Probabilistic Attention for Interactive Segmentation

Prasad Gabbur
Apple
pgabbur@apple.com

Manjot Bilkhu
Apple
mbilkhu@apple.com

Javier Movellan
Apple
movellan@apple.com

Abstract

We provide a probabilistic interpretation of attention and show that the standard dot-product attention in transformers is a special case of Maximum A Posteriori (MAP) inference. The proposed approach suggests the use of Expectation Maximization algorithms for online adaptation of key and value model parameters. This approach is useful for cases in which external agents, e.g., annotators, provide inference-time information about the correct values of some tokens, e.g., the semantic category of some pixels, and we need for this new information to propagate to other tokens in a principled manner. We illustrate the approach on an interactive semantic segmentation task in which annotators and models collaborate online to improve annotation efficiency. Using standard benchmarks, we observe that key adaptation boosts model performance ($\sim 10\%$ mIoU) in the low feedback regime and value propagation improves model responsiveness in the high feedback regime. A PyTorch layer implementation of our probabilistic attention model is available here: https://github.com/apple/ml-probabilistic-attention.

1 Introduction

Attention was first introduced as a computational primitive for natural language processing [53] and has since been widely adopted [17, 63, 13, 14] as a replacement for recurrent primitives such as LSTMs [26]. More recently it has been making inroads into computer vision [43, 69, 57, 64, 5, 19, 49] as a replacement for the long accepted convolution as the main computational primitive. Self-attention based architectures have demonstrated state-of-the-art results in fundamental vision problems including image classification [5, 43, 69, 19, 49], object detection [8, 71, 57], image and video semantic segmentation [55, 29, 57, 41] and tracking [66] to state a few.

There are a few different perspectives on the reasons for success of self-attention in computer vision and its superiority over convolution. This includes a view that the self-attention mechanism allows modeling spatially varying dynamic convolution filters [32] and at the same time enabling parameter independent scaling of receptive fields [52]. Another includes their ability to capture global context through long range interactions especially when full attention is feasible [49] at reduced spatial resolution maps or using an approximation of full attention with axial [55] or criss-cross attention [29]. A recent work [45] introduces modern Hopfield networks with continuous states where the update mechanism is shown to be equivalent to the update mechanism of standard dot-product attention [53]. They show that such a network has the capacity to store exponentially many patterns and retrieve them with high fidelity. In this work, we provide a novel interpretation of attention as a probabilistic generative model for queries and values. Specifically we hypothesize the existence of a bank of probabilistic memory units, each of which maintains a joint probability distribution over queries and values parameterized through keys. A query/value pair is generated by first sampling a unit (from a prior over units) followed by sampling the pair from the unit specific joint distribution. This is equivalent to generating the queries and values through a probabilistic mixture model over the units. A particular form for unit joint likelihoods expressed as Gaussians for both the query and value marginals, assuming their independence conditioned on a unit, turns out to be equivalent to traditional

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

dot product attention under a few constraints. As shown in Section 3.3, maximum likelihood (ML) inference for the corresponding value given a query is equivalent to standard dot-product attention.

Our probabilistic interpretation provides a systematic framework for online update of mixture model parameters based on a set of observed queries. It also allows propagation of correct values provided by an external agent for some of the units to all other units. Using Bayesian inference in the constrained case, we derive update rules for *online unsupervised adaptation* (Section 3.5) of query/key likelihood parameters based on a set of observed queries. We also derive update equations for *online value propagation* (Section 3.6) across units based on fixed externally specified values for a subset of units. The latter is specifically useful for interactive segmentation where a correction provided by an annotator has to be propagated globally to make the process more efficient. We use probabilistic attention in place of standard attention in deep architectures for interactive segmentation both within the backbone and at the network head as a classifier. Specifically we use probabilistic attention updates in the BoTNet50 [49] architecture and show that adapting keys to incoming queries leads to better model performance in the low annotator feedback regime. Using value propagation within a probabilistic attention layer at the head of the segmentation network leads to a more responsive model through effective feedback propagation in the high feedback regime. We also use both key adaptation and value propagation together and demonstrate the complementary effects of the two in both the low and high annotator feedback regimes.

