Semantic augmentation by mixing contents for semi-supervised learning

Rémy Sun, Clément Masson, Gilles Hénaff, Nicolas Thome, Matthieu Cord

Semantic Augmentation by Mixing Contents for Semi-Supervised Learning

Rémy Sun[1 3] *, Clément Masson[1], Gilles Hénaff[1]

[1]{*clement.masson; gilles.henaff*}*@fr.thalesgroup.com*

Optronics & Missile Electronics, Land & Air Systems, Thales. 2 Avenue Gay Lussac, Élancourt, France

Nicolas Thome[2]

[2]*nicolas.thome@cnam.fr*

Vertigo, CEDRIC, Conservatoire National des Arts et Métiers. 292 Rue Saint-Martin, Paris, France.

Matthieu Cord[3]

[3]{*remy.sun; matthieu.cord*}*@lip6.fr*

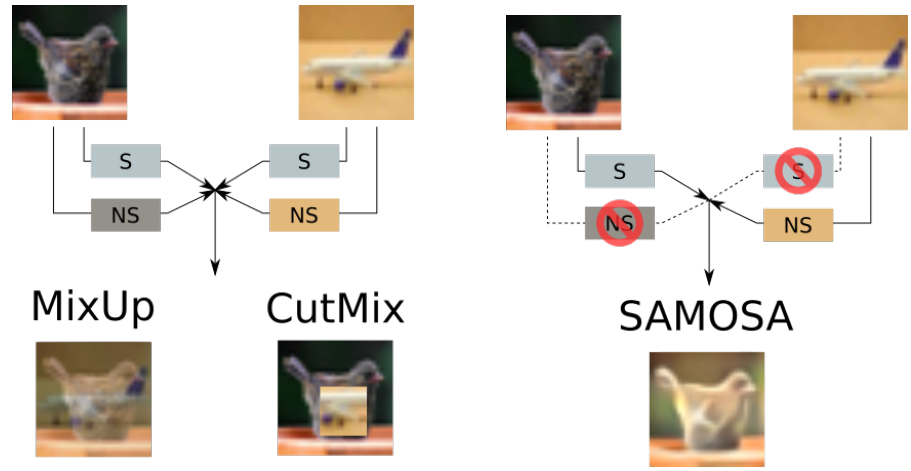
MLIA, LIP6, Sorbonne Université. Campus Pierre et Marie Curie, 4 place Jussieu, Paris, France

Abstract

Leveraging unlabeled examples is a crucial issue for boosting performances in semi-supervised learning. In this work, we introduce the SAMOSA framework based on semantic augmentation for mixing semantic components from labeled examples and non semantic characteristics from unlabeled data. Our approach is based on a novel reconstruction module that can be grafted onto most state of the art networks. The proposed approach leans on two main aspects: an architectural component optimized to disentangle semantic and auxiliary non semantic representations using an unsupervised loss, and a semantic augmentation scheme that leverages this disentangling module to generate artificially labeled examples preserving known class information while controlling auxiliary variations. We demonstrate the ability of our method to improve the performance of models trained according to standard semi-supervised procedures Mean Teacher [1], MixMatch [2] and FixMatch [3].

Keywords: Deep-Learning; Semi-Supervised Learning; Data Augmentation; Mixing augmentation

*Corresponding author



(a) Classic mixing creates between-class hybrids.

(b) SAMOSA creates in-class hybrids

Figure 1: While classical mixing combines general content (i.e. semantic ”S” + non-semantic “NS”) from both parents, SAMOSA clearly mixes semantic content (“S”) from one parent and non semantic content (“NS”) from the other.

1. Introduction

Deep architectures have proven capable of reliably solving a variety of tasks such as classification [4, 5], object detection [6] or machine translation [7]. This is however contingent on there being a large amount of labeled data to train models in a supervised fashion. This is seldom the case in practical applications where labeling comes at a significant cost.

A more realistic setting is defined by Semi-Supervised Learning (SSL), where some labeled data is provided but most of the available data is unlabeled. The unlabeled data has been leveraged to improve model performance, most notably through the use of consistency based methods [8, 1, 9]. Consistency based methods guide models towards solution stable with regard to small perturbations around samples. Recently, mixing augmentations - which mix (Fig. 1a) two samples/label pairs by interpolating samples and labels - have been used to great success [2, 10, 11, 12] in SSL by combining labeled sample/label pairs and unlabeled sample/pseudotarget pairs.

Mixing augmentations present an interesting data augmentation paradigm in that

they can be used to create new samples, something particularly appealing in low label settings. However, most mixing augmentations tend to mix semantic content from both parent samples which leads to the creation of between class samples (Fig. 1a). In fact, this justifies the use of soft targets in mixing augmentations and explains their regularizing effect observed both empirically and theoretically [13, 14, 15]. While this is a desirable effect, this prevents the generation of "true" samples.

We propose in this paper to create artificial labeled samples that only inherit the label of one parent through mixing (Fig. 1b). This could be used to expand the limited pool of labeled samples in semi-supervised learning. By mixing the semantic content from available samples with the non-semantic content (or "context") of unlabeled samples, such an augmentation method could help further leverage the many unlabeled samples provided in semi-supervised learning.

The main challenge when performing such mixing lies in the proper separation of semantic and non-semantic content. We propose a novel neural architecture that separates input information into semantic information useful to a classifier, and auxiliary information necessary for reconstruction. Furthermore, our SAMOSA framework leverages its novel asymmetrical decoder (inspired by work in generative modeling and edition [16, 17]) to mix any two extracted semantic and non-semantic content. Fig. 1b shows how SAMOSA combines a bird picture with color tones from a plane picture.

We develop three main contributions in this paper 1) A novel learning scheme and architecture - SAMOSA - that separates semantic components from non semantic components in inputs and can be grafted on top of most pre-existing SSL methods to improve classifiers 2) A new mixing data augmentation for Semi-Supervised Learning that can mix the semantic content of labeled samples with non-semantic information of unlabeled samples 3) A thorough experimental validation of how the methods developed in this paper can be used to improve three well established Semi Supervised Learning algorithms: Mean Teacher [1], MixMatch [2] and FixMatch [3].

After discussing the relevant literature in Semi-Supervised Learning and Data Augmentation (Sec. 2), we introduce our SAMOSA Framework and elaborate on how it can be used in Semi-Supervised Learning (Sec. 3). Finally, we validate experimentally the performance of our SAMOSA Framework in Sec. 4

Notations. We refer in this paper to neural networks as capitalized letters (e.g. C for a classifier). Layers in a neural network are noted using an exponent (e.g. $C^{(i)}$ refers to ith layer of C), successive layers are discussed using a range exponent (e.g. $C^{(i..j)}$ refers to the ith to jth layers of C). In general, samples are named using letter x, labels using letter y and z refers to latent variables. The $\odot$ operator refers to Hadamard product, $\circ$ is used to denote function composition.

2. Semi-Supervised Learning and Hybridization

We first introduce the semi-supervised problem before providing a quick overview of the relevant literature. In semi-supervised learning, the dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ contains two sub-datasets: a labeled $\mathcal{D}_l$ dataset and an unlabeled $\mathcal{D}_u$ dataset. At the core of semi-supervised learning therefore lies the question of how to find ways to leverage $\mathcal{D}_u$ to extract information relevant to the task of interest.

This has been achieved in a number of ways, ranging from generative modeling [18] to graph based methods [19, 20]. While generative models have mostly relied on unsupervised generative training [18], label propagation methods have been used in a number of graph based methods to infer labels for unlabeled data [19]. In general, pseudo-labeling [19, 3] has been used to guess labels for unlabeled samples [21, 22, 23], often paired with entropy minimization techniques to improve model confidence [2]. Recent advances in self-supervised learning are often hijacked for Semi-Supervised Learning by jointly performing supervised training with labeled samples and self-supervised learning on unlabeled samples [24, 22] However, deep networks have most notably leveraged unlabeled data through the use of consistency losses [1, 8] that stabilize network predictions. Early solutions explored perturbing input samples [8], model predictions [8] or even the model itself [1]. More recently, more disruptive mixing data augmentation techniques have proven very effective for consistency based training [2, 10, 12].

