Semi-Supervised Semantic Segmentation with Pixel-Level Contrastive Learning from a Class-wise Memory Bank

Inigo Alonso\textsuperscript{1} Alberto Sabater\textsuperscript{1} David Ferstl\textsuperscript{2} Luis Montesano\textsuperscript{1,3} Ana C. Murillo\textsuperscript{1}

\textsuperscript{1}RoPeRT group, at DIIS - I3A, Universidad de Zaragoza, Spain
\textsuperscript{2}Magic Leap, Zürich, Switzerland
\textsuperscript{3}Bitbrain, Zaragoza, Spain

\{inigo, asabater, montesano, acm\}@unizar.es, dferstl@magicleap.com

Abstract

This work presents a novel approach for semi-supervised semantic segmentation. The key element of this approach is our contrastive learning module that enforces the segmentation network to yield similar pixel-level feature representations for same-class samples across the whole dataset. To achieve this, we maintain a memory bank continuously updated with relevant and high-quality feature vectors from labeled data. In an end-to-end training, the features from both labeled and unlabeled data are optimized to be similar to same-class samples from the memory bank. Our approach outperforms the current state-of-the-art for semi-supervised semantic segmentation and semi-supervised domain adaptation on well-known public benchmarks, with larger improvements on the most challenging scenarios, i.e., less available labeled data. 

https://github.com/Shathe/SemiSeg-Contrastive

1. Introduction

Semantic segmentation consists in assigning a semantic label to each pixel in an image. It is an essential computer vision task for scene understanding that plays a relevant role in many applications such as medical imaging [31] or autonomous driving [2]. As for many other computer vision tasks, deep convolutional neural networks have shown significant improvements in semantic segmentation [2, 19, 1]. All these examples follow supervised approaches requiring a large set of annotated data to generalize well. However, the availability of labels is a common bottleneck in supervised learning, especially for tasks such as semantic segmentation, which require expensive per-pixel annotations.

Semi-supervised learning assumes that only a small subset of the available data is labeled. It tackles this limited labeled data issue by extracting knowledge from unlabeled samples. Semi-supervised learning has been applied to a wide range of applications [38], including semantic segmentation [11, 17, 27]. Previous semi-supervised segmentation works are mostly based on per-sample entropy minimization [17, 22, 29] and per-sample consistency regularization [11, 37, 29]. These segmentation methods do not enforce any type of structure on the learned features to increase inter-class separability across the whole dataset. Our hypothesis is that overcoming this limitation can lead to better feature learning and performance, especially when the amount of available labeled data is low.

This work presents a novel approach for semi-supervised semantic segmentation. Our approach follows a teacher-student scheme where the main component is a novel representation learning module (Figure 1). This module is based
on positive-only contrastive learning \cite{5, 14} and enforces
the class-separability of pixel-level features across different
samples. To achieve this, the teacher network produces fea-
ture candidates, only from labeled data, to be stored in a
memory bank. Meanwhile, the student network learns to
produce similar class-wise features from both labeled and
unlabeled data. The features stored in the memory bank are
selected based on their quality and learned relevance for the
contrastive optimization. Besides, our module enforces the
alignment of unlabeled and labeled data (memory bank) in
the feature space, which is another unexploited idea in semi-
supervised semantic segmentation. Our main contributions
are the following:

• A novel semi-supervised semantic segmentation
framework.
• The use of a memory bank for high-quality pixel-level
features from labeled data.
• A pixel-level contrastive learning scheme where ele-
ments are weighted based on their relevance.

The effectiveness of our method is demonstrated on
well-known semi-supervised semantic segmentation bench-
marks, reaching the state-of-the-art on different set-ups. Be-
sides, our approach can naturally tackle the semi-supervised
domain adaptation task, obtaining state-of-the-art results
too. In all cases, the improvements upon comparable meth-
ods increase with the percentage of unlabeled data.

2. Related Work

This section summarizes relevant related work for semi-
supervised learning and contrastive learning, with particular
emphasis on work related to semantic segmentation.

2.1. Semi-Supervised Learning

Pseudo-Labeling  Pseudo-labeling leverages the idea of
creating artificial labels for unlabeled data \cite{25, 33} by
keeping the most likely predicted class by an existing
model \cite{22}. The use of pseudo-labels is motivated by en-
tropy minimization \cite{13}, encouraging the network to out-
put highly confident probabilities on unlabeled data. Both
pseudo-labeling and direct entropy minimization methods
are commonly used in semi-supervised scenarios \cite{9, 20, 34,
29} showing great performance. Our approach makes use
of both pseudo-labels and direct entropy minimization.

