Supplementary Material

Table of Contents

Symbol	Definition
c	A class.
$\mathcal{C}$	The set of all c.
t	A teacher model.
$\mathcal{T}$	The set of all t.
p	A pixel in an image.
$\mathcal{I}$	The set of all p.
x_{src}	Source domain image.
y_{src}	Source domain ground truth.
x_{tgt}	Target domain image.
y_{tgt}	Target domain ground truth.
$\mathcal{D}_{src}$	Source domain dataset.
$\mathcal{D}_{tgt}$	Target domain dataset.
f^{Pixel}	Pixel-wise fusion.
$f^{Channel}$	Channel-wise fusion.
π	A fusion policy.
A	A segmentation map.
A_c	The segmentation map of class c.
A_c^{gt}	The ground truth of A_c.
A_c^{π}	$A_c^{\pi} := \{p \,\vert\, p \in \mathcal{I}, \pi(c) = t, \hat{y}^{(p,c,t)} = 1\}$ is the pseudo label of class c selected according to policy π.
A_o^{π}	$A_o^{\pi} := \bigcup_{\substack{c_1 \neq c_2, \\ c_1, c_2 \in \mathcal{C}}} (A_{c_1}^{\pi} \cap A_{c_2}^{\pi})$ is the overlapped area between A_c^{π}.
$A_{o,c}^{\pi}$	$A_{o,c}^{\pi} := A_o^{\pi} \cap A_c^{\pi}$ is the overlapped area of a class c.
Φ	The function that calculates the IoU of a segmentation map with its ground truth annotation.
$\tilde{\Phi}$	The IoU of the fused results generated using $f^{Channel}$ w.r.t. y_{tgt}.
c_0	An unlabeled symbol.
ε	A class label to be assigned in A_o^{π} under the formulation of $f^{Channel}$.
$\mathcal{C}_p^{\pi}$	$\mathcal{C}_p^{\pi} := \{c \,\vert\, c \in \mathcal{C}; p \in A_c^{\pi}\}$ is the set of class(es) that collects the class label(s) in $p \in A_c^{\pi}$.

Table S1: List of commonly-used symbols.

S1 Background Material

In this section, we walk through the background material of the previous semantic segmentation based unsupervised domain adaptation (UDA) methods. We first offer an overview of the concepts behind the adversarial domain adaptation (ADA) methods in Section S1.1. Then, we review the pseudo labeling strategy in Section S1.2. Some commonly used symbols are summarized in Table S1.

S1.1 The Concepts Behind the Adversarial Domain Adaptation Methods

For the semantic segmentation based UDA problem considered in this paper, the models are granted accesses to the image-label pairs $x_{src} \in \mathbb{R}^{|\mathcal{I}|\times 3}$, $y_{src} \in \{0,1\}^{|\mathcal{I}|\times|\mathcal{C}|}$ from a source domain dataset $\mathcal{D}_{src}$, and the images $x_{tgt} \in \mathbb{R}^{|\mathcal{I}|\times 3}$ from a target domain dataset $\mathcal{D}_{tgt}$, where $\mathcal{I}$ is the set of pixels in an image, and $\mathcal{C}$ is a given set of semantic classes. The goal is to train a model G_θ parameterized by θ, from which the semantic segmentation predictions can best estimate the target domain ground truth y_{tgt}. For example, in AdaptSegNet [1], a generator $G_\theta : \mathbb{R}^{|\mathcal{I}|\times 3} \rightarrow \mathbb{R}^{|\mathcal{I}|\times|\mathcal{C}|}$ is trained against a discriminator $D_\theta : \mathbb{R}^{|\mathcal{I}|\times|\mathcal{C}|} \rightarrow \mathbb{R}^2$ using an adversarial training scheme for minimizing the domain gap. The training objective of D_θ is to distinguish whether the semantic segmentation outputs from G_θ belong to the source domain or not. In contrast, the training objectives of G_θ is to confuse the discriminator D_θ with its predictions. Their loss functions L_G, L_D are defined as follows, respectively:

$$L_G = -\sum_{p\in\mathcal{I}} \log\, D^0(\,\hat{s}_{tgt}^{(p,c)}\,), \tag{S1}$$

$$L_D = -\sum_{p\in\mathcal{I}} (1-z)\,\log\, D^0(\,\hat{s}_{tgt}^{(p,c)}\,) + z\,\log\, D^1(\,\hat{s}_{src}^{(p,c)}\,), \tag{S2}$$

where $c \in \mathcal{C}$ denotes a class, $p \in \mathcal{I}$ denotes a pixel in an image, $\hat{s}_{src}^{(p,c)} = G_\theta(x_{src}) \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{C}|}$ is the softmax output of G_θ for x_{src}, and $\hat{s}_{tgt}^{(p,c)} = G_\theta(x_{tgt}) \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{C}|}$ is the softmax output of G_θ for x_{tgt}. D^0, D^1 denote the first and second output channels of D_θ, which represent the certainty of D_θ on whether the input is drawn from $\mathcal{D}_{tgt}$ or $\mathcal{D}_{src}$, respectively. The binary indicator z is either zero or one to indicate that the samples are drawn from the target or the source domains, respectively.

