Improving Gradient Flow with Unrolled Highway Expectation Maximization

Chonghyuk Song, Eunseok Kim, Inwook Shim[*]
Ground Technology Research Institute, Agency for Defense Development
{chonghyuk.song, a18700, iwshim}@add.re.kr

Abstract

Integrating model-based machine learning methods into deep neural architectures allows one to leverage both the expressive power of deep neural nets and the ability of model-based methods to incorporate domain-specific knowledge. In particular, many works have employed the expectation maximization (EM) algorithm in the form of an unrolled layer-wise structure that is jointly trained with a backbone neural network. However, it is difficult to discriminatively train the backbone network by backpropagating through the EM iterations as they are prone to the vanishing gradient problem. To address this issue, we propose Highway Expectation Maximization Networks (HEMNet), which is comprised of unrolled iterations of the generalized EM (GEM) algorithm based on the Newton-Rahpson method. HEMNet features scaled skip connections, or highways, along the depths of the unrolled architecture, resulting in improved gradient flow during backpropagation while incurring negligible additional computation and memory costs compared to standard unrolled EM. Furthermore, HEMNet preserves the underlying EM procedure, thereby fully retaining the convergence properties of the original EM algorithm. We achieve significant improvement in performance on several semantic segmentation benchmarks and empirically show that HEMNet effectively alleviates gradient decay.

1 Introduction

The Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) is a well-established algorithm in the field of statistical learning used to iteratively find the maximum-likelihood solution for latent variable models. It has traditionally been used for a variety of problems, ranging from unsupervised clustering to missing data imputation. With the dramatic rise in adoption of deep learning in the past few years, recent works (Jampani et al. 2018; Hinton, Sabour, and Frosst 2018; Li et al. 2019; Wang et al. 2020; Greff, Van Steenkiste, and Schmidhuber 2017) have aimed to combine the model-based approach of EM with deep neural networks. These two approaches are typically combined by unrolling the iterative steps of the EM as layers in a deep network, which takes as input the features generated by a backbone network that learns the representations. Jointly training this combined architecture discriminatively allows one to leverage the expressive power of deep neural networks and the ability of model-based methods to incorporate prior knowledge of the task at hand, resulting in potentially many benefits (Hershey, Roux, and Weninger 2014).

First, unrolled EM iterations are analogous to an attention mechanism when the underlying latent variable model is a Gaussian Mixture Model (GMM) (Hinton, Sabour, and Frosst 2018; Li et al. 2019). The Gaussian mean estimates capture long-range interactions among the inputs, just like in the self-attention mechanism (Vaswani et al. 2017; Wang et al. 2018; Zhao et al. 2018). Furthermore, EM attention (Li et al. 2019) is computationally more efficient than the original self-attention mechanism, which computes representations as a weighted sum of every point in the input, whereas EM attention computes them as a weighted sum of a smaller number of Gaussian means. The EM algorithm iteratively refines these Gaussian means such that they monotonically increase the log-likelihood of the input features, increasingly enabling them to attend to the fundamental semantics of the input while suppressing its irrelevant noise and details.

Despite the beneficial effects EM has on the forward pass, jointly training a backbone neural network with EM layers is challenging as they are prone to the vanishing gradient problem (Li et al. 2019). This phenomenon, which was first introduced in (Vaswani et al. 2017), is a problem shared by all attention mechanisms that employ the dot-product softmax operation in the computation of attention maps. Skip connections have shown to be remarkably effective at resolving vanishing gradients for a variety of deep network architectures (Srivastava, Greff, and Schmidhuber 2015; He et al. 2016b; Huang et al. 2017; Gers, Schmidhuber, and Cummins 1999; Hochreiter and Schmidhuber 1997; Cho et al. 2014), including attention-based models (Vaswani et al. 2017; Bapna et al. 2018; Wang et al. 2019). However, the question remains as *how* to incorporate skip connections in a way that maintains the underlying EM procedure of converging to a (local) optimum of the data log-likelihood.

In this paper, we aim to address the vanishing gradient problem of unrolled EM iterations while preserving the EM algorithm, thereby retaining its efficiency and convergence properties and the benefits of end-to-end learning. Instead of unrolling EM iterations, we unroll *generalized* EM (GEM) iterations, where the M-step is replaced by one step of the

[*] Inwook Shim is the corresponding author.

Newton-Rahpson method (Lange 1995). This is motivated by the key insight that unrolling GEM iterations introduces weighted skip connections, or highways (Srivastava, Greff, and Schmidhuber 2015), along the depths of the unrolled architecture, thereby improving its gradient flow during backpropgation. The use of Newton's method is non-trivial. Not only do GEM iterations based on Newton's method require minimal additional computation and memory costs compared to the original EM, but they are also guaranteed to improve the data log-likelihood, thereby inheriting the denoising capabilities of the EM algorithm. To demonstrate the effectiveness of our approach, we formulate the proposed GEM iterations as an attention module for existing backbone networks, which we refer to as Highway Expectation Maximization Networks (HEMNet), and evaluate its performance on semantic segmentation, a task that has been shown to benefit from denoising computations (Li et al. 2019).

2 Related Works

Unrolled expectation maximization. With the recent rise in the adoption of deep learning, many works have incorporated modern neural networks with the well-studied EM algorithm to leverage its clustering and filtering capabilities. SSN (Jampani et al. 2018) combines unrolled EM iterations with a neural network to learn task-specific superpixels. EMCaps (Hinton, Sabour, and Frosst 2018) use unrolled EM as an attentional routing mechanism between adjacent layers of the network. EMANet (Li et al. 2019) designs an EM-based attention module that boosts semantic segmentation accuracy of a backbone network. A similar module is used in (Wang et al. 2020) as a denoising filter for fine-grained image classification. On the other hand, NEM (Greff, Van Steenkiste, and Schmidhuber 2017) incorporates *generalized* EM (GEM) (Wu 1983) to learn representations for unsupervised clustering, where the M-step of the original EM is replaced with one gradient ascent step towards improving the data log-likelihood. Unlike original EM and our proposed GEM based on Newton's method, NEM does not guarantee an improvement of the data log-likelihood.

