InterLUDE: Interactions between Labeled and Unlabeled Data to Enhance Semi-Supervised Learning

Zhe Huang [*1] **Xiaowei Yu** [*2] **Dajiang Zhu** [2] **Michael C. Hughes** [1]

Abstract

Semi-supervised learning (SSL) seeks to enhance task performance by training on both labeled and unlabeled data. Mainstream SSL image classification methods mostly optimize a loss that additively combines a supervised classification objective with a regularization term derived *solely* from unlabeled data. This formulation often neglects the potential for interaction between labeled and unlabeled images. In this paper, we introduce InterLUDE, a new approach to enhance SSL made of two parts that each benefit from labeled-unlabeled interaction. The first part, embedding fusion, interpolates between labeled and unlabeled embeddings to improve representation learning. The second part is a new loss, grounded in the principle of consistency regularization, that aims to minimize discrepancies in the model's predictions between labeled versus unlabeled inputs. Experiments on standard closed-set SSL benchmarks and a medical SSL task with an uncurated unlabeled set show clear benefits to our approach. On the STL-10 dataset with only 40 labels, InterLUDE achieves **3.2%** error rate, while the best previous method obtains 6.3%.

1. Introduction

Deep neural networks have revolutionized various fields with their strong performance at supervised tasks. However, their effectiveness often hinges on the availability of large labeled datasets. This requirement presents a significant bottleneck, as in many applications labeled data can be scarce and expensive to obtain due to the need for manual annotation by human experts. In contrast, unlabeled data (e.g. images without corresponding labels) may be naturally far more abundant and accessible. This disparity has led to the growing importance of *semi-supervised learning* (SSL) (Zhu, 2005; Van Engelen and Hoos, 2020), which aims to learn from a small labeled set and a much larger unlabeled set. SSL can be applied to many learning tasks, such as image classification (Oliver et al., 2018), object detection (Xu et al., 2021a; Li et al., 2022), and segmentation (Chen et al., 2021; Yang et al., 2023).

This paper focuses on SSL for image classification. Over the years, numerous SSL paradigms have been proposed (Blum and Mitchell, 1998; Min et al., 2020; Kingma et al., 2014; Kumar et al., 2017; Nalisnick et al., 2019; Liu et al., 2019; Iscen et al., 2019). The current prevailing paradigm trains deep neural classifiers to jointly optimize the sum of two losses: a supervised classification objective like cross-entropy evaluated only on labeled data and a regularization term computed solely from the unlabeled data. This paradigm provides a simple yet effective framework for achieving state-of-the-art results (Laine and Aila, 2016; Tarvainen and Valpola, 2017; Sohn et al., 2020; Xu et al., 2021b; Wang et al., 2023; Chen et al., 2023).

Despite these advancements, a key limitation of this deep SSL paradigm is that labeled and unlabeled data are largely handled separately, with images from each data type routed to separate loss functions. We contend that a lack of deeper interaction fails to fully harness the potential of unlabeled data. Others have recently pointed out this *disconnect* between the two data types in deep SSL training (Huang et al., 2023a). Earlier efforts have explored some interactions between data types, such as graph-propagated pseudolabels (Iscen et al., 2019) or augmentations derived from interaction of raw features (Berthelot et al., 2019b;a). However, the effective design of representations and losses that benefit from labeled-unlabeled interaction remains underexplored.

In response to this challenge, we introduce InterLUDE, a novel SSL algorithm that facilitates direct interaction between labeled and unlabeled data in both representations and losses to enhance SSL performance. Our contributions are:

[*] Equal contribution [1] Department of Computer Science, Tufts University, Medford, MA, USA [2] Department of Computer Science, University of Texas at Arlington, Arlington, TX, USA. Correspondence to: Zhe Huang <zhe.huang@tufts.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Code: https://github.com/tufts-ml/InterLUDE/

- First, we introduce *embedding fusion* (Sec. 3.1), a part of the InterLUDE training process that interpolates between labeled and unlabeled embedding vectors to improve representation learning. Ablation studies in Fig. 3 suggest the specific utility of labeled-unlabeled interaction here.

- Second, we introduce *cross-instance delta consistency* loss (Sec. 3.2), a new loss that encourages changes (deltas) to a model's predictions to be similar *across* labeled and unlabeled inputs experiencing the same weak-to-strong change in augmentation.

- Our final contribution is that our experiments cover both standard benchmarks (Sec. 4) as well as a more realistic "open-set" medical task (Sec. 5). Typical recent work in SSL (e.g. Wang et al. (2023)) mostly assumes that labeled and unlabeled data come from the same distribution. However, in the intended real-world applications of SSL, unlabeled data will be collected automatically at scale for convenience, and thus can differ from the labeled set (Oliver et al., 2018). When unlabeled images contain extra classes beyond the labeled set, this is called "open-set" SSL (Yu et al., 2020; Guo et al., 2020). To improve SSL evaluation practices, we evaluate on the open-set Heart2Heart benchmark proposed by Huang et al. (2023b), as well as standard datasets (CIFAR and STL-10).

Ultimately, our experiments encompass **six diverse datasets** and **two architecture families**, including Convolutional Neural Networks (CNNs) with residual connections and Vision Transformers (ViTs). Across scenarios, we find InterLUDE is effective at both closed-set natural image tasks and open-set medical tasks. This latter finding is exciting because InterLUDE was not deliberately designed to handle open-set unlabeled data. We hope this work on InterLUDE opens a new avenue for future SSL research, emphasizing the extraction of training signals from the *interplay* between labeled and unlabeled data, rather than processing each modality in isolation.

2. Background and Related Work

Semi-supervised learning. In semi-supervised image classification problems, we are given a labeled dataset $\mathcal{D}^L$ of image-label pairs (x_l, y) and a much larger unlabeled dataset $\mathcal{D}^U$ containing only images x_u (i.e., $|\mathcal{D}^U| \gg |\mathcal{D}^L|$). Given both data sources, our goal is to train a classifier that maps each image to a probability vector in the C-dimensional simplex Δ^C representing a distribution over C class labels. Comprehensive reviews of SSL can be found in Zhu (2005); Van Engelen and Hoos (2020).

In this paper, we focus on the current dominant paradigm, which trains the weights θ of a deep neural network f to minimize a two-task additive loss

$$\min_\theta \sum_{x_l, y \in \mathcal{D}^L} \ell^L(y, f_\theta(x_l)) + \lambda \sum_{x_u \in \mathcal{D}^U} \ell^U(f_\theta(x_u)) \quad (1)$$

The loss ℓ^L, computed *solely* from the labeled set, is most often the standard cross-entropy loss used in supervised classifiers. The loss ℓ^U is a method-specific loss that is typically based *solely* on the unlabeled set.

Popular approaches under this paradigm include *Pseudo-labeling* that encourages the classifier to assign high probability to confidently-predicted labels for unlabeled images (Lee et al., 2013; Arazo et al., 2020; Cascante-Bonilla et al., 2021) and *Consistency regularization* (Sajjadi et al., 2016; Tarvainen and Valpola, 2017) that enforces consistent model outputs for the same unlabeled image under different transformations. Recent *hybrid* methods (Sohn et al., 2020; Wang et al., 2023) combines several techniques. While representative of successful deep SSL thus far, these approaches all have only indirect or weak interactions between labeled and unlabeled data, such as due to batch normalization (Zhao et al., 2022) or the addition of two separately-computed losses. They lack direct labeled-unlabeled interaction, especially in embedding representations or loss computation. We argue this disconnect between data types prevents fully harnessing the potential of SSL.

Direct labeled-unlabeled interaction in past methods. Several past works do engineer some direct interactions. MixMatch (Berthelot et al., 2019b) and follow-up work ReMixMatch (Berthelot et al., 2019a) both allow labeled-unlabeled interaction within augmentation procedures. Their interpolation strategy, inspired by MixUp (Zhang et al., 2017), can randomly choose to blend any pair of images (including labeled and unlabeled) and the pair's corresponding labels or pseudo-labels. However, the stated goal of MixMatch is to "unify dominant approaches" to SSL; they do not specifically argue for labeled-unlabeled interaction or design their augmentation to promote that type of interaction over other pairings.

Interacting labeled and unlabeled data is a natural concept in another branch of SSL research, namely the graph-based approach (Subramanya and Talukdar, 2022), which treats all images (labeled and unlabeled) as nodes on a graph, connected in some cases by edges that allow interaction. Graph-based methods are typically *transductive*, meaning focused on labeling the provided unlabeled data. In contrast, this work focuses on *inductive* methods, whose goal is to develop a classifier that will generalize to new images. Several recent inductive deep SSL methods have utilized the graph concept, including DeepLP (Iscen et al., 2019), CoMatch (Li et al., 2021), and SimMatchV2 (Zheng et al., 2023). Our experiments comparing InterLUDE to these graph-related methods show the superiority of our approach.

The concurrent work most similar in spirit to ours is Flat-

Match (Huang et al., 2023a). To address the disconnect between labeled and unlabeled data, FlatMatch employs sharpness-aware minimization (SAM) (Foret et al., 2020; Kwon et al., 2021; Liu et al., 2022) to ensure that predictions on unlabeled samples are consistent across both the regular model and a worst-case model, generated by parameter perturbations that maximize empirical risk on labeled data. This approach, however, increase the computation cost due to the additional back-propagation required, a challenge partially alleviated by an approximation strategy they call FlatMatch-e. In contrast, our InterLUDE avoids any additional back-propagation or lossy approximation, achieves better performance (see Tables 1 & 3), and runs much faster ($\sim$8x faster in wall-time comparison, see App. D.1.4).

Data augmentation in SSL. Data augmentation is an integral part of many modern SSL algorithms (Laine and Aila, 2016; Xie et al., 2020). Common techniques include random flip and crop (Krizhevsky et al., 2012), MixUp (Zhang et al., 2017), CutOut (DeVries and Taylor, 2017), AutoAugment (Cubuk et al., 2018), and RandAugment (Cubuk et al., 2020). In SSL, augmentation has primarily been confined to raw image features; perturbations of embeddings remain underexplored. FeatMatch (Kuo et al., 2020) is a rare example of SSL that perturbs embedding vectors (not images), using learned class prototypes. In a similar vein, our embedding fusion directly manipulates the embedding space. Yet, our approach diverges significantly in procedure (avoiding class prototypes) and yields better performance (see Table 1).

3. Method

Problem setup. Denote the overall classifier f with weights θ as a composition of two functions: $f = h \circ g$. Neural network g maps an input image x_i to an embedding representation $z_i \in \mathbb{R}^D$ with D dimensions. Practitioners can set D via architectural choices. Neural network h then maps the D-dimensional embedding to a probability vector p_i in the C-dimensional simplex Δ^C over the C classes. In practice, given an overall network f we set g as all layers from input to the second-to-last layer, following evidence from Yu et al. (2023a) suggesting that deeper layers generally yield better performance.

We train f using stochastic gradient descent, where each minibatch contains B images from the labeled set and $\mu \cdot B$ images from the unlabeled set. We fix $\mu = 7$ following past work (Sohn et al., 2020; Wang et al., 2023). Thus, each batch contains in total $R = (1 + \mu)B$ distinct images.

We apply both *weak* (random flip and crop) and *strong* augmentations (RandAugment as in FixMatch (Sohn et al., 2020)) to each image. Let Ω and Σ denote *sets* of possible weak and strong augmentations. We can draw specific realizations ω from Ω or σ from Σ. Each realization defines a specific transformation (e.g. rotate by 15 degrees). After augmentation, each batch contains $Q = 2R$ samples.