S1.2 Pseudo Labeling Strategy

Pseudo labeling was pioneered in [2] for improving the performance of classification tasks. For the semantic segmentation based UDA problem, pseudo labeling is a common measure used in the fine-tuning phase by several self-training methods. During the fine-tuning phase, a model is trained to minimize the loss between the pseudo labels ($\hat{y}_{tgt}$) and the predictions of the model on target domain instances (x_{tgt}). These pseudo labels are generated by taking the $\arg\max$ operation over the softmax predictions $\hat{s}_{tgt}$ of the model, which can be formulated as the following equation:

$$\hat{y}_{tgt}^{(p,c)} = \begin{cases} 1, & \text{if } c = \underset{c\in\mathcal{C}}{\arg\max}\,\{\hat{s}_{tgt}^{(p,c)}\} \\ 0, & \text{otherwise} \end{cases}, \tag{S3}$$

where $\hat{s}_{tgt}^{(p,c)} = m_\theta(x_{tgt}) \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{C}|}$ is the softmax output from a segmentation model $m_\theta : \mathbb{R}^{|\mathcal{I}|\times 3} \rightarrow \mathbb{R}^{|\mathcal{I}|\times|\mathcal{C}|}$, which is the model parameterized by θ to be fine-tuned in the target domain. By reducing the cross-entropy loss between the predictions $\hat{s}_{tgt}$ and the one-hot pseudo labels $\hat{y}_{tgt}$, the decision boundaries of the model m_θ are adjusted to lie in low-density regions [2]. This additional fine-tuning stage encourages the model m_θ to produce high-certainty predictions, and enhances its stability in deployment time.

S2 Theoretical Properties of Channel-Wise Fusion

In this section, we provide detailed descriptions of the theoretical properties of the proposed channel-wise fusion function (i.e., $f^{Channel}$). We first define the evaluation metric for semantic segmentation maps, i.e., mIoU, in Section S2.1. Next, we elaborate on the differences between $f^{Channel}$ and f^{Pixel} in Section S2.2. Then, we discuss how the conflict-resolving mechanism can influence the effectiveness of $f^{Channel}$ in Section S2.3. Finally, in Section S2.4, we investigate the properties of the proposed $f^{Channel}$ under the condition that $|A_o^\pi| = 0$, and derive the proofs for **Proposition 1** and **Proposition 2** mentioned in the main manuscript.

S2.1 Mean Intersection Over Union

In this section, we provide the definition and detailed explanation of the commonly used evaluation metric *mIoU* for semantic segmentation maps. Given a segmentation map $A \in 2^{\mathcal{I}} \times \mathcal{C}$ with $|\mathcal{C}|$ different class channels, its mIoU with respect to the ground truth is represented as the following:

$$\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \Phi(A_c), \tag{S4}$$

where $\Phi : 2^{\mathcal{I}} \rightarrow \mathbb{R}$ is the IoU function that calculates the per-class IoU of a segmentation map, and $A_c := \{p \,|\, p \in \mathcal{I} \text{ and the predicted label of } p \text{ is } c\} \in 2^{\mathcal{I}}$ is the segmentation map of a class $c \in \mathcal{C}$. The IoU with respect to the ground truth of a class is calculated by dividing the overlapped regions between the predicted segmentation and the ground truth, by the union of them. Therefore, given the ground truth segmentation map $A_c^{gt} := \{p \,|\, p \in \mathcal{I} \text{ and the ground truth label of } p \text{ is } c\} \in 2^{\mathcal{I}}$ of a class c, $\Phi(A_c)$ can be represented as the following:

$$\Phi(A_c) = \frac{|A_c^{gt} \cap A_c|}{|A_c^{gt} \cup A_c|}. \tag{S5}$$

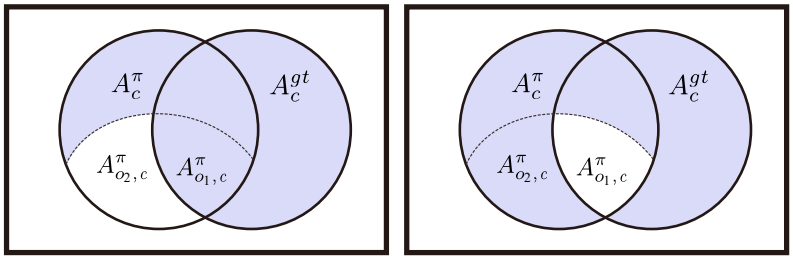
S2.2 Differences between Channel-Wise Fusion and Pixel-Wise Fusion

Channel-wise fusion ($f^{Channel}$) differs from pixel-wise fusion (f^{Pixel}) in that the mIoU's of the fused pseudo labels from $f^{Channel}$ are dependent on two additional factors: (1) the fusion policy π, and (2) the conflict-resolving mechanism that assigns the value of ε. For (1), since the fusion policy $\pi : \mathcal{C} \rightarrow \mathcal{T}$ is a mapping function that assigns each $c \in \mathcal{C}$ to a teacher model $t \in \mathcal{T}$, there may exist $|\mathcal{T}|^{|\mathcal{C}|}$ possible mappings for π. For (2), since the conflict-resolving mechanism assigns a class label $\varepsilon \in \mathcal{C}_p^\pi \cup \{c_0\}$ to each of the pixels in A_o^π given a π, there may exist $|\mathcal{C}_p^\pi \cup \{c_0\}|^{|A_o^\pi|}$ possible fusion outcomes. In order to examine how these factors can impact the mIoU's of the fused pseudo labels generated by $f^{Channel}$, in the following section, we analyze the scenarios when the effectiveness of $f^{Channel}$ is maximized and when it is minimized.