Mixing data augmentations. [13] introduced the idea of mixing content from two samples x_1 and x_2 to generate new samples to train a classifier on. In the original work, pairs of samples are drawn and a new sample as a linear interpolation between the pixel

values of the parent samples according to a certain ratio. Formally, given two sample/label pairs x_1, y_1, x_2, y_2, MixUp generates a new sample $x' = \lambda x_1 + (1 - \lambda)x_2$ and label $y' = \lambda y_1 + (1 - \lambda)y_2$ for some $\lambda \in [0, 1]$ drawn from a symmetric beta distribution.

Subsequent work on mixing augmentations has mostly focused on two main aspects: how the mixing is performed (in terms of mathematical operations) and on what features the mixing is performed (e.g. pixels vs. model features). CutMix took inspiration from another recent augmentation method - CutOut - by drawing a rectangular mask $M \in \{0; 1\}^{H \times W}$ with values 1 inside a rectangle zone and 0. Parent samples are combined by CutMix as $x' = Mx_1 + (1 - M)x_2$. Put simply, a small patch from x_2 is pasted onto x_1. This formalism has been further extended with more complex masks [25] and the use of saliency maps to guide mask selection [26].

[27] proposed performing MixUp between intermediate representations of a classifier to ensure both parent sample provide relevant content. By embedding x_1 and x_2 in the latent space of a classifier before mixing, [27] observed significant gains in the regularizing ability of mixing augmentations. This line of attack has since then been extended to use more complex mixing operations [28].

In both cases, the focus of the subsequent work has clearly leaned towards ensuring semantic content from both parent samples is present in the mixed sample. Conversely, our method provides an alternative tool where we can mix the semantic content of one sample with the non-semantic content from another. Mixed samples generated from this method could be especially useful in semi-supervised learning, as traditional mixed samples have already been used to great effect in semi-supervised learning. Concurrently to this work, SciMix [29] proposed to mix semantic and non-semantic contents but introduced explicit hybridization objectives contrarily to our approach which lets non-semantic influences emerge on their own.

Mixing augmentations in Semi-Supervised learning. MixUp, along with other mixing augmentation methods, has been used to mix labeled samples from $\mathcal{D}_l$ and unlabeled samples from $\mathcal{D}_u$ [2, 10, 12]. In general, existing approaches have relied on generating pseudo targets for unlabeled samples before performing MixUp using the inferred

pseudo targets. Interpolation consistency training [12] extend the notion of consistency targets by using MixUp on the consistency targets and inputs. MixMatch and ReMixMatch [2, 10] further extended this idea by considering pseudo-labels for unlabeled samples instead of consistency targets.

Building upon the idea of mixing samples, we propose in this paper to work on "hybrid" samples that associate semantic content of one (labeled) sample and auxiliary non semantic characteristics of another (Fig. 1b), as well as directly transfer the one-hot label target (0 1). This is in contrast to MixUp which would simply average entire input images (Fig. 1a for $\lambda = 0.5$). This however requires being able to separate semantic content from auxiliary non semantic content.

Separating Semantic Information from Irrelevant Information. Input reconstruction or generation has been used to leverage unlabeled data [30, 18] to improve the features of classifiers. [30] however points out that classifiers aim to be invariant to non-semantic information that would be required for accurate reconstruction. If, for instance, we sought to classify images of numbers on colored backgrounds, color would be required to reconstruct the images but would be superfluous noise to a classifier. Separation of semantic and non-semantic information in [30] fails to truly differentiate between the two modalities, which we address in this work through the use of a special decoder architecture.

This idea of separating semantic from non-semantic content has previously been studied in multiple domains such as Domain Generalization [31, 32] (where a model is trained to transfer well to any new domain) or unsupervised image to image translation [33, 34] (where a model learns to modify pictures to take on new characteristics in an unsupervised fashion). In most cases however, non-semantic information is either simply discarded in the case of Domain Generalization or treated mostly as a general content vs. domain issue in unsupervised image to image translation. Like in Domain Generalization techniques, we need to isolate class specific information but we also need to re-synthesize hybrid images like in image-to-image translation where simpler concepts are usually manipulated. This means no easy parallel can be drawn for such techniques in Semi-Supervised Learning: we cannot separate unlabeled data into

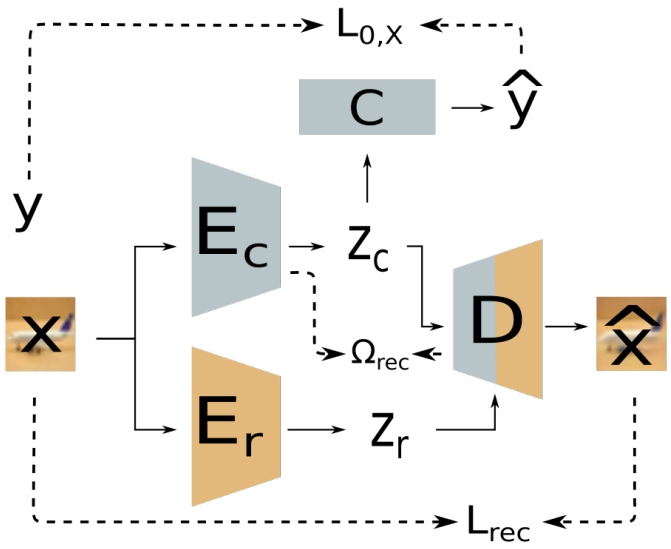


Figure 2: Overview of the SAMOSA framework. z_c and z_r are extracted from input x. z_c alone is used to classify the input (to optimize the base $\mathcal{L}_{0,X}$ loss). $\hat{x}$ (which reconstructs x due to Ω_{SAMOSA}'s sub-loss $\mathcal{L}_{rec}$) is computed from both extracted features z_c and z_r. E_c is regularized by the learned decoder (through the Ω_{rec} regularizer of Ω_{SAMOSA})

different domains like in Domain Generalization and there is absolutely no semantic notion to work with in unsupervised image-to-image translation frameworks.

Our work therefore aims to create an encoder decoder system that separates mostly independent semantic and non-semantic component. In particular, we seek to leverage such a disentangled hybrid generation process for semantically consistent mixing augmentation.

3. SAMOSA

We detail in this section our proposed SAMOSA Framework. After a brief overview of the general framework, we give a detailed account of our novel asymmetrical decoder in Sec. 3.1 and detail SAMOSA's atypical learning scheme in Sec. 3.2. Finally, we discuss how our SAMOSA framework can be used in a SSL setting in Sec. 3.3.

First and foremost, we introduce in this paper a novel architecture presented in Fig. 2. It is composed of two encoders E_c and E_r (one semantic - with regards to the

classification process - and one non semantic), a simple classifier C and a bi-modal decoder D that takes inputs from a semantic modality z_c and a non-semantic modality z_r. SAMOSA is meant to be added on top of existing semi-supervised learning algorithms for neural architectures. In this sense, an input x is mapped to a feature representation $z_c = E_c(x)$, which is then used to obtain a classifier prediction $\hat{y} = C(z_c) = C(E_c(x))$. We further elaborate on the peculiarities of our additional reconstruction modules E_r and D in Sec. 3.1.

