Sequential Ensembling for Semantic Segmentation

Rawal Khirodkar\textsuperscript{1}\textsuperscript{*}  Brandon Smith\textsuperscript{2}  Siddhartha Chandra\textsuperscript{2}  Amit Agrawal\textsuperscript{3}  Antonio Criminisi\textsuperscript{2}\textsuperscript{†}

\textsuperscript{1}Carnegie Mellon University  \textsuperscript{2}Amazon Lab 126  \textsuperscript{3}Amazon Fashion

Abstract

Ensemble approaches for deep-learning-based semantic segmentation remain insufficiently explored despite the proliferation of competitive benchmarks and downstream applications. In this work, we explore and benchmark the popular ensembling approach of combining predictions of multiple, independently-trained, state-of-the-art models at test time on popular datasets. Furthermore, we propose a novel method inspired by boosting to sequentially ensemble networks that significantly outperforms the naïve ensemble baseline. Our approach trains a cascade of models conditioned on class probabilities predicted by the previous model as an additional input. A key benefit of this approach is that it allows for dynamic computation offloading, which is useful for deploying models on mobile devices. Our proposed novel ADaptive modulatiON (ADON) block allows spatial feature modulation at various layers using previous stage probabilities. Our approach does not require any sophisticated sample selection strategies during training, and works with multiple neural architectures. We significantly improve over the naïve ensemble baseline on challenging datasets such as Cityscapes, ADE-20K, COCO-Stuff and PASCAL-Context and set a new state-of-the-art.

1. Introduction

Semantic segmentation, i.e., pixel classification, is a fundamental task in computer vision with applications in a wide variety of domains, such as autonomous driving \cite{56}, robotic navigation \cite{43}, medical analysis \cite{50}, and scene understanding \cite{81}. Over the years, deep learning based semantic segmentation methods have achieved remarkable performance \cite{73, 42, 72, 1, 45, 74, 70}. Most of these approaches focus on training a single network with novel architectures and loss functions to improve performance, overlooking the extensive classical literature on ensembles \cite{2, 54, 62}. Ensembles have been shown to improve accuracy \cite{3}, uncertainty estimation, and out-of-distribution robustness \cite{16}. We argue that deep learning approaches can benefit greatly from the pioneering work done in ensemble learning. In this paper, we methodically explore deep ensembles for semantic segmentation with the goal of further improving performance of state-of-the-art semantic segmentation models.

A naïve deep ensembling approach is to train multiple networks and fuse their outputs (e.g., average their predicted probabilities). We benchmark this ensembling approach for state-of-the-art deep semantic segmentation models across multiple datasets. Similar to \cite{37, 5, 65}, the baseline ensembling rule averages predicted probabilities using equal weight for all the models within the ensemble. We refer to this as simple ensembling (SIM-ENS). Consistent with \cite{5, 65}, and as shown in Tab. \ref{table:ensembling}, we observe diminishing returns as the number of models $N$ in the ensemble increases. For dense per-pixel predictions, simple ensembles are not effective as distinct models perform well on different regions of the image. A natural variation is to try other ensembling rules such as max-voting and weighted-averaging using model confidence estimates \cite{10}. However, these strategies result in moderate gains similar to simple en-
sembles as shown in Tab. 1 owing to challenges in obtaining reliable per pixel confidence weights (see Fig. 3).

In this work, we propose sequential ensembling as an alternative to simple ensembling. Sequential ensembling (SEQ-ENS) is a data-driven approach to learn an optimal ensemble compared to ad-hoc ensembling approaches (e.g., averaging or voting). Inspired by boosting [53] and cascaded refinement [47, 8], we train $N$ generations of deep models sequentially, as shown in Fig. 1. Generation $G_i$ uses input image $I$ and the predicted probability map $P_{i-1}$ from generation $G_{i-1}$ to predict $P_i$. Our approach only requires probability maps from the previous generation and does not require any sophisticated sample selection strategy during training. To condition on probability maps, we adopt the simple yet effective mechanism of feature modulation [49, 13, 25]. We propose a novel ADaptive modulatION (ADON) block which injects information from $P_{i-1}$ at multiple depths within the network. Dense prediction is aided by spatial context and the ADON block allows spatial modulation of intermediate features of $G_i$. Each subsequent generation learns to incorporate knowledge from the previous generation using class-specific spatial context. Unlike simple ensembles, each model can dedicate network capacity to correct mistakes made by prior generations. Our approach allows for increase in accuracy as the ensemble size is increased, as shown for multiple datasets in Fig. 2.