Skip connections. Skip connections are direct connections between nodes of different layers of a neural network that bypass, or skip, the intermediate layers. They help overcome the vanishing gradient problem associated with training very deep neural architectures (Bengio, Simard, and Frasconi 1994) by allowing gradient signals to be directly backpropagated between adjacent layers (He et al. 2016a; Srivastava, Greff, and Schmidhuber 2015; Huang et al. 2017; Hochreiter and Schmidhuber 1997; Cho et al. 2014). Skip connections are particularly crucial to attention-based models, which are also prone to vanishing gradients (Bapna et al. 2018; Wang et al. 2019; Zhang, Titov, and Sennrich 2019). The Transformer (Vaswani et al. 2017) employs an identity skip connection around each of the sub-layers of the network, without which the training procedure collapses, resulting in significantly worse performance (Bapna et al. 2018). Subsequent works (Bapna et al. 2018; Wang et al. 2019) trained deeper models by creating weighted skip connections along the depth of the Transformer's encoder, providing multiple backpropagation paths and improving gradient flow.

Our approach is motivated by the success of the above works in combating vanishing gradients and, in fact, structurally resembles Highway Networks (Srivastava, Greff, and Schmidhuber 2015) and Gated Recurrent Units (Cho et al. 2014). The key difference is that our method introduces skip connections into the network in a way that preserves the underlying EM procedure and by extension its convergence properties and computational efficiency.

3 Preliminaries

3.1 EM Algorithm for Gaussian Mixture Models

The EM algorithm is an iterative procedure that finds the maximum-likelihood solution for latent variable models. They are described by the joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$, where $\mathbf{X}$ and $\mathbf{Z}$ denote the dataset of N observed samples $\mathbf{x}_n$ and the corresponding set of latent variables $\mathbf{z}_n$, respectively, and $\boldsymbol{\theta}$ denotes the model parameters. The Gaussian mixture model (GMM) (Richardson and Green 1997) is a widely used latent variable model that models the distribution of observed data point $\mathbf{x}_n$ as a linear superposition of K Gaussians:

$$p(\mathbf{x}_n) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{1}$$

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \sigma^2\mathbf{I}) \right\}, \tag{2}$$

where the mixing coefficient $\boldsymbol{\pi}_k$, mean $\boldsymbol{\mu}_k$, and covariance $\boldsymbol{\Sigma}_k$ are the parameters for the k-th Gaussian. In this paper, we use a fixed isotropic covariance $\boldsymbol{\Sigma}_k = \sigma^2\mathbf{I}$ and drop the mixing coefficients as done in many real applications. The EM algorithm aims to maximize the resulting log-likelihood function in Eq. (2) by performing coordinate ascent on its evidence lower bound $\mathcal{L}$ (ELBO) (Neal and Hinton 1999):

$$\begin{aligned} \ln p(\mathbf{X}|\boldsymbol{\theta}) &\geq \underbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}_{Q(\boldsymbol{\theta}, q(\mathbf{Z}))} + \underbrace{\sum_{\mathbf{Z}} -q(\mathbf{Z}) \ln q(\mathbf{Z})}_{H(q(\mathbf{Z}))} \\ &= \underbrace{\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \sigma^2\mathbf{I}) - \gamma_{nk} \ln \gamma_{nk}}_{\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\gamma})}, \end{aligned} \tag{3}$$

which is the sum of the expected complete log-likelihood $Q(\boldsymbol{\theta}, q)$ and entropy term $H(q)$, for any arbitrary distribution $q(\mathbf{Z})$ defined by $\sum_k \gamma_{nk} = 1$. By alternately maximizing the ELBO with respect to q and $\boldsymbol{\theta}$ via the E- and M-step, respectively, the log-likelihood is monotonically increased in the process and converges to a (local) optimum:

$$\textbf{E-step}: \quad \gamma_{nk} = \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k^{old}, \sigma^2\mathbf{I})}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j^{old}, \sigma^2\mathbf{I})} \tag{4}$$

$$\textbf{M-step}: \quad \boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}\mathbf{x}_n \tag{5}$$

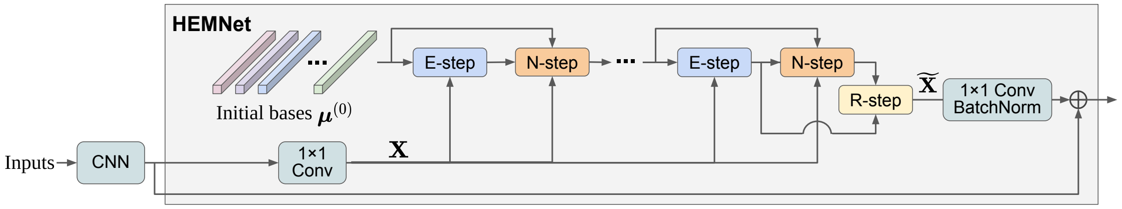

Figure 1: High-level structure of the proposed HEMNet

In the E-step, the optimal q is $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})$, the posterior distribution of $\mathbf{Z}$. Eq. (4) shows the posterior for GMMs, which is described by γ_{nk}, the responsibility that the k^{th} Gaussian basis $\boldsymbol{\mu}_k$ takes for explaining observation $\mathbf{x}_n$. In the M-step, the optimal $\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, q(\boldsymbol{\theta}^{old}))$ since the entropy term of the ELBO isn't dependent on $\boldsymbol{\theta}$. For GMMs, this $\arg\max$ is tractable, resulting in the closed-form of Eq. (5), where $N_k = \sum_{n=1}^{N} \gamma_{nk}$.

