consider two radiographic findings for simplification. The results show that our method can also mitigate the intended impact of NBF even with the class-imbalanced scenario. Interestingly, we observe better performance when FP feedback is larger than FN feedback, which makes our method suitable for the medical domain, where radiographic findings are rarely detected due to the natural prevalence of the disease.

Combining with class-imbalanced SemiSL methods. One naive way to more reliably adapt to the challenging scenario could involve combining our method with class-imbalanced SemiSL methods in stages C and D of Figure 7. To evaluate this approach, we conduct an additional simulation study in Figure 8. The simulation replicates the NBF scenario by selecting only misclassified samples within the same class. We further introduce class imbalance by varying

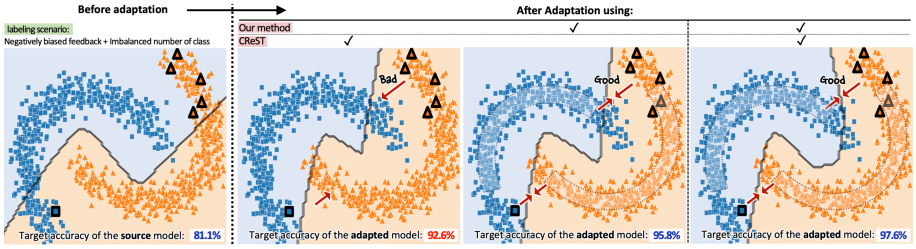

Fig. 8: Our contribution focuses on introducing NBF and analyzing the effect of NBF on adaptation. However, it may be required to consider both an NBF and a class-imbalanced scenario in real-world applications. Hence, we first simulate this scenario and perform adaptation using i) a SemiSDA method (*i.e.*, Pseudo labeling [1]) with our method and ii) a class-imbalanced SemiSL method (*i.e.*, CReST [83]).

the number of feedback points between the blue and orange classes (leftmost sub-figure).

By adapting the model with different approaches, we can find two interesting takeaways: (1) the approach proposed in CReST [83] was not designed to solve the unexpected effect of NBF, so it struggles with adaptation under the challenging scenario. (2) our method achieves better adaptation performance than using only CReST, and outperforms other results by combining with CReST. These results highlight the importance of considering an NBF case as well as a class-imbalance problem and the efficiency of our method. We hypothesize that defending the latent class space throughout adapting iterations helps the model to be robust to the effect of NBF, different from a previous generation-based approach in CReST [83]. In addition, a discussion about zero feedback for certain classes is provided in Section C.

A.3 Test-time Adaptation

To mitigate performance degradation caused by domain shift, models deployed on edge devices like smartphones and self-driving cars can be adapted to the target domain in an online manner, referred to as test-time adaptation (TTA). TTA assumes two practical settings: i) adapting without source data and ii) storing a limited amount of unlabeled target data. For instance, TENT [47, 67, 68, 77] leverages the current batch of unlabeled data to update the model's batch normalization parameters. Alternatively, methods like NOTE [19] and ContraTTA [8] employ a target memory bank where a small amount of data (*e.g.*, ***16k*** image features in ContraTTA) can be only stored and used for adaptation.

Extension to a TTA scenario. Our setup illustrated in Figure 2 also assumes a source-free setup, so it can be easily extended to a TTA scenario by employing the memory bank. In particular, on a periodic basis, when 10% of the target training data is encountered, an adaptation is executed following a TTA setup of TTT++ [39] and ContraTTA [8], where unlabeled data in the memory bank and labeled data are utilized. The memory bank size is set to ***5k*** pseudo labels, and FreeMatch [81] is used for a SemiSDA baseline algorithm. It should be noted that since previous TTA works do not consider the utilization of labeled

memory bank size = 5k			percentage of target data encountered in target domain			
method	feed.	amo.	10% ⟶	40% ⟶	70% ⟶	100%
FreeMatch [81]	RF		68.4	71.4	73.0	73.4
	NBF	368	66.9	69.5	71.0	71.5
w/ ours	NBF		68.9 (+2.0)	72.4 (+2.9)	73.6 (+2.6)	74.3 (+2.8)
FreeMatch [81]	RF		71.2	73.7	74.7	75.4
	NBF	630	69.8	72.2	73.2	73.9
w/ ours	NBF		71.5 (+1.7)	74.1 (+1.9)	75.0 (+1.8)	75.5 (+1.6)

Table 11: We evaluate our approach on a TTA scenario, where labeled data and unlabeled target data in a memory bank are only available for adaptation like ContraTTA [8]. In the real, painting, scratch, and clipart domains of DomainNet-126, 10% of the data consists of 5.5k, 2.4K, 1.9k, and 1.5k images. In the table, 40% means that the model has encountered 40% of the unlabeled target training data.

data, we can not use them as a baseline or compare the adaptation performance directly (but we attempt to alleviate this problem and implement comparisons in Section C.). The results in Table 11 show that our method works well even with a smaller amount of unlabeled data in the memory bank. We find this result very surprising and wish to continue in this direction for future research.

A.4 Learning with User Feedback

Learning with User Feedback has garnered significant attention for its effectiveness in capturing users' preferences or intentions [42, 50, 69, 84]. Reinforcement learning from human feedback is a powerful technique for model optimization based on human-provided rewards [30, 61, 71, 82]. Another application is interactive image segmentation [11, 64, 65], where users provide pixel-level annotations, enabling the model to enhance its understanding of user preferences over time.

B Further understanding with Simulation Study

In this section, we provide additional details and understanding about the simulation study in Figure 3.

Network architecture. We build the model consisting of three fully connected layers and Relu activation functions. This model takes the point coordinate as input and returns the class label as output. Please refer to example codes found in the 'sklearn.datasets.make_blobs' documents [53].