To train such an architecture, we optimize the modules necessary for classification (E_c and C) to minimize a two component loss $\mathcal{L}_{SAMOSA}$ (Eq. 1, Fig. 2). Our novel regularizer Ω_{SAMOSA} (in Eq. 1) differs substantially from standard reconstruction regularizers by leveraging peculiarities of SAMOSA's architecture. On the other hand, the base loss $\mathcal{L}_{0,X}$ term in Eq. 1 acts as a proxy to represent the base method (which we seek to improve) "X"'s training process (see Fig. 2). For instance, the three base losses we considered are $\mathcal{L}_{0,MT}$ from Mean Teacher [1], $\mathcal{L}_{0,Mix}$ from MixMatch [2] and $\mathcal{L}_{0,Fix}$ from FixMatch [3]. As the basic algorithms we consider are meant to function on standard classifier models, $\mathcal{L}_{0,X}$ is only minimized for $E_c \circ C$. The training and manipulation of the remaining modules as well as the regularizer term Ω_{SAMOSA} are specific to SAMOSA, and are elaborated upon in Sec. 3.2.

$$\begin{aligned}\mathcal{L}_{SAMOSA}(\{x_l, y_l\}_{\mathcal{D}_l} \cup \{x_u\}_{\mathcal{D}_u}) = \; & \mathcal{L}_{0,X}(\{x_l, y_l\} \cup \{x_u\}) \\ & + \Omega_{SAMOSA}(\{x_l\} \cup \{x_u\}).\end{aligned} \tag{1}$$

As such, the method trains a standard classifier $E_c \circ C$ according to a base SSL method X with loss $\mathcal{L}_{0,X}$. Our contribution consists in adding a reconstruction regularizer Ω_{SAMOSA}, a non-semantic encoder E_r and a special bi-modal decoder D to be optimized and trained simultaneously with the base SSL classifier. The bi-modal decoder in particular requires careful design to mix semantic and non-semantic content.

3.1. Adding a Non-Supervised Reconstruction Module

Our goal is to mix the semantic content of one sample with the non-semantic content of another. This requires both separating the two contents from samples and reconstructing from those contents in a modular fashion. To some extent, this has been

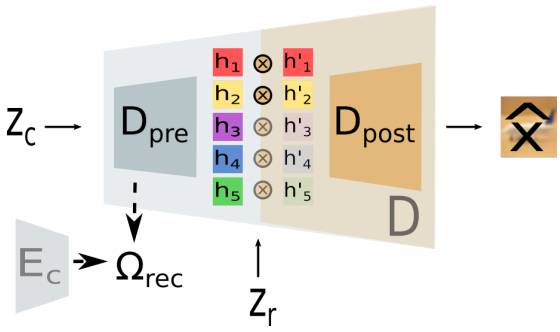


Figure 3: Our proposed asymmetrical decoder D reconstructs x from z_c, with z_r modulating which parts of D are active.

achieved in style transfer [16] and generative modeling [17]. [16] and [17] for instance have shown manipulating activation statistics of intermediate activation maps in an autoencoder can be used to train a model capable of reconstructing an input image in a number of different ways. Those methods however either explicitly define what "style" (which we liken to non semantic information) is through a specifically designed loss functions and targets [16], or perform adversarial optimization that does not allow for specific reconstructions [17]. We propose here an architecture operating along similar principles, but that can reconstruct inputs without any pre-conception of what constitutes non-semantic information.

We retain the base model's E_c as our semantic encoder, and add a separate encoder E_r for the remaining non-semantic information. A novel asymmetrical bi-modal decoder D is then used to reconstruct the input images from the outputs of the encoders E_c and E_r. Practically, an input x is mapped to an additional non semantic feature representation $z_r = E_r(x)$, which is then used in conjunction with z_c to obtain a reconstructed image $\hat{x} = D(z_c, z_r) = D(E_c(x), E_r(x))$. This last reconstruction process is facilitated by the very peculiar structure of D.

Asymmetrical decoder D. Crucially, we design a novel decoder module (Fig. 3) to combine semantic and non semantic feature spaces. An immediate concern when reconstructing from two latent spaces as we propose is that an unconstrained non semantic feature space is liable to store all the necessary information to reconstruct the

input, thereby leaving a decoder free to ignore the semantic feature space z_c. Previous work [30] ran into this issue when generating two partial reconstructions - one from semantic features and one from non-semantic features - and summing the two to obtain a complete reconstruction. This was addressed by forcefully stopping gradient flows of one partial reconstruction right before combination (depending on which partial reconstruction needs more training). However, this method led to both E_c and E_r each contributing very similar information as this process only ensures the two modalities contribute to the reconstruction. Conversely, we design an asymmetrical decoder that uses the two input modalities differently.

To prevent z_r from encoding all the information, we shift its role from affecting *what* is on the reconstruction to affecting *how* the semantic latent space z_c is translated to a reconstruction. As figured in Fig. 3, D can be broken down into two sub decoders D_{pre} and D_{post} such that $h = D_{pre}(z_c) \in \mathbb{R}^{S \times H \times W}$ can be construed as a stack of S intermediate reconstruction maps. z_r serves as a set of S gating weights $\in [0, 1]$ (through the use of a final linear projection and softmax activation) such that the final reconstruction $D_{post}(h')$ mostly relies on a few intermediate activation maps $h' = z_r \odot h$ (e.g. only the red and yellow maps remain in Fig. 3). While this can be seen as a rescaling of feature maps (like in style transfer) [16, 17, 34], the absence of style targets might lead to z_r selecting all maps with a method such as AdaIN. To address this, we ensure only a few maps are selected by z_r to contribute to the final reconstruction for each sample using a softmax activation (though no hard thresholding is applied). While it should be useful to have more intermediate maps to select from in theory, we find the method fairly robust in this regard (see Appendix. C.1).

This architecture allows us to reconstruct samples while avoiding the pitfall of forcing the classifier's feature extractor E_c to keep irrelevant information at minimal cost: computing the non-semantic attention weights requires computing z_r (equivalent to computing E_c) and performing a $\mathcal{O}(d \times S)$ linear projection which is negligible with respect to the overall model computation.Furthermore, we propose a learning scheme that pushes the semantic encoder E_c to leverage the decoder D to identify what it should keep track of through the second term Ω_{SAMOSA} of the loss given in Eq. 1.

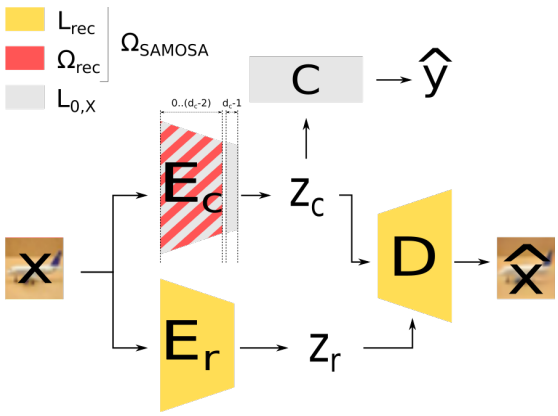


Figure 4: Optimization of $\mathcal{L}_{SAMOSA}$ in the modules. The base SSL loss $\mathcal{L}_{0,X}$ loss optimizes the classifier modules while the reconstruction loss $\mathcal{L}_{rec}$ optimizes the additional modules E_r and D. E_c is benefits from the reconstruction module through the Ω_{rec} regularizer

3.2. Learning Scheme

Model optimization. SAMOSA relies on a regularizer term Ω_{SAMOSA} to leverage its peculiar architecture:

$$\begin{aligned}\Omega_{SAMOSA}(\{x_l\}_{\mathcal{D}_l} \cup \{x_u\}_{\mathcal{D}_u}) =& \lambda_{rec}\mathcal{L}_{rec}(\{x_l\} \cup \{x_u\}) \\ &+ \lambda_{SAMOSA}\Omega_{rec}(\{x_l\} \cup \{x_u\}),\end{aligned} \tag{2}$$

with the term $\mathcal{L}_{rec}$ used to optimize E_r and D for reconstruction of inputs, and the auxiliary regularizer Ω_{rec} used to refine E_c through knowledge learned by D. This differs significantly from traditional work in SSL that uses reconstruction for regularization as we do not directly optimize the classifier for reconstruction. Rather, we leverage our asymmetrical decoder's peculiar structure to regularize the classifier so that it solely learns to reconstruct information identified as semantic by our framework.