Sequential ensembles are constructed by training a new generation with parameters of all previous generations $\{G_{i-1}, G_{i-2}, \ldots G_0\}$ frozen. This affords several important benefits. For example, the sequential chain can be interrupted at any depth, and each stage can still output pixel-wise probabilities. This is a valuable property that allows for dynamic computation offloading [46] where the ensemble size can be adjusted on the fly to meet resource limitations without any retraining. This strategy is useful for deploying models on diverse mobile platforms to best suit changing on-device constraints, e.g., diminishing battery life, applications that require accuracy vs. runtime tradeoffs (fast previews vs. slow high-quality results), responsiveness requirements in the context of changing mission-critical workloads.

Our proposed approach achieves state-of-the-art results on Cityscapes [9], ADE20K [80], PASCAL-Context [48] and COCO-Stuff [4]. For backbones with lower computational complexity, sequential ensembles ($N = 2$) outperform simple ensembles ($N = 15$) across multiple datasets, as shown in Tab. 1. In Tab. 5, we additionally show that a 2 model sequential ensemble with single-scale inference outperforms a single model with multi-scale test-time augmentation using 12 inferences (6x lower inferences). Furthermore, we show that sequential models can (a) generalize across different backbone architectures (Tab. 2), (b) exhibit self-improvement via the ability to refine their own segmentation maps iteratively yielding non-trivial gains (Tab. 7) and (c) can be used to create general neural network graphs leveraging diversity gains from ensembling (Tab. 8).

In summary:

- We explore ensembling approaches for deep neural networks to improve state-of-the-art segmentation networks and provide an ensembling benchmark of various models across multiple semantic segmentation datasets.
- Our proposed sequential ensembling outperforms naïve ensembling methods and advances the state-of-the-art on challenging benchmarks.
- Our proposed feature modulation ADON blocks efficiently incorporates information from previous generations in the sequential ensemble chains.

| Method                  | Cityscapes  | ADE20K    | COCO-Stuff |
|-------------------------|-------------|-----------|------------|
| Single Model ($N = 1$) | 76.2        | 33.1      | 32.4       |
| SIM-ENS ($N = 2$)       | 77.1 (+0.9) | 34.5 (+1.4) | 33.5 (+1.1) |
| SIM-ENS ($N = 5$)       | 77.5 (+1.3) | 34.9 (+1.8) | 33.8 (+1.4) |
| SIM-ENS ($N = 15$)      | 77.8 (+1.6) | 35.8 (+2.7) | 34.9 (+2.5) |
| Voting Ens. ($N = 15$)  | 76.9 (+0.7) | 35.3 (+2.2) | 35.1 (+2.7) |
| W-Avg. Ens. ($N = 15$) | 78.0 (+1.8) | 35.0 (+1.9) | 34.6 (+2.2) |
| SEQ-ENS ($N = 2$)       | 78.9 (+2.7) | 37.6 (+4.5) | 36.3 (+3.9) |

Table 1: Sequential ensembling (SEQ-ENS) offers significantly higher improvement (mIOU) compared to traditional ensembling approaches based on voting and simple/weighted averaging. Results are reported on the val sets using HRNetW18s-v2 [58] backbone with single-scale testing.
2. Related Work

Semantic-Segmentation: Semantic segmentation is an extension of the classification problem from image level to pixel level. Over the past few years, advances in deep learning for image classification led to significant improvements for semantic segmentation. These efforts benefited from the introduction of popular datasets such as Cityscapes, ADE20K, PASCAL-Context, PASCAL-VOC, COCO-Stuff, etc. In this work, we benchmark ensembling for segmentation.

Ensembles for Segmentation: Ensemble learning has been well-studied in machine learning; seminal works include bagging, boosting, and AdaBoost. Ensembles of deep models have been used to boost performance on many tasks such as image classification, machine translation, and uncertainty estimation. However, ensembling approaches have not been fully explored for semantic segmentation owing to computational challenges in dense pixel prediction. Our work benchmarks ensembles of state-of-the-art models across various segmentation datasets and model families to explore for semantic segmentation. We hope this will be helpful in inspiring and evaluating new ideas in field. Further, inspired by boosting paradigm, our proposed sequential ensemble methodology achieves new state-of-the-art segmentation results.