Baseline. One simple SemiSL method, Pseudo labeling [1], can be easily applied to the toy experiment. Given a mini-batch with labeled data $\{(x^b_{lb}, y^b_{lb}) : b \in [1..B]\}$ and unlabeled data $\{(x^b_{ulb}) : b \in [1..\mu{\cdot}B]\}$, we simply adapt the model with cross-entropy losses as the following:

$$\mathcal{L}_{sup} = \frac{1}{B}\sum_{b=1}^{B} \mathcal{H}(y^b_{lb}, f_\theta(x^b_{lb})), \quad \mathcal{L}_{unsup} = \frac{1}{\mu{\cdot}B}\sum_{b=1}^{\mu{\cdot}B} \mathcal{H}(\text{argmax}_c \left[f_\theta(x^b_{ulb})\right]_c, f_\theta(x^b_{ulb})). \quad (3)$$

$f_\theta(\cdot)$ is the output probability from the model and $\text{argmax}_c \left[f_\theta(x^b_{ulb})\right]_c$ refers to the pseudo label. As shown in the equation, the updating model f_θ continuously predicts pseudo labels for the unlabeled data. So, the pseudo labels can be changed based on an updated decision boundary. Figure 9 presents this phenomenon as the adapting epoch progresses.

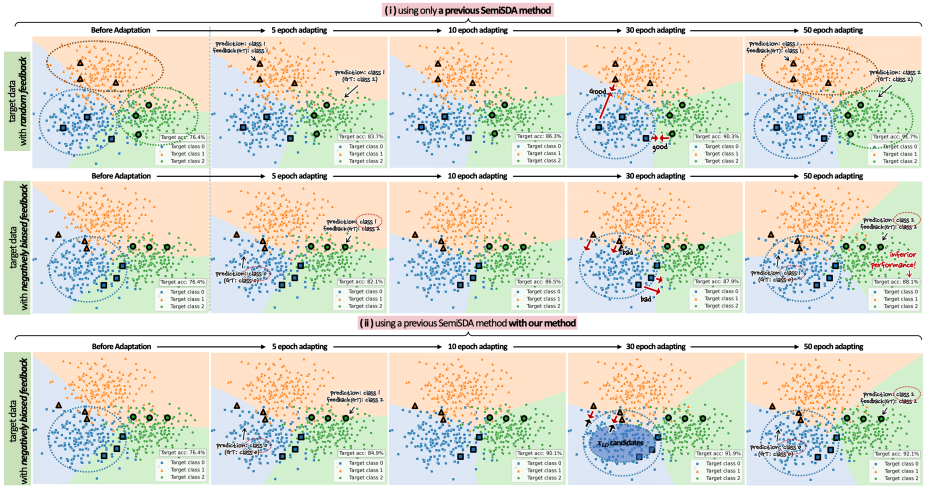

Fig. 9: During the adapting process, an updated decision boundary of the model is depicted. The details can be found in Section 3.2.

Additional study on two moon dataset. To better understand the unexpected influence of NBF on domain adaptation, we conducted additional simulations using the two moon datasets from scikit-learn [53]. As shown in Figure 10, we generate source and target data so that they have domain shifts. After pretraining a model on the source data, we evaluate its performance on the target domain, observing a performance drop due to the shift (99.9%→81.4%). After we simulate user-provided feedback under two scenarios (*i.e.*, RF and NBF), we adapt the model to the target data in a semi-supervised manner [1]. The results highlight crucial observations shown in Section 3.2: the distribution of label data significantly impacts adaptation performance. Notably, biased feedback distribution (NBF) leads to poorer performance compared to evenly distributed feedback (RF). In our main paper, we showed that this problem remained the same even with state-of-the-art SemiSDA methods and under different DA benchmarks.

C Additional Ablation Study

Reliable sample filtering. An important design of our approach is to retain only samples having reliable pseudo labels among $\{(x^n_{ulb}, \hat{y}^n_{ulb}) : n \in [1..N_{ulb}]\}$. We evaluate the adaptation performance with variations in the filtering ratio $p\%$ in Table 8. A higher p increases the likelihood of the bank being contaminated with samples with incorrect pseudo labels (*i.e.*, $y_{ulb} \neq \hat{y}_{ulb}$) while a lower p decreases the diversity of the defending samples. We observe that our approach is robust to the hyper-parameter p, yet achieves reasonable performance with $p = 0.4$.

Combining with SFDA methods. Recent SFDA methods [34,35] have shown promise in computing the unsupervised loss $\mathcal{L}_{unsup}$. So, we explore their potential

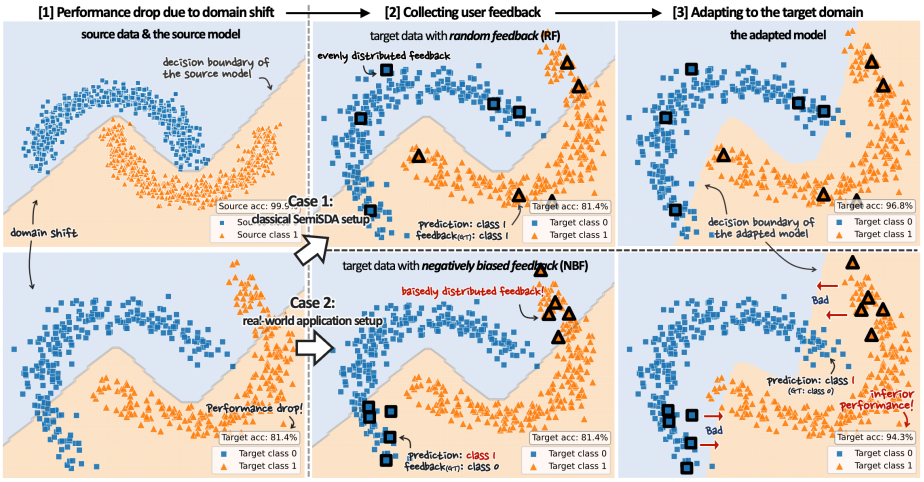

Fig. 10: Effect of negatively biased feedback. We conduct an additional simulation study with two moon dataset. We make the same observations of Figure 3, *i.e.*, NBF is biasedly distributed, leading to inferior adaptation performance compared to RF. The experimental details are provided in Section 3.2 and Section B.