$\mathcal{L}_{rec} = \frac{1}{\#\mathcal{D}} \sum_{x \in \mathcal{D}} \|D(E_c(x), E_r(x)) - x\|_2^2$ (figured on Fig. 4) tries to match inputs x to model reconstructions $D(E_c(x), E_r(x))$ through the L2 distance between the two. E_c is deliberately not optimized here as skip connections [4] in modern neural networks already let a lot of input information trickle down to their feature space. In our experiments, we found optimizing E_c for reconstruction led D to rely entirely on E_c and ignore E_r.

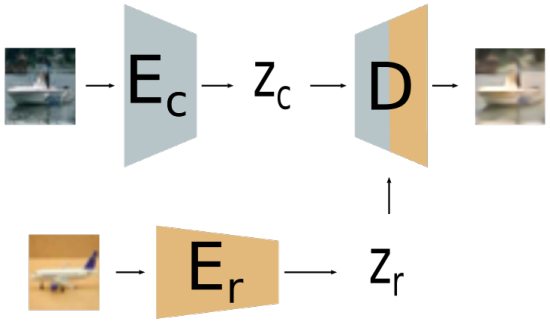


Figure 5: A trained model can then be used to combine semantic content from a boat picture and non semantic content from a plane picture.

$\Omega_{rec} = \frac{1}{\#\mathcal{D}} \sum_{x \in \mathcal{D}} \| E_c^{(0..(d_c-2))}(x) - D^{(0)}(E_c(x), E_r(x)) \|_2^2$ leverages our decoder's asymmetrical structure to regularize E_c (Fig. 4). Importantly, the first few intermediate reconstructions are purely semantic as they are prior to re-modulation by z_r (the style input $E_r(x)$ in $D(E_c(x), E_r(x))$ is of no effect). Therefore, training E_c to match these early decoder features provides a novel reconstruction regularizer for the feature extractor that is not polluted by non-semantic information (i.e. information injected by $E_r(x)$ in the reconstruction). In practice, Ω_{rec} ties the last intermediate features $E_c^{(0..(d_c-2))}(x)$ extracted by E_c (layer $E_c^{(d_c-2)}$) to the first intermediate reconstructions $D^{(0)}(E_c(x), E_r(x))$ generated by D (layer $D_{(0)}$). Here, d_c refers to the depth of E_c, $E_c^{(0..(d_c-2))}$ to the composition of the first $d_c - 1$ convolutional layers of E_c (all but the last one), and $D^{(0)}$ to the first convolutional layer of D. We tie the last intermediate features of E_c to the first intermediate reconstructions of D to regularize as much of the semantic encoder as possible. Which intermediate reconstruction precisely is used matters little as long as it provides a viable target to train the semantic encoder (see Appendix C.2).

This training process yields an architecture capable of generating hybrids that incorporates non-semantic content from a sample x_2 into a sample x_1 while preserving x_1's semantic content. We now discuss how this can be put to use in a Semi-Supervised Learning setting.

Algorithm 1 Algorithm for the hybridization procedure.

Require: Batch $\mathcal{B} = \mathcal{B}_l \cup \mathcal{B}_u$, Modules E_c, E_r, C, D

function HYBRIDIZE($\mathcal{B}_l, \mathcal{B}_u$)

$\mathcal{O} = \emptyset$

for $(x^{(1)}, y^{(1)}), x^{(2)} \in zip(\mathcal{B}_l, \mathcal{B}_u)$ **do**

$z_c^{(1)} = E_c(x^{(1)})$

$z_r^{(2)} = E_r(x^{(2)})$

$x_h = D(x^{(1)}, x^{(2)})$

if $argmax(C(E_c(x_h))) == y^{(1)}$ **then**

$x_h = x^{(1)}$

end if

$\mathcal{O} = \mathcal{O} \cup \{(x_h, y^{(1)})\}$

end for

return $\mathcal{O}$

end function

3.3. Making use of the SAMOSA framework in Semi-Supervised Learning

We introduce a novel asymmetrical decoder that is modular *by design* with regard to semantic and non semantic content, as well as propose an adapted training scheme. In practice, the learning scheme itself can be used to regularize classifiers, but the trained models can also be used to generate augmented samples to train models on. For instance, a model could be trained to optimize $\mathcal{L}_{SAMOSA}$, then used to generate a set of artificial labeled samples through SAMOSA hybridization and the model could then be re-trained on the augmented dataset.

Algorithm 2 Skeleton of SAMOSA integration with the Mean Teacher Framework. Additions to the Mean Teacher Framework are in blue.

Require: Dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_l$, Number of complete training cycles n_{cycles}, Number of epochs in a cycle n_{epochs}.

for cycle in 1...n_cycles **do**
  $\mathcal{D}_{l,0} = \mathcal{D}_l$
  for epoch in 1...n_epochs **do**
    for $\mathcal{B} = \mathcal{B}_l \cup \mathcal{B}_u$ in Batch($\mathcal{D}$) **do**
      Compute $\mathcal{L}_{SAMOSA}(\mathcal{B}) = \mathcal{L}_{0,MT}(\mathcal{B}) + \Omega_{SAMOSA}(\mathcal{B})$
      Optimization step for $\mathcal{L}_{SAMOSA}(\mathcal{B})$
    end for
  end for
  for epoch in 1...10 **do**
    $\mathcal{D}_h = \emptyset$
    for $\mathcal{B}_l, \mathcal{B}_u$ in zip(Batch($\mathcal{D}_{l,0}$), Batch($\mathcal{D}_u$)) **do**
      $\mathcal{O} = Hybridize(\mathcal{B}_l, \mathcal{B}_u)$
      $\mathcal{D}_h = \mathcal{D}_h \cup \mathcal{O}$
    end for
  end for
  $\mathcal{D}_l = \mathcal{D}_{l,0} \cup \mathcal{D}_h$
end for

Indeed, generating hybrids given a trained model is straightforward (Alg. 1 and Fig. 5). Specifically, given samples $x^{(1)}$ (with known label $y^{(1)}$) and $x^{(2)}$, we extract the relevant features $z_c^{(1)} = E_c(x^{(1)})$, $z_r^{(1)} = E_r(x^{(1)})$, $z_c^{(2)} = E_c(x^{(2)})$ and $z_r^{(2)} = E_r(x^{(2)})$. $x_h = D(z_c^{(1)}, z_r^{(2)})$ is now a sample with class $y^{(1)}$. As a conservative measure, we only keep the generated hybrid if $C(E_c(x_h)) = y^{(1)}$ to avoid disturbing decision boundaries too much. Note that with this, we generate a strong augmentation of x_1 and teach the classifier to group x_1 with its strongly augmented version in a similar line to work in contrastive representation learning [35].

Algorithm 3 Skeleton of SAMOSA integration with the MixMatch Framework. Additions to the MixMatch Framework are in blue.

Require: Dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_l$, Number of epochs during training n_{epochs}

$\mathcal{D}_{l,0} = \mathcal{D}_l$

for epoch in 1...n_epochs **do**

  for $\mathcal{B} = \mathcal{B}_l \cup \mathcal{B}_u$ in Batch($\mathcal{D}$) **do**

    $\mathcal{X}_l, \mathcal{Y}_l := \mathcal{B}_l$

    $\mathcal{X}_u := \mathcal{B}_u$

    Estimate pseudo-targets $\mathcal{Y}_u$ as per [2]

    $\mathcal{W} := Concat(\{(x_l, y_l)\}, \{x_u, y_u\})$

    $\tilde{\mathcal{W}} := Shuffle(\mathcal{W})$

    $p \sim Random(0, 1)$

    if $p < \frac{1}{5}$ **then**

      $\tilde{\mathcal{B}} = Hybridize(\mathcal{W}, \tilde{\mathcal{W}})$

    else

      $\tilde{\mathcal{B}} = MixUp_{biased}(\mathcal{W}, \tilde{\mathcal{W}})$

    end if

    Compute $\mathcal{L}_{SAMOSA}(\tilde{\mathcal{B}}) = \mathcal{L}_{0,Mix}(\tilde{\mathcal{B}}) + \Omega_{SAMOSA}(\tilde{\mathcal{B}})$

    Optimization step for $\mathcal{L}_{SAMOSA}(\tilde{\mathcal{B}})$

  end for

end for

As previously discussed, SAMOSA can be deployed in SSL systems in a variety of ways, two of which are explored experimentally paper. We study a first framework that trains a SSL model to optimize $\mathcal{L}_{SAMOSA}$, generates hybrids using labeled samples for the semantic component and unlabeled samples for the non-semantic component, and re-trains the model on the augmented set (Alg. 2). We also show a more intricate incorporation of SAMOSA in the MixMatch framework (Alg. 3) by occasionally replacing the MixUp procedure with our in-class hybridization in the training of a MixMatch model optimizing $\mathcal{L}_{SAMOSA}$. Furthermore, we also incorporate SAMOSA into the state of the art FixMatch framework (Alg. 4) by sometimes hybridizing the strongly

augmented samples used by FixMatch in the training of a FixMatch model optimizing $\mathcal{L}_{SAMOSA}$.