Segmentation Refinement: Many refinement methods have been proposed. These refinement approaches depend on the segmentation model, super-pixels, multi-scale input, data-generation, or object boundary information. In contrast to which assumptions at object boundaries, our approach does not make any such assumptions and can be applied to improve segmentation predictions at pixels away from boundaries, especially for lighter models.

Figure 3: Heatmap visualization of per-pixel confidence (max class probability) of independently trained models (blue is high confidence). We observe that confidence on a pixel is not necessarily correlated with accuracy of segmentation model. Empirically, our method performs better than confidence weighted ensembling rule.

3. Method

Semantic segmentation aims to classify pixels in an input image $I \in \mathbb{R}^{H \times W \times 3}$ into $C$ classes. Most methods, e.g., transform this problem to estimating per-pixel class probabilities to create a probability map $P \in \mathbb{R}^{H \times W \times C}$ from $I$ using a deep network $G$ such that $P = G(I)$. The segmentation network $G$ is trained in a supervised way to minimize loss $L(Y, P)$. where $Y$ is the ground truth segmentation map of image $I$. At inference, each pixel is assigned the label corresponding to the highest probability.

Simple Ensembles: Consider a pretrained model ensemble $B$ with $N$ segmentation models, $B = \{B_0, B_1, \ldots, B_{N-1}\}$, where each ensemble member $B_i$ is independently trained. The prediction $P$ of the ensemble $B$ on the input image $I$ can be defined as follows:

$$
P_i = B_i(I),
$$

$$
P = \Phi(P_0, P_1, \ldots, P_{N-1}),
$$

where $\Phi$ is a strategy to combine individual model predictions, e.g.,

$$
\Phi = \frac{1}{N} \sum_{i=0}^{N-1} w_i P_i,
$$

where $w_i$ can be uniform (average) or vary as a function of output probabilities (weighted average). Other strategies such as median averaging, majority voting, stacking, or selecting best model can also be used to define $\Phi$. Designing an optimal $\Phi$ for semantic segmentation is crucial, yet challenging since different models in an ensemble may be more accurate on different regions of an image.

3.1. Sequential Ensembles

We propose a sequential ensembling strategy that avoids the combination rule $\Phi$ and learns an optimal ensembling from the training data. Let $G = \{G_0, G_1, \ldots, G_{N-1}\}$ be $N$ segmentation models in the sequential ensemble. Each...
We implement the operations $F$ work by applying an affine transformation to the network’s ADON blocks at various depths allows conditioning on spatial probability maps at various levels within the network.

And bias the intermediate feature map $X$.

The operation $F_{\text{scale}}$ and $F_{\text{bias}}$ are used to element-wise scale and bias the intermediate feature map $X$ to produce $X'$.

Thus, ADON is a spatial generalization of channel-wise feature modulation blocks proposed in [23][31]. As the modulation parameters are adaptive to the spatially variant input probability maps, the proposed ADON block is an effective way of injecting segmentation information at multiple layers within the network in comparison to early/late fusion [18]. ADON blocks inserted early in the network capture finer details such as object boundaries and those inserted late resolve the class confusion between similar looking classes.

### 3.2. Adaptive Modulation Block

A key challenge is designing the architecture in order to take the previous generation’s probabilities as a conditioning input. A naive early fusion approach would be to simply concatenate the input image $I$ with the probability map $P_{i-1}$ corresponding to the output of the previous generation. Similarly, late fusion would concatenate feature maps from later layers within the network with appropriately down-sampled probability map $P_{i-1}$. However, both of these approaches fail to improve performance.

We describe the ADaptive modulatiON (ADON) block that can be easily introduced in any existing feature extraction backbone to overcome this issue (see Fig. 4). ADON allows spatial modulation of intermediate feature maps using the conditioning input $P_{i-1}$. The $i^{th}$ generation model $G_i$ in the sequential ensemble uses ADON blocks to leverage information from the prediction $P_{i-1}$ of the previous generation model. Similar to the Batch Normalization [27], ADON learns to adaptively influence the output of the neural network by applying an affine transformation to the network’s intermediate features based on $P_{i-1}$.