	feed. amount	378 (3 labeled data per class)			630 (5 labeled data per class)		
	method	RF	NBF	w/ ours	RF	NBF	w/ ours
ResNet-50	SHOT [34]	69.6	70.7 (+1.1)	71.5 (+0.8)	71.1	72.3 (+1.2)	73.0 (+0.7)
	NRC [93]	66.3	64.9 (-1.4)	69.3 (+4.4)	68.5	66.4 (-2.1)	69.6 (+3.2)
	ContraTTA [8]	68.6	69.2 (+0.6)	71.6 (+2.4)	70.1	70.5 (+0.4)	72.4 (+1.9)
	GuidingSP [35]	69.7	70.2 (+0.5)	**71.8** (+1.6)	70.5	71.0 (+0.5)	**72.8** (+1.8)
ViT-S	SHOT [34]	73.4	73.7 (+0.3)	74.1 (+0.4)	74.4	74.8 (+0.4)	75.4 (+0.6)
	NRC [93]	72.2	71.9 (-0.3)	72.9 (+1.0)	73.9	73.7 (-0.2)	74.6 (+0.9)
	ContraTTA [8]	72.8	73.4 (+0.6)	74.9 (+1.5)	73.9	74.8 (+0.9)	76.4 (+1.6)
	GuidingSP [35]	73.3	73.7 (+0.4)	**75.0** (+1.3)	74.1	74.9 (+0.8)	**76.4** (+1.5)

Table 12: Comparisons on DomainNet-126. We combine our method and *SFDA* methods. The average accuracy (%) of seven domain-shift scenarios is reported. We use the same pre-trained model as in Table 3.

as baselines within our framework. To construct the overall loss function $\mathcal{L}_{total}$ in Eq. (2), we simply combine their $\mathcal{L}_{unsup}$ with the supervised loss $\mathcal{L}_{sup}$ of FreeMatch [81] since SFDA methods do not take the utilization of supervised loss into account. The results are presented in Table 12. Interestingly, some SFDA works [8,34,35] using sophisticated methods, such as k-means clustering [21] and contrastive learning [22], are likely to be less susceptible to NBF. However, the trend is not consistent for all methods. NRC [93], using a strategy of nearest neighbors, shows sub-optimal performance under an NBF assumption. Notably, all SFDA methods achieve their best adaptation performance when combined with our method. This suggests that even methods that *partially* mitigate NBF's unexpected effects can further benefit from our method.

Number of appended defending samples. As mentioned in Section 4.2, we incorporate k defending samples for each labeled data point (x_{lb}^b, y_{lb}^b) to decrease the unexpected impact of NBF on the supervised signal. To understand how

		$k=1$	$k=2$	$k=3$	$k=4$	baseline
FreeMatch [81]	Res.	74.0	74.6	**74.8**	74.4	72.0
FreeMatch [81]	ViT.	75.5	**75.9**	75.7	75.4	73.9

Table 13: We ablate **the number of defending samples** k in Eq. (2). We also report the performance of the baseline without our approach (rightmost).

	only $\mathcal{L}_{unsup}$	the overall loss $\mathcal{L}_{total}$ in Eq. (??)			
pseudo-feedback per class	0	3	w/ ours	5	w/ ours
NRC [93]	63.5	63.4	64.6 (+1.2)	63.4	64.4 (+1.0)
ContrastiveTTA [8]	66.6	66.6	67.4 (+0.8)	66.5	67.2 (+0.7)

Table 14: Although out of our scope, we consider **a zero-feedback scenario** in which a user does not provide any feedback. To evaluate our method in this scenario, we leverage unlabeled target data and their pseudo label for semi-supervised adaptation.

the value of k affects performance, we conducted an ablation study in Table 13. We fix the number of labeled data points to 16 and maintain the total batch size at 128 by adjusting the ratio μ in Eq. (1). For instance, with k=4, the ratio μ is set to 3 (*i.e.*, $16 + 16\times k + 16\times\mu = 128$). Our experiments across two different architectures reveal that a k=3 value generally yields good adaptation performance. Consequently, we adopt k=3 for all experiments.

Under a zero feedback scenario. We note that, as previous SemiSDA [6, 58] and SemiSL [66, 81] works, we assume that a user provides a small amount of feedback (*i.e.*, labeled data) during their interaction with an ML application. Nevertheless, we wondered about a broader question: how can our method be used when no feedback is received? This scenario, while beyond the scope of our work, presents an intriguing area for further exploration, so we attempt to investigate the potential impact of our method under such a scenario. We initially opted to use SFDA baselines of Table 12, which have demonstrated potential in the absence of labeled target data, and assess their performance within an SFDA setup (*i.e.*, only $\mathcal{L}_{unsup}$ in Table 14). Then, pseudo-feedback is generated by randomly selecting small unlabeled data sets and their pseudo-labels from samples with high predicted probabilities. With the pseudo-feedback and unlabeled target data, we conduct SemiSDA and report the results (*i.e.*, the overall loss $\mathcal{L}_{total}$ in Table 14). We find that i) simulating pseudo-feedback has a minor influence on SFDA baselines, yet ii) the adaptation performance is enhanced by combining with our method. Based on these results, we believe that even in the absence of feedback for certain classes, SemiSDA with our method can achieve good adaptation performance by leveraging the pseudo-feedback.

D Additional Experimental Details

Details for medical experiments. We use DenseNet-121 [26] provided by the TorchXRayVision repository [13]. This architecture consists of a shared backbone and multiple classification heads for radiographic findings. When given a 256x256 image as input, it generates sigmoid values for thirteen different findings.