Overview of InterLUDE. Our proposed InterLUDE algorithm comprises two main components. First, an *embedding fusion* strategy that improves representation quality. Second, a new loss term called *cross-instance delta consistency* that makes changes (deltas) to a model's predictions similar *across* the labeled and unlabeled inputs under the same augmentation change (ω vs σ). These two components each promote labeled-unlabeled (LU) interaction and ultimately work in synergy to improve model performance. Fig. 1 illustrates the InterLUDE framework. Alg. 1 provides pseudocode. Details on each component are introduced below.

3.1. Embedding Fusion: *Better embedding via interaction*

Emerging evidence suggests that **proactively perturbing** the embedding space can yield significant performance gains across various learning tasks, such as supervised image classification (Verma et al., 2019), domain adaptation (Yu et al., 2023b;a) and natural language processing (Pereira et al., 2021; Khan et al., 2023). Our work explores deliberate perturbation of embeddings via labeled-unlabeled interaction to improve semi-supervised learning.

Given the batch of Q augmented labeled and unlabeled samples, we map via the network g to an array of embeddings $Z \in \mathbb{R}^{Q \times D}$. Key to our embedding fusion strategy is a specific **interdigitated layout** arrangement of Z (illustrated in Fig. 2). By construction, each set of $1 + \mu$ adjacent rows in Z (a labeled embedding followed by μ unlabeled embeddings) is created using the same specific augmentation ω or σ (see Alg. 1).

General Framework for Embedding Fusion. For any Z, we imagine a deterministic fusion transformation parameterized by a matrix $A \in \mathbb{R}^{Q \times Q}$:

$$Z' \leftarrow (I + A)Z \tag{2}$$

where I is the identity matrix. Each row of Z' is a linear combination of the rows of Z, thus *fusing* together original embeddings. Predicted class probabilities for each image can then be obtained via feeding each row of Z' into the classification head h. This construction is inspired by Yu et al. (2023a), who pursue embedding fusion for the supervised (not semi-supervised) case.

We impose three constraints on matrix A (Yu et al., 2023a):

$$\begin{aligned} &(i)\ \text{rank}(I + A) = Q; \\ &(ii)\ [I + A]_{ii} > [I + A]_{ij}\,, \text{for all } i \neq j; \\ &(iii)\ \left\|[I + A]_i\right\|_1 = 1, \text{for all } i. \end{aligned} \tag{3}$$

The first constraint, the full rank requirement, ensures that none of the original embeddings are eliminated during the

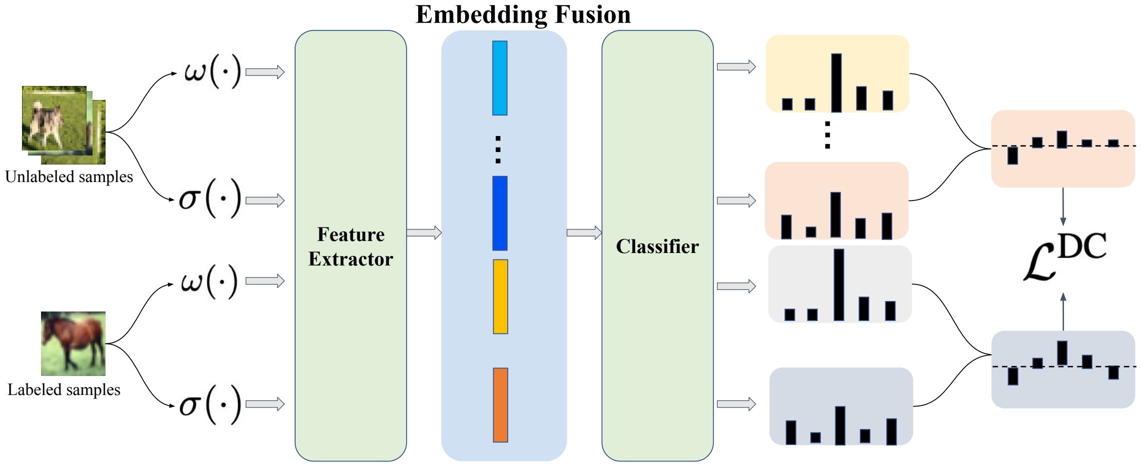

Figure 1. InterLUDE Framework. The labeled and unlabeled data are individually subjected to both weak and strong augmentations, followed by the backbone extracting feature embeddings. Embedding fusion then perturbs the embeddings as shown in Fig 2. The delta consistency loss is employed to regulate model behavior on labeled and unlabeled samples under the same augmentation changes.

fusion. *The second constraint* ensures that each new fused embedding z'_i is predominantly informed by the original z_i. Without this requirement, it is hard to ensure accurate prediction of each true label. *The final constraint* ensures that the overall magnitude of each new fused embeddings z'_i matches the magnitude of the original embedding z_i. This helps fix an otherwise unconstrained degree of freedom.

Concrete design of A: Circular Shift. Many possible A matrices could satisfy the above desiderata. Here, we adopt a concrete construction of A called *circular shift* (Yu et al., 2023a). Under this construction, each z'_i is perturbed slightly by its immediate next neighbor z_{i+1} at index $i+1$. Because our interdigitated batch layout interleaves labeled and unlabeled samples, this guarantees every labeled embedding is perturbed by an unlabeled embedding.

Formally, let $\alpha \in (0, 0.5)$ be the fusion strength. To create new embeddings Z' via Eq. (2), we set $A = \alpha * U - \alpha * I$, where $U_{i,j} = \delta_{i+1,j}$ with $\delta_{i+1,j}$ representing the Kronecker delta indicator (Frankel, 2011) and using wrap-around (aka "circular") indexing. This yields the overall $Q \times Q$ perturbation matrix:

$$I + A = \begin{bmatrix} 1-\alpha & \alpha & 0 & 0 & 0 \\ 0 & 1-\alpha & \alpha & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ \alpha & 0 & 0 & 0 & 1-\alpha \end{bmatrix} \quad (4)$$

This construction satisfies all three desiderata in Eq. (3). Each row of the new Z' can be written simply as $z'_i = (1-\alpha)z_i + \alpha z_{i+1}$, an additive perturbation toward one other image's embedding. We will show later in Sec. 4 & 5 that this embedding fusion improves accuracy across many closed-set and open-set SSL tasks.

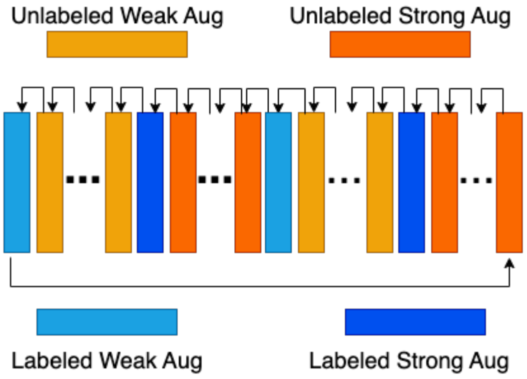

Figure 2. Illustration of Embedding Fusion. Showing the flattened embeddings with batch size 2 for clearer visualization. Each embedding is slightly perturbated by its immediate next neighbor.

Comparison with Manifold MixUp. Our embedding fusion shares similarities with Manifold MixUp (Verma et al., 2019), a regularization method for supervised image classification that extends the MixUp idea to the embedding space by linearly interpolating embedding vectors z_i (not images x_i) and corresponding labels y_i. However, there are several key differences from our approach. First, the intended use case: Manifold MixUp is proposed for supervised learning, not semi-supervised learning. Second, Manifold MixUp interpolates both embeddings and their corresponding labels, while we only perturb the embeddings since there are no labels for the unlabeled set. Additional procedures could be added to our InterLUDE to guess pseudo-labels for unlabeled samples. However, this would require more forward-passes at each batch, adding substantial complexity and runtime cost. Third and most importantly, our interdigitated batch layout (Fig. 2) deliberately enforces that each labeled embedding is always blended with an unlabeled embedding, improving diversity and robustness. Careful ablations (see Fig 3) show this deliberate labeled-unlabeled interaction leads to better classifiers than other layouts with less interaction. Empirically, we find that our embedding fusion is beneficial for semi-supervised learning. We leave improving its theoretical understanding for future work.

3.2. Delta Consistency: *Consistent deltas across L & U*

We now introduce the second key contribution of our InterLUDE: a new loss we call *delta consistency loss*. Delta consistency loss makes use of the widely-used consistency regularization principle (Tarvainen and Valpola, 2017; Sohn et al., 2020). However, as noted in prior work (Huang et al., 2023a), most consistency-regularization losses only encourage "instance-wise" consistency, that is, consistency for *each individual instance* under different transformations. In contrast, our delta consistency loss is designed to make deltas (changes) in class-prediction behavior consistent *across* labeled (L) and unlabeled (U) instances.

Recall that our approach (see Alg. 1) applies the same weak and strong augmentation to adjacent sets of $1 + \mu$ images (1 labeled image and μ unlabeled images). The key idea is that, given a specific pair of weak and strong augmentations ω and σ, the **change in predicted probabilities due to swapping weak with strong augmentation should be similar across labeled and unlabeled cases**. Intuitively, if a specific augmentation swap causes a labeled image (e.g., "cat") to look more like "dog" and less like "cat", it ought to produce a similar change for the unlabeled images of similar class.

Define index $i \in \{1, \dots B\}$ to uniquely identify a distinct labeled image in the current batch. Let ω_i, σ_i be the specific weak-strong pair of augmentations to be swapped for index i. Denote p_i^w (respectively p_i^s) as the class probabilities produced by classifier h given the weak embedding $z_i^{\prime,w}$ (strong embedding $z_i^{\prime,s}$). Let index $m \in \{1, 2, \dots \mu\}$ denote an offset from i, so subscript i, m identifies an unlabeled example that, in our interdigitated layout, received the same pair of augmentations as labeled image i. Let vector $q_{i,m}^w$ (respectively $q_{i,m}^s$) denote the class probabilities produced by classifier h given the weak embedding $z_{i,m}^{\prime,w}$ (strong embedding $z_{i,m}^{\prime,s}$) of that unlabeled image. Below we present two versions of the delta-consistency loss.

Average version. Define vectors Δ_i^L, Δ_i^U that represent the differences of predicted probabilities for the labeled and unlabeled case across weak and strong augmentations:

$$\Delta_i^L = p_i^w - p_i^s, \quad \Delta_i^U = \frac{1}{\mu} \sum_{m=1}^{\mu} (q_{i,m}^w - q_{i,m}^s) \tag{5}$$

The goal of the average version of the delta consistency loss is to encourage vector Δ_i^U, the average change in predictions across unlabeled instances experiencing the same augmentations as image i, to mimic the change vector Δ_i^L directly observed on labeled instance i. Concretely, we minimize the average squared Euclidean distances between vectors Δ_i^L and Δ_i^U:

$$\mathcal{L}_{avg}^{\text{DC}} = \frac{1}{B} \sum_{i=1}^{B} \left\| \Delta_i^L - \Delta_i^U \right\|_2^2 \tag{6}$$

Many possible distance functions could be tried; we picked this distance because it is simple, symmetric, and bounded. The bounded property may help prevent fluctuation in training dynamics (Berthelot et al., 2019b).

Class-dependent version. We also explored a class-dependent version of our delta-consistency loss. That is, we only enforce that unlabeled deltas are similar to the labeled deltas when the unlabeled image is predicted to belong to the same class. This reflects the intuition that an augmentation swap is likely to lead to predictions being more 'dog' and less 'cat' consistently across two images in a pair, only if the two images in question share the same class label.