When the M-step is *not* tractable, however, we resort to the *generalized* EM (GEM) algorithm, whereby instead of maximizing $Q(\boldsymbol{\theta}, q)$ with respect to $\boldsymbol{\theta}$ the aim is to at least increase it, typically by taking a step of a nonlinear optimization method such as gradient ascent or the Newton-Raphson method. In this paper, we use the GEM algorithm based on the Newton-Raphson method instead for its favorable properties, which are described in section 4.2:

$$\boldsymbol{\theta}^{new} = \boldsymbol{\theta}^{old} - \eta \left[\left(\frac{\partial^2 Q}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}} \right)^{-1} \frac{\partial Q}{\partial \boldsymbol{\theta}} \right]_{\boldsymbol{\theta} = \boldsymbol{\theta}^{old}} \tag{6}$$

3.2 Unrolled Expectation Maximization

In this section, we no longer consider EM iterations as an algorithm, but rather as a sequence of layers in a neural-network-like architecture. This structure, which we refer to as "unrolled EM", is comprised of a pre-defined T number of alternating E- and M-steps, both of which are considered as network layers that take as input the output of its previous step and the feature map $\mathbf{X}$ generated by a backbone CNN. For simplicity, we consider the feature map $\mathbf{X}$ of shape $C \times H \times W$ from a single sample, which we reshape into $N \times C$, where $N = H \times W$.

Given the CNN features $\mathbf{X} \in \mathbb{R}^{N \times C}$ and Gaussian bases from the t-th iteration $\boldsymbol{\mu}^{(t)} \in \mathbb{R}^{K \times C}$, the E-step computes the responsibilities $\boldsymbol{\gamma}^{(t+1)} \in \mathbb{R}^{N \times K}$ according to Eq. (4), which can be rewritten in terms of the RBF kernel $\exp(-||\mathbf{x}_n - \boldsymbol{\mu}_k||_2^2/\sigma^2)$:

$$\gamma_{nk}^{(t+1)} = \frac{\exp(-||\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)}||_2^2/\sigma^2)}{\sum_{j=1}^{K} \exp(-||\mathbf{x}_n - \boldsymbol{\mu}_j^{(t)}||_2^2/\sigma^2)} \tag{7}$$

$$\textbf{E-step:} \quad \boldsymbol{\gamma}^{(t+1)} = \text{softmax}\left(\frac{\mathbf{X}\boldsymbol{\mu}^{(t)\top}}{\sigma^2} \right) \tag{8}$$

As shown in Eq. (8), the RBF kernel can be replaced by the exponential inner dot product $\exp(\mathbf{x}_n^\top \boldsymbol{\mu}_k/\sigma^2)$, which brings little difference to the overall results (Wang et al. 2018; Li et al. 2019) and can be efficiently implemented by a softmax applied to a matrix multiplication operation scaled by the temperature σ^2. The M-step then updates the Gaussian bases according to Eq. (5), which is implemented by a matrix multiplication between normalized responsibilities $\bar{\boldsymbol{\gamma}}$ and CNN features $\mathbf{X}$:

$$\textbf{M-step:} \quad \boldsymbol{\mu}^{(t+1)} = \bar{\boldsymbol{\gamma}}^{(t+1)\top}\mathbf{X}, \tag{9}$$

After unrolling T iterations of EM, the input $\mathbf{x}_n$, whose distribution was modeled as a mixture of Gaussians, is reconstructed as a weighted sum of the converged Gaussian bases, with weights given by the converged responsibilities (Li et al. 2019; Wang et al. 2020). As a result, reconstructing the input features $\widetilde{\mathbf{X}} \in \mathbb{R}^{N \times C}$, which we call the R-step, is also implemented by matrix multiplication:

$$\textbf{R-step:} \quad \widetilde{\mathbf{X}} = \boldsymbol{\gamma}^{(T)}\boldsymbol{\mu}^{(T)} \tag{10}$$

Unrolling T iterations of E- and M-steps followed by one R-step incurs $O(NKTC)$ complexity (Li et al. 2019). However, T can be treated as a small constant for the values used in our experiments, resulting in a complexity of $O(NKC)$.

4 Highway Expectation Maximization Networks

4.1 Vanishing Gradient Problem of EM

Vanishing gradients in unrolled EM layers stem from the E-step's scaled dot-product softmax operation, shown in Eq. (8). This also happens to be the key operation of the self-attention mechanism in the Transformer (Vaswani et al. 2017), which was first proposed to address vanishing gradients associated with softmax saturation; without scaling, the magnitude of the dot-product logits grows larger with increasing number of channels C, resulting in a saturated softmax with extremely small local gradients. Therefore, gradients won't be backpropagated to layers below a saturated softmax. The Transformer counteracts this issue in self-attention layers by setting the softmax temperature $\sigma^2 = \sqrt{C}$, thereby curbing the magnitude of the logits.

However, even with this dot-product scaling operation the Transformer is still prone to gradient vanishing, making it extremely difficult to train very deep models (Bapna et al. 2018; Wang et al. 2019). In fact, the training procedure has shown to even collapse when residual connections are removed from the Transformer (Bapna et al. 2018). To make matters worse, an EM layer only has a single gradient path through $\boldsymbol{\mu}^{(t)}$ that reach lower EM layers, as opposed to the self-attention layer, which backpropagates gradients through multiple paths and therefore has shown to prevent more severe gradient vanishing (Zhang, Titov, and Sennrich 2019).

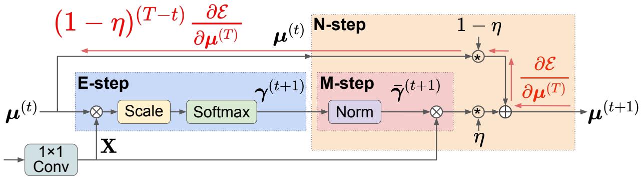
$(1-\eta)^{(T-t)}\,\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}^{(T)}}$
$\boldsymbol{\mu}^{(t)}$
N-step
$1-\eta$
$\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}^{(T)}}$
E-step
$\boldsymbol{\gamma}^{(t+1)}$
M-step
$\bar{\boldsymbol{\gamma}}^{(t+1)}$
$\boldsymbol{\mu}^{(t)}$
Scale
Softmax
Norm
$\boldsymbol{\mu}^{(t+1)}$
η
1×1 Conv
$\mathbf{X}$

Figure 2: Architecture of single HEM layer, which is comprised of one E-step and N-step operation

4.2 Unrolled Highway Expectation Maximization

In order to resolve the vanishing gradient problem in unrolled EM layers, we propose Highway Expectation Maximization Networks (HEMNet), which is comprised of unrolled GEM iterations based on the Newton-Raphson method, as shown in Fig. 1. The key difference between HEMNet and unrolled EM is that the original M-step is replaced by one Newton-Raphson step, or N-step:

N-step:

$$\boldsymbol{\mu}_k^{(t+1)} = \boldsymbol{\mu}_k^{(t)} - \eta\left(\frac{\partial^2\mathcal{Q}}{\partial\boldsymbol{\mu}_k\partial\boldsymbol{\mu}_k}\right)^{-1}\frac{\partial\mathcal{Q}}{\partial\boldsymbol{\mu}_k} \tag{11}$$

$$= \boldsymbol{\mu}_k^{(t)} - \eta\frac{-\sigma^2}{N_k^{(t+1)}}\left\{\sum_{n=1}^{N}\frac{\gamma_{nk}^{(t+1)}}{\sigma^2}\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)}\right)\right\} \tag{12}$$

$$= \underbrace{(1-\eta)\boldsymbol{\mu}_k^{(t)}}_{\text{skip connection}} + \eta\underbrace{\left(\frac{1}{N_k^{(t+1)}}\sum_{n=1}^{N}\gamma_{nk}^{(t+1)}\mathbf{x}_n\right)}_{\mathcal{F}^{EM}\left(\boldsymbol{\mu}_k^{(t)},\mathbf{X}\right)}, \tag{13}$$

where η is a hyperparameter that denotes the step size. For GMMs, the N-step update is given by Eq. (11), which rearranged becomes Eq. (13), a weighted sum of the current $\boldsymbol{\mu}_k^{(t)}$ and the output of one EM iteration $\mathcal{F}^{EM}\left(\boldsymbol{\mu}_k^{(t)},\mathbf{X}\right)$. Interestingly, the original M-step is recovered when $\eta = 1$, implying that the N-step generalizes the EM algorithm.

Eq. (13) is significant as the first term introduces a skip connection that allows gradients to be directly backpropagated to earlier EM layers, thereby alleviating vanishing gradients. Furthermore, the weighted-sum update of the N-step endows HEMNet with two more crucial properties: improving the ELBO in the forward pass and incurring negligible additional space-time complexity compared to unrolled EM.

Backward-pass properties: alleviating vanishing gradients The update equation of Eq. (13) resembles that of the Highway Network (Srivastava, Greff, and Schmidhuber 2015), which contain scaled skip connections, or highways, that facilitate information flow between neighboring layers. HEMNet also contains highways in between its unrolled iterations, or HEM layers, that bypass the gradient-attenuating E-step and allow gradients to be directly sent back to earlier HEM layers, as shown in Fig. 2. To show this, we derive expressions for the upstream gradients to each HEM layer and its (shared) input, by first recursively applying Eq. (13):

$$\boldsymbol{\mu}_k^{(T)} = (1-\eta)^{(T-t)}\boldsymbol{\mu}_k^{(t)} + \left\{\sum_{i=t}^{T-1}\eta(1-\eta)^{(T-i-1)}\mathcal{F}^{EM}\left(\boldsymbol{\mu}_k^{(i)},\mathbf{X}\right)\right\}, \tag{14}$$

and then applying the chain rule to Eq. (14), where $\mathcal{E}$, T, $\hat{\mathcal{F}}_k^{(i)} = \eta(1-\eta)^{(T-i-1)}\mathcal{F}^{EM}\left(\boldsymbol{\mu}_k^{(i)},\mathbf{X}\right)$ are the loss function, number of HEM layers, and shorthand that absorbs the scalars, respectively:

$$\begin{aligned}\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(t)}} &= \frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(T)}}\frac{\partial\boldsymbol{\mu}_k^{(T)}}{\partial\boldsymbol{\mu}_k^{(t)}} \\ &= \frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(T)}}\left\{(1-\eta)^{(T-t)} + \frac{\partial}{\partial\boldsymbol{\mu}_k^{(t)}}\sum_{i=t}^{T-1}\hat{\mathcal{F}}_k^{(i)}\right\}\end{aligned} \tag{15}$$

It can be seen that the upstream gradient to $\boldsymbol{\mu}_k$ is the sum of two gradient terms: a term $(1-\eta)^{(T-t)}\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(T)}}$ directly propagated through the skip connections and a term $\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(T)}}\frac{\partial}{\partial\boldsymbol{\mu}_k^{(t)}}\sum_{i=t}^{T-1}\hat{\mathcal{F}}_k^{(i)}$ propagated through the E-step, which is negligible due to the E-step's vanishing gradient problem and hence can be ignored. This means that as we increase the scalar $(1-\eta)^{(T-t)}$ (by reducing η), the proportion of upstream gradients backpropagated to earlier HEM layers increases as well. Furthermore, it can be seen that when $\eta = 1$ (the original EM case) the skip connection term in Eq. (15) vanishes and leaves only gradients propagated through the gradient-attenuating E-step, highlighting the vanishing gradient problem of unrolled EM.

One consequence of Eq. (15) is that as η decreases, the backbone network parameters will be increasingly influenced by earlier HEM layers, as shown in the following derivation of the upstream gradients to input point $\mathbf{x}_n$, which is generated by the backbone network:

$$\frac{\partial\mathcal{E}}{\partial\mathbf{x}_n} = \sum_{t=1}^{T}\frac{\partial\mathcal{E}}{\partial\mathbf{x}_n^{(t)}} \tag{16}$$

$$= \sum_{t=1}^{T}\sum_{k=1}^{K}\underbrace{\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(t)}}\frac{\partial\boldsymbol{\mu}_k^{(t)}}{\partial\mathbf{x}_n}}_{\text{grad. from N-step}} + \underbrace{\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(t)}}\frac{\partial\boldsymbol{\mu}_k^{(t)}}{\partial\gamma_{nk}^{(t)}}\frac{\partial\gamma_{nk}^{(t)}}{\partial\mathbf{x}_n}}_{\text{grad. from E-step}} \tag{17}$$

$$\approx \sum_{t=1}^{T}\sum_{k=1}^{K}\left\{(1-\eta)^{(T-t)}\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(T)}}\right\}\left(\eta\frac{\gamma_{nk}^{(t)}}{N_k^{(t)}}\right) \tag{18}$$

Eq. (17) shows that, ignoring the gradients from the E-step, $\frac{\partial\mathcal{E}}{\partial\mathbf{x}_n}$ becomes a weighted sum of $\frac{\partial\boldsymbol{\mu}_k^{(t)}}{\partial\mathbf{x}_n}$ weighted by $\frac{\partial\mathcal{E}}{\partial\boldsymbol{\mu}_k^{(t)}}$, which is substituted with Eq. (15). As η is reduced, the upstream gradients to earlier HEM layers grow relatively larger in magnitude, meaning that the loss gradients with respect to $\mathbf{x}_n$ become increasingly dominated by earlier HEM layers. Therefore, the backbone network can potentially learn better representations as it takes into account the effect of its parameters on not only the final HEM layers but also on earlier HEM layers, where most of the convergence of the EM procedure occurs. A full derivation is given in Appendix A.