The majority of SemiSDA methods, such as AdaMatch [6] and FreeMatch [81], depend on consistency regularization, which requires image augmentation strategies, such as ColorJitter and GaussianBlur [52]. Unfortunately, applying them

to medical images remains challenging, as most strategies have been proposed specifically for natural images. As a result, we employ Pseudo-labeling [1], a fundamental SemiSL algorithm that (i) obviates the necessity for image augmentations and (ii) can be easily implemented for a multi-finding binary classification setup. To be more specific, we substitute the cross-entropy $\mathcal{H}(\cdot,\cdot)$ in Eq. (3) with the binary cross-entropy loss. To generate pseudo labels (*i.e.*, presence or absence in Table 5 (top)), thresholds that are pre-calculated in the source domain are used. The hyper-parameters for model updates are the following.

	batch size	learning rate	optimizer	weight decay
pre-training	128	1e-3	Adam	1e-5
adaptation	128	1e-4	Adam	1e-5

Details for semantic segmentation experiments. Our experiment leverages the GTA5 [56] and Cityscapes [14] datasets as the source and target domains. To compute the supervised $\mathcal{L}_{sup}$ and unsupervised losses $\mathcal{L}_{unsup}$ in Eq. (1), we employ baseline algorithms: IAST [46] in LabOR [63] and RIPU [85]. Following previous works [63, 85], we utilize ResNet-101 as the backbone architecture and DeepLab-v2 as the segmentation model. Further details regarding implementation and hyper-parameter for adaptation can be found in the publicly available codebase of RIPU [85]. One of our method's key strengths is its simplicity, which makes it readily applicable to various tasks like semantic segmentation. To be more specific, we first identify pixel points in an image that have the top 40% probabilities for each class. Among them, we select three pixels (*i.e.*, defending pixels) for each labeled pixel in order to balance the supervised signal (*i.e.*, $\mathcal{L}_{total}$ in Eq. (2)) and obtain robust adaptation performance to the unexpected effect of NBF.

E Additional Discussion

E.1 Technique novelty

Compared to previous works, our approach, retrieval latent defending, distinguishes itself in how balancing is applied to solve the novel NBF problem.: (*i*) We initially anticipated that conventional tricks using confident pseudo labels or balancing strategy, such as CReST [83] for class-imbalance, CLUE [55], DiaNA [25] for ActiveDA, GuidSP [35] and SSNLL [10] for noisy pseudo labels, would ameliorate the NBF issue. However, as shown in the table below [3], we found these methods to fall short due to their lack of specific targeting of the novel problem by NBF, thereby underscoring the need for our tailored approach. (*ii*) Our strategy diverges from the *dataset*-level balancing approaches in [10, 55, 83]. Instead, we focus on enhancing the supervised signal within a *minibatch* through iterative retrieval of defending samples, which helps in fortifying latent spaces against the unexpected issue by NBF as illustrated in Figure 4 and Table 5. Surprisingly, this distinct method not only effectively addresses the NBF problem but also leads to substantial improvements in adaptation performance.

[3] We evaluate SSNL using the same experimental setup in Table 12.

[4] We specify the database size when the real domain of the DomainNet dataset serves as the target domain.

method	CReST (CVPR21)	CLUE (ICCV21)	DiaNA (CVPR23)	SSNLL (IROS22)	GuidSP (CVPR23)
reference	Figure 8	Table 9	Table 9	-	Table 12
accuracy	92.6	68.6	68.1	68.9	69.2
w/ ours	95.8 (+**3.2**)	71.5 (+**2.9**)	71.3 (+**3.2**)	71.4 (+**2.5**)	71.6 (+**2.4**)

E.2 Computational overhead

Our method incurs only negligible overhead, as the only additional data† that needs to be stored are pseudo labels. As shown in the following table [4], our method results in an additional 0.1 MB of memory and a 3% increase in running time compared to existing SemiSDA [6, 81] and SFDA [35] methods, but these modest increases facilitate significant performance enhancements. We adhere to the standard practices of SemiSDA and SFDA, which involve storing target images in a database (DB)‡.

method	AdaMatch (ICLR22)	w/ ours	GuidSP (CVPR23)	w/ ours	FreeMatch (ICLR23)	w/ ours
reference	Table 3	Table 3	Table 12	Table 12	Table 11	Table 11
DB size‡	55k images	55k images	55k images	55k images	5k images	5k images
add. data†	0 MB	0.1 MB	53.8 MB	53.9 MB	0 MB	0.01 MB
run. time	132 min	136 min	150 min	155 min	14 min	15 min
accuracy	64.5	72.0 (+**7.5**)	70.2	71.8 (+**1.6**)	66.9	68.9 (+**2.0**)

E.3 Limitations.

Machine learning (ML) powered products can collect target data in various ways. Beyond unlabeled data encountered in the target environment (e.g., driving scenes from a self-driving car), feedback containing valuable target information can be collected by users. For example, a radiologist can log misdiagnosed chest X-ray images in the medical application. However, leveraging effectively such feedback to enhance the deployed model has yet to be well studied. So, this paper addressed this issue by proposing a framework, domain adaptation with user feedback, as illustrated in Figure 2. Moreover, we identified potential issues (*i.e.*, the unexpected impact of NBF) and introduced a simple and scalable solution (*i.e.*, retrieval latent defending).

However, a few more considerations need to be made before this framework is applied in the real world. (1) Current SemiSDA and SemiSL works typically conduct a single adaptation round using all target training data. In practice, however, periodic adaptation may be required since the model can continuously collect new data. According to CoTTA [79], EATA [47], and EcoTTA [67], studies to make initial TTA research [8, 39, 77] more realistic, long-term adaptation can lead to catastrophic forgetting and error accumulation. They attempt to address this problem by utilizing continual learning strategies, *e.g.*, random parameter restoration and knowledge distillation. Repeated adaptation processes in our setup might result in similar issues, suggesting a potential connection to continual learning techniques within the SemiSDA methods. (2) More SemiSDA methods specializing in medical imaging still need to be developed. We employed the native SemiSDA method, Pseudo-Labeling [1], in Table 5. Developing SemiSDA methods specific to medical imaging has the potential to significantly improve adaptation performance beyond the results of Table 5. It is also a promising direction for future research.