To operationalize this idea, let $\hat{y}_{i,m}$ indicate the most likely class predicted for unlabeled image at index i, m: $\hat{y}_{i,m} = \text{argmax}_{c \in \{1, \dots C\}} q_{i,m}^w[c]$. The class-dependent delta term for unlabeled images sharing the same augmentation as labeled image i becomes

$$\Delta_{i,cls}^U = \text{mean}\left(\{q_{i,m}^w - q_{i,m}^s : m \in \{1, \dots, \mu\}, \hat{y}_{i,m} = y_i\}\right)$$

If none of the μ unlabeled images related to i share its class label, we set $\Delta_{i,cls}^U = \text{NaN}$. We then obtain the overall DC loss by averaging over all indices $i \in \{1, \dots, B\}$ where a valid vector $\Delta_{i,cls}^U$ is available:

$$\mathcal{L}_{cls}^{\text{DC}} = \text{mean}\left(\{\left\|\Delta_i^L - \Delta_{i,cls}^U\right\|_2^2 : \Delta_{i,cls}^U \neq \text{NaN}\}\right) \tag{7}$$

All results in the main paper use the simpler average version. Results for the class-dependent formulation can be found in App. E. Each version has its own advantages and disadvantages (more discussion in Sec. 7), yet their overall classification performance is comparable.

Ultimately, to better leverage unlabeled data for semi-supervised learning, it is essential to extract information that is not present in or easily derived from the labeled set alone (Van Engelen and Hoos, 2020). Our delta-consistency loss achieves this by enforcing consistency across labeled and unlabeled predictions under an augmentation change.

3.3. InterLUDE overall training objective

Overall, our proposed InterLUDE algorithm combines the classic two-term SSL objective from Eq. (1) with our new

Algorithm 1 InterLUDE and InterLUDE+

```
Input: Labeled set $\mathcal{D}^L$, Unlabeled set $\mathcal{D}^U$
Output: Trained weights $\theta$ for classifier $f$
Procedure
for iter $t \in 1, 2, \dots T$ do
    $\{x_i, y_i\}_{i=1}^{B} \leftarrow \text{DrawBatch}(\mathcal{D}^L, B)$
    $\{\bar{x}_j\}_{j=1}^{\mu B} \leftarrow \text{DrawBatch}(\mathcal{D}^U, \mu B)$
    for example $i \in 1, 2, \dots B$ do
        $\omega_i, \sigma_i \leftarrow \text{DrawAugParams}(\Omega, \Sigma)$
        $x_i^w, x_i^s, \{\bar{x}_j^w\}_{j=1}^{\mu}, \{\bar{x}_i^s\}_{j=1}^{\mu} \leftarrow \text{GetAug}(\omega_i, \sigma_i)$
    end for
    $X \leftarrow \text{InterDigitate}(\{x_i^w, x_i^s\}_{i=1}^{B}, \{\bar{x}_j^w, \bar{x}_j^s\}_{j=1}^{\mu B})$
    $Z \leftarrow g(X; \theta)$    // calc embeddings
    $Z' \leftarrow \text{CircShiftFusion}(Z, \alpha)$
    $\{p_i^w, p_i^s\}_{i=1}^{B}, \{q_j^w, q_j^s\}_{j=1}^{\mu B} \leftarrow h(Z'; \theta)$
    $\mathcal{L}_{avg}^{\text{DC}} \leftarrow$ Eq. (6)    // delta-consistency loss
    $\mathcal{L}^{\text{L}} \leftarrow$ Eq. (9)    // supervised cross-ent. loss
    $\mathcal{L}^{\text{U}} \leftarrow$ Eq. (10)    // instance-wise unlabeled loss
    if InterLUDE then
        $\mathcal{L} \leftarrow \mathcal{L}^{\text{L}} + \lambda_{\text{u}} \mathcal{L}^{\text{U}} + \lambda_{\text{DC}} \mathcal{L}_{avg}^{\text{DC}}$
    else if InterLUDE+ then
        $\mathcal{L} \leftarrow$ Eq. (11)
    end if
    $\theta = \theta - \epsilon \nabla_\theta \mathcal{L}$    // update weights via SGD
end for
return $\theta$
```

delta consistency loss. Our final loss function is:

$$\mathcal{L} = \mathcal{L}^L + \lambda_u \mathcal{L}^U + \lambda_{\text{DC}} \mathcal{L}_{avg}^{\text{DC}} \tag{8}$$

where $\lambda_u > 0$ and $\lambda_{DC} > 0$ control the relative weight of different terms. We use the following common choices for labeled loss $\mathcal{L}^L$ and unlabeled loss $\mathcal{L}^U$

$$\mathcal{L}^L = -\frac{1}{B} \sum_{i=1}^{B} \log p_i^w[y_i] \tag{9}$$

$$\mathcal{L}^U = \frac{1}{\mu B} \sum_{j=1}^{\mu B} \mathbb{I}(\max(q_j^w) > \tau) H(\hat{q}_j^w, q_j^s) \tag{10}$$

$\mathcal{L}^L$ is a standard cross-entropy loss based on labeled samples only; $\mathcal{L}^U$ is the instance-wise consistency loss exemplified by FixMatch (Sohn et al., 2020) that has been a utilized by many methods (Huang et al., 2023a; Wang et al., 2023; Chen et al., 2023) and $\hat{q}_j^w = argmax(q_j^w)$. We fix confidence threshold τ to 0.95, following Sohn et al.

3.4. InterLUDE+

Recently, the Self-Adaptive Threshold (SAT) and Self-Adaptive Fairness (SAF) ideas were introduced by Wang et al. (2023), with possible applicability to other SSL algorithms. For instance, FlatMatch (Huang et al., 2023a) has utilized these techniques. For fair comparison to such past work, we integrate these techniques into InterLUDE and name the enhanced version InterLUDE+. We provide a brief overview of SAT and SAF here; more details in App. C.

SAT is a technique for adjusting the pseudo-labels threshold over iterations based on both dataset-specific and class-specific criteria. SAF is a regularization technique that encourages diverse predictions on unlabeled data via an additional loss term $\mathcal{L}^{\text{SAF}}$. Thus, the complete loss function of InterLUDE+ (including SAT and SAF) is:

$$\mathcal{L} = \mathcal{L}^L + \lambda_u \mathcal{L}^U + \lambda_{\text{DC}} \mathcal{L}_{avg}^{\text{DC}} + \lambda_{\text{SAF}} \mathcal{L}^{\text{SAF}} \tag{11}$$

with a self-adaptive threshold replacing the fixed τ in L^U.

4. Experiments on Classic SSL Benchmarks

We evaluate InterLUDE on common closed-set SSL benchmarks with both Convolutional Neural Network (CNN) and Vision Transformer (ViT, Dosovitskiy et al. (2020)). Our comparisons encompass a **comprehensive list of 22 SSL algorithms**, including recent state-of-the-art methods.

Datasets. We use CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009) and STL-10 (Coates et al., 2011). Detailed descriptions can be found in App. A. We skip SVHN (Netzer et al., 2011) as recent SSL advances have pushed performance to near saturation (e.g., <2% error rate with only 4 labels per class (Wang et al., 2023)).

CNN experiment setting. Using CNNs as backbones, we conduct experiments across various labeled set sizes: 40, 250, 4000 for CIFAR-10; 400, 2500, 10000 for CIFAR-100 and 250, 1000 for STL-10. Following standard protocol (Wang et al., 2023; Zheng et al., 2023), we use Wide ResNet-28-2/28-8 (Zagoruyko and Komodakis, 2016) for CIFAR-10/100 and ResNet-37-2 (He et al., 2016) for STL10. We use SGD optimizer (Nesterov momentum 0.9) and a cosine learning rate schedule (Loshchilov and Hutter, 2016) $\eta = \eta_0 \cos\left(\frac{7\pi k}{16K}\right)$, where $\eta_0 = 0.03$ is the initial learning rate, $K = 2^{20}$ is the total training steps and k is the current training step. Inference is conducted using the exponential moving average of the model with a momentum of 0.999. We set the labeled batch size to 64 and the unlabeled batch size to 448 (i.e., $\mu = 7$). For hyperparameters unique to InterLUDE, we set λ_{DC} to 1.0 and fusion strength α to 0.1.

ViT experiment setting. Using ViTs as backbones, we conduct experiments across labeled set sizes of 10, 40, 250 for CIFAR-10; 200, 400, 2500 for CIFAR-100; and 10, 40, and 100 for STL-10. Following prior works (Wang et al., 2022; Li et al., 2023), we use pretrained ViT-small for CIFAR-10/100 and pretrained ViT-base for STL-10. The pretrained ViT-small and ViT-base are provided by USB (Wang et al., 2022). We use AdamW optimizer with learning rate $5e^{-4}$ for CIFAR-10/100 and $1e^{-4}$ for STL-10. The total training steps are set to 204,800. The batch size is set to 8 with $\mu = 7$. For hyperparameters unique to InterLUDE, we set λ_{DC} to 0.1 and fusion strength α to 0.1.

Results. Results are shown in Table 1 for CNNs and Table 2 for pretrained ViTs. It is worth noting that previous observers suggest it is hard for an algorithm to achieve top performance across all settings (Zheng et al., 2023).

On CNN backbones (Table 1), our methods achieve the best results in many scenarios and remain competitive in others. Notably, on CIFAR-10 with 40 labels, InterLUDE records a low error rate of 4.51%, which InterLUDE+ further pushes to 4.46%. Among methods using only 40 labeled images on CIFAR-10, InterLUDE and InterLUDE+ not only outperform all SSL alternatives but are the **only methods to surpass the fully-supervised result** of training the same network to minimize cross entropy on all 50000 training examples in the labeled set (which represents an ideal training setting for a classifier). Moreover, InterLUDE is consistently more stable than competitors like FlatMatch (see much smaller standard deviations in Tab. 1 and Tab. 3).

On ViT backbones (Tab 2), InterLUDE+ delivers the best performance in 5 out of 6 settings on CIFAR-10 and CIFAR-100, with InterLUDE close behind. On STL-10 with only 40 labels, InterLUDE (InterLUDE+) achieves an impressive low **error rate of 3.14% (4.59%), far better than the next best method of 6.25%** and remarkably better than all competitors even when they have 2.5x more labeled images (last column). Across Tables 1-2, ViT backbones exhibit superior performance, in line with observations in Wang et al. (2022). InterLUDE and InterLUDE+ perform similarly on CNNs, but with ViTs InterLUDE+ appears better on CIFAR data at low train set sizes.

5. Experiments on Heart2Heart Benchmark

Here we evaluate our method on a open-set medical imaging benchmark proposed in Huang et al. (2023b). We compare to strong baselines from Table 1 as well as two recent SOTA open-set SSL algorithms: OpenMatch (Saito et al., 2021) and Fix-A-Step (Huang et al., 2023b).

The Heart2Heart benchmark looks at three fully-deidentified medical image datasets of heart ultrasound images, collected independently by different research groups. It adopts a clinically crucial **view classification task**: *Given an ultrasound image of the heart, identify the specific anatomical view depicted*. Here we briefly describe the data (Full details in App. A). The data for training SSL methods is **TMED-2** (Huang et al., 2021; 2022), collected at one site in Boston, MA, U.S.A., with ~1700 labeled images in TMED-2's predefined train and validation set of four view classes: PLAX, PSAX, A4C and A2C. TMED-2 has a large unlabeled set of 353,500 images from 5486 routine scans that are truly *uncurated*, containing out-of-distribution classes, no known true labels, and modest feature distribution shift. The benchmark assesses classifiers on the TMED-2 test set (~2100 images) as well as the UK-based **Unity** dataset by Howard et al. (2021) (7231 images total of PLAX, A2C, and A4C classes) and France-based **CAMUS** dataset by Leclerc et al. (2019) (2000 images total; A2C and A4C classes only).