Consistency Regularization  Consistency regularization
relies on the assumption that the model should be invariant
to perturbations, \textit{e.g.}, data augmentation, made to the same
image. This regularization is commonly applied by using
two different methods: distribution alignment \cite{3, 32, 36}, or
augmentation anchoring \cite{34}. While distribution alignment
enforces the prediction of perturbed and non-perturbed
samples to have the same class distribution, augmentation
anchoring enforces them to have the same semantic la-
bel. To produce high-quality non-perturbed class distri-
bution or prediction on unlabeled data, the Mean Teacher
method \cite{37}, proposes a teacher-student scheme where the
teacher network is an exponential moving average (EMA)
of model parameters, producing more robust predictions.

2.2. Semi-Supervised Semantic Segmentation

One common approach for semi-supervised semantic
segmentation is to make use of Generative Adversarial Net-
works (GANs) \cite{12}. Hung \textit{et al.} \cite{17} propose to train the
discriminator to distinguish between confidence maps from
labeled and unlabeled data predictions. Mittal \textit{et al.} \cite{27}
make use of a two-branch approach, one branch enforc-
ing low entropy predictions using a GAN approach and an-
other branch for removing false-positive predictions using
a Mean Teacher method \cite{36}. A similar idea was proposed
by Feng \textit{et al.} \cite{10}, a recent work that introduces Dynamic
Mutual Training (DMT). DMT uses two models and the
model’s disagreement is used to re-weight the loss. DMT
method also followed the multi-stage training protocol from
CBC \cite{9}, where pseudo-labels are generated in an offline
curriculum fashion. Other works are based on data aug-
mentation methods for consistency regularization. French
\textit{et al.} \cite{11} focus on applying CutOut \cite{7} and CutMix \cite{46},
while Olsson \textit{et al.} \cite{29} propose a data augmentation tech-
nique specific for semantic segmentation.

2.3. Contrastive Learning

The core idea of contrastive learning \cite{15} is to create
positive and negative data pairs, to attract the positive and
repulse the negative pairs in the feature space. This tech-
nique has been used in supervised and self-supervised set-
ups. However, recent self-supervised methods have shown
similar accuracy with contrastive learning using positive
pairs only by performing similarity maximization with dis-
tillation \cite{5, 14} or redundancy reduction \cite{47}.

As for semantic segmentation, these techniques have been
mainly used as pre-training \cite{41, 44, 45}. Very recently,
Wang \textit{et al.} \cite{40} have shown improvements in supervised
scenarios applying standard contrastive learning in a pixel
and region level for same-class supervised samples. Van
\textit{et al.} \cite{39} have shown the advantages of contrastive learning
in unsupervised set-ups, applying it between features from
different saliency masks. Lai \textit{et al.} \cite{21} proposed to use
contrastive learning in a self-supervised fashion where posi-
tive samples were the same pixel from different views/crops
and negative samples were different pixels from a different
view, making the model invariant to context information.

In this work, we propose to follow a positive-only con-
trastive learning based on similarity maximization and dis-
tillation \cite{5, 14}. This way, we boost the performance on
semi-supervised semantic segmentation in a simpler and
The learned weights prove its segmentation performance.

from labeled samples to teach the student network labels from unlabeled samples and memory bank entries (Section 3.1).

The teacher network \( f_\xi \) is a weak augmentation of \( x_l \) (see Section 4.2 for augmentation details). For every unlabeled sample \( x_u \), the pseudo-labels \( \hat{y}_u \) are computed following this equation:

\[
\hat{y}_u = \arg \max_{y} f_\xi (x_u),
\]

where \( f_\xi \) predicts a class probability distribution. Note that pseudo-labels are computed at each training iteration.

Consistency regularization is introduced with augmentation anchoring, i.e., computing different data augmentation for each sample \( x_u \) on the same batch, helping the model to converge to a better solution [34]. The pseudo-labels loss for unlabeled data \( \mathcal{X}_u \) is calculated by the cross-entropy:

\[
L_{\text{pseudo}} = \frac{1}{A} \sum_{a=1}^{A} \mathcal{H} (f_\theta (x_u^a), \hat{y}_u),
\]

where \( x_u^a \) is a strong augmentation of \( x_u \) and \( A \) is the number of augmentations we apply to sample \( x_u \) (see Section 4.2 for augmentation details).