Algorithm 4 Skeleton of SAMOSA integration with the FixMatch Framework. Additions to the FixMatch Framework are in blue.

Require: Dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_l$, Number of epochs during training n_{epochs}

$\mathcal{D}_{l,0} = \mathcal{D}_l$

for epoch in 1...n_epochs **do**

for $\mathcal{B} = \mathcal{B}_l \cup \mathcal{B}_u$ in Batch($\mathcal{D}$) **do**

Get labeled samples $\mathcal{X}_l^w, \mathcal{Y}_l := \mathcal{B}_l$

Get weakly and strongly augmented pairs $\mathcal{X}_u^w, \mathcal{X}_u^s := \mathcal{B}_u$

Estimate pseudo-labels $\mathcal{Y}_u$ as per [3]

$\tilde{\mathcal{X}}_u^s := Shuffle(\mathcal{X}_u^s)$

$p \sim Random(0, 1)$

if $p < \frac{1}{5}$ **then**

$\mathcal{X}_u^s = Hybridize(\mathcal{X}_u^w, \tilde{\mathcal{X}}_u^s)$

end if

Compute $\mathcal{L}_{SAMOSA}(\mathcal{B}) = \mathcal{L}_{0,Fix}(\tilde{\mathcal{X}}_u^s, \mathcal{Y}_u) + \Omega_{SAMOSA}(\mathcal{B})$

Optimization step for $\mathcal{L}_{SAMOSA}(\mathcal{B})$

end for

end for

4. Experiments

We demonstrate here how the disentangling reconstruction module and resulting hybridization capabilities can be leveraged to improve upon two existing methods: Mean Teacher [1], MixMatch [2] and FixMatch [3] (refer to Sec. 4.1 for how we apply SAMOSA to these methods). We chose Mean Teacher as a reference pure consistency-based baseline. Beyond its widespread use in SSL, consistency induces a stabilization we feel would play a significant role to extract invariant semantic features. MixMatch was chosen to illustrate interactions of the method with more modern methods that make use of mixing techniques (such as CutMix, CowMix, ICT and ReMixMatch).

FixMatch finally provides an insight into how SAMOSA can interact with strong data augmentations that apply very destructive perturbations to images, and most state of the art [21, 22, 23] methods are based upon FixMatch to this day. We conduct experiments on the CIFAR10 dataset, which is a standard evaluation benchmarks in the semi-supervised learning literature. We also conduct additional experiments for the Mean Teacher backbone of SAMOSA on the SVHN dataset with very few labels and the more complex CIFAR100 dataset.

CIFAR10 dataset. The CIFAR10 dataset [36] is a subset of the TinyImages dataset comprised 32×32 RGB images from ten classes: airplane, car, truck, boat, bird, cat, deer, dog, frog and horse. Available samples are split between 50000 training samples and 10000 test samples. We mainly keep 1000 labeled training samples for our ablation studies and to compare model performances in general. In addition, we consider an intermediately difficult setting with 500 labeled samples, and perform an ablation study on a difficult 250 labeled samples setting for the Mean Teacher based SAMOSA.

SVHN dataset. The SVHN dataset [37] is comprised of 32×32 RGB images of street numbers (divided along ten classes: one per digit). Available samples are split between 73257 training samples and 26032 test images. We use the SVHN dataset to study the very challenging setting where only 100 labeled samples are available, which is unadvisable on CIFAR10 due to the very uneven quality of labeled examples in CIFAR10.

Samples are randomly flipped horizontally (only for CIFAR10) and shifted by up to 4 pixels both horizontally and vertically with reflect padding. The resulting augmented samples are then standardized channel-wise according to train set statistics. No holdout validation set is kept for either dataset but hyper-parameters are mostly directly adapted from [30, 2].

CIFAR100 dataset. The CIFAR100 dataset [36] (like CIFAR10) is a subset of the TinyImages dataset comprised 32×32 RGB images from a hundred classes. Available samples are split between 50000 training samples and 10000 test samples. This more complex dataset allows us to further study the behavior of SAMOSA in situations where

there are many different semantic classes to keep track of.

Experimental Setting. We operate on a standard WideResNet-28-2 [38] which is widely used in the Semi-Supervised Learning literature as a base model ($E_c \circ C$). E_r follows the same architecture as E_c with an additional final linear layer and softmax activation to obtain activation gates. The skeleton of D follows an inverted 13-layer 4-4-4 CNN architecture, with D_{pre} being a 4-4 block and D_{post} being made up of the last 4 block and final convolution. Hyperparameters and optimizers were generally taken to follow settings reported in the base methods' original papers [1, 2]. More details are briefly provided in the relevant incremental gains section, but exact details are provided in the supplementary material for each experiment and architecture.

We evaluate performance through classification accuracy. Results are presented as $\mu \pm \sigma$, with μ the average value and σ the standard deviation across three seeded runs (random initializations). All results presented are run from the same code base and computation servers as per [39]. In particular, the same three initializations were used for all methods. For better comparison, we perform paired one-sided t-test to evaluate improvements brought by SAMOSA and bold results where $p \leq 0.1$ as an indication (full results provided in the supplementary) in the following.

4.1. SAMOSA Gains

We show here that SAMOSA can improve performance when added to Mean Teacher, MixMatch (Tab. 1) and FixMatch (discussed below). In each table, we also report for reference the accuracy of models trained in a purely supervised fashion on the available labeled samples as a lower bound.

Mean Teacher (MT). We evaluate a first application of SAMOSA for augmentation to show improvements on Mean Teacher (the procedure is detailed in Alg. 2). We train the model normally for 300 epochs (with the reconstruction module), then we hybridize every labeled sample with 10 unlabeled samples. For every generated hybrid, we keep the artificial example only if it still gets predicted by the model as being part of the

[1]Reproduced by adapting available code (see Supplementary)

Method	CIFAR10		
	250	500	1000
Purely supervised (lower bound)	27.8 ± 0.9	35.4 ± 1.9	43.5 ± 2.4
Mean Teacher[1], [1]	61.3 ± 3.3	76.4 ± 3.1	87.6 ± 0.3
Mean Teacher + SAMOSA (ours)	$\mathbf{68.1 \pm 3.3}$	$\mathbf{82.4 \pm 1.3}$	$\mathbf{88.7 \pm 0.3}$
MixMatch[12], [2]	$\mathbf{82.4 \pm 0.4}$	86.8 ± 0.2	$\mathbf{90.4 \pm 0.1}$
MixMatch + SAMOSA (ours)	$\mathbf{84.1 \pm 2.2}$	$\mathbf{89.4 \pm 0.8}$	$\mathbf{90.7 \pm 0.3}$

Table 1: Comparative accuracies (%) with SAMOSA as an add-on module on CIFAR10.

right class, otherwise the hybrid is replaced by its semantic "parent". The model is then retrained with this additional labeled data over 300 epochs. Afterwards, another hybridization procedure is repeated and a final training is conducted, still over the same number of epochs.