Heart2Heart poses two key questions: 1. **Can we train a view classifier from limited labeled data?** (train SSL on TMED-2 then test on TMED-2; Table 3 column 1) **2. Can classifiers trained on images from one hospital transfer to hospitals in other countries?** (train SSL on TMED-2 then test on Unity and CAMUS; Table 3 columns 2-3).

Experiment setting. Our experiments adhere to the exact settings in Huang et al. (2023b). We train a separate model for each of the three predefined splits in TMED-2. We use Wide ResNet-28-2 and inherit all common hyperparameters directly from Huang et al. For FreeMatch and FlatMatch, we additionally search λ_{SAF} in [0.01, 0.05, 0.1]. For InterLUDE and InterLUDE+, we search λ_{DC} in [0.1, 1.0]. We select hyperparameters based on the validation set and report test set performance at the maximum validation checkpoint. Full hyperparameter details are in App. B.2.

Results. Table 3 presents our results, with Columns 2 and 3 essentially assessing the *zero-shot cross-hospital generalization* capabilities of each method. Across all columns, InterLUDE and InterLUDE+ consistently show competitive performance. On TMED2, InterLUDE+ emerges as the top performer, closely followed by InterLUDE. In the zero-shot generalization to CAMUS and Unity, InterLUDE continues to lead, followed by FlexMatch and FreeMatch. All algorithms show a significant performance drop and increased variance on the CAMUS dataset, a phenomenon also observed in the original paper (Huang et al., 2023b). Nevertheless, InterLUDE still outperforms all other baselines, demonstrating strong generalization. FlatMatch on the other hand, substantially underperforms (more than 2x error rate of InterLUDE, more discussion in App F).

6. Ablations and Sensitivity Analysis

Ablation of components. InterLUDE has two novel components: the cross-instance delta consistency loss and the embedding fusion. Table 4 assesses their individual impacts using both CNN and ViT backbones. We find that both components are effective. On CNN, removing embedding fusion (delta consistency loss) leads to a 0.3% (0.5%) drop in performance On ViT, removing embedding fusion (delta consistency loss) leads to performance drops of about 0.5% (0.6%). Note that in this CIFAR-10 40-label scenario, most

Table 1. Error rate (%, lower is better) with CNNs. Following Zheng et al. (2023), error rate and standard deviation are reported based on three runs. All experiments follow the same settings. Rows marked * are implemented by us using the author's code. Results of other methods are directly copied from SimMatchV2 (Zheng et al., 2023) and the original papers ("–" means the result is not available). The best results are highlighted in bold and the second-best underlined. We did not run DeepLP on STL-10 due to computation constraints.

dataset	CIFAR10			CIFAR100			STL-10	
num. labeled images	40	250	4000	400	2500	10000	250	1000
Supervised	77.18±1.32	56.24±3.41	16.10±0.32	89.60±0.43	58.33±1.41	36.83±0.21	55.07±1.83	35.42±0.48
Manifold MixUp* (Verma et al., 2019)	73.51±1.81	55.44±2.06	17.40 ±1.11	87.98±0.55	59.45±0.85	33.23±0.21	54.99±1.22	29.17±2.00
Pseudo-Labeling (Lee et al., 2013)	75.95±1.86	51.12±2.91	15.32±0.35	88.18±0.89	55.37±0.48	36.58±0.12	51.90±1.87	30.77±0.04
II-Model (Laine and Aila, 2016)	76.35 ± 1.69	48.73±1.07	13.63±0.07	87.67±0.79	56.40±0.69	36.73±0.05	52.20±2.11	31.34±0.64
DeepLP* (Iscen et al., 2019)	72.65 ± 2.04	35.80±1.43	10.58±0.29	86.11±0.61	62.87±0.42	43.03 ±1.36	–	–
Mean Teacher (Tarvainen and Valpola, 2017)	72.42±2.10	37.56±4.90	8.29±0.10	79.96±0.53	44.37±0.60	31.39±0.11	49.30±2.09	27.92±1.65
VAT (Miyato et al., 2018)	78.58±2.78	28.87±3.62	10.90±0.16	83.60±4.21	46.20±0.80	32.14±0.31	57.78±1.47	40.98±0.96
MixMatch (Berthelot et al., 2019b)	35.18±3.87	13.00±0.80	6.55±0.05	64.91±3.34	39.29±0.13	27.74±0.27	32.05±1.16	20.17±0.67
ReMixMatch (Berthelot et al., 2019a)	8.13±0.58	6.34±0.22	4.65±0.09	41.60±1.48	25.72±0.07	20.04±0.13	11.14±0.52	6.44±0.15
FeatMatch (Kuo et al., 2020)	–	7.50±0.64	4.91±0.18	–	–	–	–	–
UDA (Xie et al., 2020)	10.01±3.34	5.23±0.08	4.36±0.09	45.48±0.37	27.51±0.28	23.12±0.45	10.11±1.15	6.23±0.28
FixMatch (Sohn et al., 2020)	12.66±4.49	4.95±0.10	4.26±0.01	45.38±2.07	27.71±0.42	22.06±0.10	8.64±0.84	5.82±0.06
Dash (Xu et al., 2021b)	9.29±3.28	5.16±0.28	4.36±0.10	47.49±1.05	27.47±0.38	21.89±0.16	10.50±1.37	6.30±0.49
MPL (Pham et al., 2021)	6.62±0.91	5.76±0.24	4.55±0.04	46.26±1.84	27.71±0.19	21.74±0.09	–	6.66±0.00
CoMatch (Li et al., 2021)	6.51±1.18	5.35±0.14	4.27±0.12	53.41±2.36	29.78±0.11	22.11±0.22	7.63±0.94	5.71±0.08
FlexMatch (Zhang et al., 2021)	5.29±0.29	4.97±0.07	4.24±0.06	40.73±1.44	26.17±0.18	21.75±0.15	9.85±1.35	6.08±0.34
AdaMatch (Berthelot et al., 2021)	5.09±0.21	5.13±0.05	4.36±0.05	37.08±1.35	26.66±0.33	21.99±0.15	8.59±0.43	6.01±0.02
SimMatch (Zheng et al., 2022)	5.38±0.01	5.36±0.08	4.41±0.07	39.32±0.72	26.21±0.37	21.50±0.11	8.27±0.40	5.74±0.31
FreeMatch (Wang et al., 2023)	4.90±0.04	4.88±0.18	4.10±0.02	37.98±0.42	26.47±0.20	21.68±0.03	–	5.63±0.15
SoftMatch (Chen et al., 2023)	4.91±0.12	4.82±0.09	4.04±0.02	37.10±0.77	26.66±0.25	22.03±0.03	–	5.73±0.24
SimMatchV2 (Zheng et al., 2023)	4.90±0.16	5.04±0.09	4.33±0.16	36.68±0.86	26.66±0.38	21.37±0.20	7.54±0.81	5.65±0.26
FixMatch (w/SAA) (Gui et al., 2023)	5.24±0.99	4.79±0.07	3.91±0.07	45.71±0.73	26.82±0.21	21.29±0.20	–	–
InstanT (Li et al., 2023)	5.17±0.10	5.28±0.02	4.43±0.01	46.06±1.80	32.91±0.00	27.70±0.40	–	–
FlatMatch (Huang et al., 2023a)	5.58±2.36	**4.22±1.14**	3.61±0.49	38.76±1.62	25.38±0.85	**19.01±0.43**	–	**4.82±1.21**
FlatMatch-e (Huang et al., 2023a)	5.63±1.87	4.53±1.85	**3.57±0.50**	38.98±1.53	25.62±0.88	19.78±0.89	–	5.03±1.06
InterLUDE (ours)	4.51±0.01	4.63±0.11	3.96±0.07	**35.32±1.06**	**25.20±0.22**	20.77±0.19	7.05±0.12	5.01±0.04
InterLUDE+ (ours)	**4.46±0.11**	4.46±0.09	3.88±0.05	36.99±0.62	25.27±0.17	20.49 ±0.15	**6.99 ±0.42**	4.92±0.05
Fully-Supervised (all labeled train img.)		4.57±0.06			18.96±0.06		–	

algorithms compete for improvements of less than 0.5% over their predecessors, underscoring the effectiveness of both components. More ablations in App. Table D.6 further confirm the effectiveness of the two components.

Ablation of layout: high vs. low L-U interaction. Our interdigitated layout (Fig. 2) is specifically designed to enhance labeled-unlabeled interaction. In Fig. 3, we contrast this layout with a low-interaction alternative that adjacently places all $2B$ labeled samples together then all $2\mu B$ unlabeled samples together, changing line 8 in Alg. 1 to

$$X \leftarrow \textsc{Stack}(\{\{x_i^w\}_{i=1}^B, \{x_i^s\}_{i=1}^B, \{\bar{x}_j^w\}_{j=1}^{\mu * B}, \{\bar{x}_j^s\}_{j=1}^{\mu * B}\}).$$

Applying circular-shift fusion, the two layouts have exactly the same number of within-batch interactions, but our interdigitated layout has far more labeled-unlabeled interactions. Fig. 3 shows that our high-LU-interaction interdigitated layout is crucial in the this low label regime. More results can be found in App. Fig. D.4.

Sensitivity to Delta Consistency Loss Coefficient. We examined the impact of varying λ_{DC}, a unique hyperparameter in our method, from 0.1 to 10.0 (see details in App. Fig. D.1). We see stable performance across a wide spectrum of values. However, in extremely low label settings, very high λ_{DC} values result in diminished performance, a phenomenon similar to that observed with the unlabeled loss coefficient in other studies (Tarvainen and Valpola, 2017).

Sensitivity to Embedding Fusion strength. Another hyperparameter introduced by our method is the embedding fusion strength α. To evaluate its impact, we vary α from 0.1 to 0.4 (detailed in App. Fig. D.2). We see that α around 0.1 to 0.2 is generally a good value. Excessively high α values, nearing the 0.5 upper limit for α to satisfy Eq. (3), result in a significant drop in performance.

Sensitivity to Augmentation Strategies. Our data augmentations follow the weak-strong construction in FixMatch (Sohn et al., 2020), with weak including standard flip and crop, and strong using RandAugment (Cubuk et al., 2020). RandAugment has two hyperparameters: N (the number of augmentations applied sequentially to an input image) and M (the magnitude of these transformations). We find that when N is too large (applying multiple augmentations sequentially), performance decreases, while M has a smaller impact. Overall, our method is not overly sensitive to these hyperparameters (see details in App. D.1.2).

