Improving Model Generalization Using a Single Data Sample for Semantic Adaptation

Abstract

The limited capacity of deep networks to generalize beyond their training distribution presents a significant challenge in semantic segmentation. Traditional approaches have operated under the assumption of a fixed model post-training, with parameters remaining constant during testing. This research introduces a self-adaptive methodology for semantic segmentation that modifies the inference mechanism to accommodate each input sample individually. This adaptation involves two principal operations. First, it refines the parameters of convolutional layers based on the input image, employing a consistency-based regularization. Second, it modifies the Batch Normalization layers by dynamically blending the training distribution with a reference distribution extracted from a single test sample. Although these techniques are individually recognized in the field, their combined application establishes new benchmarks in accuracy for generalization from synthetic to real-world data. The empirical evidence from this study indicates that self-adaptation can effectively enhance deep network generalization to out-of-domain data, serving as a valuable complement to the established methods of model regularization during training.

1 Introduction

State-of-the-art models in semantic segmentation exhibit a notable deficiency in robustness when confronted with out-of-distribution data, where the distributions of training and testing sets diverge. While numerous studies have examined this challenge, with a predominant focus on image classification, it has been observed that Empirical Risk Minimization (ERM), which presumes independent and identically distributed training and testing samples, remains remarkably competitive. This contrasts with the evident advancements in domain adaptation for both image classification and semantic segmentation. The domain adaptation setup, however, typically requires access to an unlabeled test distribution during training. In the generalization scenario considered here, only a single test sample is accessible during inference, and no information sharing must occur between subsequent test samples.

This study investigates the generalization challenge in semantic segmentation, specifically from synthetic data to real-world scenarios, by employing an adaptive approach. Unlike prior research that has concentrated on modifying model architecture or training procedures, this work revises the standard inference procedure using a technique derived from domain adaptation methods. Termed self-adaptation, this technique utilizes a self-supervised loss function to facilitate adaptation to individual test samples through a limited number of parameter updates. In addition to these loss-based updates, self-adaptation incorporates feature statistics from the training data with those of the test sample within the Batch Normalization layers.

2 Related Work

This research contributes to the ongoing investigation into the generalization capabilities of semantic segmentation models and is related to explorations of feature normalization and online learning.

.

In contrast to previous studies that focused on training strategies and model design, this study specifically examines the inference process during test time. Prior research has attempted to improve generalization by augmenting synthetic training data with styles transferred from real images, or by utilizing a classification model trained on real images to ensure feature proximity between models via distillation, often seeking layer-specific learning rates. Some approaches have added instance normalization (IN) layers heuristically to the network. Recent studies have sought to extract domain-invariant feature statistics through instance-selective whitening loss or frequency-based domain randomization. Others have aimed to learn style-invariant representations using causal frameworks or have augmented single-domain data to simulate a multi-source scenario to increase source domain diversity. Some techniques involve swapping channel-wise statistics in feature normalization layers and learning adapter functions to adjust the mean and variance based on the input. Another method enforces consistency of output logits across multiple images of the same class. To improve generalization in federated learning, researchers have explored training clients locally with sharpness-aware minimization and averaging stochastic weights. However, these methods either assume access to a distribution of real images during training or require modifications to the network architecture. The technique presented in this work does not require either, making it applicable post-hoc to already trained models to improve their generalization.

Batch Normalization (BN) and other normalization techniques have been increasingly associated with model robustness. The most common methods, including BN, Layer Normalization (LN), and Instance Normalization (IN), also impact the model's expressive capacity, which can be further improved by combining these techniques within a single architecture. In domain adaptation, some studies use source-domain statistics during training and replace them with target-domain statistics during inference. Recent work has explored combining source and target statistics during inference, weighted by the number of samples they aggregate. Others propose using batch statistics from the target domain during inference instead of training statistics from the source domain. This study complements these findings by demonstrating improved generalization of semantic segmentation models.