Let $X \in \mathbb{R}^{H \times W \times D}$ be an intermediate feature map in the segmentation network $G_i$. The ADON block consists of operations $F_{\text{shared}}$, $F_{\text{scale}}$ and $F_{\text{bias}}$ on the probability map $P_{i-1}$ from the previous generation $G_{i-1}$. The modulation outputs $\sigma$ and $\beta$ are used to element-wise scale and bias the intermediate feature map $X$ to produce $X'$. We implement the operations $F_{\text{shared}}$, $F_{\text{scale}}$ and $F_{\text{bias}}$ using a simple two-layer convolutional network, whose design is in the supplemental material. First, the probability map $P_{i-1} \in \mathbb{R}^{H \times W \times C}$ is spatially downsampled to match the 2D resolution of $X$. The operation $F_{\text{shared}}$ then maps $P_{i-1} \in \mathbb{R}^{H' \times W' \times D}$ to a $K$-dimensional latent space, $e \in \mathbb{R}^{H' \times W' \times K}$. We use this latent space to predict the scale and bias parameters jointly using $F_{\text{scale}}$ and $F_{\text{bias}}$ where $\sigma, \beta \in \mathbb{R}^{H' \times W' \times D}$.

Thus, ADON is a spatial generalization of channel-wise feature modulation blocks proposed in [23][31]. As the modulation parameters are adaptive to the spatially variant input probability maps, the proposed ADON block is an effective way of injecting segmentation information at multiple layers within the network in comparison to early/late fusion [18]. ADON blocks inserted early in the network capture finer details such as object boundaries and those inserted late resolve the class confusion between similar looking classes.

### 4. Experiments

We evaluate our approach on the following datasets.

**Cityscapes** [9] is a real world driving dataset that consists of 2975 train, 500 val and 1525 test images with resolution 2048 × 1024. The dataset contains 19 semantic categories for the segmentation task.

**ADE20K** [80] is used in ImageNet scene parsing challenge 2016 consisting of around 20k train, 2k val, 3k test images spanning 150 fine-grained semantic categories and diverse scenes.

**COCO-Stuff** [4] is a challenging scene parsing dataset that contains 171 semantic classes. We use the smaller version with 10k images. The train set and test set consists of 9k and 1k images respectively.

**PASCAL-Context** [43] consists of 59 semantic classes and 1 background label. The dataset contains 4998 train and 5105 test images. We follow the standard testing
Table 2: Comparison of SEQ-ENS (Ours) with SIM-ENS using N = 2 or 3 variants. R-@ and H-@ stand for ResNet-@ and HRNet-W@ respectively. D8 is the output stride of DeepLabv3+. s denotes the small version of the backbone. We report class-wise mIoU for 11 rare classes in Cityscapes. For rare classes such as motor-cycle (mcycl), SEQ-ENS improves mIoU by ≈ 2× compared to SIM-ENS.

Procedure. The image is resized to 480 × 480 and then fed into our network. The resulting 480 × 480 label maps are then resized to the original image size.

Implementation Details. We use the mmsegmentation\footnote{https://github.com/open-mmlab/mmsegmentation} codebase for the implementation of various model families. We use the available pretrained models as $G_0$ in the sequential ensemble, and focus on improving their performance by adding additional generations. For fairness, the models in the later generations are trained with the same hyper-parameters as the $G_0$ model. We use the imagenet pretrained backbone for all our experiments. For SIM-ENS, the segmentation head of the backbone is randomly initialized. In contrast, we report results using inference at single-scale and no rescaling.