S2.3 Influences of the Conflict-Resolving Mechanism

The conflict-resolving mechanism is a method that assigns a class label for $\varepsilon \in \mathcal{C}_p^\pi \cup \{c_0\}$. Based on the definition of IoU and the formulation of $f^{Channel}$, the IoU's of the fused pseudo labels generated using $f^{Channel}$ w.r.t. y_{tgt} for a given class $c \in \mathcal{C}$ and an arbitrary fusion policy π are maximized when the following conditions are met. An illustration of these conditions is plotted in Fig. S1 (a).

- **Condition a.1:** The conflict-resolving mechanism assigns class label c to the pixels under the area $A_{o_1,c}^\pi := A_{o,c}^\pi \cap A_c^{gt}$, where $A_{o,c}^\pi := A_o^\pi \cap A_c^\pi$ is the overlapped area of class c and the other class(es).



(a) The scenario when the IoU is maximized. (b) The scenario when the IoU is minimized.

Figure S1: An illustration of the scenarios under which the IoU's of the fused pseudo label generated using $f^{Channel}$ can be maximized or minimized.

- **Condition a.2:** The conflict-resolving mechanism assigns class label $c^{'} \in \mathcal{C}_p^{\pi} \cup \{c_0\}$, $c^{'} \neq c$ to the pixels under the area $A_{o_2,c}^{\pi} := A_{o,c}^{\pi} \setminus A_c^{gt}$.

Under such conditions, the IoU w.r.t. the target domain ground truth (y_{tgt}) for class c is given by:

$$\tilde{\Phi}^{*(c,\pi(c))} := \frac{|A_c^{gt} \cap (A_c^{\pi} \setminus A_{o_2,c}^{\pi})|}{|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_2,c}^{\pi})|} = \frac{|A_c^{gt} \cap A_c^{\pi}|}{|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_2,c}^{\pi})|}. \tag{S6}$$

Eq. (S6) suggests that $\tilde{\Phi}^{*(c,\pi(c))} \geq \Phi^{(c,\pi(c))}, \forall c \in \mathcal{C}$, where $\Phi^{(c,\pi(c))} := \Phi(A_c^{\pi}) = \frac{|A_c^{gt} \cap A_c^{\pi}|}{|A_c^{gt} \cup A_c^{\pi}|}$ is the IoU w.r.t. y_{tgt} for class c before applying the conflict resolving mechanism.

In contrast, the IoU's of the fused pseudo labels generated using $f^{Channel}$ w.r.t. y_{tgt} for a given class $c \in \mathcal{C}$ and an arbitrary fusion policy π are minimized when the following conditions are met. An illustration of these conditions is plotted in Fig. S1 (b).

- **Condition b.1:** The conflict-resolving mechanism assigns the class label $c^{'} \in \mathcal{C}_p^{\pi} \cup \{c_0\}$, $c^{'} \neq c$ to pixels under the area $A_{o_1,c}^{\pi}$.
- **Condition b.2:** The conflict-resolving mechanism assigns the class label c to pixels under the area $A_{o_2,c}^{\pi}$.

Under such conditions, the IoU w.r.t. the target domain ground truth (y_{tgt}) for class c is given by:

$$\tilde{\Phi}^{'(c,\pi(c))} := \frac{|A_c^{gt} \cap (A_c^{\pi} \setminus A_{o_1,c}^{\pi})|}{|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_1,c}^{\pi})|}. \tag{S7}$$

Based on the definition in Eq. (S7), the inequality $\tilde{\Phi}^{'(c,\pi(c))} \leq \Phi^{(c,\pi(c))}, \forall c \in \mathcal{C}$ holds.

Proposition S1. $\forall c \in \mathcal{C}, \tilde{\Phi}^{*(c,\pi(c))} = \Phi^{(c,\pi(c))} = \tilde{\Phi}^{'(c,\pi(c))}$ if and only if $|A_o^{\pi}| = 0$.

Proof. ($\Rightarrow$) $\forall c \in \mathcal{C}, \tilde{\Phi}^{*(c,\pi(c))} = \Phi^{(c,\pi(c))} = \tilde{\Phi}^{'(c,\pi(c))}$, the following equality holds:

$$\forall c \in \mathcal{C}, \tilde{\Phi}^{*(c,\pi(c))} = \frac{|A_c^{gt} \cap A_c^{\pi}|}{|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_2,c}^{\pi})|} = \frac{|A_c^{gt} \cap (A_c^{\pi} \setminus A_{o_1,c}^{\pi})|}{|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_1,c}^{\pi})|} = \tilde{\Phi}^{'(c,\pi(c))}$$

$$\Rightarrow \forall c \in \mathcal{C}, |A_c^{gt} \cap A_c^{\pi}|\,|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_1,c}^{\pi})| = |A_c^{gt} \cap (A_c^{\pi} \setminus A_{o_1,c}^{\pi})|\,|A_c^{gt} \cup (A_c^{\pi} \setminus A_{o_2,c}^{\pi})|$$

