Appendix

A. More Related Works

Unsupervised Domain Adaptation. UDA aims to tackle the challenge of generalizing a model trained on a large number of labeled samples from the source domain to the target domain. UDA has been extensively explored for many practical applications, including image classification [39–41], semantic segmentation [42–44], object detection [45–47], and time series forecasting [48–50]. However, UDA methods assume that both the labeled source domain and the unlabeled target domain are available at the same time, which is not available in some scenarios where the source domain data cannot be accessed during training due to privacy-preserving policies.

Source-free Domain Adaptation (White-box Predictors). Compared to the UDA setting, SFDA has higher privacy protection because it does not need to touch any source data. Many SFDA methods have been proposed recently [6–8, 51], which only require access to unlabeled target data and a trained source domain model during training. Most existing research on SFDA tasks is mainly based on self-training [6, 8, 51], class prototypes [8, 52], contrastive learning [7, 53], and generative models [54, 25]. Although the SFDA task has contributed to the mitigation of privacy protection issues to some extent, recent research [12] has found that exposing the details of the white-box predictive model training is quite dangerous due to certain reverse generation techniques like [9, 10].

Setting Comparison and Method Improvement. As shown in Figure 6, the respective processes and the differences among UDA, SFDA, and DABP are presented. Compared with SFDA, DABP provides better data privacy protection with more flexible portability, which only needs to upload target data to the cloud API and then download predictions before training. Figure 7 illustrates the advantages of CVH-TDN and the differences from previous DABP methods. Different from the previous DABP methods [11–14, 31], CVH-TDN leverages sample spatial similarity instead of suppressing targeted sample information, thereby improving model generalization and class discrimination.

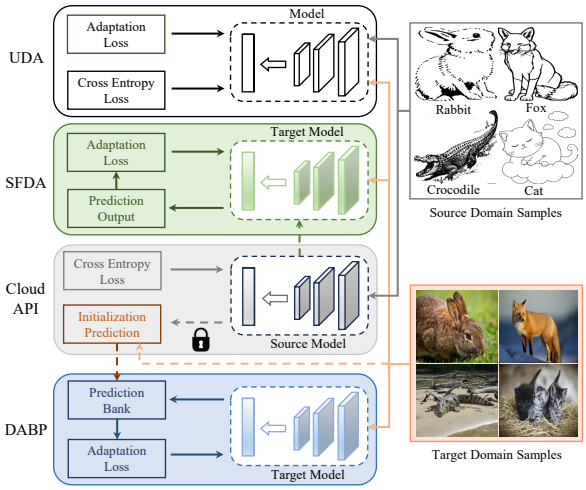


Figure 6: Illustration of the differences among UDA, SFDA, and DABP. The dotted lines in the figure indicate the operations performed by the cloud API with the source model under different settings. SFDA requires the entire source model to be obtained from the cloud API before training. DABP outperforms SFDA in data privacy protection and portability, simply requiring uploading target data to the cloud API and then downloading predictions before training.

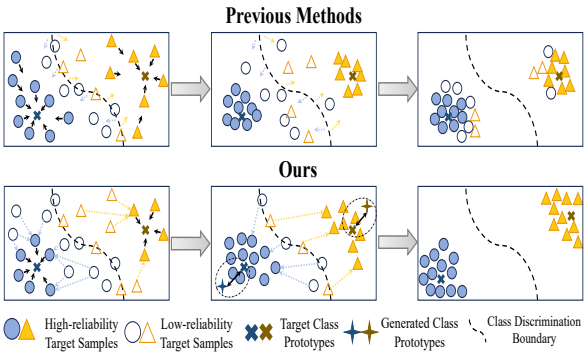


Figure 7: Comparison between existing DABP methods and CVH-TDN. Top: Existing DABP methods focus on learning high-reliability knowledge and force low-reliability sample aggregation. Bottom: CVH-TDN proposes hallucination alignment to investigate the spatial connection between samples and hallucinations and introduces hallucination calibration to explore relationships among spatially similar samples. Our algorithm enhances the capabilities of model generalization and class discrimination.

B. Theoretical Analysis

We provide theoretical justifications grounded in the generalization bound of reasoning to clarify the working mechanism of our algorithm.