3.1. Supervised Segmentation: \( L_{\text{sup}} \)

Our supervised semantic segmentation optimization, applied to the labeled data \( \mathcal{X}_l \), follows the standard optimization with the weighted cross-entropy loss. Let \( \mathcal{H} \) be the weighted cross-entropy loss between two lists of \( N \) per-pixel class probability distributions \( y_1, y_2 \):

\[
\mathcal{H}(y_1, y_2) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_1^{n,c} \log(y_2^{n,c}) \alpha^c \beta^n,
\]

where \( C \) is the number of classes to classify, \( N \) is the number of elements, i.e., pixels in \( y_1, \) \( \alpha^c \) is a per-class weight, and, \( \beta^n \) is a per-pixel weight. Specific values of \( \alpha^c \) and \( \beta^n \) are detailed in Section 4.2. The supervised loss (see top part of Figure 2) is defined as follows:

\[
L_{\text{sup}} = \mathcal{H} (f_\theta (x_l^a), y_l),
\]

where \( x_l^a \) is a weak augmentation of \( x_l \) (see Section 4.2 for augmentation details).

3.2. Learning from Pseudo-labels: \( L_{\text{pseudo}} \)

The key to the success of semi-supervised learning is to learn from unlabeled data. One idea our approach exploits is to learn from pseudo-labels. In our case, pseudo-labels are generated by the teacher network \( f_\xi \) (see Figure 2). For every unlabeled sample \( x_u \), the pseudo-labels \( \hat{y}_u \) are computed following this equation:

\[
L_{\text{pseudo}} = \frac{1}{A} \sum_{a=1}^{A} \mathcal{H} (f_\theta (x_u^a), \hat{y}_u),
\]

where \( x_u^a \) is a strong augmentation of \( x_u \) and \( A \) is the number of augmentations we apply to sample \( x_u \) (see Section 4.2 for augmentation details).
3.3. Direct Entropy Minimization: $\mathcal{L}_{\text{ent}}$

Direct entropy minimization is applied on the class distributions predicted by the student network from unlabeled samples $x_u$ as a regularization loss:

$$\mathcal{L}_{\text{ent}} = -\frac{1}{A} \frac{1}{N} \sum_{a=1}^{A} \sum_{n=1}^{N} \sum_{c=1}^{C} f_{\theta}(x_{u}^{a,n,c}) \log f_{\theta}(x_{u}^{a,n,c}),$$

(7)

where $C$ is the number of classes to classify, $N$ is the number of pixels and $A$ is the number of augmentations.

3.4. Contrastive Learning: $\mathcal{L}_{\text{contr}}$

Figure 3 illustrates our proposed contrastive optimization inspired by positive-only contrastive learning works based on similarity maximization and distillation [5, 14]. In our approach, a memory bank is filled with high-quality feature vectors from the teacher $f_t$ (right part of Figure 3). Concurrently, the student $f_s$ extracts feature vectors from either $X_l$ or $X_u$. In a per-class fashion, every feature is passed through a simple self-attention module that serves as per-feature weighting in the contrastive loss. Finally, the loss enforces the weighted feature vectors from the student to be similar to feature vectors from the memory bank. As the memory bank contains high-quality features from all labeled samples, the contrastive loss helps to create a better class separation in the feature space across the whole dataset as well as aligning the unlabeled data distribution with the labeled data distribution.

Optimization. Let $f_{\theta^-}$ be the student network without the classification layer and $\{x, y\}$ a training sample either from $\{X_l, Y_l\}$ or $\{X_u, \hat{Y_u}\}$. The first step is to extract all feature vectors: $V = f_{\theta^-}(x)$. The feature vectors $V$ are then fed to a projection head, $Z = g_{\theta}(V)$, and a prediction head, $P = g_{\theta}(Z)$, following [5, 14], where $g_{\theta}$ and $g_{\theta}$ are two different Multi-Layer Perceptrons (MLPs). Next, $P$ is grouped by the different semantic classes in $y$.

Let $P_c = \{p_c\}$ be the set of prediction vectors from $P$ of a class $c$. Let $Z'_c = \{z'_c\}$ be the set of projection vectors of class $c$ obtained by the teacher, $Z'_c = g_t(f_t(x))$ from the labeled examples stored in the memory bank.

Next, we learn which feature vectors ($p_c$ and $z'_c$) are beneficial for the contrastive task, by assigning per-feature learned weights (Equation 8) that will serve as a weighting factor (Equation 10) for the contrastive loss function (Equation 11). These per-feature weights are computed using class-specific attention modules $S_{c, \theta}$ (see Section 4.2 for further details) that generate a single value ($w \in [0, 1]$) for every $z'_c$ and $p_c$ feature. Following [35] we L1 normalize these weights to prevent converging to the trivial all-zeros solution. For the prediction vectors $P_c$ case, the weights $w_{p_c}$ are then computed as follows:

$$w_{p_c} = \frac{N_{p_c}}{\sum_{p_t \in P_c} S_{c, \theta}(p_t)},$$

(8)

where $N_{p_c}$ is the number of elements in $P_c$. Equation 8 is used to compute $w_{z'_c}$ too, changing $Z'_c$ and $z'_c$ for $P_c$ and $p_c$.