Several previous studies have updated model parameters during inference, particularly in object tracking where the object detector must adapt to the changing appearance of the tracked instance. Conditional generative models have been employed to learn from single image samples for super-resolution and scene synthesis. Recently, this principle has been extended to improve the robustness of image classification models, though the self-supervised tasks developed for image classification do not always extend well to dense prediction tasks like semantic segmentation. Recent research has proposed more suitable alternatives for self-supervised loss in domain adaptation, and several works have developed domain-specific approaches for medical imaging or first-person vision.

Most of the related works focus on domain adaptation in image classification, typically assuming access to multiple samples from the target distribution during training. This work addresses semantic segmentation in the domain generalization setting, requiring only a single datum from the test set. In this context, simple objectives like entropy minimization improve baseline accuracy only moderately. In contrast, the self-adaptation method presented here, which uses pseudo-labels to account for prediction uncertainty, proves significantly more effective. The task is distinct from few-shot learning, where the model may adapt during testing using a small annotated set of samples. Here, no such annotation is available; the model adjusts to the test sample in an unsupervised manner, without requiring proxy tasks or prior knowledge of the test distribution.

3 Methodology

In traditional inference, the parameters of the segmentation model are assumed to remain fixed. In contrast, adaptive systems are capable of learning to specialize to their environment. Analogously, this study allows the segmentation model to update its parameters during inference. It is important to note that this setup differs from domain adaptation scenarios, as the updated parameters are discarded after processing each sample, aligning with the principles of domain generalization.

The proposed approach creates mini-batches of images for each test sample using data augmentation. Starting with the original test image, a set of N augmented images is generated through multi-scaling, horizontal flipping, and grayscaling. These images form a mini-batch that is processed by the CNN. The resulting softmax probabilities are transformed back to the original pixel space using inverse

affine transformations, producing multiple predictions for each pixel. The mean of these probabilities is computed along the mini-batch dimension for each class and pixel on the spatial grid.

A threshold value is computed from the maximum probability of every class to create a class-dependent threshold. For each pixel, the class with the highest probability is extracted. Low-confidence predictions are ignored by setting pixels with a softmax probability below the threshold to an ignore label, while the remaining pixels use the dominant class as the pseudo-label. This pseudo ground truth is used to fine-tune the model for a set number of iterations using gradient descent with the cross-entropy loss. After this self-adaptation process, a single final prediction is produced using the updated model weights. The weights are then reset to their initial values before processing the next test sample, ensuring that the model does not accumulate knowledge about the entire target data distribution.

Batch Normalization (BN) has become an integral part of modern CNNs. Although originally designed to improve training convergence, it is now recognized for its role in model robustness, including domain generalization. During training, BN computes the mean and standard deviation across the batch and spatial dimensions. The normalized features are derived using these statistics. At test time, it is common practice to normalize feature values with running estimates of the mean and standard deviation across training batches, rather than using test-batch statistics. This is referred to as train BN (t-BN).

In the context of out-of-distribution generalization, the running statistics derived from the source data can differ substantially from those computed using target images, a problem known as covariate shift. Domain adaptation methods often mitigate this issue by replacing source running statistics with those of the target, a technique known as Adaptive Batch Normalization (AdaBN). Recent studies have also explored prediction-time BN (p-BN), which uses the statistics of the current test batch instead of running statistics from training.

This study assumes the availability of only a single target sample during inference. Alternatives like AdaBN and p-BN are not directly applicable in this scenario. Instance Normalization (IN) layers could replace BN layers, but this might lead to covariate shift issues, as sample statistics may only approximate the complete test distribution. Additionally, such a replacement could interfere with the statistics of activations in intermediate layers.

Self-adaptive normalization (SaN) is proposed as a solution. It combines the inductive bias from the source domain's running statistics with statistics extracted from a single test instance. The source mean and variance are averaged with sample statistics from the target domain, weighted by a parameter 1. This parameter represents the shift from the source domain (1 = 0) to a reference domain (1 = 1). During inference, new mean and variance are computed using this weighted average, and these are used to normalize the features of the single test sample. This approach does not affect the behavior of BN layers during training and applies only during testing.