Since our algorithm is trained in an unlabeled target domain and generates the controlled samples based on our hallucination generation, we denote $x \sim D_T$ as the real sample distribution of the target domain. Based on the existing theories [55], the error of CVH-TDN can be formulated as a convex combination of the errors of the reliable subdomain $x_R \sim D_R$, the unreliable subdomain $x_{UR} \sim D_{UR}$, and the generated subdomain $x_G \sim D_G$ that represents the generated sample distribution of the reliable subdomain. And denote y_R, y_{UR} and $\widehat{y}_R$, $\widehat{y}_{UR}$ as the true labels and the predicted labels of x_R and x_{UR}, respectively. Denote y_G as the true label and $\widehat{y}_G$ as the predicted label of x_G. y_T and $\widehat{y}_T$ are the true labels and the predicted labels of target domain, $y_G = y_R$, $y_T = y_R + y_{UR}$, and $\widehat{y}_T = \widehat{y}_R + \widehat{y}_{UR}$. Let H denote a hypothesis, which can be expressed as:

$$\epsilon_R\left(H, \widehat{y}_R\right) \leq \epsilon_G\left(H, \widehat{y}_G\right) + d_{n\Delta n}(D_R, D_G) + \varepsilon_G, \tag{19}$$

$$\epsilon_{\widehat{T}}\left(H, \widehat{y}_T\right) = \alpha\epsilon_R\left(H, \widehat{y}_R\right) + (1-\alpha)\epsilon_{UR}\left(H, \widehat{y}_{UR}\right), \tag{20}$$

$$\epsilon_T\left(H, y_T\right) = \alpha\epsilon_R\left(H, y_R\right) + (1-\alpha)\epsilon_{UR}\left(H, y_{UR}\right), \tag{21}$$

where $d_{n\Delta n}(D_R, D_G) = 2\sup_{H,H' \in n} |\mathbb{E}_{x_R \sim D_R}\big[H(x_R) \neq H'(x_R)\big] - \mathbb{E}_{x_G \sim D_G}\big[H(x_G) \neq H'(x_G)\big]|$; $\epsilon_G\left(H, \widehat{y}_G\right)$ is the expected error of the generated sample distribution; $\varepsilon_G = min(\epsilon_R\left(H, \widehat{y}_R\right) + \epsilon_G\left(H, \widehat{y}_G\right))$; $\epsilon_R\left(H, \widehat{y}_R\right)$ is the expected error of the reliable subdomain; α is the trade-off parameter that is controlled by λ in Eq. (7); $\epsilon_{UR}\left(H, \widehat{y}_{UR}\right)$ is the expected error of the unreliable subdomain; $\epsilon_R\left(H, y_R\right)$ and $\epsilon_{UR}\left(H, y_{UR}\right)$ denote the oracle errors of reliable and unreliable samples, respectively.

For $\epsilon_R\left(H, \widehat{y}_R\right) \leq \epsilon_G\left(H, \widehat{y}_G\right) + d_{n\Delta n}(D_R, D_G) + \varepsilon_G$, we analyze each component in detail in this paragraph:

(1) $\epsilon_G\left(H, \widehat{y}_G\right)$ is the expected error of the generated sample distribution, which can be minimized with a cross-entropy loss in the former term of Eq. (15). With the guidance of the black-box predictors, in the initial stage of training, the distribution of the reliable samples selected by the target model in the feature space is relatively close to that of the samples in the source domain. We can obtain good training results for these samples with similar features through the black-box predictors. As the training progresses, the model gradually adapts to the distribution of the target domain. the target model continuously selects more reliable samples. Thus, D_G can better learn the knowledge of these reliable samples through Eq. (15), and align the distribution of the target features through Eq. (12), so as to continuously adapt to the target domain. Therefore, $\epsilon_G\left(H, \widehat{y}_G\right)$ is small in the whole training.

(2) ε_G is the shared error of the ideal joint hypothesis which is considered to be a sufficiently small constant to represent the complexity of the generated sample hypothesis space.

(3) $d_{n\Delta n}(D_R, D_G)$ depends on the expected error of the disagreement between two hypothesis on the reliable subdomain and the generated sample distribution of the reliable subdomain. In the early stages of training, it is easy to find two hypotheses that both H and H' correctly predict the reliable samples. As the training progresses, hallucination alignment explores the spatial relationship between the samples and the hallucinations generated by hallucination generation through bidirectional alignment to encourage clustering of samples with similar features. Hallucination alignment helps to maintain the discrimination and generalization ability of the model in the target domain. Therefore, $\mathbb{E}_{x_G \sim D_G}\big[H(x_G) \neq H'(x_G)\big]$ is always small during adaptation phase. Hallucination calibration uses a hierarchical way to aggregate samples with similar signature features and separates ambiguous samples with common features. Hallucination calibration helps to improve model reasoning ability and prevent reliable sample overfitting. The joint learning of hallucination alignment and hallucination calibration is conducted on reliable samples. As a result, $\mathbb{E}_{x_R \sim D_R}\big[H(x_R) \neq H'(x_R)\big]$ always maintaining a small value during adaptation phase.