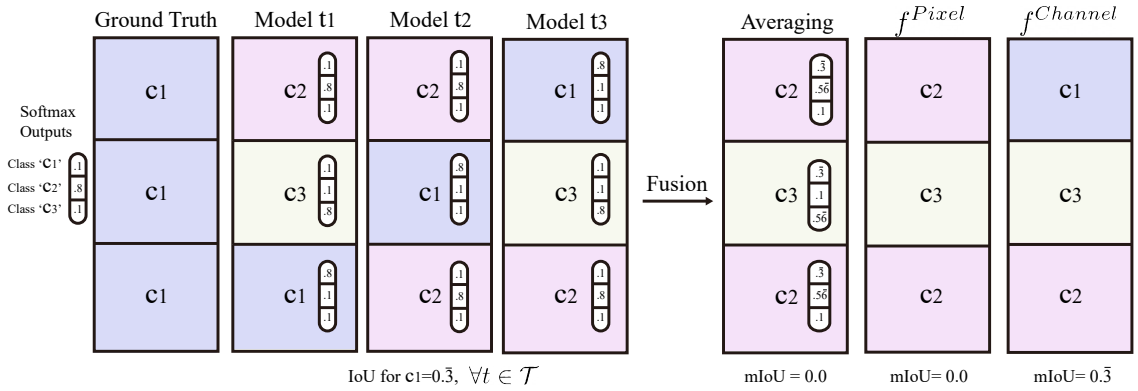


Figure S2: An illustration of the counter example described in **Proposition 2**. Each column in the figure represents a segmentation map with three pixels. The notations 't_1', 't_2', 't_3' represent three different teacher models in $\mathcal{T}$, and 'c_1', 'c_2', 'c_3' represent the class labels in $\mathcal{C}$. The small stripes with three digits indicate the softmax outputs for those classes. In the illustrated example, the IoU's of class 'c_1' for the models 't_1','t_2','t_3' are all greater than a positive constant α (e.g., 0.3). However, after fusion, the mIoU's of the fused results generated using averaging or f^{Pixel} are equal to zero. On the contrary, the mIoU of the fused results generated by $f^{Channel}$ is greater than $\frac{n\alpha}{|\mathcal{C}|}$ (e.g., $\frac{1\times 0.3}{3}$), when a constant fusion policy $\pi(c) = t_3,\ \forall c \in \{c_1, c_2, c_3\}$ is adopted.

Based on the definition of $A^{\pi}_{o_1,c}$ and $A^{\pi}_{o_2,c}$, the above equation can be re-formulated as follows:

$$\forall c \in \mathcal{C}, |A^{gt}_c \cap A^{\pi}_c| \left(|A^{gt}_c \cup A^{\pi}_c| - |A^{\pi}_{o_1,c}|\right) = \left(|A^{gt}_c \cap A^{\pi}_c| - |A^{\pi}_{o_1,c}|\right) \left(|A^{gt}_c \cup A^{\pi}_c| - |A^{\pi}_{o_2,c}|\right)$$

$$\Rightarrow \forall c \in \mathcal{C}, |A^{\pi}_{o_1,c}|\,|A^{\pi}_{o_2,c}| = |A^{gt}_c \cap A^{\pi}_c|\,|A^{\pi}_{o_2,c}| + |A^{\pi}_{o_1,c}| \left(|A^{gt}_c \cup A^{\pi}_c| - |A^{gt}_c \cap A^{\pi}_c|\right)$$

Since $|A^{gt}_c \cap A^{\pi}_c| > |A^{\pi}_{o_1,c}|$ and $(|A^{gt}_c \cup A^{\pi}_c| - |A^{gt}_c \cap A^{\pi}_c|) > 0$, $|A^{\pi}_{o_1,c}| = |A^{\pi}_{o_2,c}| = 0, \forall c \in \mathcal{C}$. This implies $|A^{\pi}_o| = 0$, as $A^{\pi}_o = \bigcup_{c\in\mathcal{C}} A^{\pi}_{o,c} = \bigcup_{c\in\mathcal{C}} (A^{\pi}_{o_1,c} \cup A^{\pi}_{o_2,c})$.

($\Leftarrow$) If $|A^{\pi}_o| = 0$, then $\forall c \in \mathcal{C}, |A^{\pi}_{o_1,c}| = |A^{\pi}_{o_2,c}| = 0$. This implies the following:

$$\forall c \in \mathcal{C}, \tilde{\Phi}^{*(c,\pi(c))} = \frac{|A^{gt}_c \cap A^{\pi}_c|}{|A^{gt}_c \cup (A^{\pi}_c \setminus A^{\pi}_{o_2,c})|} = \frac{|A^{gt}_c \cap A^{\pi}_c|}{|A^{gt}_c \cup A^{\pi}_c|} = \frac{|A^{gt}_c \cap (A^{\pi}_c \setminus A^{\pi}_{o_1,c})|}{|A^{gt}_c \cup (A^{\pi}_c \setminus A^{\pi}_{o_1,c})|} = \tilde{\Phi}'^{(c,\pi(c))}.$$

Therefore, the equality $\tilde{\Phi}^{*(c,\pi(c))} = \Phi^{(c,\pi(c))} = \tilde{\Phi}'^{(c,\pi(c))}, \forall c \in \mathcal{C}$ holds. This also implies that, under such a condition, the IoU $\tilde{\Phi}^{(c,\pi(c))}$ of the fused results achieved by $f^{Channel}$ is solely determined by the fusion policy π. □

S2.4 Proofs for the Propositions in the Main Manuscript

In this section, we provide proofs for the two propositions in Section 4.2.3 of the main manuscript based on the discussions in Section S2.3.