F Results of all domain shifts

In addition to Table 3, Table 4, and Table 12, we report the adaptation results for all domain shift scenarios in Table 15, Table 16, Table 17, Table 18, Table 19, and Table 20.

	method	feedback	average	real→clip.	real→pain.	pain.→clip.	clip.→scat.	scat.→pain.	real→scat.	pain.→real
ResNet-50 [23]	Source model	-	56.5	56.1	63.7	55.2	48.0	51.7	45.8	74.7
	FixMatch [66]	RF	67.6	66.2	68.3	68.2	61.0	69.8	58.7	80.8
		NBF	63.4	62.4	65.1	64.8	55.8	64.6	52.7	78.4
	w/ ours	NBF	73.2	75.0	74.3	74.7	66.9	71.8	65.4	84.1
	UDA [87]	RF	69.2	68.7	70.0	69.8	62.8	70.9	60.0	82.0
		NBF	64.9	64.5	66.0	67.3	57.2	66.3	53.8	79.5
	w/ ours	NBF	73.4	76.2	74.0	74.7	67.4	71.9	65.7	84.1
	FlexMatch [96]	RF	73.3	76.7	74.0	75.6	66.9	73.2	64.4	82.5
		NBF	71.4	74.8	72.2	74.5	63.8	71.1	61.7	81.4
	w/ ours	NBF	74.7	77.9	74.8	77.8	68.9	72.2	66.9	84.4
	FreeMatch [81]	RF	73.8	76.6	74.2	75.5	67.7	73.5	65.1	84.0
		NBF	72.0	75.5	72.9	74.6	65.0	72.3	62.0	81.7
	w/ ours	NBF	74.8	78.1	74.5	77.1	68.8	72.4	67.3	85.0
	MME [58]	RF	69.5	70.0	71.2	69.3	63.5	69.6	61.7	81.5
		NBF	68.4	69.5	70.7	69.1	61.5	69.0	58.8	80.2
	w/ ours	NBF	70.8	72.9	71.6	72.9	64.0	68.4	62.1	83.5
	CDAC [33]	RF	68.3	67.1	69.0	68.9	62.6	69.9	59.5	81.1
		NBF	64.6	64.5	66.2	66.3	56.9	65.8	53.6	78.6
	w/ ours	NBF	73.2	76.1	73.9	74.4	67.0	71.2	65.8	84.1
	AdaMatch [6]	RF	67.6	66.6	68.5	68.5	60.3	69.2	58.7	81.5
		NBF	64.5	64.3	66.1	65.6	56.9	65.6	54.2	78.9
	w/ ours	NBF	72.0	74.5	72.7	73.9	65.5	70.0	64.3	83.2
	Fully sup.	-	83.6	85.6	81.4	85.6	80.4	81.4	80.4	90.1
ViT-S [15]	Source model	-	64.5	63.6	70.2	61.6	56.7	65.5	53.5	80.5
	FixMatch [66]	RF	74.6	75.5	77.1	73.8	67.7	75.9	67.1	85.1
		NBF	73.0	73.8	75.4	74.0	65.1	72.8	66.1	83.8
	w/ ours	NBF	75.6	77.1	77.7	77.3	67.8	76.8	68.0	84.7
	UDA [87]	RF	74.8	75.5	77.1	74.0	67.9	76.1	67.4	85.4
		NBF	73.3	74.1	75.6	74.3	65.4	73.2	66.3	83.9
	w/ ours	NBF	75.8	77.1	77.8	77.6	68.2	77.1	68.2	84.9
	FlexMatch [96]	RF	74.9	75.5	77.0	74.7	68.4	76.2	66.7	85.7
		NBF	73.9	74.5	76.6	75.1	66.1	74.5	66.4	84.1
	w/ ours	NBF	75.8	77.2	77.5	77.9	68.3	77.0	67.9	85.0
	FreeMatch [81]	RF	74.9	75.3	76.8	74.5	68.1	76.5	67.0	86.0
		NBF	73.9	74.6	76.4	75.0	66.0	74.5	66.5	84.1
	w/ ours	NBF	75.7	76.9	77.5	77.9	68.1	76.7	67.8	85.2
	MME [58]	RF	73.2	74.0	74.8	73.0	66.5	74.6	65.2	84.3
		NBF	72.7	73.2	74.8	73.8	65.3	73.0	64.8	83.8
	w/ ours	NBF	74.1	75.4	75.9	76.2	66.2	74.7	66.4	84.2
	CDAC [33]	RF	74.2	74.8	76.3	73.8	67.5	75.5	66.6	84.9
		NBF	72.8	73.6	74.9	73.9	65.0	72.8	65.4	83.8
	w/ ours	NBF	75.4	76.7	77.6	77.2	67.6	76.2	67.9	84.6
	AdaMatch [6]	RF	74.7	75.3	76.9	73.8	68.0	76.3	67.1	85.5
		NBF	73.7	74.7	76.2	74.7	65.7	74.0	66.8	84.0
	w/ ours	NBF	75.9	76.9	77.8	77.8	68.5	76.6	68.3	85.1
	Fully sup.	-	85.4	87.8	83.4	87.8	81.3	83.4	81.3	92.7

Table 15: Adaptation results with SemiSL and SemiSDA methods on DomainNet-126. The adaptation performance on various domain shifts is reported, where the number of labeled data per class is 3. The details can be found in Table **??**.

