

CAA : Category-specific Augmentation Search Using Attentional Interpolation for Unsupervised Domain Adaptation

Technical Supplementary Material

Table 6. Augmentation Types. Consider the six types of augmentation strategies to be selected by our method.

No.	List of augmentation strategies	Description
1	Without Augmentation	No augmentation is employed
2	Posterize	50% chance (bits=2)
3	Center-crop	50% chance (size=224)
4	Horizontal flip	50% chance
5	Solarization	50% chance (threshold=192.0)
6	Equalization	50% chance

A Additional Experimental Result

A.1 Implementation Details

For fair comparison against diverse UDA paradigms, we selected six baselines (distance-based: DAN [23], JAN [24], TSA [18]; adversarial: DANN [1], AFN [41]; mixup-based: PMTrans-DeiT [46]) and used their original published code and settings. We evaluate our method on standard UDA benchmarks: Office-31 [31] (3 domains, 31 categories, viewpoint robustness), VisDA-17 [29] (2 domains, 12 categories, large sim-to-real, scalability), and DomainNet [30] (6 domains, 345 categories, large-scale multi-domain). For our augmentation search phase, we used a pre-trained ResNet-18 backbone, trained for 70 epochs with Adam (LR=0.5, weight decay=0.001), MI $\beta = 0.1$, $\lambda = 0.9$, and 500-1000 iterations per epoch, employing six strategies (Table 6). Images were normalized using standard ImageNet mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225] after converting to tensors.

As outlined in the main text, our method’s evaluation involved comparisons against six diverse UDA baselines (DAN [23], JAN [24], TSA [18], DANN [1], AFN [41], PMTrans-DeiT [46]) on standard benchmarks including Office-31 [31] and VisDA-17 [29]. Results and further analysis concerning the large-scale DomainNet [30] benchmark are presented in the Table 7. We use the six types of augmentation strategies specified in Table 6. All experiments were performed using NVIDIA RTX-3090 GPUs. The pseudocode is shown in the Algorithm 1.

A.1.1 Datasets

The office-31 dataset [31] contains 31 object categories spanning three domains—Amazon, DSLR, and Webcam. On average, three different angles of each object were recorded. The 795 low-resolution (640×480) webcam images have a lot of noise, color, and white balance artifacts. Each object captures different viewpoints on average three times to create various situations, which would be useful for the evaluation of methods’ robustness to domain shifts. In addition, it also evaluates the robustness of various domain combinations. The VisDA-17 [29] is a scalable simulation-to-real dataset with over 280,000 images spread over 12 categories for visual domain adaptation. While the validation images are gathered from MSCOCO [21], the training images are synthesized using a 3D object model under various capturing environments. The VisDA-17 dataset

Algorithm 1 Category-specific Augmentation Search

Input: $S = \{(\mathbf{I}_i^s, y_i)\}_{\forall i}$, $U = \{(\mathbf{I}_i^t)\}_{\forall i}$

Parameter: v_n^k where $k \in \{1, \dots, K\}$, $n \in \{1, \dots, N\}$, B : Batch size,

λ_0 : User-defined hyperparameter that balances the loss between the source and target domains.

Output: Category-specific Augmentation Methods

Search Phase:

- 1: Initialize \mathbf{v} by 1
- 2: **for** $t=1$ to T **do**
- 3: $\lambda = (t/T) \times \lambda_0$
- 4: $\alpha_n \leftarrow \text{softmax}([v_n^1, \dots, v_n^K])$
- 5: $S \leftarrow \{(\mathbf{I}_b^s, y_b)\}_{b \in (1 \dots B)}$
- 6: $\hat{S}^n \leftarrow n\text{-th augmentation to } S \text{ for } n \in \{1, \dots, N\}$
- 7: $U \leftarrow \{(\mathbf{I}_i^t)\}_{i \in (1 \dots B)}$
- 8: Compute probabilistic outputs of target samples: $\{\mathbf{y} = \text{softmax}(\mathbf{y})\}$, generate target pseudo labels: $\{T(\mathbf{y}) = \arg \max(\mathbf{y})\}$
- 9: Loss estimation by Eq. (4) [Full paper part]
- 10: Update the model and $\forall v_n^k$
- 11: **end for**
- 12: Update weight parameters α_n^k by minimizing the loss in Eq. (4) with an optimizer.
- 13: **return** $\forall v_n^k$

Training Phase:

- 1: $\mathbf{v} \leftarrow$ Search phase
- 2: $\Theta \leftarrow$ Pre-trained model weights
- 3: Fine-tune Θ using augmentations from \mathbf{v}

can be reflected in actual data when adapting the synthetic data generated in multiple environments for the same object to the real domain. A large dataset will be meaningful in scalability verification. Six unique domains and the shared items comprise the DomainNet [30]. Three hundred forty-five types, or categories, of objects, such as cello, plane, bird, and bracelet, are present in all domains. These include quickdraw drawings of players of the global game ‘Quick Draw,’ painting artistic representations of objects in the form of paintings, real: pictures taken from photographs and the actual world; sketch: drawings of certain objects; infographics: pictures of infographics featuring particular objects; and clipart: a collection of clipart pictures.

A.1.2 Additional Quantitative Results for Office-31

Further analysis across domains reveals additional insights into CAA’s effectiveness (Table 1). The D→A adaptation task exhibited the highest average gain among all tasks, reaching approximately +6.4%. This high average was significantly influenced by a substantial +15.2% improvement when CAA was applied to the strong

Table 7. Evaluate Generalization on DomainNet for UDA (single-source).

Method	c→p	c→r	c→s	p→c	p→r	p→s	r→c	r→p	r→s	s→c	s→p	s→r	Avg.	Gain
MCC [17]	37.7	55.7	42.6	45.4	59.8	39.9	54.4	53.1	37.0	58.1	46.3	56.2	48.9	
DAN [23]	38.8	55.2	43.9	45.9	59.0	40.8	50.8	49.8	38.9	56.1	45.9	55.5	48.4	
+ CAA(ours)	40.4	55.7	46.3	46.3	59.0	41.5	50.1	50.3	40.4	56.1	45.9	55.5	49.0	(+0.6%)
JAN [24]	40.5	56.7	45.1	47.2	59.9	43.0	54.2	52.6	41.9	56.6	46.2	55.5	50.0	
+ CAA(ours)	42.8	57.6	48.4	47.6	59.3	45.1	55.0	53.7	45.5	57.0	47.6	56.1	51.3	(+1.3%)

Table 8. Variant Augmentation Strategies on VisDA-17, Synthetic → Real(TSA). 1) a-type : basically selected six augmentation strategies, 2) b-type : different types of six augmentation strategies, 3) c-type : totally ten augmentation strategies. (add the 4-type augmentation strategies)

Method	a-type(%)	b-type(%)	c-type(%)
Baseline	78.6	78.6	78.6
CAA(Ours)	81.5	79.5	80.1

PMTrans-DeiT baseline, showcasing CAA’s potential even on advanced models for certain tasks. Conversely, the A→D task presented more varied results; while achieving a solid average gain of +4.8% and demonstrating significant improvement for most baselines (e.g., +13.4% for DAN), applying CAA to PMTrans-DeiT resulted in a slight performance decrease (-2.5%) on this specific task. This highlights that while generally beneficial, the interaction between CAA and the baseline can vary depending on the specific domain transfer. Tasks where baseline performance was already near saturation, such as W→D (where most baselines achieve 100.0%) and D→W (showing only +0.45% average gain), naturally yielded minimal absolute improvements from CAA, aligning with expected behavior on less challenging transfers. Furthermore, the low standard deviations generally reported for CAA results in Table 1 (mostly below ± 1.0) suggest that the performance enhancements provided by our method are typically stable.

A.1.3 Additional Quantitative Results for DomainNet

To further assess the scalability and robustness of CAA on a significantly larger and more diverse benchmark, we conducted experiments on DomainNet, which encompasses more domains (6) and categories (345) than Office-31 and VisDA-17. We evaluated CAA by applying it to the DAN and JAN baselines under the single-source UDA setting across 12 challenging domain transfer tasks, representing a demanding test scenario.

As shown in Table 7, CAA demonstrated consistent effectiveness by improving the average performance across all 12 tasks for both baselines. Specifically, CAA enhanced DAN’s average accuracy by +0.6% (from 48.4% to 49.0%) and JAN’s by +1.3% (from 50.0% to 51.3%). While these average gains are more modest compared to those observed on smaller datasets like VisDA-17, a crucial finding is that these improvements were achieved without any dataset-specific hyperparameter re-tuning for either CAA or the underlying baseline methods. This is particularly noteworthy given the substantially increased number of categories and the inherent complexity of the DomainNet benchmark.