Then, during the transition from the early stage to the middle stage of adaptation, we derive an upper bound of how the error $\epsilon_{\widehat{T}}\left(H, \widehat{y}_R\right)$ is close to $\epsilon_T\left(H, y_T\right)$ though hallucination calibration, which is the oracle error with the truth label y_t of the target domain.

Theorem 1. Let H be a hypothesis in class n, we have:

$$
\begin{aligned}
\left|\epsilon_{\widehat{T}}(H,\widehat{y}_T)-\epsilon_T(H,y_T)\right| &= \Big|\alpha\,\epsilon_R(H,\widehat{y}_R)+(1-\alpha)\,\epsilon_{UR}(H,\widehat{y}_{UR}) \\
&\quad -\alpha\,\epsilon_R(H,y_R)-(1-\alpha)\,\epsilon_{UR}(H,y_{UR})\Big| \\
&\le \alpha\left(|\epsilon_R(H,y_R)-\epsilon_{UR}(H,y_{UR})|+|\epsilon_R(H,\widehat{y}_R)-\epsilon_{UR}(H,\widehat{y}_{UR})|\right) \\
&\quad +|\epsilon_{UR}(H,\widehat{y}_{UR})-\epsilon_{UR}(H,y_{UR})| \\
&\le \alpha\left(d_{n\Delta n}(D_R,D_{UR})+\varepsilon+\widehat{\varepsilon}\right)+\rho_{UR}, \qquad (22)
\end{aligned}
$$

where the ideal risk in this hypothesis H is the combinatorial error of the ideal joint hypothesis $\varepsilon = \epsilon_R\left(H^*,y_R\right)+\epsilon_{UR}\left(H^*,y_{UR}\right)$ with $H^* = \arg\min_H \left(\epsilon_R\left(H,y_R\right)+\epsilon_{UR}\left(H,y_{UR}\right)\right)$; $\widehat{\varepsilon} = \epsilon_R\left(H^*,\widehat{y}_R\right)+\epsilon_{UR}\left(H^*,\widehat{y}_{UR}\right)$ is the predicted risk; the distribution discrepancy between reliable and unreliable subdomains is $d_{n\Delta n}(D_R,D_{UR}) = 2\sup_{H,H'\in n}\left|\mathbb{E}_{x_R\sim D_R}\left[H(x_R)\neq H'(x_R)\right]-\mathbb{E}_{x_{UR}\sim D_{UR}}\left[H(x_{UR})\neq H'(x_{UR})\right]\right|$; and ρ_{UR} is the predicted label rate of $\widehat{y}_{UR}$, $\rho_{UR} = \epsilon_{UR}\left(\widehat{y}_{UR},y_{UR}\right)$. Then, we set $\epsilon_1 = \left|\epsilon_R\left(H,y_R\right)-\epsilon_{UR}\left(H,y_{UR}\right)\right|$, $\epsilon_2 = \left|\epsilon_R\left(H,\widehat{y}_R\right)-\epsilon_{UR}\left(H,\widehat{y}_{UR}\right)\right|$, and $\epsilon_3 = \left|\epsilon_{UR}\left(H,\widehat{y}_{UR}\right)-\epsilon_{UR}\left(H,y_{UR}\right)\right|$, making $\left|\epsilon_{\widehat{T}}\left(H,\widehat{y}_T\right)-\epsilon_T\left(H,y_T\right)\right| \le \alpha(\epsilon_1+\epsilon_2)+\epsilon_3$. By applying the triangle inequality for classification errors [56] as presented in **Lemma 1**, we can prove the upper bound of ϵ_1, ϵ_2, and ϵ_3.