	method	feedback	average	real→clip.	real→pain.	pain.→clip.	clip.→scat.	scat.→pain.	real→scat.	pain.→real
ResNet-50 [23]	Source model	-	56.5	56.1	63.7	55.2	48.0	51.7	45.8	74.7
	FixMatch [66]	RF	71.5	71.3	70.9	73.1	65.5	71.9	65.1	83.0
		NBF	66.1	66.2	67.6	67.6	57.4	67.4	56.5	79.8
	w/ ours	NBF	75.1	77.2	75.7	77.2	69.8	73.9	68.0	84.1
	UDA [87]	RF	72.9	73.4	72.6	74.6	67.1	73.1	65.9	83.4
		NBF	68.8	70.3	68.7	71.1	60.3	70.3	60.5	80.5
	w/ ours	NBF	75.3	78.3	75.2	77.9	69.6	73.9	68.1	84.3
	FlexMatch [96]	RF	75.3	78.5	74.6	77.5	70.3	73.8	68.7	83.8
		NBF	73.9	77.3	74.0	76.3	66.2	73.8	67.2	82.6
	w/ ours	NBF	76.0	79.5	75.6	78.7	70.2	74.3	69.0	84.7
	FreeMatch [81]	RF	75.6	78.6	74.9	77.6	70.2	74.3	69.0	84.7
		NBF	74.4	77.6	74.5	76.3	66.8	73.8	68.2	83.5
	w/ ours	NBF	76.1	79.6	75.5	78.6	70.4	74.5	69.3	84.9
	MME [58]	RF	71.2	71.3	72.1	71.8	65.6	70.7	64.6	82.6
		NBF	70.1	71.4	71.4	70.4	62.1	70.7	62.7	81.8
	w/ ours	NBF	72.5	74.5	72.7	74.9	66.4	70.7	64.6	83.8
	CDAC [33]	RF	71.7	71.5	71.7	73.0	66.1	72.0	64.8	82.9
		NBF	68.1	69.5	68.9	69.3	59.8	69.4	59.7	80.0
	w/ ours	NBF	74.9	77.0	74.9	77.0	69.6	73.4	67.9	84.2
	AdaMatch [6]	RF	70.9	70.6	70.4	72.7	65.3	70.8	63.7	83.0
		NBF	67.7	69.0	68.7	69.7	59.5	67.6	58.8	80.4
	w/ ours	NBF	74.3	76.7	74.4	76.8	68.8	72.8	66.2	84.1
	Fully sup.	-	83.6	85.6	81.4	85.6	80.4	81.4	80.4	90.1
ViT-S [15]	Source model	-	64.5	63.6	70.2	61.6	56.7	65.5	53.5	80.5
	FixMatch [66]	RF	75.7	76.5	77.4	76.2	69.6	76.9	67.9	85.8
		NBF	74.3	75.7	75.6	75.6	67.4	74.2	66.7	84.7
	w/ ours	NBF	76.5	78.0	77.9	78.3	70.2	76.9	68.7	85.4
	UDA [87]	RF	75.9	76.7	77.4	76.4	69.8	76.9	68.1	85.9
		NBF	74.5	75.9	76.0	76.0	67.6	74.4	67.0	84.9
	w/ ours	NBF	76.7	78.2	78.2	78.8	70.6	76.9	68.8	85.5
	FlexMatch [96]	RF	76.0	76.5	77.2	76.8	70.1	77.3	68.1	86.2
		NBF	75.1	76.2	76.6	76.2	68.9	75.5	67.4	85.1
	w/ ours	NBF	76.9	78.9	77.9	79.1	70.4	77.6	68.6	86.0
	FreeMatch [81]	RF	76.0	76.7	77.1	76.6	69.9	77.1	68.0	86.3
		NBF	75.1	76.2	76.4	76.3	69.0	75.6	67.4	85.1
	w/ ours	NBF	76.8	78.5	77.8	78.5	70.5	77.4	68.8	85.9
	MME [58]	RF	74.5	75.3	75.4	75.2	68.2	75.4	66.5	85.1
		NBF	74.0	74.9	75.2	75.2	67.3	74.1	66.3	84.7
	w/ ours	NBF	75.2	76.4	76.4	77.3	68.9	75.4	66.9	85.1
	CDAC [33]	RF	75.4	76.3	76.9	75.5	69.2	76.4	67.8	85.6
		NBF	74.1	75.1	75.3	75.4	67.4	73.9	66.5	84.6
	w/ ours	NBF	76.2	77.8	77.4	78.3	70.0	76.4	68.5	85.3
	AdaMatch [6]	RF	75.9	76.6	77.1	76.6	70.0	76.9	68.2	86.1
		NBF	75.1	76.2	76.7	76.3	68.1	75.5	67.5	85.2
	w/ ours	NBF	76.7	78.6	78.0	78.8	69.6	77.2	68.8	86.0
	Fully sup.	-	85.4	87.8	83.4	87.8	81.3	83.4	81.3	92.7

Table 16: Adaptation results with SemiSL and SemiSDA methods on DomainNet-126. The adaptation performance on various domain shifts is reported, where the number of labeled data per class is 5. The details can be found in Table **??**.

method	feedback	average	a→c	a→p	a→r	c→a	c→p	c→r	p→a	p→c	p→r	r→a	r→c	r→p
Source	-	57.6	44.2	65.6	71.6	47.3	60.2	58.2	47.9	40.8	69.8	60.6	46.5	78.1
FreeMatch [81]	RF	71.4	56.6	79.7	76.3	67.9	83.2	74.5	65.5	58.6	78.3	69.4	62.4	84.8
	NBF	68.6	53.0	76.1	75.3	65.3	78.5	74.8	62.5	56.7	74.7	66.7	56.2	83.7
w/ ours	NBF	73.7	60.8	80.3	80.5	69.2	84.0	78.6	67.7	62.3	80.1	70.0	64.1	87.2
UDA [87]	RF	72.2	56.1	81.0	76.8	68.0	83.4	75.6	67.1	59.7	79.7	69.8	62.7	86.4
	NBF	69.5	53.3	78.6	75.7	66.3	79.7	75.8	63.7	57.2	75.7	66.7	57.2	83.9
w/ ours	NBF	74.1	61.1	80.7	80.3	69.0	85.9	79.2	68.0	62.3	80.7	70.4	63.9	87.4
FlexMatch [96]	RF	73.7	58.0	84.6	79.3	68.4	84.7	78.8	68.4	62.8	79.8	70.6	62.9	86.3
	NBF	72.1	56.1	79.0	77.8	68.4	83.4	77.6	67.5	60.1	79.2	68.8	60.5	86.2
w/ ours	NBF	74.7	60.8	81.7	81.1	70.0	85.8	79.8	68.8	61.4	81.4	70.2	65.7	89.4
FreeMatch [81]	RF	74.0	58.5	85.0	79.4	68.2	84.7	79.2	68.4	62.5	80.4	71.0	63.7	87.0
	NBF	72.2	56.4	79.3	77.7	67.7	83.4	78.5	67.3	60.5	79.1	69.2	61.0	86.9
w/ ours	NBF	74.8	60.6	81.4	81.5	70.8	86.7	80.0	68.6	61.6	81.7	69.8	66.2	89.2
MME [58]	RF	71.2	56.2	80.4	75.7	65.1	81.0	76.7	64.5	59.0	79.8	69.0	62.0	85.1
	NBF	70.2	55.0	77.6	76.8	65.1	82.2	77.7	61.1	57.1	77.1	68.8	58.1	85.4
w/ ours	NBF	73.4	60.5	81.4	80.0	68.6	84.8	78.4	65.3	61.3	79.8	69.8	62.8	87.5
CDAC [33]	RF	71.2	55.5	80.0	76.4	67.1	82.4	75.8	64.5	58.7	79.0	69.2	61.5	84.4
	NBF	69.0	54.1	76.2	75.4	64.1	79.5	75.4	63.9	57.9	75.2	66.5	55.8	83.6
w/ ours	NBF	74.3	63.7	81.3	80.4	70.0	85.4	79.0	67.9	62.2	80.3	69.6	65.1	86.9
AdaMatch [6]	RF	70.9	55.4	80.4	75.9	65.7	81.5	74.6	65.9	58.7	78.4	68.8	61.5	84.3
	NBF	69.3	54.2	76.6	75.3	65.9	79.3	75.5	63.7	57.4	75.9	66.7	56.8	84.2
w/ ours	NBF	73.8	62.2	81.0	79.7	68.8	85.4	78.6	67.7	61.7	79.5	69.0	64.1	88.2
Fully sup.	-	87.4	84.5	95.1	89.0	80.9	95.1	89.0	80.9	84.5	89.0	80.9	84.5	95.1