7. Discussion

We introduced InterLUDE, an SSL method that fosters direct interactions between labeled and unlabeled data via embedding fusion and a new delta-consistency loss term. We

Table 2. Error rate (%, lower is better) with ViT backbone. The error rate and 95% confidence interval are reported based on three random seeds (Li et al., 2023). Rows marked * are implemented by us using the author's code. Other results directly copied from Li et al. (2023). The best results are highlighted in bold and the second-best underlined.

dataset	CIFAR10			CIFAR100			STL10		
num. labeled images	10	40	250	200	400	2500	10	40	100
PL (Lee et al., 2013)	62.35±3.1	11.79±5.3	4.58±0.4	36.66±2.0	26.87±0.9	15.72±0.1	69.26±6.7	42.84±4.2	26.56±1.5
MT (Tarvainen and Valpola, 2017)	35.43±4.9	12.85±2.5	4.75±0.5	40.50± 0.8	30.58±0.9	17.09±0.4	57.28±7.8	33.20±3.4	22.29±1.8
MixMatch (Berthelot et al., 2019b)	34.96±2.6	2.84±0.9	2.05±0.1	39.64± 1.3	27.74±0.1	16.16±0.3	89.32±1.1	72.42±16.2	38.15±11.3
VAT (Miyato et al., 2018)	39.93±6.3	6.67±6.6	2.33±0.2	34.11±1.8	24.67±0.4	16.58±0.4	79.43±4.4	34.82±7.0	19.06±1.0
UDA (Xie et al., 2020)	21.24±3.6	2.08±0.2	2.04±0.1	34.51±1.6	24.15±0.6	16.19±0.2	51.63±4.3	20.33±4.9	10.60±1.0
FixMatch (Sohn et al., 2020)	33.50±15.1	2.56±0.9	2.05±0.1	34.71±1.4	24.48±0.1	16.02±0.1	59.87±3.4	22.28±4.4	11.59±1.6
FlexMatch (Zhang et al., 2021)	29.46±9.6	2.22±0.3	2.12±0.2	36.24±0.9	25.99±0.5	16.28±0.2	39.37±12.9	21.83±3.7	10.46±1.3
Dash (Xu et al., 2021b)	25.65±4.5	3.37±2.0	2.10±0.3	36.67±0.4	25.46±0.2	15.99±0.2	58.94±4.4	21.97±3.9	10.44±2.0
AdaMatch (Berthelot et al., 2021)	14.85±20.4	2.06±0.1	2.08±0.1	26.39±0.1	21.41±0.4	15.51±0.1	31.83±7.7	16.50±4.2	10.75±1.5
FlatMatch* (Huang et al., 2023a)	**11.95±7.3**	3.17±0.3	2.33±0.1	26.56±1.0	21.80±1.0	13.83±0.3	**23.90±8.9**	6.25±0.3	4.74±0.2
InstanT (Li et al., 2023)	12.68±10.2	2.07±0.1	1.92±0.1	25.83±0.3	21.20±0.4	15.72±0.5	30.61±7.4	14.91±2.8	10.65±1.9
InterLUDE (ours)	31.90±4.1	1.78±0.1	1.55±0.1	35.66±1.9	21.19±0.2	13.39±0.1	27.49±6.6	**3.14±0.2**	**2.66±0.1**
InterLUDE+ (ours)	12.29±7.3	**1.55±0.1**	**1.49±0.1**	**23.60±1.2**	**16.32±0.3**	**12.93±0.2**	25.83±9.9	4.56±0.9	3.23±0.3

Table 3. Heart2Heart Benchmark. Error rate and standard deviation are reported based on three pre-defined data splits. We re-implemented FlexMatch, FreeMatch and FlatMatch using the author's codes. Other results are copied from Huang et al. (2023b). Best results highlighted in bold and the second-best underlined.

	TMED2	CAMUS	UNITY
Supervised	7.35±0.79	30.23±7.77	9.55±1.68
Pi-model	7.41±0.63	38.52±0.96	9.90±1.31
VAT	6.43±0.11	32.40±8.11	9.21±2.16
FixMatch	5.66±0.68	22.12±9.08	7.63±1.42
FlexMatch	3.56±0.32	17.58±6.53	5.20±0.52
FreeMatch	3.52±0.25	16.67±6.04	5.45±0.18
FlatMatch	7.84±0.48	28.93±10.38	10.05±1.15
FixAStep	4.79±0.49	18.78±10.20	6.04±1.07
OpenMatch	5.88±0.63	22.07±5.89	7.17±1.89
InterLUDE (ours)	3.45±0.39	**13.75±6.22**	**4.86±0.48**
InterLUDE+ (ours)	**3.25±0.17**	18.12±8.37	5.53±0.85

Table 4. Ablations to isolate the effect of InterLUDE's two key components. Showing error rate (%) on CIFAR-10 with 40 labels.

	CNN	ViT
InterLUDE	**4.51±0.01**	**1.78±0.06**
w/o Embedding Fusion	4.82±0.12	2.31±0.92
w/o $\mathcal{L}_{avg}^{\mathrm{DC}}$	5.00±0.35	2.37±0.86

show these changes lead to superior performance across multiple SSL benchmarks. Of particular note is InterLUDE's success with the open-set Heart2Heart benchmark and recent ViT architectures.

Limitations. Improved understanding of precisely *why* embedding fusion works well is needed. Although we have many experimental results suggesting the practical value of embedding fusion, more theoretical work is needed to understand the underlying mechanisms in a more principled manner. Li (2022) analyse the impact of injecting noise to a learning system from an information theory perspective. They show that certain perturbations in image space help reduce task complexity. Extending such analysis to the embedding space might be an interesting future work.

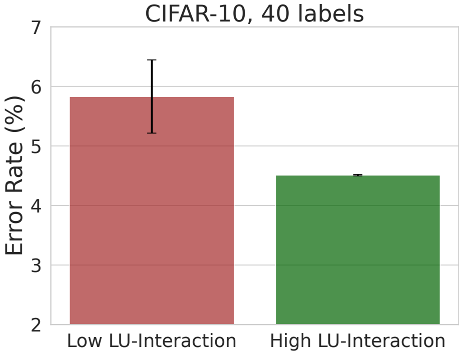

Figure 3. Ablation on Different Batch Layout

Improved understanding of delta-consistency loss would also be beneficial. While we primarily present the average version of the delta-consistency loss in the main paper for its simplicity, this formulation lacks fine-grained class dependencies. Conversely, the class-dependent version considers class labels when deciding how changes in predictions should be consistent across labeled and unlabeled examples. However, this approach relies on the quality of pseudo-labels, which can be error-prone during early training.

Furthermore, in the extreme case where the regular instance-wise consistency loss leads the model to make exactly the same predictions on the weak and strong augmentations for both the unlabeled and labeled data, our proposed delta-consistency loss would approach zero. If this occurs early in training, the impact of the delta-consistency loss would be limited. However, our empirical evaluations across several datasets have not yet observed such an extremity.

Outlook. We hope this work suggests the untapped potential of using labeled-unlabeled interactions to improve SSL classifiers. We also encourage future SSL evaluation to consider medically-inspired benchmarks like Huang et al. (2023b)'s Heart2Heart alongside traditional datasets.

Impact Statement

This paper presents work whose goal is to advance the field of semi-supervised learning. We hope our work improves practitioners' ability to train accurate classifiers from limited labeled data, especially in medical applications where acquiring labeled data is prohibitively costly. Even though the Heart2Heart cross-hospital generalization task we examine is fully deidentified and open-access by the respective dataset creators, we implore future work pursuing it to remember the real human patients the data represents and take proper care. A key concern in translating SSL-trained classifiers to practice is fairness to different subpopulations, which cannot yet be assessed with the available data in that benchmark.

References

E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019b.

D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *arXiv preprint arXiv:2106.04732*, 2021.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6912–6920, 2021.

H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023.

X. Chen, Y. Yuan, G. Zeng, and J. Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with main. *arXiv preprint arXiv:1708.04552*, 2017.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

T. Frankel. *The geometry of physics: an introduction*. Cambridge university press, Cambridge, 2011.

G. Gui, Z. Zhao, L. Qi, L. Zhou, L. Wang, and Y. Shi. Enhancing sample utilization through sample adaptive augmentation in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15880–15889, 2023.

L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*, pages 3897–3906. PMLR, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

J. P. Howard, C. C. Stowell, G. D. Cole, K. Ananthan, C. D. Demetrescu, K. Pearce, R. Rajani, J. Sehmi, K. Vimalesvaran, G. S. Kanaganayagam, et al. Automated left ventricular dimension assessment using artificial intelligence developed and validated by a uk-wide collaborative. *Circulation: Cardiovascular Imaging*, 14(5): e011951, 2021.

Z. Huang, G. Long, B. Wessler, and M. C. Hughes. A new semi-supervised learning benchmark for classifying view and diagnosing aortic stenosis from echocardiograms. In *Machine Learning for Healthcare Conference*, pages 614–647. PMLR, 2021.

Z. Huang, G. Long, B. Wessler, and M. C. Hughes. Tmed 2: a dataset for semi-supervised classification of echocardiograms. In *In DataPerf: Benchmarking Data for Data-Centric AI Workshop*, 2022.

Z. Huang, L. Shen, J. Yu, B. Han, and T. Liu. Flatmatch: Bridging labeled data and unlabeled data with cross-sharpness for semi-supervised learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Z. Huang, M.-J. Sidhom, B. Wessler, and M. C. Hughes. Fix-a-step: Semi-supervised learning from uncurated unlabeled data. In *International Conference on Artificial Intelligence and Statistics*, pages 8373–8394. PMLR, 2023b.

S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021.

A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5070–5079, 2019.

M. A. Khan, N. Yadav, M. Jain, and S. Goyal. The art of embedding fusion: Optimizing hate speech detection. *arXiv preprint arXiv:2306.14939*, 2023.

D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

A. Kumar, P. Sattigeri, and T. Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. *Advances in neural information processing systems*, 30, 2017.

C.-W. Kuo, C.-Y. Ma, J.-B. Huang, and Z. Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 479–495. Springer, 2020.

J. Kwon, J. Kim, H. Park, and I. K. Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.

S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

S. Leclerc, E. Smistad, J. Pedrosa, A. Østvik, F. Cervenansky, F. Espinosa, T. Espeland, E. A. R. Berg, P.-M. Jodoin, T. Grenier, et al. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE transactions on medical imaging*, 38(9):2198–2210, 2019.

D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

H. Li, Z. Wu, A. Shrivastava, and L. S. Davis. Rethinking pseudo labels for semi-supervised object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1314–1322, 2022.

J. Li, C. Xiong, and S. C. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021.

M. Li, R. Wu, H. Liu, J. Yu, X. Yang, B. Han, and T. Liu. Instant: Semi-supervised learning with instance-dependent thresholds. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

X. Li. Positive-incentive noise. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1 – 7, 2022.

B. Liu, Z. Wu, H. Hu, and S. Lin. Deep metric transfer for label propagation with limited annotated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

Y. Liu, S. Mai, X. Chen, C.-J. Hsieh, and Y. You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12360–12370, 2022.

I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

A. Madani, R. Arnaout, M. Mofrad, and R. Arnaout. Fast and accurate view classification of echocardiograms using deep learning. *NPJ digital medicine*, 1(1):6, 2018.

S. Min, X. Chen, H. Xie, Z.-J. Zha, and Y. Zhang. A mutually attentive co-training framework for semi-supervised recognition. *IEEE Transactions on Multimedia*, 23:899–910, 2020.

C. Mitchell, P. S. Rahko, L. A. Blauwet, B. Canaday, J. A. Finstuen, M. C. Foster, K. Horton, K. O. Ogunyankin, R. A. Palma, and E. J. Velazquez. Guidelines for performing a comprehensive transthoracic echocardiographic examination in adults: recommendations from the american society of echocardiography. *Journal of the American Society of Echocardiography*, 32(1):1–64, 2019.

T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pages 4723–4732. PMLR, 2019.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018.

L. K. Pereira, Y. Taya, and I. Kobayashi. Multi-layer random perturbation training for improving model generalization efficiently. *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2021.

H. Pham, Z. Dai, Q. Xie, and Q. V. Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021.

K. Saito, D. Kim, and K. Saenko. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. *arXiv preprint arXiv:2105.14148*, 2021.

M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016.

K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

A. Subramanya and P. P. Talukdar. *Graph-based semi-supervised learning*. Springer Nature, 2022.

A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

J. E. Van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019.

Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, and R. Wang. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems*, 2022.

Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2023.

Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021a.

Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR, 2021b.

L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7236–7246, 2023.

Q. Yu, D. Ikami, G. Irie, and K. Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 438–454. Springer, 2020.

X. Yu, Y. Xue, L. Zhang, L. Wang, T. Liu, and D. Zhu. Exploring the influence of information entropy change in learning systems. *arXiv preprint arXiv:2309.10625*, 2023a.

X. Yu, L. Zhang, D. Zhu, and T. Liu. Robust core-periphery constrained transformer for domain adaptation. *arXiv preprint arXiv:2308.13515*, 2023b.