η T_{train}	0.1	0.2	0.4	0.8	EMANet
1	77.17	77.50	77.92	77.26	77.31
2	77.16	77.80	77.50	**78.10**	**77.55**
3	77.05	77.82	77.81	77.94	76.51
4	77.64	77.73	**78.11**	77.10	76.63
6	77.46	**78.22**	77.83	77.40	76.26
8	77.48	78.14	77.60	78.04	76.48
12	**77.74**	78.11	77.73	77.84	76.17
16	77.64	77.65	77.81	77.06	76.29

Table 1: Ablation study on training iteration number T_{train} and step size η on PASCAL VOC. The rightmost column denotes EMANet (Li et al. 2019), where $\eta = 1.0$.

η T_{eval}	0.1	0.2	0.4	0.8	EMANet
1	64.52	70.30	73.20	77.16	76.26
2	70.72	75.50	76.94	78.10	77.55
3	73.72	77.33	77.85	78.34	**77.73**
4	75.34	77.96	78.11	**78.42**	77.70
6	76.88	**78.22**	78.24	78.37	77.50
8	77.47	78.13	**78.24**	78.30	77.37
12	77.74	77.98	78.19	78.26	77.21
16	**77.84**	77.90	78.17	78.23	77.16

Table 2: Ablation study on evaluation iteration number T_{eval}. For each η, we perform ablations on the best T_{train} (underlined). The best T_{eval} is highlighted in bold.

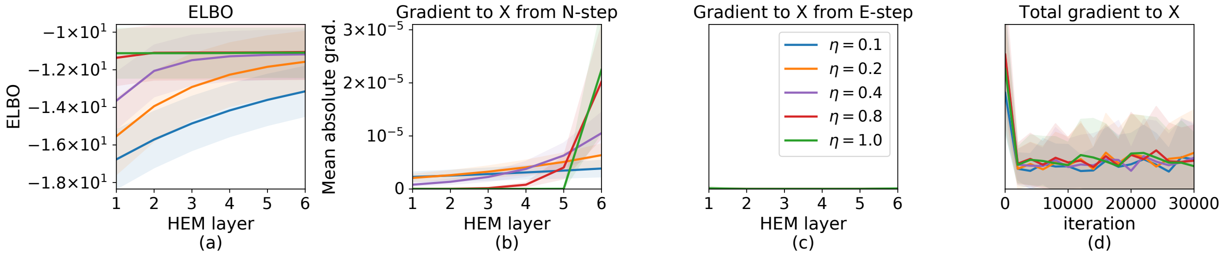

Figure 3: The effects of changing η on (a) the ELBO, (b) the gradient to input $\mathbf{X}$ from N-step and (c) from E-step at the beginning of training, and (d) the total gradient to $\mathbf{X}$ during training. The E-step gradients in (c) are on the order of 10^{-7} and is therefore not displayed. The mean absolute gradients were computed by backpropagating with respect to the same set of 100 randomly chosen training samples, every 2000 training iterations.

Forward-pass properties: improvement of ELBO In the forward pass, the N-step increases the ELBO for fractional step sizes, as shown in the following proposition:

Proposition 1. *For step size $\eta \in (0, 1]$, the N-step of a HEM iteration given by Eq. (13) updates $\boldsymbol{\mu}$ such that*

$$\mathcal{L}(\boldsymbol{\mu}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}) > \mathcal{L}(\boldsymbol{\mu}^{(t)}, \boldsymbol{\gamma}^{(t+1)}), \tag{19}$$

unless $\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)}$, where $\mathcal{L}$ is the ELBO for GMMs in Eq. (3), $\boldsymbol{\gamma}^{(t+1)}$ are the responsibilities computed by the E-step using $\boldsymbol{\mu}^{(t)}$ in Eq. (4), and t is the iteration number.

The proof is given in Appendix B. Since HEMNet is comprised of alternating E-steps and N-steps, it increases the ELBO, just as does the original unrolled EM. This is a nontrivial property of HEMNet. Increasing the ELBO converges the unrolled HEM iterations to a (local) maximum of the data-log likelihood. In other words, with every successive HEM iteration, the updated Gaussian bases $\boldsymbol{\mu}_k$ and attention weights γ_{nk} can better reconstruct the original input points, as their important semantics of the input increasingly distilled into the GMM parameters.

Computational complexity Eq. (13) shows that computing the N-step update requires one additional operation to the original M-step: a convex summation of the M-step output $\mathcal{F}^{EM}\left(\boldsymbol{\mu}^{(t)}, \mathbf{X}\right)$ and the current $\boldsymbol{\mu}^{(t)}$ estimate. This operation incurs minimal additional computation and memory costs compared to the matrix multiplications in the E- and M-step, which dominates unrolled EM. Therefore, HEMNet has $\mathcal{O}(NKC)$ space-time complexity, as does unrolled EM.