The model is trained using a SGD optimizer with cosine learning rate (base 0.2 learning rate) for 300 epochs over the unlabeled samples for CIFAR10 for one training cycle. After each training and subsequent augmentation step, the learning rate is reset and training resumes (for an overall 900 epochs). In the following training passes, the model is only optimized over the augmented dataset (instead of the true dataset) from epoch 150 to 250 of each cycle (following discussions in [40, 41]) with $\lambda_{recons} = 0.25$ and $\lambda_{SAMOSA} = 0.5/0.1$. Exact details in the supplementary material.

As a baseline, we check the performance of the model trained under the same procedure (3 training cycles) but with no reconstruction regularizer, and no artificial samples. Tab. 1 shows improvements from using the SAMOSA framework on top of Mean Teacher. Notably, we have very noticeable gains for 250 labels, which suggests the method is particularly useful when labeled information is lacking. To explore performance with very few labels, we furthermore tested the model with 100 labels on SVHN. An important accuracy gain from 62.5 ± 3.7 to $\mathbf{66.2 \pm 2.2}$ is observed which is significant given such very low label settings are especially interesting in applied settings. The same gains are also reliably observed on the more complex CIFAR100 datasets where SAMOSA improves the Mean Teacher baseline from 24.9 ± 0.7 to $\mathbf{27.6 \pm 2.3}$

with 1000 labels (and from 48.1 ± 0.8 to $\mathbf{51.0 \pm 0.5}$ with 2500 labels).

MixMatch (Mix). We showcase a more intricate use of SAMOSA on MixMatch by directly incorporating our augmentation process in MixMatch's native hybridization (MixUp) as detailed in Alg. 3. We train the reconstruction module along the base classifier model, as well as optimize for the reconstruction regularizer. Every batch, with probability $p = 0.2$, we replace the MixUp examples with Hybrids generated from our reconstruction module. For every reconstructed hybrid, we keep the label/pseudolabel corresponding to its semantic "parent". Contrarily to the Mean Teacher case, we generate hybrids that have both labeled and unlabeled samples as semantic "parents" (as is done by MixUp in MixMatch) and leverage the pseudo-label MixMatch naturally generates throughout its course for MixUp. Exact details are given in the supplementary material, but are similar to the Mean Teacher ones. We follow [2] and report results from a weight averaged model.

As a baseline comparison, we check the performance of the model trained under the same procedure (which is basically the normal training procedure) but with no reconstruction regularizer, and no artificial samples. As can be seen from Tab. 1, sizable gains are achieved on both the 500 and 250 labels CIFAR10 settings, and are consistent even when 3 runs are not enough to definitely verify improvements. Interestingly, adding SAMOSA to MixMatch increases the variance of the model which is not the case in the Mean Teacher case. The use of MixUp hybrids in MixMatch strongly influences the hybrids generated by SAMOSA (discussed in Sec. 4.3). This and the random nature of MixUp could lead to a stronger variability in the quality of hybrids learned by a SAMOSA generator. Considering how reliant SAMOSA's classifier is on the quality of the generated hybrids for regularization, we believe this explains the higher variance on MixMatch + SAMOSA.

FixMatch (Fix). We additionally demonstrate SAMOSA can combine with strong data augmentation techniques by also studying a FixMatch based version of SAMOSA as detailed in Alg. 4. We train the reconstruction module along the base classifier model,

[2] Different setting from [2] for fast training.

as well as optimize for the reconstruction regularizer. Every batch, with probability $p = 0.2$, we further perturb the strongly augmented samples in FixMatch by hybridizing them through our reconstruction module. For every generated "strong" hybrid, we keep the label/pseudolabel corresponding to its semantic "parent". We follow [2] and report results from a weight averaged model.

As a baseline comparison, we check the performance of the model trained under the same procedure (which is basically the normal training procedure) but with no reconstruction regularizer, and no artificial samples on a very challenging low label setting (CIFAR 10 with 100 labels). The newly described Algorithm 4 leads to significant gains on FixMatch (from 91.5 ± 0.6 to $\mathbf{92.1 \pm 0.2}$) which demonstrates SAMOSA can combine a state-of-the-art framework based on strong data augmentation and pseudo-labeling.

With SAMOSA's ability to improve existing methods like Mean Teacher, MixMatch and FixMatch well established, we now study the individual relevance of its internal components.

4.2. Ablation Study: General Components of SAMOSA

We now validate our two main contributions on CIFAR 10, ie the incorporation of a reconstruction module that allows mixing semantic and non semantic information from different samples, and the use of hybrid samples as data augmentation to further refine model features. We mainly study the ablations on challenging settings so that performance gains can be as clear as possible. As such, we consider the 500 label setting on CIFAR10 for both the Mean Teacher backbone and the MixMatch backbone (due to high MixMatch variance at 250 labels).

Results from Tab. 2 show that both the reconstruction regularization loss and the augmented hybrids provide significant gains in accuracy. Moreover, the best performance is attained when stacking the two components. Interestingly, the gain from using the regularizer Ω_{rec} is the greatest influence for both experiments. This shows the relevance of the reconstruction scheme to regularize training, which is especially pronounced in low data regimes. Nevertheless, improvements can consistently be observed from adding augmented samples to the regularized model. Interestingly, opti-

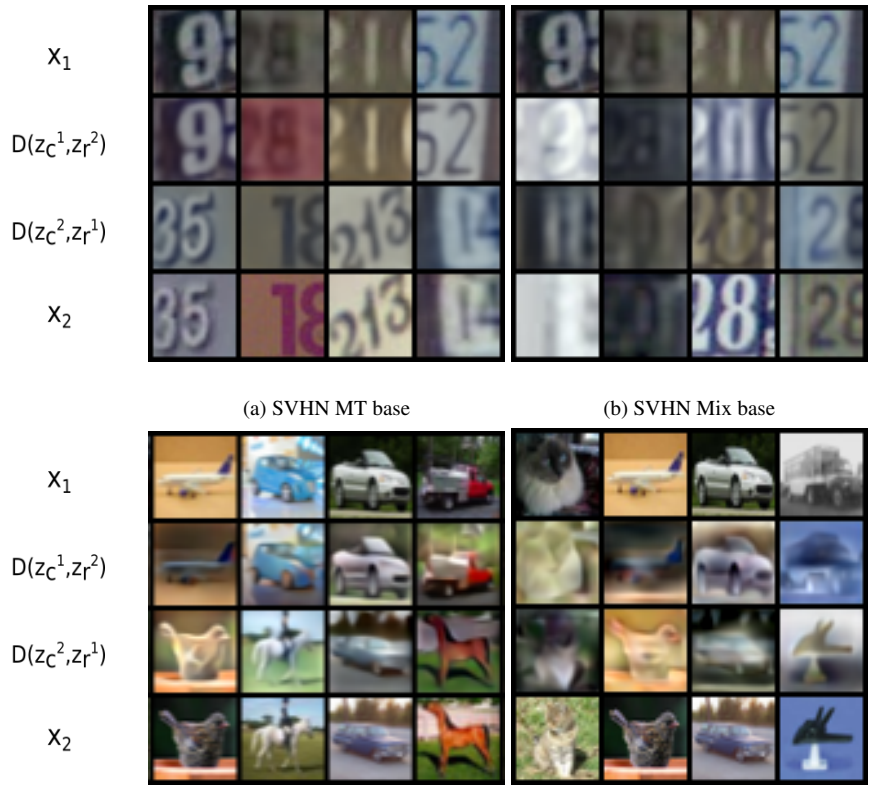
$\mathcal{L}_0$	Ω_{rec}	Aug	Mean Teacher	MixMatch
			500	500
✓	✗	✗	76.4 ± 3.1	86.8 ± 0.2
✓	✓	✗	82.1 ± 1.5	88.5 ± 1.2
✓	✓	✓	$\mathbf{82.4 \pm 1.3}$	$\mathbf{89.4 \pm 0.8}$

Table 2: Ablation Study on the components of SAMOSA. Accuracies (%) on the CIFAR10 settings for both studied backbones

mizing E_c with $\mathcal{L}_{rec}$ leads to hybrids identical to the semantic parent and poor accuracy of the trained classifier.