S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. jianup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

X. Zhao, K. Krishnateja, R. Iyer, and F. Chen. How out-of-distribution data hurts semi-supervised learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 763–772. IEEE, 2022.

M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14471–14481, 2022.

M. Zheng, S. You, L. Huang, C. Luo, F. Wang, C. Qian, and C. Xu. Simmatchv2: Semi-supervised learning with graph consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16432–16442, 2023.

X. Zhu. Semi-Supervised Learning Literature Survey. Technical Report Technical Report 1530, Department of Computer Science, University of Wisconsin Madison., 2005.

Supplementary Material

In this supplement, we provide:

- Sec. A: Dataset Details
- Sec. B: Hyperparameter Details
- Sec. C: InterLUDE+ Details
- Sec. D: Additional Albation and Sensitivity Analysis
- Sec. E: Additional Results on Class-dependent Delta Consistency Loss
- Sec. F: Additional Discussion

A. Additional Dataset Details

Here, we provide more details on the datasets used in the paper. For the medical image datasets used in the Heart2Heart Benchmark (Huang et al., 2023b), we emphasize that all three medical image datasets are deidentified and accessible to academic researchers.

A.1. Classic Benchmarks

- The CIFAR-10 dataset is a collection of images commonly used to train machine learning and computer vision algorithms, and has been a classic benchmark to use for SSL. It contains 60,000 32x32 color images in 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 testing images. The classes include various objects and animals like cars, birds, dogs, and ships..
- Similar to CIFAR-10, the CIFAR-100 dataset is another common SSL benchmark. CIFAR-100 has same total number of images as CIFAR-10, i.e., 60,000 32x32 color images, but they are spread over 100 classes, each containing 600 images, with 500 training images and 100 testing images per class. The dataset features a more diverse set of classes compared to CIFAR-10
- The STL-10 dataset is also popular in benchmarking SSL algorithms. The dataset contains 5,000 labeled training images and 8,000 test images, with each image being a higher resolution of 96x96 pixels. It contains 10 classes: airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck. Additionally, the dataset provides 100,000 unlabeled images for unsupervised learning, which are drawn from a similar but broader distribution than the labeled images.

A.2. Heart2Heart Benchmark

The Heart2Heart Benchmark contains three medical image datasets of heart ultrasound images, collected independently by different research groups in the world. Thanks to common device standards, these images are interoperable. All data used in the benchmark are resized to 112x112 (Huang et al., 2023b).

The benchmark adopts a view classification task: *Given an ultrasound image of the heart, identify the specific anatomical view depicted.* Such task is of great clinical importance, as determining the view type is a prerequisite for many clinical measurements and diagnoses (Madani et al., 2018).

- The TMED-2 dataset provides set of labeled images of four specific view types: Parasternal Long Axis (PLAX), Parasternal Short Axis (PSAX), Apical Four Chamber (A4C) and Apical Two Chamber (A2C), gathered from certified annotators, and a truly *uncurated* unlabeled set of 353,500 images from routine scans of 5486 patient-studies. With routine Transthoracic Echocardiograms (TTEs) commonly presenting at least nine canonical view types (Mitchell et al., 2019), this unlabeled set likely contains classes not found in the labeled set. Moreover, the unlabeled dataset consists of a wide variety of patient scans, in contrast to the labeled dataset, which specifically contains a higher percentage of patients with a heart disease called aortic stenosis (AS), particularly severe AS. About 50% of patients in the labeled set have severe AS, a significant increase compared to its less than 10% occurrence in the general population. This discrepancy in sampling results in noticeable feature differences. For instance, PLAX and PSAX images from patients with severe AS typically exhibit more pronounced calcification (thickening) of the aortic valve.

Note that the TMED-2 data used in Heart2Heart Benchmark is only **a subset of the full TMED-2 dataset**, more details should be referred to the original papers (Huang et al., 2022; 2021)

- The UNITY dataset contains ultrasound images of the heart collected from 17 hospitals in the UK. The original dataset contains other view such as A3C and A5C, but only PLAX, A2C and A4C are used in the Heart2Heart Benchmark. More details should be referred to the original paper (Howard et al., 2021).
- The CAMUS dataset contains ultrasound images of the heart collected from a hospital in France. The dataset contains A2C and A4C. Beside the view label, the original dataset also provide labels for the cardiac cycle when the image is acquired (e.g., end diastolic or end systolic). More details should be referred to the original paper (Leclerc et al., 2019).

B. Additional Hyperparameter Details

Here we provide complete list of the hyperparameters used in the paper:

B.1. Classic SSL Benchmarks

CNN Backbones. Here we list the hyperparameters used for CNN backbone experiments in Sec 4. We closely follow the established setting from prior studies (Wang et al., 2023).

Table B.1. Algorithm-independent hyperparameters on CNNs

Dataset	CIFAR-10	CIFAR-100	STL-10
Model	WRN-28-2	WRN-28-8	WRN-37-2
Weight decay	5e-4	1e-3	5e-4
Batch size	64	64	64
Learning rate	0.03	0.03	0.03
SGD momentum	0.9	0.9	0.9
EMA decay	0.999	0.999	0.999

For algorithm dependent hyperparameters, we set λ_{DC} to 1.0, embedding fusion strength α to 0.1. We set the unlabeled batch ratio μ to 7 following convention (Sohn et al., 2020; Zheng et al., 2023; Wang et al., 2023). Note that μ is a common hyperparameter to many SSL algorithms, not unique to ours.

ViT Backbones. Here we list the hyperparameters used for the ViT backbone experiments in Sec 4. We closely follow the established setting from prior studies (Li et al., 2023). For the ViT-small experiments, we utilized the pretrained weights from ViT-small as provided by (Wang et al., 2022). In the case of ViT-base experiments, we employed the pretrained weights from PyTorch image models, as documented in (Imambi et al., 2021).

Table B.2. Algorithm-independent hyperparameters on ViTs

Dataset	CIFAR-10	CIFAR-100	STL-10
Model	ViT-small	ViT-small	ViT-base
Weight decay	5e-4	5e-4	5e-4
Batch size	8	8	8
Learning rate	5e-4	5e-4	1e-4
Optimizer	AdamW	AdamW	AdamW
EMA decay	0.999	0.999	0.999

For algorithm dependent hyperparameters, we set λ_{DC} to 0.1, embedding fusion strength α to 0.1. We set the unlabeled batch ratio μ to 7 following convention (Sohn et al., 2020; Zheng et al., 2023; Wang et al., 2023). Note that μ is a common hyperparameter to many SSL algorithms, not unique to ours.

B.2. Heart2Heart Benchmark

Our experiments closely follow the protocol from (Huang et al., 2023b). We directly inherit all the common hyperparameters from (Huang et al., 2023b) without tunning. For FreeMatch and FlatMatch, we additionally search λ_{SAF} in [0.01, 0.05, 0.1]. For InterLUDE and InterLUDE+ we search λ_{DC} in [0.1, 1.0].

Table B.3. Heart2Heart Benchmark Hyperparameters

	Data Split 1	Data Split 2	Data Split 3
Weight decay	5e-4	5e-4	5e-4
Learning rate	0.1	0.1	0.1
Batch size	64	64	64
Optimizer	SGD	SGD	SGD
		FreeMatch	
λ_{SAF}	0.05	0.01	0.05
		FlatMatch	
λ_{SAF}	0.05	0.1	0.01
		InterLUDE	
λ_{DC}	1.0	0.1	0.1
		InterLUDE+	
λ_{DC}	1.0	0.1	0.1

C. InterLUDE+ Details

Recently, Self-Adaptive Threshold (SAT) and Self-Adaptive Fairness (SAF) were introduced by Wang et al. (2023), with potential applicability to other SSL algorithms. Here, we integrate SAT and SAF into our InterLUDE framework, and named the enhanced version InterLUDE+.

SAT aim to adjust the pseudo-label threshold by considering both the global (dataset-specific) threshold and local (class-specific) threshold at different time step, each estimated via the model's current learning status. Similar techniques such as Distribution Alignment (DA) (Berthelot et al., 2019a) and its variant (Berthelot et al., 2021) has been proposed and used by various SSL algorithms (Li et al., 2021; Zheng et al., 2022).

$$\tau_t(c) = \frac{\tilde{p}_t(c)}{\max\{\tilde{p}_t(c) : c \in [C]\}} \cdot \tau_t, \tag{12}$$

where τ_t is the gloabl threshold at time t and $\tilde{p}_t(c)$ is the local threshold for class c at time t.

$$\tau_t = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tau_{t-1} + (1-\lambda) \frac{1}{\mu B} \sum_{j=1}^{\mu B} \max(q_j^w), & \text{otherwise}, \end{cases} \tag{13}$$

$$\tilde{p}_t(c) = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tilde{p}_{t-1}(c) + (1-\lambda) \frac{1}{\mu B} \sum_{j=1}^{\mu B} q_j^w(c), & \text{otherwise}, \end{cases} \tag{14}$$

SAF is a regularization term that encourages diverse model predictions on unlabeled data.

$$\mathcal{L}^{SAF} = -H\left[\text{SumNorm}\left(\frac{\tilde{p}_t}{\tilde{h}_t}\right), \text{SumNorm}\left(\frac{\bar{p}}{\bar{h}}\right)\right] \tag{15}$$

where

$$\bar{p} = \frac{1}{\mu B} \sum_{j=1}^{\mu B} \mathbf{1}(\max(q_j^w) \geq \tau_t(\arg\max(q_j^w))q_j^s, \tag{16}$$

$$\bar{h} = \text{Hist}_{\mu B}\left(\mathbf{1}(\max(q_j^w) \geq \tau_t(\arg\max(q_j^w))\hat{q_j^s}\right), \tag{17}$$

$$\tilde{h}_t = \lambda \tilde{h}_{t-1} + (1-\lambda)\text{Hist}_{\mu B}(\hat{q_j^w}) \tag{18}$$

SumNorm $= (\cdot)/\sum(\cdot)$. Hist means the histogram distribution. $\hat{q_j^w}$ and $\hat{q_j^s}$ are the one-hot encoding of q_j^w and q_j^s. More details of the SAT and SAF should be referred to in the original paper (Wang et al., 2023).

Incorporating SAT and SAF with InterLUDE, the loss function of InterLUDE+ can be written as:

$$\mathcal{L} = \mathcal{L}^L + \lambda_u \mathcal{L}^U + \lambda_{DC}\mathcal{L}_{avg}^{DC} + \lambda_{SAF}\mathcal{L}^{SAF} \tag{19}$$

where $\tau_t(c)$ is used to replaced the fixed threshold τ in Eq. (10).

D. Additional Experiments

D.1. CNN backbones

D.1.1. Sensitivity Analysis

The sensitivity analysis of hyperparameters unique to our method: delta consistnecy loss λ_{DC} and embedding fusion strength α with CNN backbone on the CIFAR10 dataset are presented in Fig. D.1 and D.2, respectively. We can observe that the error remains stable over a wide range of λ_{DC}, leading us to conclude that InterLUDE is not overly sensitive to λ_{DC}. When considering the embedding fusion strength, we see that α around 0.1 to 0.2 is generally a good value. Excessively high α values, nearing the 0.5 upper limit defined in Eq. (3) result in a significant drop in performance. More sensitivity analysis on the ViT backbones can be found in sec. D.2.2.