Proposition 1. Consider an arbitrary fusion policy π. Given a constant $\alpha \in (0, 1)$ and classes $c_1, ..., c_n \in \mathcal{C}$. If $\Phi^{(c_i,t)} \geq \alpha, \forall i \in \{1, ..., n\},\ \forall t \in \mathcal{T}$ and $|A^{\pi}_o| = 0$, we have:

$$\text{mIoU} = \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \tilde{\Phi}^{(c,\pi(c))} \geq \frac{n\alpha}{|\mathcal{C}|}. \tag{S8}$$

Proof. As discussed in **Proposition S1**, given an arbitrary fusion policy π, if $|A_o^\pi| = 0$, then the IoU $\tilde{\Phi}$ of the fused results achieved by $f^{Channel}$ is solely determined by π since $\tilde{\Phi}^{*(c,\pi(c))} = \Phi^{(c,\pi(c))} = \tilde{\Phi}'^{(c,\pi(c))}$. Therefore, $\tilde{\Phi}^{(c,\pi(c))} = \Phi^{(c,\pi(c))}$ holds for all $c \in \mathcal{C}$. If $\Phi^{(c_i,t)} \geq \alpha, i \in \{1, ..., n\}$, $\forall t \in \mathcal{T}$, the mIoU of the fused results according to Eq. (S4) and the definition of π can be expressed as the following:

$$\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\tilde{\Phi}^{(c,\pi(c))} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\Phi^{(c,\pi(c))} \geq \frac{1}{|\mathcal{C}|}(n\alpha + \sum_{c\in\mathcal{C}\setminus\{c_1,...,c_n\}}\Phi^{(c,\pi(c))}) \geq \frac{n\alpha}{|\mathcal{C}|}. \tag{S9}$$

As a result, the mIoU achieved by $f^{Channel}$ with any π is ensured to be greater than or equal to $\frac{n\alpha}{|\mathcal{C}|}$.

On the other hand, the mIoU's achieved by either averaging or f^{Pixel} are not guaranteed to be greater than $\frac{n\alpha}{|\mathcal{C}|}$ under the same condition (i.e., $\Phi^{(c_i,t)} \geq \alpha, i \in \{1, ..., n\}$, $\forall t \in \mathcal{T}$). As demonstrated in the counter example in Fig. S2, the IoU's of class 'c_1' for every teacher model 't_1', 't_2', 't_3' are greater than a constant $\alpha \in (0, 1)$. However, the mIoU's of the fused results generated by averaging and f^{Pixel} are below $\frac{n\alpha}{|\mathcal{C}|}$. □

Proposition 2. Consider an optimal fusion policy $\pi^*(c) = \arg\max_{t\in\mathcal{T}}\{\Phi^{(c,t)}\}$. Assume $|A_o^{\pi^*}| = 0$, we have:

$$\text{mIoU} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\tilde{\Phi}^{(c,\pi^*(c))} \geq \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\Phi^{(c,t)}, \forall t \in \mathcal{T}. \tag{S10}$$

Proof. As discussed in **Proposition S1**, given an arbitrary fusion policy π, if $|A_o^\pi| = 0$, then the IoU $\tilde{\Phi}$ of the fused results achieved by $f^{Channel}$ is solely determined by π since $\tilde{\Phi}^{*(c,\pi(c))} = \Phi^{(c,\pi(c))} = \tilde{\Phi}'^{(c,\pi(c))}$. Therefore, $\tilde{\Phi}^{(c,\pi(c))} = \Phi^{(c,\pi(c))}$ holds for all $c \in \mathcal{C}$. Under such a condition, the optimal IoU's for every class can be reached by following a policy $\pi^*(c) = \arg\max_{t\in\mathcal{T}}\{\Phi^{(c,t)}\}$. Such a policy is a greedy one that selects $t \in \mathcal{T}$ to maximize the target domain per-class IoU's $\Phi^{(c,t)}$ w.r.t. y_{tgt} for all $c \in \mathcal{C}$. This suggests that the inequality Eq. (S10) holds for $t \in \mathcal{T}$, since:

$$\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\tilde{\Phi}^{(c,\pi^*(c))} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\Phi^{(c,\pi^*(c))} \geq \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\Phi^{(c,t)}, \forall t \in \mathcal{T}. \tag{S11}$$