4.1. Comparisons with Simple Ensembles

We compare SEQ-ENS and SIM-ENS in Tab.2 for $N = 2$. We benchmark three model families, MobileNetv3 [22], HRNet [58], and DeepLabv3+ [71], with different backbones, e.g., HRNet consistently improves over SIM-ENS in all settings; relative improvements over baseline with respect to SIM-ENS range from 2x to 8x. On Cityscapes, SEQ-ENS shows consistent gains across all backbones over SIM-ENS. Gains from SEQ-ENS tend to be more exaggerated for scenarios susceptible to underfitting compared to single models or SIM-ENS, for example, mIoU improves most significantly for rare categories, such as train and motorcycle, and categories with fine structures, such as pole and fence. This shows SEQ-ENS distributes model capacity intelligently to focus on underrepresented classes and finer details. Our improvements are especially pronounced when ensembling light models like MobileNetv3 or evaluating on complex datasets with more classes. For example, for H-18s, SEQ-ENS outperforms SIM-ENS by +1.6 mIoU for Cityscapes (19 categories) and +1.6 for PASCAL-Context (59 categories), but by +3.1 for ADE-20K (150 categories) and +2.7 for COCO-Stuff (171 categories). We show qualitative results in Fig.7.
Further, we report various metrics like mIoU, parameters (M), GFLOPs, test-time FPS and total training epochs with varying ensemble size (N) for SIM-ENS and SEQ-ENS in Tab. 3 using MobileNetv2 [52]. As can be seen, later generations require fewer epochs for training in SEQ-ENS due to quicker convergence (495 vs. 127 epochs for 10th vs. 1st generation). Finally, our method allows for a dynamic trade-off between accuracy and latency by varying the ensemble size during inference — a desirable property for edge devices.

4.2. Improving State-of-the-Art

We improve the state-of-the-art among methods that do not use extra data on the Cityscapes and ADE20K val sets. We construct a sequential ensemble of size 3 using Segformer MiT-B5 [71] as $G_0$ followed by 2 generations of HRNet-W48. Our models are trained at the same resolution and learning rate schedule as the Segformer and use test-time augmentation at 7 scales with left-right flipping. SEQ-ENS improves the prior art by 0.8 on Cityscapes and 2.2 on ADE20K, as shown in Tab. 4. SEQ-ENS with fewer parameters outperforms Trans-21 [44].

### Table 4: Improving the state-of-the-art methods using sequential ensembling on Cityscapes and ADE20K val set.

| Method     | Cityscapes mIoU | ADE20K mIoU | SIM-ENS | ADE20K |
|------------|-----------------|-------------|---------|--------|
| FCN [52]   | 76.2            | 33.1        | -       | -      |
| EncNet [76] | 76.2            | 33.1        | -       | -      |
| PSPNet [77] | 76.2            | 33.1        | -       | -      |
| CCNet [59]  | 76.2            | 33.1        | -       | -      |
| DeepLabV3+ [78] | 76.2         | 33.1        | -       | -      |
| OCRNet [82] | 76.2            | 33.1        | -       | -      |
| GSCNN [79]  | 76.2            | 33.1        | -       | -      |
| Ax-DeepLab [83] | 76.2       | 33.1        | -       | -      |
| Dynamic Routing [84] | 76.2       | 33.1        | -       | -      |
| Auto-DeepLab [85] | 76.2       | 33.1        | -       | -      |
| SEIR [70]  | 76.2            | 33.1        | -       | -      |
| SegFormer [71] | 76.2        | 33.1        | -       | -      |
| Swin-Trans [44] | 76.2       | 33.1        | -       | -      |

| Method     | Cityscapes mIoU | ADE20K mIoU | SIM-ENS | ADE20K |
|------------|-----------------|-------------|---------|--------|
| FCN [52]   | 76.2            | 33.1        | -       | -      |
| EncNet [76] | 76.2            | 33.1        | -       | -      |
| PSPNet [77] | 76.2            | 33.1        | -       | -      |
| CCNet [59]  | 76.2            | 33.1        | -       | -      |
| DeepLabV3+ [78] | 76.2         | 33.1        | -       | -      |
| OCRNet [82] | 76.2            | 33.1        | -       | -      |
| GSCNN [79]  | 76.2            | 33.1        | -       | -      |
| Ax-DeepLab [83] | 76.2       | 33.1        | -       | -      |
| Dynamic Routing [84] | 76.2       | 33.1        | -       | -      |
| Auto-DeepLab [85] | 76.2       | 33.1        | -       | -      |
| SEIR [70]  | 76.2            | 33.1        | -       | -      |
| SegFormer [71] | 76.2        | 33.1        | -       | -      |
| Swin-Trans [44] | 76.2       | 33.1        | -       | -      |