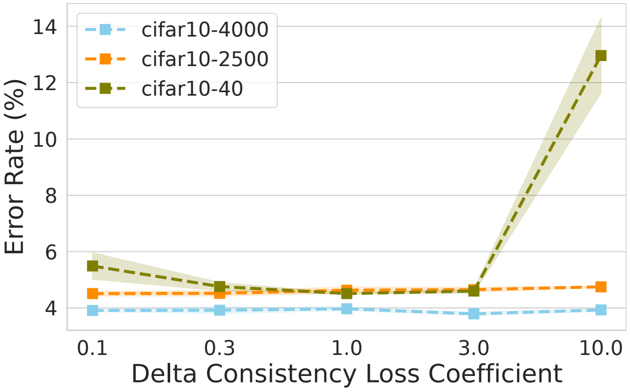

Figure D.1. Sensitivity of Delta Consistency Loss Coefficient. Showing CNN performance on CIFAR-10.

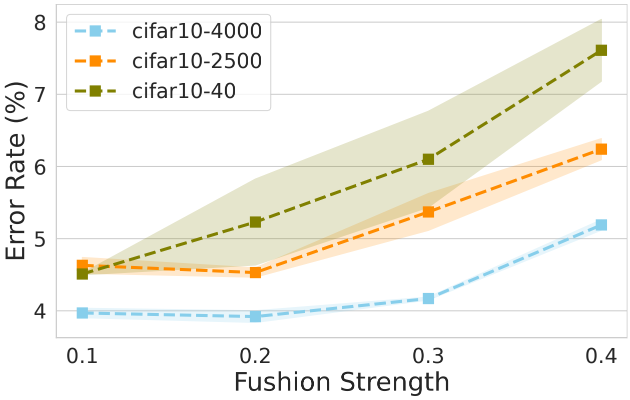

Figure D.2. Sensitivity of Embedding Fusion Strength. Showing CNN performance on CIFAR-10.

D.1.2. Augmentation Strategies

RandAugment involves two hyperparameter: N and M. N represents the number of augmentation transformations applied sequentially to an input image, and M denotes the magnitude of these transformations. Higher values for N and M result in stronger distortion to the input.

Below, we evaluate InterLUDE on CIFAR-10 with 40 labels (CNN) using various N and M. The default setting from prior SSL literature is N=2 and M=10. Experiments in our paper use this default choice. Here, due to computation constraints, we only run the experiment with 1 run.

Table D.3. Sensitivity to Augmentation Strength

n	2*	2	2	3	3	3	5	5	5
m	10*	8	5	10	8	5	10	8	5
Results	4.51	5.03	5.66	4.76	5.65	5.22	6.37	6.55	7.24

The results show that the value of N (number of transformations sequentially applied to the input image) has a larger impact on the performance than M. When N=5, the performance notably decreases, which is reasonable, as composing 5 sequential transformations to the input severely destroys the semantic meaning of the original input. Please note that this decrease in performance due to augmentation being too strong is not unique to our method, other works have obtained similar conclusions (Gui et al., 2023). On the other hand, given the same N, larger M seems to work slightly better.

We also explored the impact of deviating from the standard weak-strong augmentation structure with either a weak-weak augmentation or strong-strong augmentation approach. We find that deviation from such established weak-strong format notably decreases performance. These observations are in line with those reported in FixMatch, which highlighted the effectiveness of adopting the weak-strong structure. This is discussed in section 5.2 of their paper.

D.1.3. Additional Ablation of layout: high vs. low L-U interaction.

Our interdigitated batch layout (Fig 2) is specifically designed to enhance labeled-unlabeled interaction. Here we conduct more ablations to analysis the effect of high vs. low L-U interaction under the same circular shift embedding fusion operation. We contrast the low L-U interaction batch layout with 3 high L-U interaction layouts (including the one we used as default in the paper: Fig 2)

The different layouts are defined as follows:

$$\text{Low-I} =: \textsc{Stack}(\{\{x_i^w\}_{i=1}^{B}, \{x_i^s\}_{i=1}^{B}, \{\bar{x}_j^w\}_{j=1}^{\mu*B}, \{\bar{x}_j^s\}_{j=1}^{\mu*B}\}) \tag{20}$$

$$\text{High-I1} =: \textsc{Stack}(\{\{x_i^w\}_{i=1}^{B/2(\mu+1)}, \{x_i^s\}_{i=1}^{B/2(\mu+1)}, \{\bar{x}_j^w\}_{j=1}^{(\mu*B)/2(\mu+1)}, \{\bar{x}_j^s\}_{j=1}^{(\mu*B)/2(\mu+1)}\}^{2(\mu+1)}) \tag{21}$$

$$\text{High-I2} =: \textsc{Stack}(\{\{x_i^w\}, \{x_i^s\}, \{\bar{x}_j^w\}_{j=1}^{\mu}, \{\bar{x}_j^s\}_{j=1}^{\mu}\}^{B}) \tag{22}$$

$$\text{High-I3 (default)} =: \textsc{Stack}(\{\{x_i^w\}, \{\bar{x}_j^w\}_{j=1}^{\mu}, \{x_i^s\}, \{\bar{x}_j^s\}_{j=1}^{\mu}\}^{B}) \tag{23}$$

$$\tag{24}$$

Note that applying the same circular-shift fusion, these different layout have exactly the same number of within-batch interactions, the high-L-U-interaction design has more labeled-unlabeled interactions.

We can see from Fig D.4 that having more L-U interactions is *especially important in the low label regime*. However, as the labeled examples becoming more available (e.g., CIFAR-10 with 4000 labels scenario), the advantage of having L-U interactions over L-L interactions diminishes, which is intuitive: *if we have enough labeled data, we might not need the unlabeled data*.

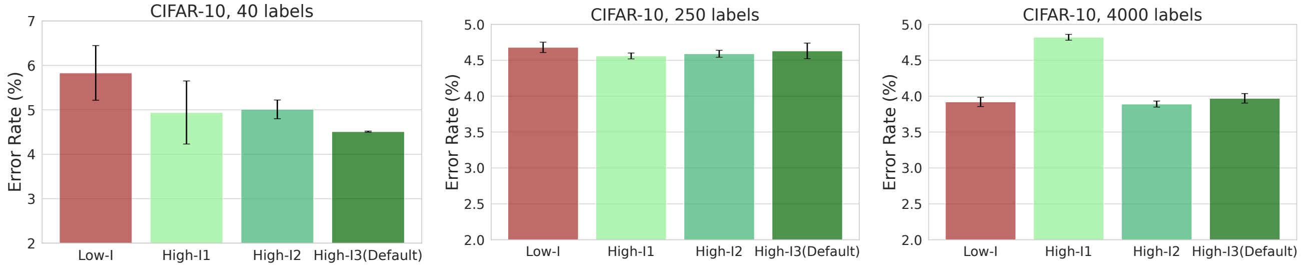

Figure D.4. Effect of Different Batch Layout. From left to right are CIFAR-10 with 40 labels, 250 labels and 4000 labels.

D.1.4. Wall Time Comparison

Here we compare the wall time of InterLUDE, FreeMatch (Wang et al., 2023), FlexMatch (Zhang et al., 2021) and FlatMatch (Huang et al., 2023a) on the TMED2 dataset using the **exact same hardware** (NVIDIA A100 with 80G Memory). We can see that InterLUDE has a substantially lower run time cost per iteration. For each of our two key innovations, here is a brief efficiency analysis: Embedding fusion is a linear transformation of the feature representations in the embedding space and can be implemented efficiently with 2 lines of code in Pytorch. Delta-consistency loss calculation can be done using existing logits already computed for the supervised loss term and the instance-wise consistency loss term.

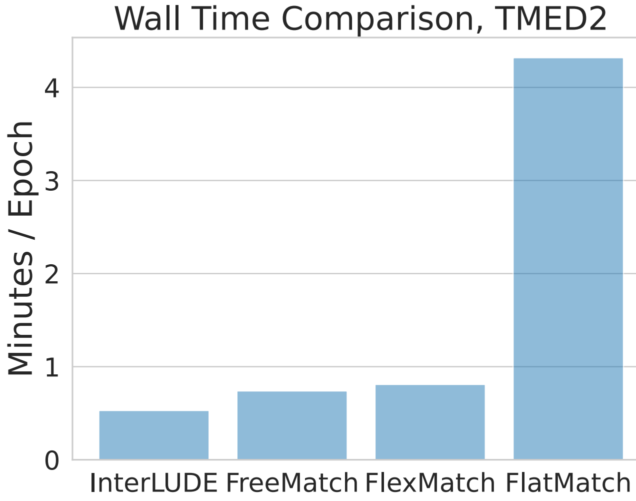

Figure D.5. Wall Time Comparison

D.2. ViT backbones

D.2.1. Ablation

The ablation of InterLUDE and InterLUDE+ with ViT backbone are presented in Table D.6. We can observe that the delta consistency loss and embedding fusion both contribute to the performance improvement of the proposed method.

Table D.6. Ablation study on ViT backbone. Error rates (%) are averaged with three random seeds and reported with a 95% confidence interval.

Dataset	CIFAR10		CIFAR100		STL10	
#Label	40	250	400	2500	40	100
InterLUDE	1.78±0.1	1.55±0.1	21.19±0.2	13.39±0.1	3.14±0.2	2.66±0.1
InterLUDE (w/o Embedding Fusion)	2.31±0.9	1.64±0.1	23.70±1.9	13.88±0.3	5.23±3.7	3.23±0.4
InterLUDE (w/o $\mathcal{L}_{avg}^{\mathrm{DC}}$)	2.37±0.9	1.75±0.2	23.20±1.7	14.07±0.6	3.75±0.4	3.36±0.3
InterLUDE+	1.55±0.1	1.49±0.1	16.32±0.3	12.93±0.2	4.56±0.9	3.23±0.3
InterLUDE+ (w/o Embedding Fusion)	1.78±0.1	1.61±0.1	16.63±1.6	14.00±0.6	4.25±1.0	4.37±0.1
InterLUDE+ (w/o $\mathcal{L}_{avg}^{\mathrm{DC}}$)	1.57±0.1	1.60±0.1	16.33±0.8	13.15±0.1	4.71±1.0	4.55±0.3

D.2.2. Sensitivity Analysis

Due to limited computational resources, we conduct a single experiment for sensitivity analysis on the ViT backbone. The sensitivity analysis of the hyperparameters of the delta consistency loss λ_{DC} and embedding fusion strength α with ViT backbone on the CIFAR10 dataset are presented in Fig. D.7, and D.8 respectively.

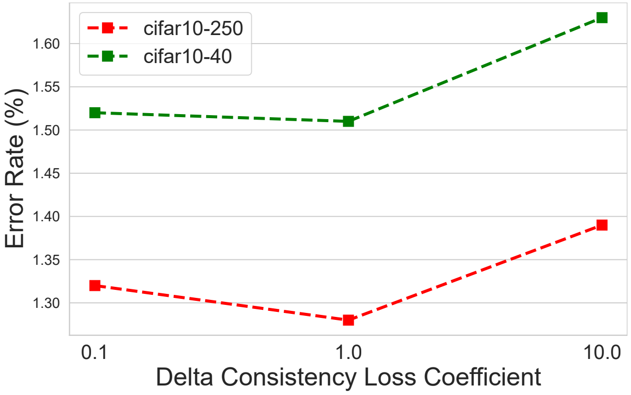

Figure D.7. Sensitivity of Delta Consistency Loss Coefficient. Showing ViT performance on CIFAR-10.