□

Hyperparameter Settings	
CBST [3]	
Learning Rate	1×10^{-4}
Weight Decay Factor	5×10^{-3}
Momentum	0.9
Batch Size	2
Epochs	6 with early stopping
Image Crop Size	500×500
Data Augmentation	Random Multi-scale Resizing (0.7∼1.3) and Horizontal Flip
Class Balancing Maximum Weighting	7
MRKLD [4]	
Learning Rate (Phase 1)	1×10^{-3}
Learning Rate (Phase 2)	1×10^{-4}
Weight Decay Factor	5×10^{-4}
Momentum	0.9
Batch Size	32
Epochs	6 with early stopping
Image Crop Size	500×500
Data Augmentation	Random Cropping, Multi-scale Resizing (0.7∼1.3) and Horizontal Flip
R-MRNet [5]	
Learning Rate	1×10^{-4}
Weight Decay Factor	5×10^{-3}
Momentum	0.9
Dropout Rate	0.5
Batch Size	9
Epochs	35 with early stopping
Image Crop Size	512×256
Data Augmentation	Random Cropping, Multi-scale Resizing (0.8∼1.2) and Horizontal Flip
Inference Re-weighting Factor (α)	1
Inference Re-weighting Factor (β)	0.5
DACS [6]	
Learning Rate	2.5×10^{-4}
Weight Decay Factor	5×10^{-4}
Momentum	0.9
Batch Size	2 (For Both the Source and the Target Domain)
Epochs	80 with early stopping
Image Crop Size	512×512
Data Augmentation	Random Cropping
EnD [7] and EnD^2 [8]	
Learning Rate	2.5×10^{-4}
Weight Decay Factor	5×10^{-3}
Momentum	0.9
Batch Size	10
Epochs	35 with early stopping
Image Crop Size	Original Image Size (1024×2048 For Cityscapes)
Data Augmentation	Random Horizontal Flip
Temperature (T)	1
Ours	
Learning Rate	2.5×10^{-4}
Weight Decay Factor	5×10^{-3}
Momentum	0.9
Batch Size	10
Epochs	35 with early stopping
Image Crop Size	Original Image Size (1024×2048 For Cityscapes)
Data Augmentation	Random Horizontal Flip
Kernel Size (κ)	13

Table S2: A summary of the hyperparameters used in the proposed method and the baseline methods.

S3 A Detailed Training Guide for Reproduction

In this section, we provide a detailed training guide for reproducing our work. In Section S3.1, we offer the pseudo code as well as the link to the source code for training the proposed framework. Then, in Section S3.2, we summarize the hyper-parameters for training the proposed framework and the baselines.

S3.1 Pseudo Code and Source Code

The pseudo code for training the proposed framework is presented in Algorithm S1. For more details about the source codes, please refer to the GitHub repository: `https://github.com/Chao-Chen-Hao/Rethinking-EnD-SegUDA`.

Algorithm S1 The Proposed Ensemble-Distillation Method

Input: Ensemble $\mathcal{T}$, and dataset $\mathcal{D}_{tgt}$
Output: Student model m_θ
// Certainty-Aware Policy Selection Strategy
Split $\mathcal{D}_{tgt}$ into $\mathcal{D}_{tgt}^{train}$ and $\mathcal{D}_{tgt}^{val}$
for $t \in \mathcal{T}$ **do**
    Initialize the weights of a student model m_θ.
    Sample x_{tgt} from $\mathcal{D}_{tgt}^{train}$, and generate the fused pseudo labels $\tilde{y}^{(p,c)}$ using $f^{Channel}$ with the constant policy $\forall c \in \mathcal{C}, \pi^{const}(c) = t$.
    Train m_θ with the loss in Eq. (9) in the manuscript.
    Evaluate the average per-class output certainty values $\rho^{(c,t)}$ of m_θ with instances in $\mathcal{D}_{tgt}^{val}$.
end for
// Ensemble-Distillation
Initialize the weights of a student model m_θ.
Sample x_{tgt} from $\mathcal{D}_{tgt}$, and generate the fused pseudo labels $\tilde{y}^{(p,c)}$ using $f^{Channel}$ with π selected based on Eq. (8) in the manuscript.
Train m_θ with the loss in Eq. (9) in the manuscript.

S3.2 Detailed Hyper-Parameter Settings

The detailed hyperparameters for training each of the teacher models in $\mathcal{T}$, EnD [7], EnD^2 [8], and the proposed framework are summarized in Table S2.

S4 Additional Experimental Results

In this section, we report the additional experimental results and provide discussions on them. We first demonstrate the performance of the proposed framework under different backbone settings in Section S4.1. Next, we showcase the reproducibility and the stability of the proposed framework in Section S4.2. Finally, we present some additional visualized results of our framework in Section S4.3.

S4.1 A Comparison of the Backbone of the Student Model

Table S3 compares the performance of our framework using different backbone architectures in the student model. The first, second, and third columns correspond to the backbone architectures, the number of trainable parameters, and the average inference speed (denoted as *IS*), respectively. The column 'Before Distillation' denotes the mIoU of the fused pseudo labels generated by $f^{Channel}$. The column 'After Distillation' refers to the student model's performance after being trained with the fused pseudo labels. As suggested in [9],