Table 17: Adaptation results with SemiSL and SemiSDA methods on OfficeHome. The adaptation performance on various domain shifts is reported, where the number of labeled data per class is 3. The details can be found in Table **??**.

method	feedback	average	a→c	a→p	a→r	c→a	c→p	c→r	p→a	p→c	p→r	r→a	r→c	r→p
Source	-	57.6	44.2	65.6	71.6	47.3	60.2	58.2	47.9	40.8	69.8	60.6	46.5	78.1
FreeMatch [81]	RF	73.9	59.8	83.9	80.0	69.2	84.6	77.8	66.7	63.7	80.5	71.2	63.3	85.8
	NBF	72.2	57.6	82.4	76.3	68.2	82.0	76.6	65.3	61.4	78.0	71.4	60.9	86.6
w/ ours	NBF	75.3	63.9	84.6	79.0	70.0	85.7	79.1	68.6	64.8	81.1	73.0	65.4	88.6
UDA [87]	RF	74.4	60.2	84.5	79.8	68.4	85.1	79.8	66.5	64.4	80.7	72.2	64.4	86.1
	NBF	73.0	58.7	82.6	77.4	68.6	82.5	77.3	66.9	61.9	78.8	71.4	62.0	87.4
w/ ours	NBF	76.0	64.9	84.5	79.4	71.2	85.8	79.8	71.4	65.4	80.5	74.2	66.3	88.9
FlexMatch [96]	RF	75.9	64.3	84.9	82.1	69.6	85.7	80.4	69.2	65.7	82.3	74.2	65.4	87.3
	NBF	74.9	62.9	83.2	77.6	70.2	84.7	80.5	69.8	62.9	79.5	74.4	64.4	87.7
w/ ours	NBF	76.6	63.3	86.7	79.5	71.6	86.9	81.0	72.0	65.7	81.3	75.0	67.5	88.9
FreeMatch [81]	RF	75.8	63.2	85.2	81.8	70.0	86.3	80.6	69.0	65.8	82.1	73.2	65.6	87.0
	NBF	75.0	63.2	83.6	77.4	70.0	84.9	80.5	70.4	62.6	79.8	74.6	63.9	88.9
w/ ours	NBF	76.6	63.4	85.6	79.8	71.8	86.3	81.2	71.8	65.3	81.8	74.8	67.5	89.6
MME [58]	RF	73.5	59.6	82.4	78.7	67.3	83.6	79.2	67.3	62.4	80.5	71.4	63.2	86.6
	NBF	73.1	59.5	83.2	77.2	66.5	82.5	78.3	65.1	61.5	79.1	72.8	62.8	88.3
w/ ours	NBF	75.6	63.6	84.2	77.3	69.8	85.5	80.3	70.8	65.2	80.5	74.6	66.6	88.9
CDAC [33]	RF	73.5	59.7	83.4	79.3	68.6	84.5	78.1	66.3	63.4	80.5	69.8	63.4	85.1
	NBF	72.3	59.5	81.7	76.6	67.7	81.9	76.7	65.9	62.4	77.4	70.8	60.2	86.4
w/ ours	NBF	75.7	64.2	84.7	79.0	72.2	85.5	79.6	70.4	65.3	80.4	73.4	65.4	88.6
AdaMatch [6]	RF	73.4	60.0	83.8	78.7	68.0	84.4	77.6	66.5	62.5	80.0	71.0	63.2	85.1
	NBF	72.7	60.2	81.7	76.9	67.1	81.5	77.2	66.3	61.8	78.7	71.2	62.0	87.1
w/ ours	NBF	75.5	63.4	84.4	78.8	70.0	86.0	79.4	70.2	65.3	80.6	72.8	66.6	88.4
Fully sup.	-	87.4	84.5	95.1	89.0	80.9	95.1	89.0	80.9	84.5	89.0	80.9	84.5	95.1

Table 18: Adaptation results with SemiSL and SemiSDA methods on OfficeHome. The adaptation performance on various domain shifts is reported, where the number of labeled data per class is 5. The details can be found in Table **??**.