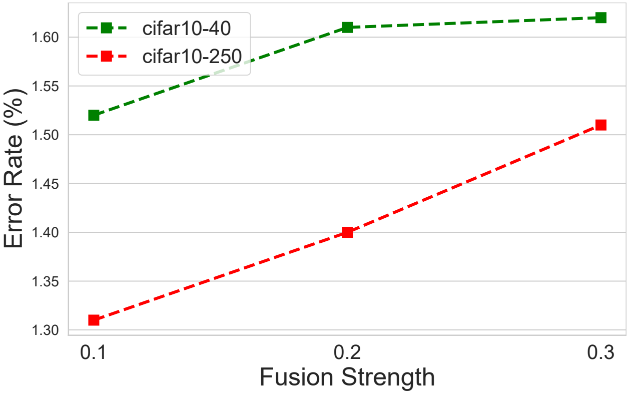

Figure D.8. Sensitivity of Embedding Fusion Strength. Showing ViT performance on CIFAR-10.

E. Additional Results on class-dependent delta-consistency loss

In this section, we present the result with the class-dependent formulation of the delta consistency loss on the classic SSL benchmarks. All the experiment setting here are the same as in the main paper, except the delta consistency loss term is from Eq. 7 instead of Eq. 6.

Table E.1 show results with CNN backbones. Table E.2 show results with ViT backbones. Overall, the two delta consistency loss formulation perform similarly, but the class-dependent version is more complex in implementation.

Table E.1. Error rate (%) with CNNs. Following (Zheng et al., 2023), error rate and standard deviation are reported based on three runs. All experiments follow the same settings. Rows marked * are implemented by us using the author's code. Results of other methods are directly copied from SimMatchV2 (Zheng et al., 2023) and the original papers ("–" means the result is not available). The best results are highlighted in bold and the second-best underlined. We did not run DeepLP on STL-10 due to computation constraints.

Dataset	**CIFAR10**			**CIFAR100**			**STL-10**	
#Label	40	250	4000	400	2500	10000	250	1000
Fully-Supervised		4.57±0.06			18.96±0.06		–	
Supervised	77.18±1.32	56.24±3.41	16.10±0.32	89.60±0.43	58.33±1.41	36.83±0.21	55.07±1.83	35.42±0.48
Manifold MixUp* (Verma et al., 2019)	73.51±1.81	55.44±2.06	17.40 ±1.11	87.98±0.55	59.45±0.85	33.23±0.21	54.99±1.22	29.17±2.00
Pseudo-Labeling (Lee et al., 2013)	75.95±1.86	51.12±2.91	15.32±0.35	88.18±0.89	55.37±0.48	36.58±0.12	51.90±1.87	30.77±0.04
Π-Model (Laine and Aila, 2016)	76.35 ± 1.69	48.73±1.07	13.63±0.07	87.67±0.79	56.40±0.69	36.73±0.05	52.20±2.11	31.34±0.64
DeepLP* (Iscen et al., 2019)	72.65 ± 2.04	35.80±1.43	10.58±0.29	86.11±0.61	62.87±0.42	43.03 ±1.36	–	–
Mean Teacher (Tarvainen and Valpola, 2017)	72.42±2.10	37.56±4.90	8.29±0.10	79.96±0.53	44.37±0.60	31.39±0.11	49.30±2.09	27.92±1.65
VAT (Miyato et al., 2018)	78.58±2.78	28.87±3.62	10.90±0.16	83.60±4.21	46.20±0.80	32.14±0.31	57.78±1.47	40.98±0.96
MixMatch (Berthelot et al., 2019b)	35.18±3.87	13.00±0.80	6.55±0.05	64.91±3.34	39.29±0.13	27.74±0.27	32.05±1.16	20.17±0.67
ReMixMatch (Berthelot et al., 2019a)	8.13±0.58	6.34±0.22	4.65±0.09	41.60±1.48	25.72±0.07	20.04±0.13	11.14±0.52	6.44±0.15
FeatMatch (Kuo et al., 2020)	–	7.50±0.64	4.91±0.18	–	–	–	–	–
UDA (Xie et al., 2020)	10.01±3.34	5.23±0.08	4.36±0.09	45.48±0.37	27.51±0.28	23.12±0.45	10.11±1.15	6.23±0.28
FixMatch (Sohn et al., 2020)	12.66±4.49	4.95±0.10	4.26±0.01	45.38±2.07	27.71±0.42	22.06±0.10	8.64±0.84	5.82±0.06
Dash (Xu et al., 2021b)	9.29±3.28	5.16±0.28	4.36±0.10	47.49±1.05	27.47±0.38	21.89±0.16	10.50±1.37	6.30±0.49
MPL (Pham et al., 2021)	6.62±0.91	5.76±0.24	4.55±0.04	46.26±1.84	27.71±0.19	21.74±0.09	–	6.66±0.00
CoMatch (Li et al., 2021)	6.51±1.18	5.35±0.14	4.27±0.12	53.41±2.36	29.78±0.11	22.11±0.22	7.63±0.94	5.71±0.08
FlexMatch (Zhang et al., 2021)	5.29±0.29	4.97±0.07	4.24±0.06	40.73±1.44	26.17±0.18	21.75±0.15	9.85±1.35	6.08±0.34
AdaMatch (Berthelot et al., 2021)	5.09±0.21	5.13±0.05	4.36±0.05	37.08±1.35	26.66±0.33	21.99±0.15	8.59±0.43	6.01±0.02
SimMatch (Zheng et al., 2022)	5.38±0.01	5.36±0.08	4.41±0.07	39.32±0.72	26.21±0.37	21.50±0.11	8.27±0.40	5.74±0.31
FreeMatch (Wang et al., 2023)	4.90±0.04	4.88±0.18	4.10±0.02	37.98±0.42	26.47±0.20	21.68±0.03	–	5.63±0.15
SoftMatch (Chen et al., 2023)	4.91±0.12	4.82±0.09	4.04±0.02	37.10±0.77	26.66±0.25	22.03±0.03	–	5.73±0.24
SimMatchV2 (Zheng et al., 2023)	4.90±0.16	5.04±0.09	4.33±0.16	36.68±0.86	26.66±0.38	21.37±0.20	7.54±0.81	5.65±0.26
FixMatch (w/SAA) (Gui et al., 2023)	5.24±0.99	4.79±0.07	3.91±0.07	45.71±0.73	26.82±0.21	21.29±0.20	–	–
InstanT (Li et al., 2023)	5.17±0.10	5.28±0.02	4.43±0.01	46.06±1.80	32.91±0.00	27.70±0.40	–	–
FlatMatch (Huang et al., 2023a)	5.58±2.36	**4.22**±**1.14**	3.61±0.49	38.76±1.62	25.38±0.85	**19.01**±**0.43**	–	**4.82**±**1.21**
FlatMatch-e (Huang et al., 2023a)	5.63±1.87	4.53±1.85	**3.57**±**0.50**	38.98±1.53	25.62±0.88	19.78±0.89	–	5.03±1.06
InterLUDE (avg)	4.51±0.01	4.63±0.11	3.96±0.07	35.32±1.06	**25.20**±**0.22**	20.77±0.19	7.05±0.12	5.01±0.04
InterLUDE+ (avg)	**4.46**±**0.11**	4.46±0.09	3.88±0.05	36.99±0.62	25.27±0.17	20.49 ±0.15	6.99 ±0.42	4.92±0.05
InterLUDE (cls)	4.68±0.18	4.57±0.09	3.82±0.02	35.61±1.63	25.50±0.18	20.63±0.09	7.14±0.36	5.08±0.06
InterLUDE+ (cls)	4.58±0.23	4.45±0.09	3.76±0.07	**35.13**±**0.48**	25.43±0.09	20.30 ±0.19	**6.98** ±**0.51**	4.98±0.05

Table E.2. Error rate (%) with ViT backbone. The error rate and 95% confidence interval are reported based on three random seeds (Li et al., 2023). Rows marked * are implemented by us using the author's code. Other results directly copied from Li et al. (2023). The best results are highlighted in bold and the second-best underlined.

Dataset	CIFAR10			CIFAR100			STL10		
#Label	10	40	250	200	400	2500	10	40	100
PL (Lee et al., 2013)	62.35±3.1	11.79±5.3	4.58±0.4	36.66±2.0	26.87±0.9	15.72±0.1	69.26±6.7	42.84±4.2	26.56±1.5
MT (Tarvainen and Valpola, 2017)	35.43±4.9	12.85±2.5	4.75±0.5	40.50± 0.8	30.58±0.9	17.09±0.4	57.28±7.8	33.20±3.4	22.29±1.8
MixMatch (Berthelot et al., 2019b)	34.96±2.6	2.84±0.9	2.05±0.1	39.64± 1.3	27.74±0.1	16.16±0.3	89.32±1.1	72.42±16.2	38.15±11.3
VAT (Miyato et al., 2018)	39.93±6.3	6.67±6.6	2.33±0.2	34.11±1.8	24.67±0.4	16.58±0.4	79.43±4.4	34.82±7.0	19.06±1.0
UDA (Xie et al., 2020)	21.24±3.6	2.08±0.2	2.04±0.1	34.51±1.6	24.15±0.6	16.19±0.2	51.63±4.3	20.33±4.9	10.60±1.0
FixMatch (Sohn et al., 2020)	33.50±15.1	2.56±0.9	2.05±0.1	34.71±1.4	24.48±0.1	16.02±0.1	59.87±3.4	22.28±4.4	11.59±1.6
FlexMatch (Zhang et al., 2021)	29.46±9.6	2.22±0.3	2.12±0.2	36.24±0.9	25.99±0.5	16.28±0.2	39.37±12.9	21.83±3.7	10.46±1.3
Dash (Xu et al., 2021b)	25.65±4.5	3.37±2.0	2.10±0.3	36.67±0.4	25.46±0.2	15.99±0.2	58.94±4.4	21.97±3.9	10.44±2.0
AdaMatch (Berthelot et al., 2021)	14.85±20.4	2.06±0.1	2.08±0.1	26.39±0.1	21.41±0.4	15.51±0.1	31.83±7.7	16.50±4.2	10.75±1.5
FlatMatch* (Huang et al., 2023a)	**11.95±7.3**	3.17±0.3	2.33±0.1	26.56±1.0	21.80±1.0	13.83±0.3	23.90±8.9	6.25±0.3	4.74±0.2
InstanT (Li et al., 2023)	12.68±10.2	2.07±0.1	1.92±0.1	25.83±0.3	21.20±0.4	15.72±0.5	30.61±7.4	14.91±2.8	10.65±1.9
InterLUDE (avg)	31.90±4.1	1.78±0.1	1.55±0.1	35.66±1.9	21.19±0.2	13.39±0.1	27.49±6.6	**3.14±0.2**	**2.66±0.1**
InterLUDE+ (avg)	12.29±7.3	1.55±0.1	1.49±0.1	23.60±1.2	16.32±0.3	12.93±0.2	25.83±9.9	4.56±0.9	3.23±0.3
InterLUDE (cls)	31.66±10.7	1.67±0.1	1.48±0.1	34.96±1.6	21.38±0.2	13.21±0.3	40.83±22.5	3.80±0.9	3.78±0.8
InterLUDE+ (cls)	17.31±10.2	**1.45±0.1**	**1.31±0.1**	**20.45±1.7**	**15.78±0.6**	**12.36±0.2**	**14.4±1.8**	3.56±0.5	3.48±0.3

F. Additional Discussion

F.1. Additional Discussion on Heart2Heart Benchmark Results

While our InterLUDE and InterLUDE+ are not specifically designed for open-set SSL, the performance on the Heart2Heart Benchmark is supprisingly strong. On the other hand, FlatMatch that is competitive on the classic benchmarks substantially underperform in this open-set medical imaging benchmark, we *hypothesize* that this due to FlatMatch's cross-sharpness objective's goal of pulling model toward direction that is "beneficial to generalization on unlabeled data". The unlabeled set of TMED-2 is uncurated, containing both out-of-distribution classes as well as feature distribution shift. More study is needed to understands the challenges in this uncurated unlabeled set and the limitation of current SSL algorithms under this challenging real-world scenario.