Model (Backbone)	Parameters	IS	Before Distillation mIoU (train)	After Distillation mIoU (train)	After Distillation mIoU (val)	Oracle mIoU (val)
Deeplabv2 (ResNet-101)	43.9 M	33.1 ms	56.31	51.76	52.29	62.54
Deeplabv2 (DRN-D-54)	35.6 M	18.8 ms		54.14	55.25	70.25
Deeplabv2 (MobileNetV2)	2.0 M	16.5 ms		48.83	50.98	60.18
Deeplabv3+ (ResNet-101)	59.3 M	35.1 ms		51.71	54.75	67.43
Deeplabv3+ (DRN-D-54)	40.7 M	22.1 ms		55.46	57.98	72.32
Deeplabv3+ (MobileNetV2)	5.8 M	20.9 ms		52.75	54.00	65.25

Table S3: A comparison of the performance of the proposed framework using different backbone architectures (ResNet-101, DRN-D-54, and MobileNetV2) in the student model. The numerical results are evaluated on the GTA5→Cityscapes benchmark. The inference speed is derived based on the average over 500 inferences. '*IS*' denotes the inference speed evaluated on an NVIDIA GTX TITAN V GPU. 'mIoU (train)' refers to the mIoU evaluated on the training set of Cityscapes, which includes 2975 instances. 'mIoU (val)' represents the mIoU evaluated on the validation set of Cityscapes, which includes 500 instances. The column 'Before Distillation' refers to the mIoU of the fused pseudo labels generated by $f^{Channel}$, while 'After Distillation' represents the mIoU of the student's predictions. 'Oracle' refers to the experimental setting that the student is trained directly with y_{tgt} in the training set of Cityscapes and evaluated on the validation set of Cityscapes.

GTA5 → Cityscapes

Model (Backbone)	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
Deeplabv2	92.89	55.61	84.42	41.09	36.53	26.16	37.39	46.14	82.82	44.68	81.96	56.27	32.94	83.27	54.82	46.59	0.00	34.27	50.72	52.07
(ResNet-101)	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±
	0.10	1.15	0.10	0.75	0.45	0.11	0.15	0.10	0.07	0.34	0.30	0.21	0.38	0.10	1.25	0.46	0.00	0.46	0.54	0.24
Deeplabv3+	93.32	59.17	86.20	33.58	37.85	37.45	43.67	52.36	86.34	43.54	86.34	62.81	34.53	86.72	46.07	45.81	0.00	32.00	53.74	53.63
(MobileNetV2)	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±
	0.06	1.05	0.17	1.19	1.21	0.32	0.43	0.79	0.12	1.11	0.45	0.26	0.42	0.86	2.02	1.18	0.00	3.60	3.43	0.45
Deeplabv3+	94.50	61.58	87.91	35.87	39.68	40.74	48.90	55.13	88.20	48.93	88.57	67.06	38.78	89.26	55.00	50.48	0.02	40.03	54.91	57.13
(DRN-D-54)	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±
	0.22	1.55	0.15	0.85	0.89	0.35	0.67	0.44	0.05	0.47	0.39	0.53	1.12	0.20	2.74	1.25	0.06	0.95	1.20	0.28

SYNTHIA → Cityscapes

Model (Backbone)	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
Deeplabv2	87.83	43.42	81.17	18.85	3.69	26.07	27.65	34.05	80.78	-	82.60	54.82	18.78	83.63	-	46.09	-	20.08	49.05	47.41
(ResNet-101)	±	±	±	±	±	±	±	±	±	-	±	±	±	±	-	±	-	±	±	±
	0.04	0.31	0.11	0.37	0.29	0.10	0.86	0.27	0.10	-	0.19	0.33	0.16	0.16	-	1.38	-	0.64	0.21	0.15
Deeplabv3+	88.72	46.91	82.90	18.68	3.89	34.4	29.61	36.93	84.13	-	88.25	60.18	19.35	87.01	-	49.01	-	16.0	52.30	49.89
(MobileNetV2)	±	±	±	±	±	±	±	±	±	-	±	±	±	±	-	±	-	±	±	±
	0.18	0.35	0.16	0.53	0.16	0.24	1.18	0.15	0.17	-	0.13	0.24	0.23	0.24	-	1.67	-	2.66	0.15	0.26
Deeplabv3+	88.64	47.04	83.59	19.43	3.03	36.11	32.15	37.87	84.39	-	87.56	63.35	21.12	87.94	-	52.58	-	21.93	53.76	51.28
(DRN-D-54)	±	±	±	±	±	±	±	±	±	-	±	±	±	±	-	±	-	±	±	±
	0.19	0.36	0.08	0.39	0.31	0.14	2.57	0.29	0.35	-	0.44	0.41	0.58	0.20	-	1.10	-	1.97	0.80	0.13

Table S4: Validation of the stability and the reproducibility for the proposed framework on the GTA5 → Cityscapes and the SYNTHIA → Cityscapes benchmarks. The middle columns and the last column report the per-class IoU's and the mIoU's, respectively. Different rows correspond to different backbone configurations. Each of the numerical results are obtained from five models trained with different initial random seeds without early-stopping.

the distillation process typically requires a larger backbone to fully learn the knowledge from the teachers. However, adopting a larger backbone contradicts the core idea of ensemble-distillation, as the objective is to reduce the model size so that the computational cost at deployment time is affordable. Therefore, in our experiments, a stronger backbone 'Deeplabv3+ (DRN-D-54)' is adopted, as its number of parameters is comparable with 'Deeplabv2 (ResNet-101)' adopted by the members in $\mathcal{T}$, while performing predictions with better effectiveness. Under such a setting, it is observed that the student model is able to effectively approximate the fused pseudo labels, as mIoU's (train) only degrade slightly (0.85%) after distillation.