5 Experiments

5.1 Implementation Details

We use ResNet (He et al. 2016a) pretrained on ImageNet (Russakovsky et al. 2015) with multi-grid (Chen et al. 2017) as the backbone network. We use ResNet-50 with an output stride (OS) of 16 for all ablation studies and Resnet-101 with OS = 8 for comparisons with other state-of-the-art approaches. We set the temperature $\sigma^2 = \sqrt{C}$ following (Vaswani et al. 2017), where C is the number of input channels to HEMNet, which is set to 512 by default. The step size is set to $\eta = 0.5$ for PASCAL VOC (Everingham et al. 2010) and PASCAL Context (Mottaghi et al. 2014), and $\eta = 0.6$ for COCO Stuff (Caesar, Uijlings, and Ferrari 2018). We set the training iteration number to $T_{train} = 3$ for all datasets. We use the moving average mechanism (Ioffe and Szegedy 2015; Li et al. 2019) to update the initial Gaussian bases $\boldsymbol{\mu}^{(0)}$. We adopt the mean Intersection-over-Union (mIoU) as the performance metric for all experiments across all datasets. Further details, including the training regime, are outlined in Appendix C.

5.2 Ablation Studies

Ablation study on step size In Fig. 3, we analyze the effects of the step size η on the ELBO in the forward pass and the gradient flow in the backward pass. It can be seen that as η is reduced, the convergence of the ELBO slows down, requiring more unrolled HEM layers than would otherwise

Method	SS	MS	FLOPs	Memory	Params
ResNet-101	-	-	370G	6.874G	40.7M
DeeplabV3+	77.62	78.72	+142G	+318M	+16.0M
PSANet	78.51	79.77	+185G	+528M	+26.4M
EMANet*	79.73	80.94	**+45.2G**	**+236M**	**+5.15M**
HEMNet*	80.93	81.44	**+45.2G**	**+236M**	**+5.15M**
EMANet**	80.05	81.32	+92.3G	+329M	+10.6M
HEMNet**	**81.33**	**82.23**	+92.3G	+331M	+10.6M

Table 3: Comparisons on PASCAL VOC val set in mIoU (%). All results are computed for a ResNet-101 backbone, where OS = 8 for training and evaluation. FLOPs and memory are computed for input size of 513 × 513. SS: Single-scale input testing. MS: Multi-scale input testing augmented by left-right flipped inputs. * and ** denote 256 and 512 input channels, respectively, to EMANet and HEMNet.

Method	Backbone	mIoU (%)
DeeplabV3 (Chen et al. 2017)	ResNet-101	85.7
PSANet (Zhao et al. 2018)	ResNet-101	85.7
EncNet (Zhang et al. 2018a)	ResNet-101	85.9
Exfuse (Zhang et al. 2018b)	ResNet-101	86.2
SDN (Fu et al. 2019)	ResNet-101	86.6
CFNet (Zhang et al. 2019)	ResNet-101	87.2
EMANet (Li et al. 2019)	ResNet-101	87.7
HEMNet	ResNet-101	**88.0**

Table 4: Comparisons on the PASCAL VOC test set.

(Fig. 3a) . On the other hand, the gradients backpropagated to input $\mathbf{X}$ from each HEM layer, which is dominated by the N-step, become more evenly distributed as a greater proportion of upstream gradients are sent to earlier HEM layers (Fig. 3b). The subtlety here is that reducing η does not seem to necessarily increase the magnitude of the total upstream gradients to $\mathbf{X}$ (Fig. 3d), since the gradients sent back to $\mathbf{X}$ from the t-th HEM layer is proportional to $\eta(1-\eta)^{(T-t)}$, as shown in Eq. (18). This suggests that the change in performance from changing η is likely due to the resulting change in relative weighting among the different HEM layers when computing the gradients with respect to $\mathbf{x}_n$, not the absolute magnitude of those gradients.

In other words, there is a trade-off, controlled by η, between the improvement of the ELBO in the forward pass and how evenly the early and later HEM layers contribute to backpropagating gradients to $\mathbf{X}$ in the backward pass. Table 1 shows this trade-off, where the best performance for each value of T_{train} is achieved by intermediate values of η, suggesting that they best manage this trade-off. Furthermore, the best η decreases for larger T_{train}, since later HEM layers become increasingly redundant as they converge to an identity mapping of the GMM parameter estimates as the underlying EM procedure converges as well. Decreasing η reduces the relative weighting on these redundant layers by increasing the proportion of upstream gradients sent to earlier HEM layers, resulting in potentially better-learned representations.

Method	Backbone	mIoU (%)
MSCI (Lin et al. 2018)	ResNet-152	50.3
SGR (Liang et al. 2018)	ResNet-101	50.8
CCL (Ding et al. 2018)	ResNet-101	51.6
EncNet (Zhang et al. 2018a)	ResNet-101	51.7
DANet (Fu et al. 2019)	ResNet-101	52.6
ANLNet (Zhu et al. 2019)	ResNet-101	52.8
EMANet (Li et al. 2019)	ResNet-101	53.1
CFNet (Zhang et al. 2019)	ResNet-101	54.0
HEMNet	ResNet-101	**54.3**

Table 5: Comparisons on the PASCAL Context test set.

Method	Backbone	mIoU (%)
RefineNet (Lin et al. 2017)	ResNet-101	33.6
CCL (Ding et al. 2018)	ResNet-101	35.7
ANLNet (Zhu et al. 2019)	ResNet-101	37.2
SGR (Liang et al. 2018)	ResNet-101	39.1
DANet (Fu et al. 2019)	ResNet-101	39.7
EMANet (Li et al. 2019)	ResNet-101	39.9
HEMNet	ResNet-101	**40.1**

Table 6: Comparisons on the COCO Stuff test set.

Ablation study on iteration number We further investigate the effect of changing the training iteration number T_{train}. It can be seen in Table 1 that for all step sizes η the mIoU generally increases with the T_{train} up to a certain point, after which it decreases. This is likely attributed to the vanishing gradient problem as an exponentially less proportion of the upstream gradients reach earlier HEM layers as T_{train} increases, meaning that the backbone network parameters are increasingly influenced by the later HEM layers, which amounts to a mere identity mapping of the GMM parameter estimates for high values of T_{train}. This is corroborated by the observation that the performance peaks at larger T_{train} for smaller η, which can be explained by the fact that smaller η slows down the exponential decay of the upstream gradients to earlier HEM layers, allowing us to unroll more HEM layers. In the case of EMANet (Li et al. 2019) the performance peaks at a low value of $T_{train} = 2$, most likely because gradients aren't backpropagated through the unrolled EM iterations, preventing the rest of the network from learning representations optimized for the EM procedure.