4 Experiments

In this study, the evaluation protocol is revised to adhere to principles of robustness and generalization. The supplier has access to two data distributions: the source data for model training and a validation set for model validation. The generalization ability of the model is assessed on three distinct target sets, providing an estimate of the expected model accuracy for out-of-distribution deployment. The datasets used are restricted to traffic scenes for compatibility with previous research.

Source data for model training comes from two synthetic datasets, GTA and SYNTHIA, which offer low-cost ground truth annotation and exhibit visual discrepancies with real imagery. The validation set used is WildDash, which is understood to be of limited quantity but bears a closer visual resemblance to potential target domains. The model is evaluated on three target domains: Cityscapes, BDD, and IDD, chosen for their geographic diversity and differences in data acquisition. The average accuracy across these target domains estimates the expected model accuracy. Additionally, the Mapillary dataset is used for comparison with previous works, although it does not disclose the geographic origins of individual samples.

The framework is implemented in PyTorch, and the baseline model is DeepLabv1 without CRF post-processing. The models are trained on the source domains for 50 epochs using an SGD optimizer with a learning rate of 0.005, decayed polynomially. Data augmentation techniques include random-size

crops, random aspect ratio adjustments, random horizontal flipping, color jitter, random blur, and grayscaling.

Experiments were conducted to investigate the influence of the parameter 1 in Self-adaptive Normalization (SaN) on segmentation accuracy and the Intersection over Union (IoU) for both source domains (GTA, SYNTHIA) and all main target domains (Cityscapes, BDD, IDD). The optimal 1 was determined based on the IoU on the WildDash validation set. The segmentation accuracy with this optimal 1 was reported, showing that SaN improves the mean IoU over both the established t-BN baseline and the more recent p-BN. The improvement was consistent across different backbone architectures and target domains. Additionally, model calibration, measured by the expected calibration error (ECE), was found to improve with SaN, which was competitive with the MC-Dropout method and showed complementary effects when used jointly.

Self-adaptation was compared to Test-Time Augmentation (TTA), which involves augmenting test samples with flipped and grayscaled versions at multiple scales and averaging the predictions. Self-adaptation outperformed TTA by a clear margin, aligning with reported ECE scores and demonstrating that self-adaptation effectively uses calibrated confidence to generate reliable pseudo-labels.

Self-adaptation was compared with state-of-the-art domain generalization methods, showing consistent improvements over carefully tuned baselines, regardless of backbone architecture or source data. The method outperformed previous methods without modifying the model architecture or training process, altering only the inference procedure.

A comparison with Tent, which also updates model parameters at test time but minimizes entropy instead of using pseudo-labels, showed that self-adaptation outperformed Tent substantially. This was demonstrated by training HRNet-W18 on GTA and comparing the IoU on Cityscapes, where self-adaptation achieved a 7.5% improvement in IoU.

The influence of the number of iterations for self-adaptation was investigated, showing that self-adaptation balances accuracy and inference time by adjusting iteration numbers and layer choices. It was found to be more efficient and accurate than model ensembles. Self-adaptation can trade off accuracy vs. runtime by using fewer update iterations or updating fewer upper network layers.

Hyperparameter sensitivity analysis revealed that self-adaptation is robust to the choice of hyperparameters 1, 8, and 7. The optimal values were determined using the validation set, and the model accuracy declined moderately with deviations from these values. Qualitative results showed that self-adaptation improves segmentation quality and reduces pathological failure modes.

The integration of self-adaptation with state-of-the-art architectures like DeepLabv3+, HRNet-W18, HRNet-W48, and UPerNet with a Swin-T backbone demonstrated substantial improvements in segmentation accuracy across all target domains. Evaluation on the ACDC dataset, which includes adverse weather conditions, showed that self-adaptation outperformed the baseline by 13.57% on average.