The contrastive loss enforces prediction vectors $p_c$ to be similar to projection vectors $z'_c$ as in [5, 14] (in our case, projection vectors are in the memory bank). For that, we use the cosine similarity as the similarity measure $C$:

$$C(p_c, z'_c) = \frac{(p_c, z'_c)}{\|p_c\|_2 \cdot \|z'_c\|_2},$$

(9)

where, the weighted distance between predictions and memory bank entry is computed by:

$$D(p_c, z'_c) = w_{p_c} w_{z'_c} (1 - C(p_c, z'_c)),$$

(10)

and, our contrastive loss is computed as follows:

$$\mathcal{L}_{\text{contr}} = \frac{1}{C} \frac{1}{N_{p_c}} \frac{1}{N_{z'_c}} \sum_{c=1}^{C} \sum_{p_t \in P_c} \sum_{z'_c \in Z'_c} D(p_c, z'_c).$$

(11)
Memory Bank. The memory bank is the data structure that maintains the target feature vectors \( z'_c \), \( \psi \) for each class \( c \), used in the contrastive loss. As there is not infinite space for saving all pixels of the labeled data, we propose to store only a subset of the feature vectors from labeled data with the highest quality. As shown in Figure 3, the memory bank is updated on every training iteration with a subset of \( z'_c \in Z' \) generated by the teacher. To select what subset of \( Z' \) is included in the memory bank, we first perform a Feature Quality Filter (FQF), where we only keep features that lead to an accurate prediction when the classification layer is applied, \( y = \arg \max f_\xi(x_l) \), having confidence higher than a threshold, \( f_\xi(x_l) > \phi \). The remaining \( Z' \) are grouped by classes \( z'_c \). Finally, instead of picking randomly a subset of every \( Z'_c \) to update the memory bank, we make use of the class-specific attention modules \( S_{c} \). We get ranking scores \( R_c = S_{c}(Z'_c) \) to sort \( Z'_c \) and we update the memory bank only with the top-K highest-scoring vectors. The memory bank is a First In First Out (FIFO) queue per class for computation and time efficiency. This way it maintains recent high-quality feature vectors in a very efficient fashion computation-wise and time-wise. Detailed information about the hyper-parameters is included in Section 4.2.

4. Experiments

This section describes evaluation set-up and the evaluation of our method on different benchmarks for semi-supervised semantic segmentation, including a semi-supervised domain adaptation, and a detailed ablation study.

4.1. Datasets

- Cityscapes [6]. It is a real urban scene dataset composed of 2975 training and 500 validation samples, with 19 semantic classes.
- PASCAL VOC 2012 [8]. It is a natural scenes dataset with 21 semantic classes. The dataset has 10582 and 1449 images for training and validation respectively.
- GTA5 [30]. It is a synthetic dataset captured from a video game with realistic urban-like scenarios with 24966 images in total. The original dataset provides 33 different categories but, following [42], we only use the 19 classes that are shared with Cityscapes.

4.2. Implementation details

Architecture. We use DeepLab networks [4] in our experiments. For the ablation study and most benchmarking experiments, DeepLabv2 with a ResNet101 backbone is used to have similar settings to previous works [29, 9, 17, 27]. DeepLabv3+ with Resnet50 backbone is also used to equal comparison with [26, 21]. \( \tau \) is set from 0.995 to 1 during training in (Equation 2).

The prediction and projection heads follow [14]: Linear (256) \( \rightarrow \) BatchNorm [18] \( \rightarrow \) Relu [28] \( \rightarrow \) Linear (256). The proposed class-specific attention modules follow a similar architecture: Linear (256) \( \rightarrow \) BatchNorm \( \rightarrow \) LeakyRelu [24] \( \rightarrow \) Linear (1) \( \rightarrow \) Sigmoid. We use \( 2 \times N_{classes} \) attention modules since they are used in a class-wise fashion. In particular, two modules per class are used because we have different modules for projection or prediction feature vectors.