Table 2 shows the effect of changing the evaluation iteration number, *after* training. It can be seen that for all values of η except 0.2, increasing T beyond T_{train} during evaluation, where vanishing gradients is no longer an issue, can further improve performance. The observation can be attributed to the improvement in the ELBO with more HEM layers, which is consistent with previous findings (Jampani et al. 2018; Li et al. 2019). We suspect that the reason for the performance deterioration at high values of T_{eval} is that the Gaussian bases have not fully converged at the chosen values of T_{train} and that there is still room for the Gaussian bases' norms to change, making it difficult for HEMNet to generalize beyond its training horizon (David Krueger 2016).

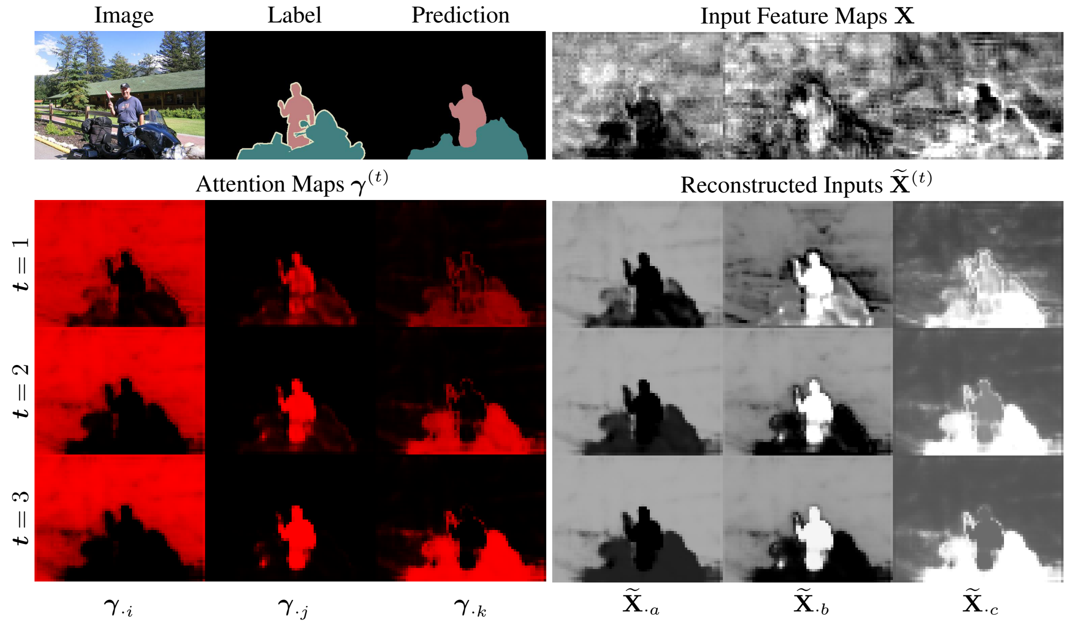
Image
Label
Prediction
Input Feature Maps $\mathbf{X}$
Attention Maps $\boldsymbol{\gamma}^{(t)}$
Reconstructed Inputs $\widetilde{\mathbf{X}}^{(t)}$
$t = 1$
$t = 2$
$t = 3$
$\gamma_{\cdot i}$
$\gamma_{\cdot j}$
$\gamma_{\cdot k}$
$\widetilde{\mathbf{X}}_{\cdot a}$
$\widetilde{\mathbf{X}}_{\cdot b}$
$\widetilde{\mathbf{X}}_{\cdot c}$

Figure 4: Visualization of the attention maps γ, input feature maps $\mathbf{X}$ from the backbone CNN, and reconstructed inputs $\widetilde{\mathbf{X}} = \gamma\boldsymbol{\mu}$ for a sample image from the PASCAL VOC 2012 val set. The images in the top-left corner contain the original image, label, and the prediction made by HEMNet with a ResNet-101 backbone. The three rows below show the attention maps (left) and reconstructed inputs (right) at each HEM iteration. $\gamma_{\cdot k}$ denotes the attention map w.r.t. the k-th Gaussian basis $\boldsymbol{\mu}_k$ and $\widetilde{\mathbf{X}}_{\cdot c}$ denotes the c-th channel of the reconstructed input, where $1 \leq i, j, k \leq K$ and $1 \leq a, b, c \leq C$.

5.3 Comparisons with State-of-the-arts

We first compare our approach to EMANet (Li et al. 2019) and other baselines such as DeeplabV3+ (Chen et al. 2018) on the PASCAL VOC validation set. Table 3 shows that HEMNet outperforms all baselines by a large margin. Most notably, HEMNet outperforms EMANet while incurring virtually the same computation and memory costs and using the same number of parameters. Furthermore, HEMNet with 256 input channels exceeds EMANet with 512 input channels, and HEMNet's single-scale (SS) performance is on par with EMANet's *multi-scale* (MS) performance, which is a robust method for improving semantic segmentation accuracy (Zhao et al. 2017; Chen et al. 2018; Li et al. 2019). We also display significant improvement over EMANet on the PASCAL VOC, PASCAL Context and COCO Stuff test set, as shown in Table 4, Table 5, and Table 6.

5.4 Visualizations

In Fig. 4, we demonstrate the effect of improving the ELBO in HEMNet by visualizing the responsibilities γ, which act as attention weights, and the feature map $\widetilde{\mathbf{X}}$ reconstructed from those attention weights. It can be seen that the attention map with respect to each Gaussian basis attends to a specific aspect of the input image, suggesting that each basis corresponds to a particular semantic concept. For instance, the attention maps with respect to bases i, j, and k highlights the background, person, and motorbike, respectively.

Furthermore, after every HEM iteration the attention maps grow sharper and converge to the underlying semantics of each basis. As a result, every feature map reconstructed from the converged attention maps γ and Gaussian bases $\boldsymbol{\mu}$ recovers the fundamental semantics of the noisy input $\mathbf{X}$, while suppressing irrelevant concepts and details. The denoising capability stemming from this underlying EM procedure is the reason why HEMNet was applied to semantic segmentation, a task that requires denoising unnecessary variations in the input image to capture its fundamental semantics (Li et al. 2019). HEMNet facilitates this denoising of the backbone CNN features as the module's constituent HEM iterations and R-step, which estimates the optimal GMM parameters and reconstructs $\mathbf{X}$ from these estimates, respectively, amount to a special case of the GMM Symmetric Smoothing Filter (Chan, Zickler, and Lu 2017).