Lemma 1. For any hypotheses H_1, H_2, and H_3 in class n,

$$
\epsilon\left(H_1,H_2\right) \le \epsilon\left(H_1,H_3\right)+\epsilon\left(H_2,H_3\right). \qquad (23)
$$

Therefore, for ϵ_1 , we can prove that:

$$
\begin{aligned}
\epsilon_1 &= |\epsilon_R(H,y_R)-\epsilon_{UR}(H,y_{UR})| \\
&\le |\epsilon_R(H,y_R)-\epsilon_R(H,H^*)|+|\epsilon_{UR}(H,H^*)-\epsilon_{UR}(H,y_{UR})|+|\epsilon_R(H,H^*)-\epsilon_{UR}(H,H^*)| \\
&\le \epsilon_R(H^*,y_{UR})+\epsilon_{UR}(H^*,y_{UR})+|\epsilon_R(H,H^*)-\epsilon_{UR}(H,H^*)| \\
&\le \frac{1}{2}d_{n\Delta n}(D_R,D_{UR})+\varepsilon \qquad (24)
\end{aligned}
$$

Similarly, for ϵ_2,

$$
\begin{aligned}
\epsilon_2 &= |\epsilon_R(H,\widehat{y}_R)-\epsilon_{UR}(H,\widehat{y}_{UR})| \\
&\le |\epsilon_R(H,\widehat{y}_R)-\epsilon_R(H,H^*)|+|\epsilon_{UR}(H,H^*)-\epsilon_{UR}(H,\widehat{y}_{UR})|+|\epsilon_R(H,H^*)-\epsilon_{UR}(H,H^*)| \\
&\le \epsilon_R(H^*,\widehat{y}_R)+\epsilon_{UR}(H^*,\widehat{y}_{UR})+|\epsilon_R(H,H^*)-\epsilon_{UR}(H,H^*)| \\
&\le \frac{1}{2}d_{n\Delta n}(D_R,D_{UR})+\epsilon_R(H^*,\widehat{y}_R)+\epsilon_{UR}(H^*,\widehat{y}_{UR}) \\
&\le \frac{1}{2}d_{n\Delta n}(D_R,D_{UR})+\widehat{\varepsilon}. \qquad (25)
\end{aligned}
$$

For ϵ_3,

$$
\epsilon_3 = |\epsilon_{UR}(H,\widehat{y}_{UR})-\epsilon_{UR}(H,y_{UR})| \le \epsilon_{UR}(\widehat{y}_{UR},y_{UR}) = \rho_{UR}. \qquad (26)
$$

By proving ϵ_1, ϵ_2, and ϵ_3, we can derive **Theorem 1**,

$$
\begin{aligned}
\left|\epsilon_{\widehat{T}}(H,\widehat{y}_T)-\epsilon_T(H,y_T)\right| &\le \alpha(\epsilon_1+\epsilon_2)+\epsilon_3 \\
&\le \alpha\left(\frac{1}{2}d_{n\Delta n}(D_R,D_{UR})+\varepsilon\right)+\alpha\left(\frac{1}{2}d_{n\Delta n}(D_R,D_{UR})+\widehat{\varepsilon}\right)+\rho_{UR} \\
&= \alpha\left(d_{n\Delta n}(D_R,D_{UR})+\varepsilon+\widehat{\varepsilon}\right)+\rho_{UR}. \qquad (27)
\end{aligned}
$$

when the reliable subdomain is mostly correct, y_R and $\widehat{y}_R$ are extremely similar with the ideal risk ε that is negligibly small [57], in which case $\rho_R \approx 0$, $\widehat{\varepsilon}$ is bounded by the predicted label rate of unreliable subdomain ρ_{UR}. Empirical results demonstrate that $d_{n\Delta n}(D_R,D_{UR})$ is usually small across the two subdomains, which plays a significant role in tightening the upper bound though hallucination calibration. Therefore, our method can theoretically reduce the expected error of the model on the target domain.

C. Algorithm Details

The whole training process is shown in Algorithm 1. In addition, the experimental code and the main code are available in the Supplementary Materials.

Algorithm 1 CVH-TDN for DABP task.

Input: Target samples $D_t = \{(x_i)\}_{i=1}^{N_t}$, black-box hard predictions P_s, and training model $\mathcal{M}_\theta \in \{F, C\}$;
Parameter: The model parameter θ and the hyperparameters λ, μ, $\tilde{\mu}_s$, ϕ, and r;