The ability to achieve performance gains on such a large-scale and complex dataset without specific tuning underscores the robustness and generalization capability of our category-specific augmentation approach. It suggests that the principles guiding the attentional search mechanism and the synergy with MI loss are effective even when faced with greater diversity and scale, without relying on costly dataset-specific optimization. This validates CAA as a generally applicable method that can provide benefits across different scales of

Table 9. Single or Multi Augmentation Strategies Test on Office-31, A→D (AFN) for our Method. 1) single-strategy : use only one augmentation strategy randomly, 2) multi-strategies : use of all augmentation strategies simultaneously, 3) CAA(ours) : performance of our method.

Component	Avg(%)	Gain(%)
baseline	87.1(± 0.2)	
1) single-strategy	92.4(± 0.6)	(+5.4%)
2) multi-strategies	92.1(± 0.3)	(+5.0%)
CAA(ours)	93.6(± 0.1)	(+6.5%)

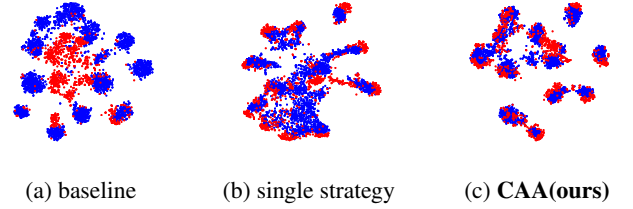


Figure 7. Visualize CAA Performance results on VisDA-17, synthetic(blue)→ real(red)(DAN). (a) baseline(DAN), (b) single-strategy: use only one augmentation strategy randomly, (c) CAA(ours): augmentation search network.

domain adaptation challenges.

B Additional Analysis

B.1 Variant Augmentation Strategies

We conducted additional experiments to rigorously evaluate the robustness and flexibility of our proposed augmentation selection method against varying augmentation candidates. Specifically, we defined three distinct configurations of augmentation strategies to examine their influence on performance: *a-Type* corresponds to our default augmentation set, which includes six basic augmentations as detailed in Table 6. *b-Type* explores an alternative combination, substituting the default augmentations with a different set comprising *w/o aug*, *ColorJitter*, *Autocontrast*, *Sharpness*, *GaussianBlur*, and *Affine*. Lastly, *c-Type* expands upon the default set by integrating additional augmentations, namely *Autocontrast*, *Sharpness*, *GaussianBlur*, and *Affine*, thus totaling ten potential augmentation strategies. Table 8 summarizes the results of these experiments on the VisDA-17 dataset (Synthetic → Real). Our method consistently outperformed the baseline across all three augmentation types. Specifically, for *a-Type*, our method achieved the highest performance increase (81.5%) compared to the baseline (78.6%). In the *b-Type* experiment, despite employing a different set of augmentations, our method still demonstrated noticeable robustness by improving performance (79.5%). Furthermore, when we expanded the augmentation pool to ten strategies (*c-Type*), our approach maintained robust and stable performance (80.1%), clearly indicating its capability to effectively select optimal augmentations from a larger and diverse candidate pool.

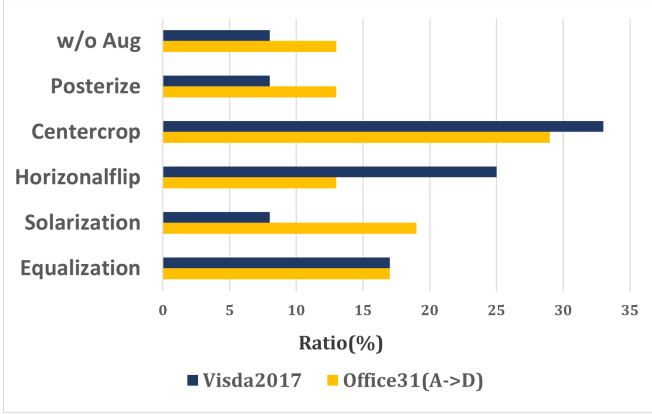


Figure 8. Ratio of Optimal Augmentation Strategies determined for Each Dataset. Optimal augmentation strategy ratios differ by dataset.

Viewpoint strategies dominate for VisDA-17 (diverse views), while Office-31 selects viewpoint, denoising, and color inversion relatively equally.