6 Conclusion

In this work, we proposed Highway Expectation Maximization Networks (HEMNet) in order to address the vanishing gradient problem present in expectation maximization (EM) iterations. HEMNet is comprised of unrolled iterations of the generalized EM algorithm based on the Newton-Rahpson method, which introduces skip connections that ameliorate gradient flow while preserving the underlying EM procedure and incurring negligible additional computation and memory costs. We performed extensive experiments on several semantic segmentation benchmarks to demonstrate that our approach effectively alleviates vanishing gradients and enables better performance.

Ethical Impact

Our research can be broadly described as an attempt to leverage what has been two very successful approaches to machine learning: model-based methods and deep neural networks. Deep learning methods have shown state-of-the-art performance in a variety of applications due to their excellent ability for representation learning, but they are considered as black box methods, making it extremely difficult to interpret their inner workings or incorporate domain knowledge of the problem at hand. Combining deep neural networks with well-studied, classical model-based approaches provide a straightforward way to incorporate problem specific assumptions, such as those from the physical world like three-dimensional geometry or visual occlusion. Furthermore, this combined approach also provides a level of interpretability to the inner workings of the system, through the lens of well-studied classical methods. In our paper we consider a semantic segmentation task, which has become a key component of the vision stack in autonomous driving technology. Combining deep neural networks with the model-based approaches can be crucial for designing and analyzing the potential failure modes of such safety-critical applications of deep-learning based systems. For instance in our approach, the Gaussian bases converge to specific semantics with every passing EM iteration, and the corresponding attention maps give an indication of the underlying semantics of a given input image. This can help researchers better understand what the system is actually learning, which can be invaluable, for example, when trying to narrow down the reason for an accident induced by a self-driving vehicle. We hope that our work encourages researchers to incorporate components inspired by classical model-based machine learning into existing deep learning architectures, enabling them to embed their domain knowledge into the design of the network, which in turn may provide another means to better interpret its inner mechanism.

References

Bapna, A.; Chen, M.; Firat, O.; Cao, Y.; and Wu, Y. 2018. Training deeper neural machine translation models with transparent attention. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3028–3033.

Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5(2): 157–166.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1209–1218.

Chan, S. H.; Zickler, T.; and Lu, Y. M. 2017. Understanding symmetric smoothing filters: A gaussian mixture model perspective. *IEEE Transactions on Image Processing (TIP)* 26(11): 5107–5121.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* .

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 801–818.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

David Krueger, R. M. 2016. Regularizing rnns by stabilizing activations. In *International Conference on Learned Representations (ICLR)*.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B* 39(1): 1–38.

Ding, H.; Jiang, X.; Shuai, B.; Liu, A. Q.; and Wang, G. 2018. Context contrasted feature and gated multi-scale aggregation for scene segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2393–2402.

Everingham, M.; Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)* 88(2): 303–338.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3146–3154.

Fu, J.; Liu, J.; Wang, Y.; Zhou, J.; Wang, C.; and Lu, H. 2019. Stacked deconvolutional network for semantic segmentation. *IEEE Transactions on Image Processing (TIP)* 1–1.

Gers, F. A.; Schmidhuber, J.; and Cummins, F. 1999. Learning to forget: Continual prediction with lstm. In *International Conference on Artificial Neural Networks (ICANN)*, 850–855.

Greff, K.; Van Steenkiste, S.; and Schmidhuber, J. 2017. Neural expectation maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6691–6701.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 630–645.

Hershey, J. R.; Roux, J. L.; and Weninger, F. 2014. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574* .

Hinton, G. E.; Sabour, S.; and Frosst, N. 2018. Matrix capsules with em routing. In *International Conference on Learned Representations (ICLR)*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8): 1735–1780.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 448–456.

Jampani, V.; Sun, D.; Liu, M.-Y.; Yang, M.-H.; and Kautz, J. 2018. Superpixel sampling networks. In *European Conference on Computer Vision (ECCV)*, 352–368.

Lange, K. 1995. A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 57(2): 425–437.

Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; and Liu, H. 2019. Expectation-maximization attention networks for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 9167–9176.

Liang, X.; Hu, Z.; Zhang, H.; Lin, L.; and Xing, E. P. 2018. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1853–1863.

Lin, D.; Ji, Y.; Lischinski, D.; Cohen-Or, D.; and Huang, H. 2018. Multi-scale context intertwining for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 603–619.

Lin, G.; Milan, A.; Shen, C.; and Reid, I. 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1925–1934.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 891–898.

Neal, R. M.; and Hinton, G. E. 1999. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, 355–368. Cambridge, MA, USA: MIT Press.

Richardson, S.; and Green, P. J. 1997. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)* 59(4): 731–792.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)* 115(3): 211–252.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Training very deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2377–2385.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 5998–6008.

Wang, Q.; Li, B.; Xiao, T.; Zhu, J.; Li, C.; Wong, D. F.; and Chao, L. S. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787* .

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7794–7803.

Wang, Z.; Wang, S.; Yang, S.; Li, H.; Li, J.; and Li, Z. 2020. Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, C. J. 1983. On the convergence properties of the em algorithm. *The Annals of statistics* 95–103.

Zhang, B.; Titov, I.; and Sennrich, R. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 898–909.

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018a. Context encoding for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7151–7160.

Zhang, H.; Zhang, H.; Wang, C.; and Xie, J. 2019. Co-occurrent features in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 548–557.

Zhang, Z.; Zhang, X.; Peng, C.; Xue, X.; and Sun, J. 2018b. Exfuse: Enhancing feature fusion for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2881–2890.

Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C. C.; Lin, D.; and Jia, J. 2018. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision (ECCV)*, 267–283.

Zhu, Z.; Xu, M.; Bai, S.; Huang, T.; and Bai, X. 2019. Asymmetric non-local neural networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 593–602.