Initialize: $\mathcal{M}_\theta$ simple tests on D_t to initialize θ and memory storage M; initialize smooth adaptive storage S with P_s and $\tilde{\mu}$ (determined by $\tilde{\mu}_s$ and μ);
while Adaptation **do**
  Get sample batch B using $\mathcal{M}_\theta$ from D_t;
  Hallucination Generation:
  Control hallucination image x_i^g generated by F evaluating x_i in B using Eqs. (1)-(4).
  Hallucination Alignment:
  Dynamic divide the samples according to their reliability using Eqs. (6)-(8);
  Update M using Eq. (5) to extract the latest information;
  Calculate bidirectional alignment weights w using Eq. (9).
  Hallucination Calibration:
  Update S and $\tilde{\mu}$ using Eq. (13) to mimic the brain cognitive processes;
  Retrieve M to explore the relationship between features and space and calculate the spatial similarity.
  Model Training:
  Optimize target model $\mathcal{M}_\theta$ by minimizing Eq. (18).
end while

Table 6: Accuracies (%) on the *DomainNet* using the ResNet-50 backbone. The rows represent the source domain and the columns represent the adapted target domain.

ResNet	clp	inf	pnt	qdr	rel	skt	Mean	**DINE**	clp	inf	pnt	qdr	rel	skt	Mean
clp	–	16.5	36.0	10.1	52.8	41.8	31.4	clp	–	12.1	29.6	11.1	60.4	37.3	29.4
inf	32.1	–	32.0	2.7	47.4	26.4	28.1	inf	29.5	–	37.6	3.4	53.8	26.5	30.1
pnt	29.6	23.2	–	4.9	36.7	27.8	24.4	pnt	37.3	12.9	–	4.2	60.5	34.7	29.9
qdr	11.2	1.1	1.9	–	4.3	7.7	5.3	qdr	9.4	0.7	3	–	8.3	6.6	5.6
rel	48.2	19.6	47.9	4.3	–	35.6	31.1	rel	45.1	14.4	49.7	5.5	–	35.0	29.9
skt	49.1	13.5	35.5	11.5	47.1	–	31.3	skt	43.3	10.0	39.3	11.6	57.2	–	32.2
Mean	34.0	14.8	30.7	6.7	37.7	27.9	25.3	Mean	32.9	10.0	31.8	7.2	48.0	28.0	26.2
BETA	clp	inf	pnt	qdr	rel	skt	Mean	**CVH-TDN**	clp	inf	pnt	qdr	rel	skt	Mean
clp	–	13.4	41.2	13.0	61.8	41.1	34.1	clp	–	19.0	41.8	11.3	57.2	42.7	34.4
inf	34.9	–	41.6	3.7	56.8	30.7	33.6	inf	37.8	–	39.9	2.9	53.5	32.1	33.2
pnt	47.3	18.4	–	3.2	62.5	41.9	34.7	pnt	45.2	18.9	–	4.0	60.3	37.9	33.3
qdr	11.7	0.9	2.1	–	9.1	8.1	6.4	qdr	15.0	0.9	2.9	–	6.0	11.1	7.2
rel	46.5	15.8	50.9	5.6	–	37.7	31.3	rel	52.7	21.7	52.5	5.7	–	39.2	34.4
skt	47.3	12.3	42.3	14.8	59.9	–	35.3	skt	54.1	17.3	44.3	12.4	54.3	–	36.5
Mean	37.5	12.2	35.6	8.1	50.0	31.9	28.2	Mean	41.0	15.6	36.3	7.3	46.3	32.6	**29.8**

D. Specific Dataset Details

Four standard benchmark datasets are used for evaluating our method and comparison, including *Office-31* [27], *Office-Home* [28], *VisDA-17* [29], and *DomainNet* [30]. *Office-31* is a small-scale benchmark dataset, which contains 4,110 images with 31 categories from 3 domains, Amazon (A), Dslr (D), and Webcam (W). *Office-Home* is a widely used medium-scale benchmark that contains a total of 15.5K images with 65 categories from 4 distinct domains, Real World (R), Clipart (C), Art (A), and Product (P). *VisDA-17* is a challenging large-scale benchmark with 12 categories that include 152k synthetic source domain images and 55k target images of real objects, presenting a greater challenge due to a large synthetic-to-real domain gap. *DomainNet* is the largest domain adaptation benchmark dataset, which consists of about 600K with 345 categories across 6 domains: Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr), Real (rel), and Sketch (skt).