### Table 5: (Left) Effect of inserting ADON blocks at various layers in the HRNet-W18s [53] backbone with single scale testing. (Right) Test-time augmentation using six scales and left-right flipping for $N = 2$. Note that our $G_1$ model with single scale testing (two inferences) outperforms $G_0$ with multi-scale testing (12 inferences).

| Method     | Cityscapes mIoU | ADE20K mIoU | SIM-ENS | ADE20K |
|------------|-----------------|-------------|---------|--------|
| FCN [52]   | 76.2            | 33.1        | -       | -      |
| EncNet [76] | 76.2            | 33.1        | -       | -      |
| PSPNet [77] | 76.2            | 33.1        | -       | -      |
| CCNet [59]  | 76.2            | 33.1        | -       | -      |
| DeepLabV3+ [78] | 76.2         | 33.1        | -       | -      |
| OCRNet [82] | 76.2            | 33.1        | -       | -      |
| GSCNN [79]  | 76.2            | 33.1        | -       | -      |
| Ax-DeepLab [83] | 76.2       | 33.1        | -       | -      |
| Dynamic Routing [84] | 76.2       | 33.1        | -       | -      |
| Auto-DeepLab [85] | 76.2       | 33.1        | -       | -      |
| SEIR [70]  | 76.2            | 33.1        | -       | -      |
| SegFormer [71] | 76.2        | 33.1        | -       | -      |
| Swin-Trans [44] | 76.2       | 33.1        | -       | -      |

5. Analysis

We first show ablation studies with respect to ADON block placement and parameters and then discuss several interesting features of our framework.

### ADON Block Placement:
We analyse the effect of adding ADON blocks at various depths in the HRNet-W18s back-

---

**Figure 5**: Sequential ensembles allows segmentation prediction conditioning between generations in various ways. 

**Table 5**: (Left) Effect of inserting ADON blocks at various layers in the HRNet-W18s [53] backbone with single scale testing. (Right) Test-time augmentation using six scales and left-right flipping for $N = 2$. Note that our $G_1$ model with single scale testing (two inferences) outperforms $G_0$ with multi-scale testing (12 inferences).

**Multi-Model Generalization**: Our SEQ-ENS method can be generalized to multiple architectures, where $G_i$ is trained to randomly condition on probability inputs $P_{i-1}$ from different $G_0$ segmentation models (see Fig. 5 (left)). By training a single $G_i$, during inference we see improved performance for all the different $G_0$s. In Tab. 6 we show that using HRNet-W18s [53] backbone, $G_i$, improves the performance of all three backbones in the HRNet family on Cityscapes and ADE20K datasets. The generalized sequential ensembles also outperforms simple ensemble baseline and have comparable performance to the default scenario of individual sequential ensembles using one-third number of parameters.

**Self-Improvement**: Fig. 6 (top row) shows that $G_i$ attempts to fix segmentation errors in the input probability map $P_{i-1}$. For this example, networks improves pixel accuracy to 18.9% when conditioned on a random probability map (accuracy: 1.8%). Interestingly, when conditioned on the ground-truth, $G_i$ attempts to utilize it. Inspired by this, we evaluate self-refinement as described in Fig. 5 (middle), where we feed the predicted probabilities $P_i$ of $G_i$ as its own input.
during inference. Tab. 7 shows non-negligible improvement in performance using self-refinement.

**Ensemble of Chains:** Our method allows creation of general graphs where a node is a segmentation network and the edges define the conditional dependence using the predicted probabilities. We showed that adding models/nodes in a sequential manner in a chain reduces dataset error. We can further reduce the error by adding multiple chains and

| Arch | Cityscapes | ADE20K |
|------|------------|----------------|
|      | $G_0$ | $G_1$ | $G_1$ | $G_0$ | $G_1$ | $G_1$ |
| H-18s | 76.2 | 79.1 | 78.1 | 33.1 | 37.6 | 37.0 |
| H-18 | 78.7 | 79.4 | 79.9 | 36.8 | 39.1 | 38.5 |
| H-48 | 80.5 | 80.7 | 80.5 | 42.0 | 42.6 | 42.3 |

Table 6: Comparison of generalized vs default sequential ensembles using HRNet-W18 backbone as $G_1$. Our approach allows improving performance by training $G_1$ models for each of the three architectures separately (default scenario). A single generalized $G_1$ model can give similar performance with 33% parameters.