B.2 Effectiveness of Category-specific Augmentation

To validate the effectiveness of our category-specific augmentation scheme, we conducted a set of comparative experiments using uniform augmentation strategies. Specifically, we explored two scenarios: *Single-strategy* and *Multi-strategies*. In the *Single-strategy* setup, one of six augmentation strategies is randomly selected and applied uniformly to all categories. Conversely, the *Multi-strategies* scenario simultaneously employs all six augmentation strategies uniformly across categories. Table 9 and Fig. 7 show the results of these experiments conducted on the Office-31 dataset (A→D scenario) using the AFN method. The *Single-strategy* method shows a performance improvement of 5.4% compared to the baseline without any augmentation, indicating that even a single randomly selected augmentation can enhance performance significantly. However, the *Multi-strategies* method, despite applying all augmentations simultaneously, results in a slightly lower improvement of 5.0%, likely due to redundant augmentations leading to underfitting.

In contrast, our proposed method, Category-specific Augmentation Adaptation (CAA), achieved a substantial performance gain of 6.5%. This demonstrates that our approach effectively selects and applies optimal augmentation strategies tailored specifically to each category. The visualization provided in Fig. 7 further confirms that our category-specific approach generates more distinct and well-separated feature distributions compared to uniform augmentation strategies. Thus, the results confirm the clear advantage and effectiveness of our category-specific augmentation method in enhancing domain adaptation performance.

B.3 Optimal Augmentation Strategy Ratio

The ratio of the strategies after the category-specific augmentation search phase is shown in Fig. 8. The results for the A→D task in Office-31 are in the ratio of RandomCentercrop (29%), RandomSolarize (19%), RandomHorizontalFlip (13%), and RandomPosterize (13%). The strategies in the Synthetic→Real task in VisDA-17 rank in this order: No-augmentation (33%), RandomSolarize (25%), RandomPosterize (17%), and RandomEqualize (17%). As observed, a different optimal strategy is chosen for each dataset and task. The Centercrop and HorizontalFlip strategy is most frequently selected in Fig. 8; each object in the image within each domain has a different viewpoint, so it judges that the inverting technique is selected for viewpoint matching. Fig. 5 qualitatively show that the pro-

Table 10. Estimation of Average Runtime (Office31, A→D).

Method / Phase	Searching (sec)	Adaptation (sec)	Total (sec)
DAN + CAA	728	8,550	9,278
JAN + CAA	731	12,834	13,565
AFN + CAA	722	10,912	11,634
TSA + CAA	747	7,201	7,948
AVG	732	9,874	10,606

Table 11. Comparison of Augmentation Search Network Runtime(sec),(Office31, A→D).

Method	6-strategies (ours)	6-strategies (other)	10-strategies
Performance(%)	81.5	79.5	80.1
Time(sec)/GPU(MB)	747/8,133	800/8,133	985/8,235

posed search mechanism enhances the category-specific augmentation technique per their visual properties.

C Discussion

We evaluate the effectiveness of our class-specific augmentation approach by analyzing computational efficiency, and augmentation selection quality across various datasets. We also emphasize challenges in certain categories where complex domain differences persist. Finally, we assess the general applicability of our method as a preprocessing step for state-of-the-art domain adaptation models, demonstrating its potential versatility.

C.1 Component-wise Training Duration Analysis

To further analyze the computational overhead associated with our CAA framework, we measured the runtime for the initial augmentation search phase versus the main domain adaptation phase. Table 10 details these timings when applying CAA to four distinct UDA baselines (DAN, JAN, AFN, TSA) on the Office-31 A→D task.

The results highlight the efficiency of our search mechanism. While the adaptation phase duration naturally varies depending on the complexity of the baseline algorithm (ranging from approximately 7,201s to 12,834s), the augmentation search phase consistently required only a brief period, averaging 732 seconds across all four methods. Consequently, this dedicated search constitutes just approximately 6.9% of the total average training time (10,606 seconds). This minimal relative overhead clearly validates the computational efficiency of our proposed attentional search approach.

This demonstrated efficiency provides strong justification for our design choice to decouple the augmentation search and domain adaptation processes into two distinct phases. Attempting to merge these phases or find optimal augmentations through naive methods (like repeated random trials, see Table 5) would drastically increase the overall optimization time due to the need for repeated, costly search iterations within the main training loop. By employing separate phases enabled by our efficient search mechanism, CAA effectively identifies beneficial category-specific augmentations while imposing only a minor computational burden relative to the primary adaptation task, confirming the practicality and efficiency of our method.