4.3. Qualitative Study: Generated Hybrids

We show in Fig. 6 hybrids generated by the method. As can be observed, SAMOSA learns to isolate a variety of visually identifiable non semantic characteristics (number colors, light exposition, color hues, some irrelevant colorations) without any more supervision than simple L2 reconstruction. Interestingly, the MixMatch based variant is more aggressive in combining samples (the background in particular) while still preserving the outline of the relevant semantic content. This can be attributed to interplay between the MixUp procedure and our hybridization procedure, and suggests there is indeed complementarity between our method and other mixing augmentation methods. Figures presented in the supplementary material suggest SAMOSA's ability to identify such characteristics is correlated with model performance, which can account for the lack of strong effect of hybrid augmentation in the low label CIFAR10 setting. This actually reinforces the intuition behind SAMOSA: the non semantic encoder picks up redundancies discarded by the semantic encoder. When the classifier's accuracy is not very high, it fails to discard redundant information which leads to little dependence on z_r. While this means the model will not benefit from SAMOSA, it is fortunately unlikely this will significantly deteriorate its performance. Indeed, such failure cases tend to result in hybrids being complete reconstructions of their semantic parent and therefore have no effect.



(a) SVHN MT base (b) SVHN Mix base

(c) CIFAR10 MT base (d) CIFAR10 Mix base

Figure 6: Hybrids between true samples x_1 and x_2. Results for both Mean-Teacher and MixMatch based SAMOSA trained on SVHN (100 labels) and CIFAR10 (1000 labels).

In addition to verifying z_r does indeed cause non-semantic changes in reconstruction/hybridization, we also observe that it does not contain a lot of semantic information. Training a classification head (with all 50000 labeled training samples) on the non-semantic space of a trained MT-SAMOSA model (1000 label CIFAR 10 setting) does lead to low classification accuracy (about 30%). This contrasts with the 88% accuracy obtained by a linear layer trained on the semantic space z_c with only 1000 labeled samples in previous experiments. As such, we verify that z_r does not extract specifically semantic features in accordance with SAMOSA's design. What little semantic information remains in z_r is ignored by the decoder D as generated hybrids only inherit the class of their non-semantic parents in about 10% of cases (random chance). Furthermore, we can verify that expunging semantic content from z_r is pointless. z_r can be made wholly non-semantic by training a linear classifier to classify from z_r, and training E_r to fool this classifier. While a model trained this way retains almost no semantic information in non-semantic space z_r (about 18% accuracy for a linear classifier trained on the frozen projection), such a model fails to better classify samples (accuracy of 88.5 ± 0.1 vs. 88.7 ± 0.3).

To verify these intuitions, we investigate this from a more quantitative point of view.

4.4. Case Study: Component Separation

We now investigate the composition of generated hybrids on reduced settings (MT Base) to verify the model's ability to generate hybrids that correctly inherit their parent's semantic and non-semantic components. To this end, at various points during training, we generate a study dataset of hybrid samples $\mathcal{D}_H$. The dataset is generated by mixing every sample in the labeled set with ten random unlabeled samples such that $\#\mathcal{D}_H = 10 \times \#\mathcal{D}_l$.

Inheritance of semantic and non-semantic features. We start by assessing how well generated hybrids inherit semantic/non-semantic features with respect to our model's learned projections. The quality of the inherited semantic component can be approximated straightforwardly by considering the accuracy s_c of our trained classifier on

hybrids (ie, checking how many hybrids are correctly classified as belonging to the same class as their semantic parent). As we do not have access to such a clear criterion for the non-semantic component, we use a proxy metric in the non-semantic latent space. We consider the distance $d_l := \|z_r^h - z_r^1\|_2^2$ (resp. $d_r := \|z_r^h - z_r^2\|_2^2$) between the extracted non-semantic feature $z_r^h = E_r(x_h)$ of a hybrid x_h and those of its semantic parent z_r^1 (resp. non-semantic parent z_r^2). If $d_l \geq d_r$, then we conclude the hybrid correctly inherited its non-semantic parent's style component. As such, we can define a non-semantic separation accuracy s_r by the proportion of hybrids in $\mathcal{D}_H$ correctly identified as being closer to their non-semantic parent. In other words, we monitor whether the hybrid's non-semantic content is indeed closer to its non-semantic parent's.

The accuracy of the semantic and non-semantic separation tasks are presented in (Tab. 3a) along with the average distances in non-semantic space to the hybrid's parent samples d_l and d_r at the end of training. On both datasets, we can observe that hybrids mostly inherit the correct semantic and non semantic characteristics at the end of training. In particular, non-semantic features of hybrids are about 10 times closer to their non-semantic parents' compared to their semantic parents'.

Importantly, the observed inheritance of semantic/non-semantic features significantly improves over the course of the entire training. For instance, with 1000 labels on CIFAR10, semantic accuracy s_c on generated hybrids at the end of the first training cycle (300 epochs, no hybrid augmentation yet) is 74.1 ± 2.5, 93.0 ± 2.5 at the end of the second (trained with hybrid augmentation) and 97.3 ± 0.6 at the end of training. In theory, two inputs reconstructed from the same semantic features but different non-semantic features should lead to extracting the same semantic features. However, in practice generated hybrids constitute new samples an overfit model could have trouble accommodating, or present combinations of semantic/non-semantic features that interfere with each other. As per the previous results, our augmentation strategy helps the model deal with those new problematic samples by presenting them as training samples.

Method	CIFAR10	SVHN
	1000	250
Accuracy s_c (%)	97.3 ± 0.6	100 ± 0.0
Accuracy s_r (%)	100 ± 0	98.2 ± 0.3
Ratio of mean $\frac{d_c}{d_r}$	15.5 ± 2.6	7.6 ± 1.4

(a) Component separation (CIFAR10 and SVHN).

Method	MNIST-M
	100
Accuracy s_c (%)	99.9 ± 0.2
Accuracy s_r (%)	96.8 ± 3.1
Ratio of mean $\frac{d_c}{d_r}$	11.0 ± 4.0

(b) Background inheritance (MNIST-M).

Table 3: Identification of semantic and non-semantic parents on a hybrid dataset $\mathcal{D}_H$ at the end of training on multiple datasets. Both the semantic separation s_c and the non-semantic separation s_r accuracies show the model properly incorporates semantic and non-semantic information during hybridization. The ratio of the average non-semantic distances $\frac{d_c}{d_r}$ between hybrids and their semantic/non-semantic parent is given to complement non-semantic separation scores s_r.

Inheritance of non-semantic background in MNIST-M. To better understand non-semantic features, we run an additional experiment by generating an MNIST-M-style dataset [42] by combining each digit picture in the MNIST [43] dataset with a random crop from the BSD 500 dataset [44] (Fig. 7). A model is trained following our standard procedure over 50000 training samples (100 labeled samples), and we track the hybrids generated during training as outlined previously.

Once again, we assess the correct inheritance of semantic content from the seman-

Figure 7: Hybrids for MNIST-M (format: see Fig. 6).

tic parent by tracking the classification accuracy s_c over hybrids. This experiment however differs from the previous one in how the non-semantic distances d_l and d_r are computed. Instead, of considering the distances in z_r latent space we leverage the construction of MNIST-M to propose a more interpretable criterion.

The MNIST-M dataset presents one known non-semantic feature: the background of the samples. We therefore verify experimentally that the background of hybrids generated by our procedure closely matches the background of their non-semantic parents (more closely than the one of their semantic parent) instead of considering z_r distances. By construction of MNIST-M, we know which pixels in images correspond to a digit and which correspond to a BSD 500 background: we know which MNIST sample was used to generate the sample. As such, we can have access to a mask that zeroes out pixels corresponding to the digit and does not alter background pixels for MNIST-M images. As qualitative studies (Fig. 7) suggest hybrid samples do correctly inherit digit outline from their semantic parent, we approximate the background of hybrids to be the same as that of their semantic parent. Therefore, we calculate the background of hybrid samples $b_h = m_1 * x_h$ by applying a mask m_1 that zeroes out pixels corresponding to the digit in the semantic parent (known by construction). The backgrounds b_1 and b_2 of the parent samples are also known by construction (corresponding to the BSD 500 backgrounds used to generate samples). Similarly to our previous procedure, if $d_l := \|b_h - m_1 * b_1\|_2^2 \geq d_r := \|b_h - m_1 * b_2\|_2^2$, then we conclude the hybrid correctly inherited its non-semantic parent's style component.