Additional qualitative results and failure cases were discussed, showing that self-adaptation can struggle with cases of mislabeling regions with incorrect but semantically related classes. However, these failure cases were relatively rare, and the majority of image samples benefited from self-adaptation, with accuracy improvements of up to 35% IoU compared to the baseline.

5 Results

The empirical results demonstrate that self-adaptive normalization (SaN) consistently enhances segmentation accuracy in out-of-distribution scenarios. For instance, when training on the GTA dataset and testing on Cityscapes, BDD, and IDD, SaN improved the mean IoU by 4.1% with ResNet-50 and 5.1% with ResNet-101 compared to the t-BN baseline. Furthermore, SaN outperformed the more recent p-BN method, showing improvements irrespective of the backbone architecture and the target domain tested. In terms of calibration quality, measured by the expected calibration error (ECE), SaN not only improved the baseline but also showed competitiveness with the MC-Dropout method, even exhibiting complementary effects when both methods were combined.

Self-adaptation was found to outperform traditional Test-Time Augmentation (TTA) across both source domains (GTA, SYNTHIA) and three target domains (Cityscapes, BDD, IDD). Despite TTA improving the baseline, self-adaptation provided a clear and consistent margin of 2.19% IoU on

average. This aligns with the reported ECE scores, demonstrating that self-adaptation effectively exploits the calibrated confidence of predictions to yield reliable pseudo-labels.

In comparison to state-of-the-art domain generalization methods, self-adaptation showed substantial improvements even over carefully tuned baselines. It outperformed methods like DRPC and FSDR on most benchmarks, despite these methods using individual models for each target domain and resorting to target domains for hyperparameter tuning. Self-adaptation achieved superior segmentation accuracy without requiring access to a distribution of real images for training or modifying the model architecture, unlike previous methods such as ASG, CSG, DRPC, and IBN-Net.

The study also compared self-adaptation with Tent, which updates model parameters at test time by minimizing entropy. Self-adaptation, which constructs pseudo-labels based on well-calibrated predictions, substantially outperformed Tent. Specifically, when training HRNet-W18 on GTA and evaluating on Cityscapes, self-adaptation achieved a 7.5% improvement in IoU compared to Tent under a comparable computational budget.

Further analysis revealed that self-adaptation provides a flexible mechanism for trading off accuracy and runtime by varying the number of update iterations and the layers to adjust. It was found to be more efficient and accurate than model ensembles. Hyperparameter sensitivity analysis indicated that self-adaptation is robust to the choice of hyperparameters, with optimal values determined using the validation set.

Qualitative results demonstrated that self-adaptation visibly improves segmentation quality, reducing artifacts and mislabeling compared to the baseline. The method's effectiveness was consistent across different architectures, including DeepLabv3+, HRNet-W18, HRNet-W48, and UPerNet with a Swin-T backbone, showing substantial improvements in segmentation accuracy on all target domains.

6 Conclusion

The traditional learning principle of Empirical Risk Minimization (ERM) assumes independent and identically distributed training and testing data, which often results in models that are not robust to domain shifts. To address this, a self-adaptive inference process was introduced, bypassing the need for explicit assumptions about the test distribution. This study also outlined four principles for a rigorous evaluation process in domain generalization, adhering to best practices in machine learning research.

The analysis demonstrated that even a single sample from the test domain can significantly improve model predictions. The self-adaptive approach showed substantial accuracy improvements without altering the training process or model architecture, unlike previous works. These results suggest that self-adaptive techniques could be valuable in other application domains, such as panoptic segmentation or monocular depth prediction.

While the presented self-adaptation method is not yet real-time, it offers a favorable trade-off between accuracy and computational cost compared to model ensembles. Future research could explore reducing the latency of self-adaptive inference through adaptive step sizes, higher-order optimization, or low-precision computations. Overall, this work demonstrates the potential of self-adaptation to enhance model generalization and robustness in various applications.