E. Additional Dataset *DomainNet*

For the *DomainNet* dataset, we compare our algorithm with previous SOTA methods [11, 13] under the ResNet-50 backbone. As shown in Table 6, compared with other algorithms, CVH-TDN achieves the highest average accuracy of 29.8% on the *DomainNet*, which is full of on-hard tasks (whose source-only accuracies are below 65%). These results demonstrate that, in the on-hard tasks, our

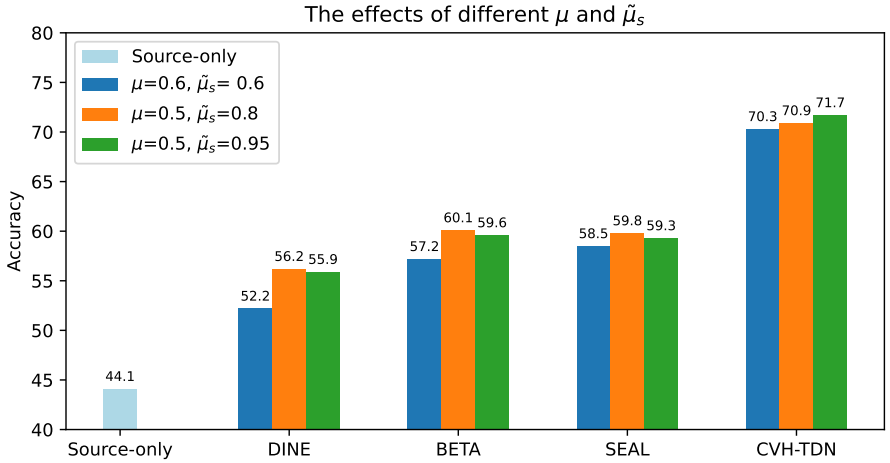


Figure 8: The effect display of different μ and $\tilde{\mu}_s$ in the A→C subtask of the *Office-Home*. Each result is reported when the best accuracy is achieved.

spatial exploration method based on hallucination control is more effective than methods that rely on suppressing specific sample information.

F. More Parameter Analysis and Motivation Demonstration

As shown in Figure 8, we report the performance of different methods [11, 13, 31] under various conditions of μ and $\tilde{\mu}_s$ on the *Office-Home*. In the brain cognitive processes [58], when humans encounter something they have never seen before, their brain cognition is primarily influenced by external information, with a small portion derived from their own understanding. And as their understanding deepens, they can balance the weight of external information and their own understanding, increasingly trusting their own understanding of the matter. In this work, we mimic the brain cognitive process to improve the learning and updating process of Eq. (13), where $\tilde{\mu}_s$ simulates the initial external information and μ simulates human understanding. Additionally, when $\mu = \tilde{\mu}_s = 0.6$, it indicates the use of the previously common adaptive label smoothing update [11] instead of our update strategy. Experimental results show that this improvement significantly enhances the compared DABP methods.

Moreover, we conducted further analytical experiments to demonstrate the advantage of our method for more effectively leveraging both reliable and unreliable samples compared to existing approaches. First, we elaborate on some concepts: the higher the model prediction accuracy of a class, the higher the proportion of reliable samples in that class. Meanwhile, the samples exhibiting features that clearly distinguish their class tend to have a higher probability of being reliable. Because different methods have different strategies for discriminating reliability, we have selected sample instances with consistent initial reliability judgments between BETA [13] and SEAL [31]. Specifically, in Figure 9, the classes corresponding to high-reliability samples are: Bus, Skateboard, and Person; and the classes corresponding to low-reliability samples are: Car, Motorcycle, and Truck.

As shown in Figure 9, the samples corresponding to the classes of Bus and Skateboard were consistently regarded as reliable samples during the training of BETA and SEAL. They focus on learning from these reliability samples, and Figure 9 demonstrates that they effectively concentrate and lock the feature regions of interest on the salient features. Meanwhile, some low-reliability samples (corresponding to the class Car) have also been correctly rectified from the wrong judgments of the black-box predictors through high sample knowledge. However, for some low-reliability samples (corresponding to the class Motorcycle and Truck), the previous methods all failed: BETA didn't lock onto the effective features, and SEAL mistakenly locked onto the features of other classes. In addition, SEAL even misjudged some samples that were judged as high-reliability samples (corresponding to the class Person) as belonging to other classes. These observations are sufficient to show the limitations of using high-reliability samples to constrain the attention-locked area. Moreover, the