S4.2 The Reproducibility and the Stability of the Proposed Framework

Table S4 demonstrates the reproducibility and the stability of the proposed ensemble-distillation framework. Each row in the table corresponds to a backbone configuration. Each of the numerical results is obtained from five models trained with different initial random seeds. From Table S4, it is observed that both the per-class IoU's and the mIoU's show only slight fluctuations in terms of their variances, indicating that the proposed method is relatively stable and thus is reproducible.

S4.3 Visualization

Fig. S3 shows a few additional visualized results that qualitatively demonstrate the effectiveness of the proposed framework.

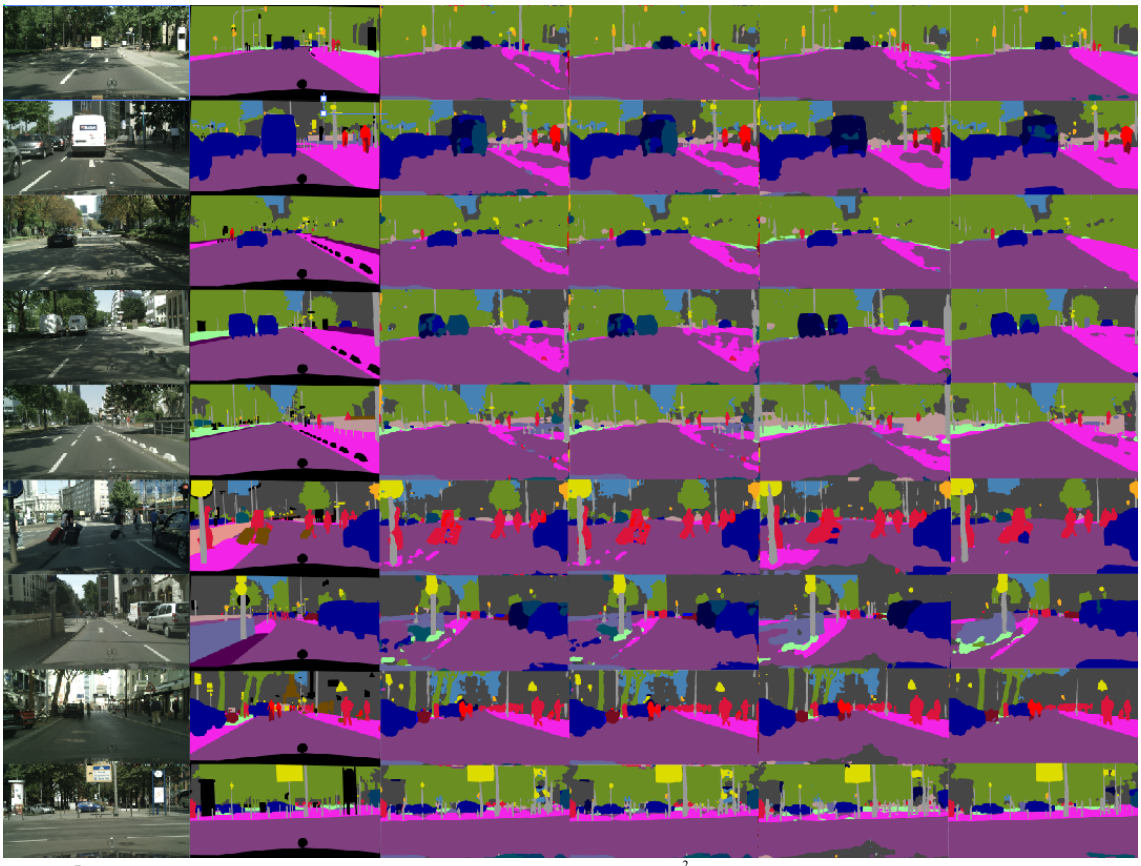


Figure S3: The visualized results evaluated on the validation set of Cityscapes. These figures are presented for qualitatively comparing the student models trained by EnD [7], EnD^2 [8], as well as those trained by the proposed framework with pixel-wise fusion (i.e., Ours (Pixel)) and channel-wise fusion (i.e., Ours (Channel)).

References

[1] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7472–7481, 2018.

[2] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, Int. Conf. on Machine Learning (ICML)*, volume 3, Jun. 2013.

[3] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proc. European Conf. Computer Vision (ECCV)*, pages 289–305, Sep. 2018.

[4] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang. Confidence regularized self-training. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 5982–5991, Oct. 2019.

[5] Z. Zheng and Y. Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *Int. Journal of Computer Vision (IJCV)*, 2020. doi: 10.1007/s11263-020-01395-y.

[6] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson. DACS: Domain adaptation via cross-domain mixed sampling. In *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 1379–1389, Jan. 2021.

[7] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, Mar. 2015.

[8] A. Malinin, B. Mlodozeniec, and M. Gales. Ensemble distribution distillation. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[9] Q. Xie, M.-T. Luong, E. Hovy, and Q. V Le. Self-training with noisy student improves imagenet classification. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698, 2020.