2 Related Work

2.1 Attention

Natural language processing has seen the rise [3, 53] and widespread adoption of attention in recent years [17, 63, 13, 14, 59]. One of the first works on visual attention was on learning to attend to image regions for caption generation [60]. Since then there has been a steady progress on using attention primitives within vision models for recognition and classification [27, 57, 5, 54, 43, 69, 19, 52, 49], detection [8, 71], segmentation [55, 29], tracking [66] and video analysis [42, 41]. There have been numerous works interpreting the attention mechanism as a form of computing non-local means [7, 57], approximating dynamic convolution filters [32, 52], and capturing global context through long range interactions [27, 43, 55]. The standard dot-product attention [53] update was also formulated as emerging from the update rule of modern Hopfield networks [45]. Our work introduces attention mechanism from a novel perspective as that of inferring from a probabilistic memory bank. To our knowledge, the only work that is closest to our approach is [18], which also proposes a similar interpretation but only for the queries in order to study the explaining away effect of attention update. Other notable works modeling attention in a probabilistic framework have done so using neural processes [25] or variational approximations for modeling alignments as latent variables [16] or to overcome the bypassing phenomenon of deterministic attention [4] in sequence tasks. Being different from these approaches, our formulation encapsulates queries and values in a single generative model and provides an interpretation of standard dot-product attention as constrained Bayesian inference. Doubly normalized attention scheme [18] also emerges as a special case of key adaptation in our framework.

2.2 Interactive Segmentation

Deep neural networks have set state-of-the art in semantic segmentation through the use of fully convolutional architectures [36, 70, 10, 11, 12, 50, 56, 67] and more recently using hybrid convolution and self-attention [64] or stand-alone self-attention architectures [55]. The input domain of interactive segmentation includes user input in the form of clicks or scribbles in addition to the visual signal (images or videos). The earliest works in interactive segmentation were based on algorithmic approaches for incorporating human inputs into region [46, 51, 65] or boundary [40, 20] processing pipelines. [44] provides a comprehensive survey of interactive segmentation approaches. More recently deep networks have been used to incorporate user feedback to guide their output predictions at the pixel level. Following a similar taxonomy, these can be roughly categorized into region based [62, 61, 37, 33, 6, 2, 28, 68, 34, 31] or boundary based approaches [9, 1, 35, 58]. Deep Extreme Cut (DEXTR) [38] demonstrated that user guidance in the form of extreme points could be used in addition to the input channels to accurately localize the object of interest. More recently [68] argued that three points are sufficient as input guidance to localize the object but additional corrective clicks could be used to further refine the prediction. Other works have used the corrective clicks to adapt the network inputs [30], embeddings [48] or parameters [31] online. Different from these previous approaches, we use corrective clicks as providing fixed values for a subset of units in the proposed probabilistic attention framework. These values are propagated globally through the attention mechanism to directly and more effectively influence the outputs towards user intended values.

3 Method

We provide a probabilistic interpretation of attention as a generative model for queries and values through a set of memory units. Using this formulation, traditional attention in transformers [53] reduces to the special case of maximum a posteriori (MAP) inference of values given queries, assuming Gaussians for the likelihoods. Using Bayesian inference, we provide a systematic approach to adapt keys online as a locally ML update of the corresponding model parameters. Our formulation also allows to fix the values of certain units and propagate their influence to other units online by conditioning on the fixed unit values. The following sections provide more details on the probabilistic model.