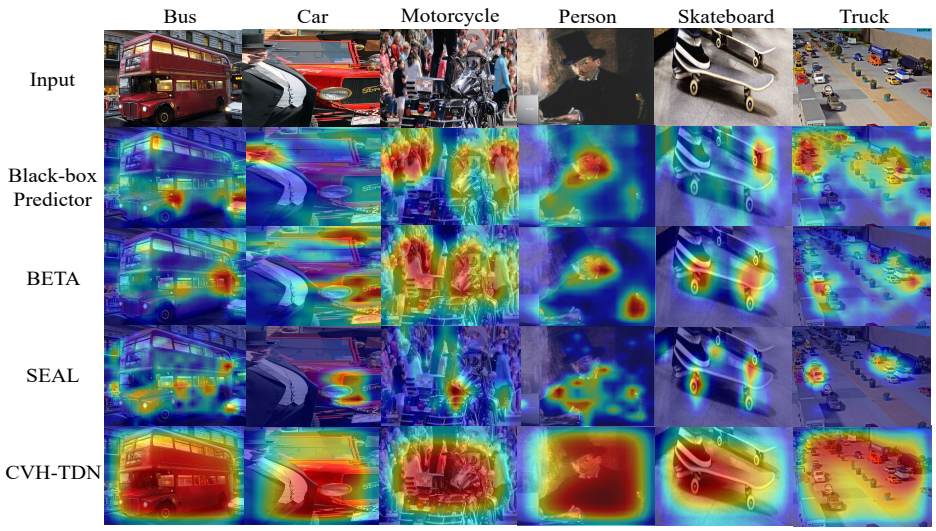
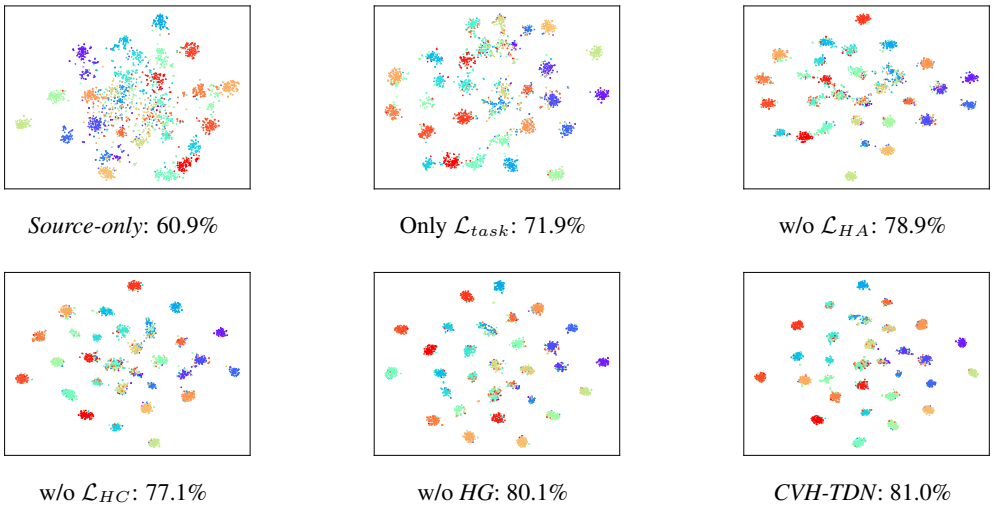


Figure 9: The visualization results of the heat map on the *VisDA-17*. The redder the area, the higher the model's level of attention.



Source-only: 60.9%

Only $\mathcal{L}_{task}$: 71.9%

w/o $\mathcal{L}_{HA}$: 78.9%

w/o $\mathcal{L}_{HC}$: 77.1%

w/o *HG*: 80.1%

CVH-TDN: 81.0%

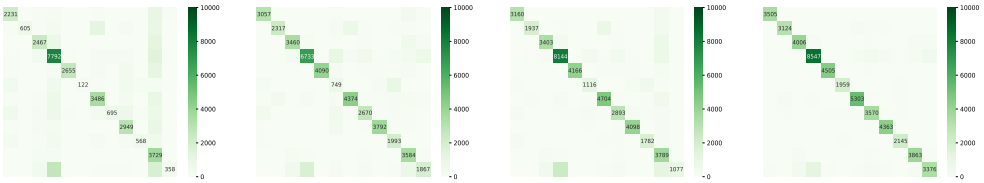
Figure 10: The ablation visualizations of the feature distribution in the W→A subtask of the *Office-31* using t-SNE [18]. Herein, the points represent target samples and the different colors correspond to their true classifications.

feature visualization of Figure 1 can also serve as evidence for the inadequacy of their clustering ability.