Figure 6: $G_1$ refines/utilizes conditional probabilities $P_{i-1}$. Top row shows an extreme example where $G_1$ refines random input probabilities. If ground-truth probabilities are provided as conditioning, $G_1$ utilizes them. We use this observation for self-improvement and achieve a gain of 0.6% when using 3 self-loops with HRNet-W18s backbone as shown in Tab. 7.

| Number of Chains (C) |
|----------------------|
| $C = 1$ | $C = 2$ | $C = 3$ | $C = 4$ | $C = 5$ |
| $N = 1$ | 61.3 | 62.2 | 62.4 | 62.7 | 62.7 |
| $N = 2$ | 64.2 | 65.7 | 66.3 | 65.8 | 66.5 |
| $N = 3$ | 66.1 | 67.2 | 67.4 | 67.8 | 68.1 |

Table 8: Multiple sequential-ensemble chains can be used to improve performance. We show performance by varying the number of chains ($C$) and the number of model generations ($N$) using MobileNetv2-D8 backbone on the Cityscapes val set.

**Limitations and Future Work:** Sequential ensembling trains one generation at a time. This improves accuracy compared to simple ensembles, but at the cost of increased training time. Though, empirically, we observe quicker convergence for later generation when training. This is a one-time, offline cost, with potential ways to speed-up, e.g. several generations could be trained in parallel using latest prior generation predictions, possibly further reducing training time. Additional improvements could come by injecting feature maps from prior generations as inputs to future generations along with the probability map. This can further decrease the numbers of parameters required in the sequential ensemble to improve performance.

An open question for future work is whether large models give superior performance in comparison to ensembles with same number of parameters. [69] shows that ResNet-101 outperforms ResNet-152 for semantic segmentation, indicating that naively increasing parameters may lead to overfitting. Our sequential strategy of building complexity gradually allows training using fewer resources and enables us to modulate the accuracy vs complexity tradeoff at test time.

**6. Conclusion**

In this work, we explore deep ensembles to improve the performance of segmentation models. We benchmark ensembles of state-of-the-art deep models for multiple datasets. Inspired by boosting approaches, we propose sequential ensembling—a strategy to gradually increase model complexity to improve performance. This an alternative to the standard practice of training large models and compressing them for on-device deployment, and is suitable for problems with dynamic resource constraints. Our proposed ADON block utilizes feature modulation to efficiently connect multiple generations in the sequential ensemble. Sequential ensembles demonstrate state-of-the-art results on challenging datasets. We hope that our work will inspire future research in ensembling for semantic segmentation as well as other dense prediction tasks such as depth and pose estimation.
Figure 7: Qualitative comparison of Sequential Ensembles and Simple Ensembles. Cityscapes (first three rows) and ADE20K (last six rows) for HRNet-W48(N=2). The white eclipses highlight the fine-grained details that our approach captures in comparison to the baseline.
Figure 8: Qualitative comparison of Sequential Ensembles (SEQ-ENS) with the Simple Ensembles (SIM-ENS) on Cityscapes val set using HRNet-W48 backbone (N=2). The white eclipses highlight the fine-grained details that our approach captures in comparison to the baseline. Zoom in for details.
Figure 9: Qualitative comparison of Sequential Ensembles (SEQ-ENS) with the Simple Ensembles (SIM-ENS) on ADE20K val set using HRNet-W48 backbone (N=2). SEQ-ENS is especially effective on a complex dataset like ADE-20K with 150 categories. We set a new state-of-art on ADE20K val set. Zoom in for details.
References

[1] 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.

[2] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[5] 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.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[8] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017.

[9] 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 the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[10] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[11] Philipe A Dias and Henry Medeiros. Probabilistic semantic segmentation refinement by monte carlo region growing. *arXiv preprint arXiv:2005.05856*, 2020.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[14] Suso Dzeroski and Bernard Zenko. Is combining classifiers better than selecting the best one? In *Machine Learning*, pages 255–273. Morgan Kaufmann, 2004.

[15] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[16] Stanislaw Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. In *arXiv preprint arXiv:1912.02757*, 2020.

[17] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[18] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetsche. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6. IEEE, 2020.

[19] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017.