C.2 Analysis of Augmentation Strategy Set Choice

We further investigated the impact of the specific set and number of augmentation strategies employed during the search phase. Table 11 presents a comparison on the Office-31 A→D task, evaluating our proposed set of 6 strategies against both an alternative set of

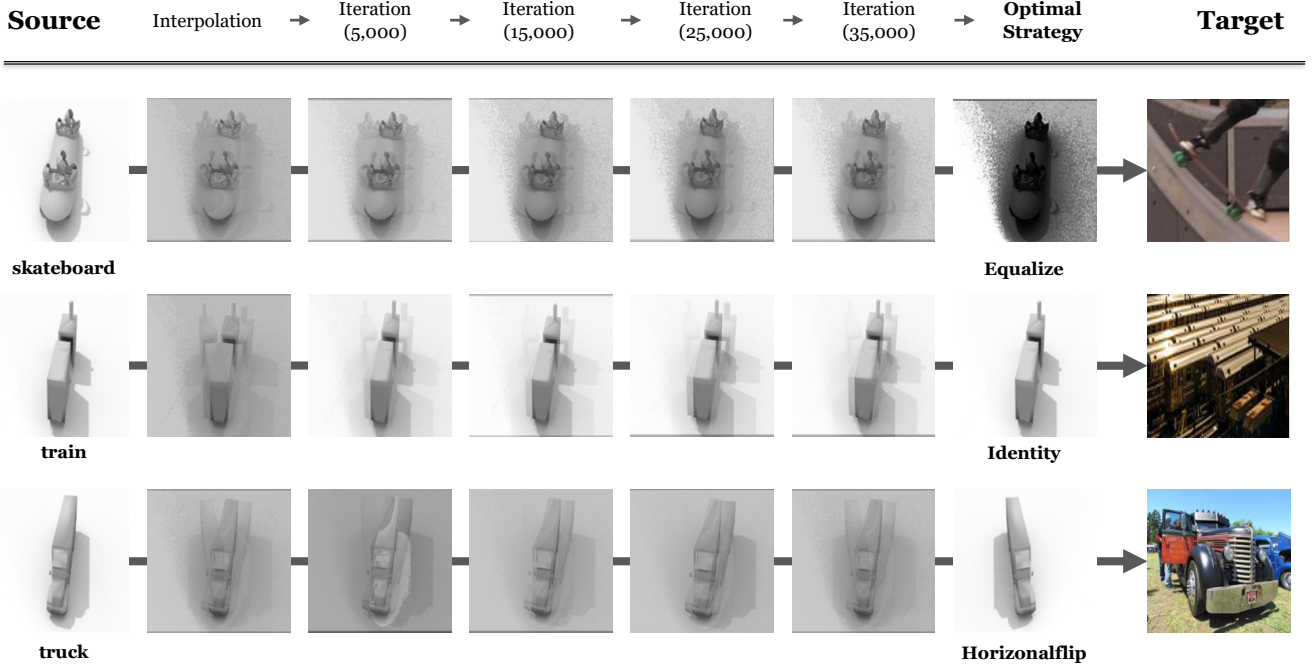


Figure 9. Effectiveness for Hard Categories on VisDA-17. Each category image approaches the target image by employing the optimal strategy.

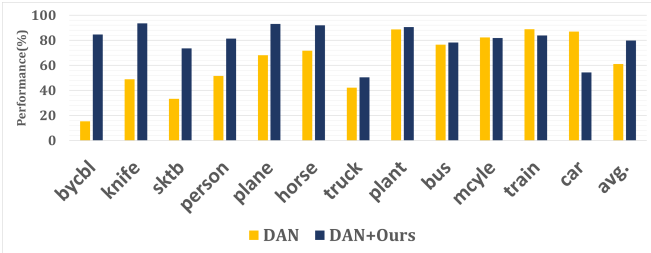


Figure 10. Effectiveness for Hard Categories on VisDA-17, from left to right, we rank in order of most significant performance gain. Our approach resulted in a considerable improvement in performance for the bicycle, knife, and skateboard categories, which are the hard categories in the baseline. [x-axis:categories, y-axis:performance(%)]

6 common augmentations (including ColorJitter, RandomAutocontrast, etc.) and an expanded set of 10 strategies. The results clearly indicate that our chosen set of 6 strategies yields the highest performance (81.5%) among the tested configurations. Utilizing the alternative set of 6 strategies not only resulted in lower accuracy (79.5%) but also slightly increased the search runtime (800s vs. 747s) without changing GPU memory requirements. Expanding the pool to 10 diverse strategies proved even less effective, further decreasing performance (80.1%) while significantly increasing the search time by approximately 32% (to 985s) and requiring slightly more GPU memory. This analysis strongly suggests that simply incorporating more or different augmentation strategies into the search space does not necessarily lead to better adaptation performance and can incur additional computational costs. The superior accuracy achieved by our carefully selected set of 6 strategies, coupled with its relative efficiency in search time, validates our choice as an effective and well-balanced set for the category-specific augmentation search process demonstrated on this task.