	method	feedback	average	real→clip.	real→pain.	pain.→clip.	clip.→scat.	scat.→pain.	real→scat.	pain.→real
ResNet-50 [23]	Source model	-	56.5	56.1	63.7	55.2	48.0	51.7	45.8	74.7
	SHOT [34]	RF	69.6	70.2	70.9	69.6	63.4	69.1	61.4	82.8
		NBF	70.7	71.7	72.7	71.0	64.1	69.7	62.0	83.6
	w/ ours	NBF	71.5	73.8	72.8	73.5	64.6	69.8	62.6	83.6
	NRC [93]	RF	66.3	66.1	69.3	64.8	58.0	67.9	57.6	80.6
		NBF	64.9	63.1	68.4	63.6	56.9	67.1	55.1	80.4
	w/ ours	NBF	69.3	70.2	71.4	69.7	62.1	68.2	62.0	81.4
	ContraTTA [8]	RF	68.6	72.3	70.4	70.7	60.0	65.1	61.6	80.1
		NBF	69.2	72.8	70.9	71.1	60.2	66.5	62.1	80.7
	w/ ours	NBF	71.6	74.6	72.1	75.3	64.1	69.7	62.7	82.7
	GuidingSP [35]	RF	69.7	66.6	68.5	68.5	60.3	69.2	58.7	81.5
		NBF	70.2	64.3	66.1	65.6	56.9	65.6	54.2	78.9
	w/ ours	NBF	71.8	74.5	72.7	73.9	65.5	70.0	64.3	83.2
	Fully sup.	-	83.6	85.6	81.4	85.6	80.4	81.4	80.4	90.1
ViT-S [15]	Source model	-	64.5	63.6	70.2	61.6	56.7	65.5	53.5	80.5
	SHOT [34]	RF	73.4	73.9	74.9	73.2	66.8	74.8	65.4	84.7
		NBF	73.7	74.6	75.6	74.2	67.0	74.4	65.4	84.6
	w/ ours	NBF	74.1	75.1	75.7	74.9	67.6	74.6	66.0	84.7
	NRC [93]	RF	72.2	73.0	73.9	72.3	65.6	73.6	63.8	83.0
		NBF	71.9	73.1	73.8	72.1	65.2	73.0	64.1	82.3
	w/ ours	NBF	72.9	73.9	74.9	73.9	65.5	73.4	64.5	84.3
	ContraTTA [8]	RF	72.8	73.0	74.1	74.7	66.7	73.2	62.9	84.8
		NBF	73.4	74.3	75.1	74.6	67.6	73.8	63.7	84.9
	w/ ours	NBF	74.9	75.4	75.8	76.7	69.2	75.6	66.6	85.0
	GuidingSP [35]	RF	73.3	73.9	74.5	75.0	66.9	73.7	63.4	85.1
		NBF	73.7	74.8	75.5	74.6	67.8	73.9	63.9	85.1
	w/ ours	NBF	75.0	75.6	75.8	76.9	69.1	75.6	66.5	85.2
	Fully sup.	-	85.4	87.8	83.4	87.8	81.3	83.4	81.3	92.7

Table 19: Adaptation results with SFDA methods on DomainNet-126. The adaptation performance on various domain shifts is reported, where the number of labeled data per class is 3. The details can be found in Table 12.

	method	feedback	average	real→clip.	real→pain.	pain.→clip.	clip.→scat.	scat.→pain.	real→scat.	pain.→real
ResNet-50 [23]	Source model	-	56.5	56.1	63.7	55.2	48.0	51.7	45.8	74.7
	SHOT [34]	RF	71.1	71.9	72.6	70.8	65.3	70.1	63.6	83.1
		NBF	72.3	73.3	74.0	73.1	65.8	71.5	64.4	84.2
	w/ ours	NBF	73.0	75.2	74.2	74.3	66.3	71.4	65.1	84.5
	NRC [93]	RF	68.5	68.6	70.1	68.3	61.1	68.6	61.5	81.2
		NBF	66.4	65.4	69.0	65.7	58.9	67.0	58.6	80.7
	w/ ours	NBF	69.6	70.6	72.2	70.2	61.9	68.1	62.4	81.6
	ContraTTA [8]	RF	70.1	73.7	71.0	72.4	61.8	67.0	64.0	81.0
		NBF	70.5	74.4	71.8	72.3	61.4	67.8	64.2	81.3
	w/ ours	NBF	72.4	76.0	73.3	73.1	64.8	71.3	65.0	83.2
	GuidingSP [35]	RF	70.5	70.9	70.6	70.4	72.7	65.3	70.8	63.7
		NBF	71.0	67.7	69.0	68.7	69.7	59.5	67.6	58.8
	w/ ours	NBF	72.8	74.3	76.7	74.4	76.8	68.8	72.8	66.2
	Fully sup.	-	83.6	85.6	81.4	85.6	80.4	81.4	80.4	90.1
ViT-S [15]	Source model	-	64.5	63.6	70.2	61.6	56.7	65.5	53.5	80.5
	SHOT [34]	RF	74.4	75.1	75.6	74.6	68.5	75.2	67.0	85.0
		NBF	74.8	75.9	76.3	75.1	68.7	75.8	66.7	85.3
	w/ ours	NBF	75.4	77.3	76.5	75.9	69.2	76.1	67.1	85.4
	NRC [93]	RF	73.9	75.1	75.1	73.8	67.4	74.2	66.3	85.5
		NBF	73.7	74.8	74.9	73.8	67.2	73.8	66.2	85.0
	w/ ours	NBF	74.6	76.0	75.9	75.5	67.9	74.5	66.7	85.3
	ContraTTA [8]	RF	73.9	74.3	74.9	76.2	68.5	74.1	64.7	84.9
		NBF	74.8	74.9	75.7	76.2	69.2	75.3	66.7	85.5
	w/ ours	NBF	76.4	77.2	76.4	79.0	70.9	76.8	67.8	86.5
	GuidingSP [35]	RF	74.1	74.2	75.0	76.5	68.9	74.2	64.9	85.0
		NBF	74.9	74.9	75.8	76.3	69.1	75.2	66.8	85.9
	w/ ours	NBF	76.4	77.4	76.4	79.1	70.9	76.8	67.7	86.6
	Fully sup.	-	85.4	87.8	83.4	87.8	81.3	83.4	81.3	92.7

Table 20: Adaptation results with SFDA methods on DomainNet-126. The adaptation performance on various domain shifts is reported, where the number of labeled data per class is 5. The details can be found in Table 12.