Optimization. For all experiments, we train for 150K iterations using the SGD optimizer with a momentum of 0.9. The learning rate is set to \( 2 \times 10^{-4} \) for DeepLabv2 and \( 4 \times 10^{-4} \) for DeepLabv3+ with a poly learning rate schedule. For the Cityscapes and GTA5 datasets, we use a crop size of 512 \( \times \) 512 and batch sizes of 5 and 7 for DeepLabv2 and DeepLabv3+, respectively. For Pascal VOC, we use a crop size of 321 \( \times \) 321 and batch sizes of 14 and 20 for DeepLabv2 and DeepLabv3+, respectively. Cityscapes images are resized to 512 \( \times \) 1024 before cropping when DeepLabv2 is used for a fair comparison with [29, 9, 17, 27]. The different loss weights in (Equation 1) are set as follows for all experiments: \( \lambda_{sup} = 1, \lambda_{pseudo} = 1, \lambda_{ent} = 0.01, \lambda_{contr} = 0.1 \). An exception is made for the first 2K training iterations where \( \lambda_{contr} = 0 \) and \( \lambda_{pseudo} = 0 \) to make sure predictions have some quality before being taken into account. Regarding the per-pixel weights (\( \beta^n \)) from \( H \) in (Equation 3), we set it to 1 for \( L_{sup} \). For \( L_{pseudo} \), we follow [9] weighting each pixel with its corresponding pseudo-label confidence with a sharpening operation, \( f_\xi(x_u)^+, \) where we set \( s = 6 \). As for the per-class weights \( \alpha^c \) in (Equation 3), we perform a class balancing for the Cityscapes and GTA5 datasets by setting \( \alpha^c = \sqrt{\frac{f_c}{f_{\text{median}}}} \) with \( f_c \) being the frequency of class \( c \) and \( f_{\text{median}} \) the median of all class frequencies. In semi-supervised settings the amount of labels, \( Y_l \), is usually small. For a more meaningful estimation, we compute these frequencies not only from \( Y_l \) but also from \( Y_u \). For the Pascal VOC we set \( \alpha^c = 1 \) as the class balancing does not have a beneficial effect.

Other details. DeepLab’s output resolution is \( \times 8 \) lower than the input resolution. For feature comparison during training, we keep the output resolution and downsample the labels reducing memory requirements and computation.

The memory bank size is fixed to \( \psi = 256 \) vectors per class (see Section 4.4 for more details). The confidence threshold \( \phi \) for accepting features is set to 0.95. The number of vectors added to the memory bank at each iteration, for each image, and for each class is set as \( \max(1, \frac{\psi}{|X_l|}) \), where \( |X_l| \) is the number of labeled samples.

A single NVIDIA Tesla V100 GPU is used for all experiments. All our reported results are the mean of three different runs with different labeled/unlabeled data splits.

Following [29, 37], the segmentation is performed with the student \( f_\theta \) in the experimental validation, although the teacher would lead to a slightly better performance [34].
Data augmentation. We use two different augmentation set-ups, a weak one for labeled samples and a strong set-up for unlabeled samples, following [29] with minor modifications (Table 1 describes the followed data augmentation scheme in our method). Besides, we set $A = 2$ (Equation 6) as the number of augmentations for each sample.

| Parameter | Weak | Strong |
|-----------|------|--------|
| Flip probability | 0.50 | 0.50 |
| Resize $\times [0.75, 1.75]$ probability | 0.50 | 0.80 |
| Color jittering probability | 0.20 | 0.80 |
| Brightness adjustment max intensity | 0.15 | 0.30 |
| Contrast adjustment max intensity | 0.15 | 0.30 |
| Saturation adjustment max intensity | 0.075 | 0.15 |
| Hue adjustment max intensity | 0.05 | 0.10 |
| Gaussian blurring probability | 0 | 0.20 |
| ClassMix probability | 0.20 | 0.80 |

Table 1. Strong and weak data augmentation set-ups

4.3. Benchmark Experiments

Following experiments compare our method with state-of-the-art methods in different semi-supervised scenarios.

4.3.1 Semi-supervised Semantic Segmentation

Cityscapes. Table 4.1 compares different methods on the Cityscapes benchmark for different labeled-unlabeled rates: $\frac{1}{30}$, $\frac{1}{20}$ and $\frac{1}{8}$. Fully Supervised (FS) scenario, where all images are labeled, is also shown as a reference. As shown in the table, our approach outperforms the current state-of-the-art by a significant margin. The performance difference is increasing as less labeled data is available, demonstrating the effectiveness of our approach. This is particularly important since the goal of semi-supervised learning is to learn with as little supervision as possible. Note that the upper bound for each method is shown in the fully supervised setting (FS). Figure 4 shows a visual comparison of the top-performing methods on different relevant samples from Cityscapes. Note that Lai et al. [21] have a higher FS baseline since they use a higher batch size and crop size among other differences in the set-up.