C.3 Quality of Selected Augmentation

Investigating the outcome of the search process reveals further insights. The selection of optimal strategies shows consistency across different random seeds indicating a stable learning outcome. While certain generally useful strategies like CenterCrop and HorizontalFlip are frequently selected (Fig. 8), the crucial aspect is the learned dataset-specific adaptivity, visualized via attention weights in Fig. 6. The high weight variance on VisDA-17 (std: 11.65) signifies that CAA effectively learned diverse, category-specific needs crucial for the complex sim-to-real task, correlating strongly with the large average performance gain observed (+7.4%). Conversely, the low weight variance on Office-31 (std: 2.49) reflects learned uniformity appropriate for its context, aligning with the more moderate gain (+3.5%). This confirms that the attention mechanism effectively tailors augmentation specificity to the demands of the dataset.

C.4 Limitations and Future Works

In Fig. 10, we found that accuracy decreased in the train and car categories. We found that domain difference for these categories is complex, with images in the source resembling toys and in the target containing more detail. This complex difference cannot be effectively handled by augmentations alone, suggesting style-transfer methods for future research. Additionally, as shown in Table 3, the performance improvement for state-of-the-art (SOTA) models [8, 46] is relatively small, likely due to their already sophisticated and technologically advanced design. However, it is noteworthy that our approach significantly reduces the performance disparity across different baseline models. Our result indicates that the proposed scheme can serve as a general preprocessing step to reduce the domain gap, thereby enhancing the effectiveness of domain adaptation techniques.




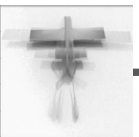
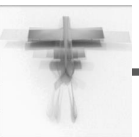
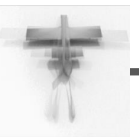









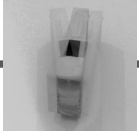







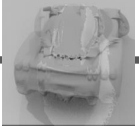
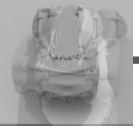
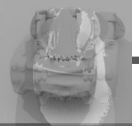
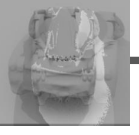
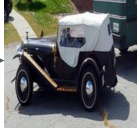


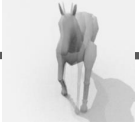
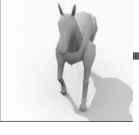
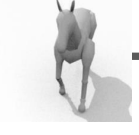

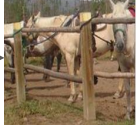


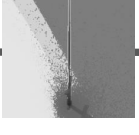
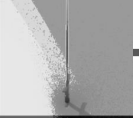
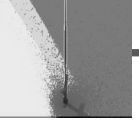
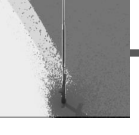




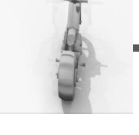

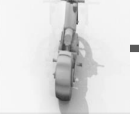




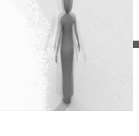
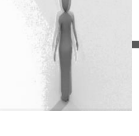
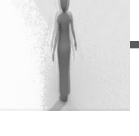

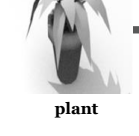
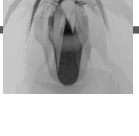
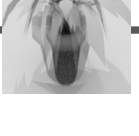
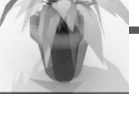

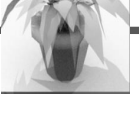
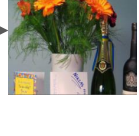
Source	Interpolation	→	Iteration (5,000)	→	Iteration (15,000)	→	Iteration (25,000)	→	Iteration (35,000)	→	Optimal Strategy	Target
 aeroplane											Horizonalflip	
 bicycle											Horizonalflip	
 bus											Posterize	
 car											Solarize	
 horse											Posterize	
 knife											Solarize	
 motorcycle											Centercrop	
 person											Horizonalflip	
 plant											Centercrop	

Figure 11. Effectiveness for Hard Categories on VisDA-17.