[20] Spyros Gidaris and Nikos Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5248–5257, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[23] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[24] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[25] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[26] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Cnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[28] Md Amirul Islam, Shujon Naha, Mrigank Rochan, Neil Bruce, and Yang Wang. Label refinement network for coarse-to-fine semantic segmentation. *arXiv preprint arXiv:1703.00551*, 2017.

[29] Shun Iwase, Xingyu Liu, Rawal Khirdokar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8732–8741, 2020.
[30] Shruti Jadon. A survey of loss functions for semantic segmentation. In 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pages 1–7. IEEE, 2020.

[31] Rawal Khirodkar, Visesh Charu, Amit Agrawal, and Ambrish Tyagi. Multi-hypothesis pose networks: Rethinking top-down pose estimation. arXiv preprint arXiv:2101.11223, 2021.

[32] Rawal Khirodkar and Kris M Kitani. Adversarial domain randomization. arXiv preprint arXiv:1812.00491, 2018.

[33] Rawal Khirodkar and Kris Kitani. Occluded human mesh recovery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1715–1725, 2022.

[34] Rawal Khirodkar, Donghyun Yoo, and Kris Kitani. Domain randomization for scene-specific car detection and pose estimation. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1932–1940. IEEE, 2019.

[35] Taesup Kim, Inchul Song, and Yoshua Bengio. Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition. arXiv preprint arXiv:1707.06065, 2017.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25:1097–1105, 2012.

[38] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9207–9216, 2019.

[39] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), 2017.

[40] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3659–3667, 2016.

[41] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8553–8562, 2020.

[42] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1925–1934, 2017.

[43] Janice Lin, Wen-June Wang, Sheng-Kai Huang, and Hsiang-Chieh Chen. Learning based semantic segmentation for robot navigation in outdoor environment. In 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), pages 1–5. IEEE, 2017.

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030, 2021.

[45] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[46] Yuyi Mao, Jun Zhang, and Khaled B Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12):3590–3605, 2016.

[47] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Posefix: Model-agnostic general human pose refinement network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7773–7781, 2019.

[48] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[49] Ethan Perez, Florian Strub, Harm De Vries, Vincent Du-moulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[50] 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.

[51] Sebastian Ruder. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04746, 2016.

[52] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018.

[53] Robert E Schapire. The strength of weak learnability. Machine learning, 5(2):197–227, 1990.

[54] Robert E Schapire and Yoav Freund. A decision-theoretic generalization of on-line learning and an application to boosting. In Second European Conference on Computational Learning Theory, pages 23–37, 1995.

[55] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. International journal of computer vision, 81(1):2–23, 2009.

[56] Mennatullah Siam, Sara Elkerdawy, Martin Jagersand, and Senthil Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In 2017 IEEE 20th international conference on intelligent transportation systems (ITSC), pages 1–8. IEEE, 2017.

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[58] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514, 2019.

[59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[60] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5229–5238, 2019.

[61] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019.

[62] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, pages I–I. Ieee, 2001.

[63] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In European Conference on Computer Vision, pages 108–126. Springer, 2020.

[64] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 2020.

[65] Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M Kitani, Yair Movshovitz-Attias, and Elad Eban. On the surprising efficiency of committee-based models. arXiv preprint arXiv:2012.01988, 2020.

[66] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. arXiv preprint arXiv:2002.06715, 2020.

[67] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. arXiv preprint arXiv:2006.13570, 2020.

[68] David H. Wolpert. Neural Networks, 5, 1992.

[69] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. High-performance semantic segmentation using very deep fully convolutional networks. arXiv preprint arXiv:1604.04339, 2016.

[70] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition, 90:119–133, 2019.

[71] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv preprint arXiv:2105.15203, 2021.

[72] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122, 2015.

[73] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 472–480, 2017.

[74] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16, pages 173–190. Springer, 2020.

[75] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In European Conference on Computer Vision, pages 489–506. Springer, 2020.

[76] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 7151–7160, 2018.

[77] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2881–2890, 2017.

[78] Ying Zhao, Jun Gao, and Xuezhi Yang. A survey of neural network ensembles. In 2005 International Conference on Neural Networks and Brain, volume 1, pages 438–442. IEEE, 2005.

[79] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6881–6890, 2021.

[80] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[81] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision, 127(3):302–321, 2019.