Pascal VOC. Table 4.1 shows the comparison of different methods on the Pascal VOC val set for different labeled-unlabeled ratios and, in parentheses, the difference w.r.t. the corresponding fully supervised (FS) result.

| Architecture: DeepLabv2 with ResNet-101 backbone | 1/50 | 1/20 | 1/8 | FS |
|-----------------------------------------------|------|------|-----|----|
| Adversarial [17]+ | 57.2 (-17.7) | 64.7 (-10.2) | 69.5 (-5.4) | 74.9 |
| s4GAN [27]+ | 63.3 (-4.9) | 67.2 (-6.4) | 71.4 (-2.2) | 73.6 |
| French et al. [11]+ | 64.8 (-7.7) | 66.5 (-6.0) | 67.6 (-4.9) | 72.5 |
| CBC [9]+ | 65.5 (-8.1) | 69.3 (-4.3) | 70.7 (-2.9) | 73.6 |
| ClassMix [29]+ | 66.2 (-7.9) | 67.8 (-6.3) | 71.0 (-3.1) | 74.1 |
| DMT [10]+ | 67.2 (-7.6) | 69.9 (-4.9) | 72.7 (-2.1) | 74.8 |
| Ours* | 65.6 (-7.0) | 67.8 (-4.8) | 69.9 (-2.7) | 72.6 |
| Ours+ | 68.2 (-5.9) | 70.1 (-4.0) | 71.8 (-2.3) | 74.1 |

Table 2. Performance (Mean IoU) for the Cityscapes val set for different labeled-unlabeled ratios and, in parentheses, the difference w.r.t. the corresponding fully supervised (FS) result.

| Error-corr [26]+ | 70.7 (-4.1) | 70.7 (-4.1) | 74.8 |
| Lai et al. [21]+ | 69.7 (-7.8) | 72.7 (-4.8) | 77.5 |
| Ours* | 64.9 (-9.3) | 70.1 (-4.1) | 71.7 (-2.5) | 74.2 |

* ImageNet pre-training, + COCO pre-training

Table 3. Performance (Mean IoU) for the Pascal VOC val set for different labeled-unlabeled ratios and, in parentheses, the difference w.r.t. the corresponding fully supervised (FS) result.

| City | ASS [42] | Liu et al. [23] | Ours with domain adaptation | Ours no adaptation |
|------|----------|----------------|-----------------------------|-------------------|
| Labels | | | | |
| 1/30 | 54.2 | 55.2 | 59.9 | 58.0 |
| 1/15 | 56.0 | 57.0 | 62.0 | 59.9 |
| 1/6 | 60.2 | 60.4 | 64.2 | 63.7 |
| 1/3 | 64.5 | 64.6 | 65.6 | 65.1 |

Table 4. Mean IoU in Cityscapes val set. Central columns evaluate the semi-supervised domain adaptation task (GTA5 → Cityscapes). The last column evaluates a semi-supervised setting in Cityscapes (no adaptation). Different labeled-unlabeled ratios for Cityscapes are compared. All methods use ImageNet pre-trained DeepLabv2 with ResNet-101 backbone.
Our method can naturally tackle this task by evenly sampling from both \( X_l \) and \( X_d \) as our labeled data when optimizing \( \mathcal{L}_{sup} \) and \( \mathcal{L}_{contr} \). However, the memory bank only stores features from the target domain \( X_l \). In this way, both the features from unlabeled data \( X_u \), and the features from the other domain \( X_d \) are aligned with those from \( X_l \).

Following ASS [42, 23], we take the GTA5 dataset as \( X_d \), where all elements are labeled, and the Cityscapes is the target domain consisting of a small set of labeled data \( X_l \) and a large set of unlabeled samples \( X_u \). Table 4.3.1 compares the results of our method with previous methods [42, 23] where all methods use ImageNet pre-training. For reference, we also show the results of our approach with no adaptation, i.e., only training on the target domain Cityscapes, as we do for the semi-supervised set up from the previous experiment (Table 4.3.1). We can see that our approach benefits from the use of the other domain data (GTA5), where there is little labeled data available (\( \frac{1}{30} \)), as it could be expected. Our method outperforms ASS by a large margin in all the different set-ups. As in previous experiments, our improvement is more significant when the amount of available labeled data is smaller.

### 4.4. Ablation Experiments

The following experiments study the impact of the different components of the proposed approach. The evaluation is done on the Cityscapes data, since it provides more complex scenes compared to Pascal VOC. We select the challenging labeled data ratio of \( \frac{1}{30} \).