In our method, driven by the hallucination alignment, all type-UR samples will become samples of type R in the middle of training. We conceptually blurred the distinction between high-reliability and low-reliability samples during the learning process: samples are no longer constrained by high- or low-reliability ones. As shown in Figures 4 and 9, CVH-TDN enhances the model's reasoning ability by expanding the coverage area of the model's region of interest through hallucination generation and hallucination alignment, resulting in more effective use of all samples.

G. More Ablation Visualization and Experimental Comparison

In Figure 10, we present the t-SNE visualizations [18] of the ablation study on the *Office-31*. Experimental results show that each component of our method improves the discrimination capacity of classes. It is worth noting that, in the small-scale dataset *Office-31*, the impact of removing HG



Source-only: 48.9% *DINE*: 75.6% *BETA*: 85.1% *CVH-TDN*: 90.6%

Figure 11: Classification visualization with Confusion Matrix to compare different methods on the *VisDA-17*. (Zooming in for a clear view)

Table 7: Results of cost comparison on the *VisDA-17* with the ResNet-101 backbone.

Method	**Time (s/epoch)**	**Space (MiB)**	**Accuracy (%)**
DINE	124s	9881MiB	75.6
BETA	1101s	20247MiB	85.1
SEAL	-	Over 24G	89.2
CVH-TDN	205s	10589MiB	90.6

or $\mathcal{L}_{HA}$ is not as significant as shown in the heatmap visualizations [38] of the large-scale dataset *VisDA-17* in Figure 4. This indicates that in datasets with small domain gaps, an efficient clustering algorithm is more important than an algorithm designed to prevent model overfitting, while the opposite holds for datasets with large domain gaps. Furthermore, $\mathcal{L}_{HA}$ focuses on enhancing the model reasoning ability, while $\mathcal{L}_{HC}$ emphasizes improving the class discrimination ability.

For a fair comparison, we set the same running conditions (*e.g.*, batch size = 64, num workers = 4, *etc.*) in the compared works on a machine with an NVIDIA GeForce RTX4090 GPU. Figure 11 shows the Confusion Matrix visualization, in which our method consistently outperforms in discriminating target samples of each class on the *VisDA-17*. This demonstrates that exploring the spatial relationships among samples by controlling hallucinations is more effective in improving class discrimination ability than suppressing specific sample information.

In Table 7, we record the average runtime cost, the maximum GPU space usage, and the best accuracy of each comparison method. When adapting *VisDA-17*, it is worth noting that BETA [13] is divided into two stages that are highly computationally intensive: the first stage is the initialization, which requires initialization of the two models due to their mutually-distilled network structures; the second stage is the two-step process, which requires distillation and fine-tuning for each epoch. SEAL [31] is highly resource-intensive, and its official code cannot complete the adaptation task on *VisDA-17* under the same conditions with 24GB GPU memory. Compared to other methods, CVH-TDN calculates the model area of interest and generates attention-specific masking blocks with minimal cost, which does not require fine-tuning or a computationally expensive generation network.

To ensure robustness, we report the performances across multiple runs with different random seed initializations. As shown in Table 8, we maintained a small average accuracy gap (0.4%) among different seeds, reflecting the stability and superiority of our method.

Table 8: Results under different random seeds on the *Office-Home* with the ResNet-50 backbone.

Seed	**A→C**	**A→P**	**A→R**	**C→A**	**C→P**	**C→R**	**P→A**	**P→C**	**P→R**	**R→A**	**R→C**	**R→P**	**Mean**
2022	71.7	88.4	83.6	70.1	86.7	82.8	70.8	69.2	83.3	74.7	74.2	91.6	78.9
2023	72.2	89.3	83.7	69.9	87.7	82.9	69.6	68.7	83.3	74.9	73.8	91.6	79.0
2024	71.7	88.7	83.3	69.7	86.1	83.3	70.2	68.9	83.7	73.8	72.6	91.3	78.6
2025	72.1	88.6	83.5	70.0	87.9	82.4	70.1	67.2	83.7	75.4	72.1	91.4	78.7

H. Broader Impacts and Limitations

Our work CVH-TDN focuses on the problem of Domain Adaptation of Black-Box Predictors (DABP), which provides better data privacy protection with more flexible portability compared with other DA settings. Inspired by research in pathology and neuroscience, CVH-TDN is specifically designed for the DABP classification task. While its effectiveness has been demonstrated through extensive experiments and its theoretical soundness established, its applicability to other tasks remains an open question. Therefore, we plan to further explore the practical utility of this algorithm in a broader range of task scenarios.