The separation accuracies s_c and s_r as well as the distances between the hybrid's background and its parents' are given in Tab. 3b. Results suggest a clear separation of semantic and non-semantic content in hybrids. The nature of the pixel distances tracked in this experiment strongly correlate the model's notion of non-semantic features with the known background modularity as expected. As such, the results strongly suggest that at least in simple cases, SAMOSA is capable of correctly identifying and separating the semantic and non-semantic factors in training data.

5. Discussion

In this paper, we introduced SAMOSA, a framework that improves existing SSL algorithms by refining classifier features through unsupervised reconstruction, and by generating hybrid samples for data augmentation. Thanks to its separation of semantic and non semantic components, SAMOSA generates hybrids mixing the semantic content and non semantic characteristics of different samples.

We verified experimentally the framework improves the performance of the Mean Teacher, MixMatch and FixMatch algorithms, with noticeable gains given little labeled data. We also demonstrated the usefulness both of our reconstruction module for classifier regularization, and of the semantically consistent hybrid augmentation. Furthermore, we displayed convincing hybrids by human standard, showing our asymmetrical decoder's ability to hybridize samples. Further investigation demonstrated the good quality of the generated hybrids and provided an interpretation of non-semantic content on the toy MNIST-M dataset.

We explored in this paper content hybridization of samples (as opposed to pixel interpolation) in Semi-Supervised Learning. We believe mixing augmentations with hard predictive labels is currently insufficiently studied relative to strong augmentations for consistency optimization. In particular, how this new hybridization - which behaves somewhat like a pseudo-labeling scheme - should be leveraged for model hybridization remains to be explored in more details.

Funding

This work was conducted using HPC resources from GENCI–IDRIS (Grant 2020-AD011011781), and under a CIFRE grant between Thales Land and Air Systems and Sorbonne University.

References

[1] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 1195–1204.

[2] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 5049–5059.

[3] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, C.-L. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 596–608.

[4] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 630–645. doi:10.1007/978-3-319-46493-0_38.

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, Vol. 25, 2012, pp. 1097–1105.

[6] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, Vol. 28, 2015, pp. 91–99.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 5998–6008.

[8] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: International Conference on Learning Representations, 2017.

[9] T. Miyato, S. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: A regularization method for supervised and semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (2019) 1979–1993. doi:10.1109/TPAMI.2018.2858821.

[10] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, Remixmatch: Semi-supervised learning with distribution matching and aug-

mentation anchoring, in: International Conference on Learning Representations, 2020.

[11] R. Hataya, H. Nakayama, Unifying semi-supervised and robust learning by mixup, Limited Labeled Data (LLD) Workshop at ICLR 2020.

[12] V. Verma, A. Lamb, J. Kannala, Y. Bengio, D. Lopez-Paz, Interpolation consistency training for semi-supervised learning, in: International Joint Conference on Artificial Intelligence, 2019, pp. 3635–3641. doi:10.24963/ijcai.2019/504.

[13] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[14] L. Carratino, M. Cissé, R. Jenatton, J.-P. Vert, On mixup regularization, in: ArXiv preprint, 2020.

[15] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, S. Michalak, On mixup training: Improved calibration and predictive uncertainty for deep neural networks, in: Advances in Neural Information Processing Systems, 2019, pp. 13888–13899.

[16] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017.

[17] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. doi:10.1109/CVPR.2019.00453.

[18] C. LI, T. Xu, J. Zhu, B. Zhang, Triple generative adversarial nets, in: Advances in Neural Information Processing Systems, Vol. 30, 2017.

[19] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. doi:10.1109/CVPR.2019.00521.

[20] O. Chapelle, B. Schlkopf, A. Zien, Semi-Supervised Learning, The MIT Press, 2006. doi:10.5555/1841234.

[21] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, T. Shinozaki, Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling, in: Advances in Neural Information Processing Systems, 2021.

[22] B. Kim, J. Choo, Y. Kwon, S. Joe, S. Min, Y. Gwon, Selfmatch: Combining contrastive self-supervision and consistency for semi-supervised learning, in: ArXiv preprint, 2021.

[23] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, R. Jin, Dash: Semi-supervised learning with dynamic thresholding, in: International Conference on Machine Learning, 2021.

[24] X. Zhai, A. Oliver, A. Kolesnikov, L. Beyer, S4l: Self-supervised semi-supervised learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.

[25] K. Baek, D. Bang, H. Shim, Gridmix: Strong regularization through local context mapping, Pattern Recognition 109 (2021) 107594. doi:10.1016/j.patcog.2020.107594.

[26] J.-H. Kim, W. Choo, H. O. Song, Puzzle mix: Exploiting saliency and local statistics for optimal mixup, in: Proceedings of the 37th International Conference on Machine Learning, Vol. 119, 2020, pp. 5275–5285.

[27] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, Y. Bengio, Manifold mixup: Better representations by interpolating hidden states, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 6438–6447.

[28] Cutmix: Regularization strategy to train strong classifiers with localizable features, in: 2019 IEEE/CVF International Conference on Computer Vision, 2019, pp. 6022–6031. doi:10.1109/ICCV.2019.00612.

[29] R. Sun, C. Masson, G. Hénaff, N. Thome, M. Cord, Swapping semantic contents for mixing images, in: Proceedings of the 26th International Conference on Pattern Recognition, IEEE, 2022, pp. 1280–1286. doi:10.1109/ICPR56361.2022.9956602.

[30] T. Robert, N. Thome, M. Cord, Hybridnet: Classification and reconstruction cooperation for semi-supervised learning, in: Proceedings of the European Conference on Computer Vision, 2018.

[31] D. Guan, J. Huang, S. Lu, A. Xiao, Scale variance minimization for unsupervised domain adaptation in image segmentation, Pattern Recognition 112 (2021) 107764. doi:https://doi.org/10.1016/j.patcog.2020.107764.

[32] X. Peng, Z. Huang, X. Sun, K. Saenko, Domain agnostic learning with disentangled representations, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 5102–5112.

[33] A. H. Liu, Y.-C. Liu, Y.-Y. Yeh, Y.-C. F. Wang, A unified feature disentangler for multi-domain image translation and manipulation, in: Advances in Neural Information Processing Systems, Vol. 31, 2018.

[34] X. Huang, M.-Y. Liu, S. Belongie, J. Kautz, Multimodal unsupervised image-to-image translation, in: Proceedings of the European Conference on Computer Vision, 2018.

[35] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. doi:10.1109/CVPR42600.2020.00975.

[36] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Tech. rep. (2009).

[37] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: Advances in Neural Information Processing Systems, 2011.

[38] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of the British Machine Vision Conference, 2016, pp. 87.1–87.12. doi:10.5244/C.30.87.

[39] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, I. Goodfellow, Realistic evaluation of deep semi-supervised learning algorithms, in: Advances in Neural Information Processing Systems, Vol. 31, 2018.

[40] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, in: ArXiv preprint, 2017.

[41] R. Gontijo-Lopes, S. J. Smullin, E. D. Cubuk, E. Dyer, Affinity and diversity: Quantifying mechanisms of data augmentation, in: ArXiv preprint, 2020.

[42] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: Proceedings of the 32nd International Conference on Machine Learning, Vol. 37, 2015, pp. 1180–1189. doi:10.5555/3045118.3045244.

[43] Y. Lecun, The mnist database of handwritten digits, http://yann.lecun.com/exdb/mnist/.

[44] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 898–916. doi:10.1109/TPAMI.2010.161.