#### Losses impact

Table 4.4 shows the impact of every loss used by the proposed method. We can observe that the four losses are complementary, getting a 10 mIoU increase over our baseline model, using only the supervised training when

| \( \mathcal{L}_{sup} \) | \( \mathcal{L}_{pseudo} \) | \( \mathcal{L}_{ent} \) | \( \mathcal{L}_{contr} \) | mIoU |
|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | 59.4 |
| ✓ | ✓ | ✓ | ✓ | 59.0 |
| ✓ | ✓ | ✓ | ✓ | 57.4 |
| ✓ | ✓ | ✓ | ✓ | 54.4 |
| ✓ | ✓ | ✓ | ✓ | 52.2 |
| ✓ | ✓ | ✓ | ✓ | 56.7 |
| ✓ | ✓ | ✓ | ✓ | 49.5 |

Note that at testing time, our approach only uses the student network \( f_\theta \), adding zero additional computational cost. At training time, for the experiment of Table 4.4 having an input resolution of \( 512 \times 512 \) with a forward pass cost of 372.04 GFLOPs for one training step using one labeled image and one unlabeled image, compared to the 1488.16 GFLOPs from [10] or 1116.12 GFLOPs from [29]. The total number of GFLOPs come from 372.04 for computing labeled image predictions, 372.04 for the unlabeled image predictions, 372.04 for computing the pseudo-labels and, 35.07 for our contrastive module, which mainly include the computation of the prediction and projection heads (8.59), the class-
Table 6. Effect of different values for the factor $\lambda_{\text{contr}}$ (Equation 1) that weights the effect of the contrastive loss $L_{\text{contr}}$. Results on Cityscapes benchmark ($\frac{1}{30}$ available labels, Deeplabv2-ResNet101 COCO pre-trained).

| $\lambda_{\text{contr}}$ | $10^4$ | $10^2$ | $10^3$ | $10^5$ | $10^{-1}$ | $10^{-2}$ | $10^{-4}$ |
|-------------------------|-------|-------|-------|-------|----------|----------|----------|
| mIoU                    | 50.3  | 51.4  | 54.8  | 59.1  | 59.4     | 58.7     | 57.6     |

Table 7. Effect of our memory bank size (features per-class), $\psi$. Results on Cityscapes benchmark ($\frac{1}{30}$ available labels, Deeplabv2-ResNet101 COCO pre-trained).

| $\psi$ | 32  | 64  | 128  | 256  | 512 |
|-------|-----|-----|------|------|-----|
| mIoU  | 58.7| 58.9| 59.2 | 59.4 | 59.3 |

Contrastive learning module. Table 4.4 shows an ablation on the influence of the contrastive learning module for different values of $\lambda_{\text{contr}}$ (Equation 1). As expected, if this value is too low, the effect gets diluted, with similar performance as if the proposed loss is not used at all (see Table 4.4). High values are also detrimental, probably because it acts as increasing the learning rate vastly, which hinders the optimization. The best performance is achieved when this contrastive loss weight is a little lower than the segmentation losses $L_{\text{sup}}$ and $L_{\text{pseudo}}$ ($\lambda_{\text{contr}} = 10^{-1}$).

The effect of the size (per-class) of our memory bank is studied in Table 4.4. As expected, higher values lead to stronger performances, although from 256 up they tend to maintain similarly. Because all the elements from the memory bank are used during the contrastive optimization (Equation 11) the larger the memory bank is, the more computation and memory it requires. Therefore, we select 256 as our default value.

Table 4.4 studies the effect of the main components used in the proposed contrastive learning module. The base configuration of the module which includes our simplest implementation of the per-pixel contrastive learning using the memory bank, still presents a performance gain compared to not using the contrastive learning module (57.4 mIoU from 4.4). Generating and selecting good quality prototypes is the most important factor. This is done both by the Feature Quality Filter (FQF), i.e., checking that the feature leads to an accurate and confident prediction, and extracting them with the teacher network $f_\xi$. Then, using the class-specific attention $S_{c,\theta}$ to weight every sample (both from the memory bank and input sample) is also beneficial, acting as a learned sampling method.

Future direction. Our proposed approach could potentially be applied to other semi-supervised tasks like object detection or instance segmentation. The straightforward way is to perform the proposed contrastive learning using the features from the semantic head of the detection or instance segmentation networks, i.e., the part of the network that outputs the semantic class of the object or instance. The method is currently restricted by the number of classes and number of memory bank entries per class. A future step to solve this problem could be to cluster the feature vectors per class and save only cluster centers of the class features, similar to the very recent work from Zhang et. al [48] for domain adaptation based on prototypical learning.

5. Conclusion

This paper presents a novel approach for semi-supervised semantic segmentation. Our work shows the benefits of incorporating positive-only contrastive learning techniques to solve this semi-supervised task. The proposed contrastive learning module boosts the performance of semantic segmentation in these settings. Our new module contains a memory bank that is continuously updated with selected features from those produced by a teacher network from labeled data. These features are selected based on their quality and relevance for the contrastive learning. Our student network is optimized for both labeled and unlabeled data to learn similar class-wise features to those in the memory bank. The use of contrastive learning at a pixel-level has been barely exploited and this work demonstrates the potential and benefits it brings to semi-supervised semantic segmentation and semi-supervised domain adaptation. Our results outperform state-of-the-art on several public benchmarks, with particularly significant improvements on the more challenging set-ups, i.e., when the amount of available labeled data is low.

6. Acknowledgments

This work was partially funded by FEDER/ Ministerio de Ciencia, Innovación y Universidades/ Agencia Estatal de Investigación/RTC-2017-6421-7, PGC2018-098817-A-I00 and PID2019-105390RB-I00, Aragón regional government (DGA T45 17R/FSE) and the Office of Naval Research Global project ONRG-NICOP-N62909-19-1-2027.
References

[1] Íñigo Alonso, Luis Riazuelo, and Ana C Murillo. Mininet: An efficient semantic segmentation convnet for real-time robotic applications. IEEE Transactions on Robotics (T-RO), 2020. 1

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12):2481–2495, 2017. 1

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In Advances in Neural Information Processing Systems, pages 5049–5059, 2019. 2

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017. 5

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566, 2020. 2, 4

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In Proceedings of IEEE Conference on CVPR, pages 3213–3223, 2016. 5

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017. 2

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–332, 2010. 5

[9] Zhengyang Feng, Qianyu Zhou, Guangliang Cheng, Xin Tan, Jianping Shi, and Lizhuang Ma. Semi-supervised semantic segmentation via dynamic self-training and class-balanced curriculum. arXiv preprint arXiv:2004.08514, 2020. 2, 5, 6

[10] Zhengyang Feng, Qianyu Zhou, Qiji Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Dmt: Dynamic mutual training for semi-supervised learning. arXiv preprint arXiv:2004.08514, 2020. 2, 6, 7

[11] Geoff French, Samuli Laine, Timo Aila, and Michal Mackiewicz. Semi-supervised semantic segmentation needs strong, varied perturbations. In 29th British Machine Vision Conference, BMVC 2020, 2019. 1, 2, 6

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27:2672–2680, 2014. 2

[13] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. Advances in neural information processing systems, 17:529–536, 2004. 2

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733, 2020. 2, 4, 5

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), volume 2, pages 1735–1742. IEEE, 2006. 2

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9729–9738, 2020. 3

[17] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yan-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. arXiv preprint arXiv:1802.07934, 2018. 1, 2, 5, 6

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015. 5

[19] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In CVPR Workshops. IEEE, 2017. 1

[20] Tarun Kalluri, Girish Varma, Mannohaman Chandraker, and CV Jawahar. Universal semi-supervised semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, pages 5259–5270, 2019. 2

[21] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1205–1214, 2021. 2, 5, 6

[22] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, volume 3, 2013. 1, 2

[23] Weizhe Liu, David Ferstl, Samuel Schuster, Lukas Zebedin, Pascal Fua, and Christian Leistner. Domain adaptation for semantic segmentation via patch-wise contrastive learning. arXiv preprint arXiv:2104.11056, 2021. 6, 7

[24] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In Proceedings icml, volume 30, page 3, 2013. 5

[25] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. Journal of the American Statistical Association, 70(350):365–369, 1975. 2

[26] Robert Mendel, Luis Antonio de Souza, David Rauber, João Paulo Papa, and Cristoph Palm. Semi-supervised segmentation based on error-correcting supervision. In European Conference on Computer Vision, pages 141–157. Springer, 2020. 5, 6
[27] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 2, 5, 6.

[28] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 5

[29] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021. 1, 2, 5, 6, 7.

[30] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 5

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[32] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pages 1163–1171, 2016. 2

[33] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 2

[34] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 2, 3, 5

[35] Shuyang Sun, Liang Chen, Gregory Slabaugh, and Philip Torr. Learning to sample the most useful training patches from images. *arXiv preprint arXiv:2011.12097*, 2020. 4

[36] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. 2

[37] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 1, 2, 3, 5

[38] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 1

[39] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. *arXiv preprint arXiv:2102.06191*, 2021. 2

[40] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Endre Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. *arXiv preprint arXiv:2101.11939*, 2021. 2, 3

[41] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. *arXiv preprint arXiv:2011.09157*, 2020. 2

[42] Zhonghao Wang, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-Mei Hwu, Thomas S Huang, and Honghui Shi. Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 936–937, 2020. 5, 6, 7

[43] Zhifeng Wu, Yujian Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3

[44] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pancontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. 2

[45] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. *arXiv preprint arXiv:2011.10043*, 2020. 2

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 2

[47] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *International Conference on Machine Learning, ICML*, 2021. 2

[48] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12414–12424, 2021. 8