3.1 Probabilistic attention

We assume that there are n memory units, indexed by i, each of which can be queried through a vector $q_i \in R^d$ to yield an output value vector $v_i \in R^m$. The queries and the corresponding values may depend on an input x. For example, each memory unit may represent a pixel x_i in an image x. The joint distribution of queries q_i and values v_i conditioned on the input x is assumed to factorize over memory units

$$p(q_{1:n}, v_{1:n}|x) = \Pi_{i=1}^{n} p_i(q_i, v_i \mid x), \tag{1}$$

where x is the conditioning input, $q_{1:n} = \{q_1, \cdots, q_n\}$, $v_{1:n} = \{v_1, \cdots, v_n\}$, and $q_i \in R^d, v_i \in R^m$ are the query, value vectors for unit i respectively. The per-unit joint likelihood $p_i(q_i, v_i \mid x)$ is a probabilistic mixture model given by

$$p_i(q, v \mid x) = \sum_{j=1}^{n} p_i(q, v, u_j \mid x) = \sum_{j=1}^{n} \pi_{i,j}(x)\, p_i(q, v \mid u_j, x), \tag{2}$$

where we have dropped the subscript i from q and v for simplicity. In the above, u_j indexes unit j, $\pi_{i,j}(x)$ is the probability of activating unit j when unit i is queried, $p_i(q, v \mid u_j, x)$ is the likelihood of observing the pair (q, v), given the pair is generated through unit j in the mixture, conditioned on the input x.

3.2 Value inference

Using the above model, it is possible to find the most likely value $\hat{v}$ given a query q to unit i

$$\hat{v} = \underset{v}{\operatorname{argmax}}\, p_i(v|q, x). \tag{3}$$

We use Expectation Maximization (EM) [15] to achieve this: starting with an initial estimate v^0 of the most probable value and iterating over the standard EM auxiliary function Q_i. Given the latest known estimate v^t, the M step produces a new estimate v^{t+1} that increases Q_i by maximizing it w.r.t. v^{t+1}. This guarantees local maximization of $p_i(v|q, x)$.

$$Q_i(v^t, v^{t+1} \mid x) = \sum_j w_{i,j}^t \log p_i(u_j, q, v^{t+1} \mid x), \tag{4}$$

where

$$w_{i,j}^t = p_i(u_j \mid q, v^t, x) = \frac{\pi_{i,j}(x)\, p_i(q, v^t \mid u_j, x)}{\sum_j \pi_{i,j}(x)\, p_i(q, v^t \mid u_j, x)}. \tag{5}$$

The $n \times n$ matrix formed by the entries $w_{i,j}$ corresponds to the *attention* matrix in standard transformers. The optimal value $\hat{v}$ is obtained by taking the gradient with respect to v^{t+1} and setting it to zero

$$\nabla_{v^{t+1}} Q_i(v^t, v^{t+1} \mid x) = \sum_j w_{i,j} \nabla_{v^{t+1}} \log p_i(q, v^{t+1}, u_j \mid x), \tag{6}$$

where $\nabla_{v^{t+1}} \log p_i(q, v_{t+1}, u_j \mid x)$ is the Fisher Score for unit i with respect to v^{t+1}.

3.3 Relationship to standard attention

We show that standard attention in transformers [53] solves Eq. (6) under the special case of a constrained Gaussian mixture model (GMM). Assuming isotropic Gaussians with conditionally independent queries and values given input and mixture component

$$p_i(q, v \mid u_j, x) = p_i(q \mid u_j, x)\, p_i(v \mid u_j, x) \tag{7}$$

$$p_i(q \mid u_j, x) = \Big(\frac{\alpha_j(x)}{2\pi}\Big)^{d/2}\, e^{-\frac{\alpha_j(x)}{2}\|q - \xi_j(x)\|^2} \tag{8}$$

$$p_i(v \mid u_j, x) = \Big(\frac{\beta_j(x)}{2\pi}\Big)^{m/2}\, e^{-\frac{\beta_j(x)}{2}\|v - \mu_j(x)\|^2}, \tag{9}$$

where $\alpha_j(x), \beta_j(x) > 0$ are precision parameters, $\xi_j(x) \in R^d$, $\mu_j(x) \in R^m$ are the key and expected value parameters for unit j given the input x. The dependency of $p_i(q, v \mid u_j, x)$ on x is through the fact that the parameters $\alpha_j, \beta_j, \pi_{i,j}, \xi_j, \mu_j$ are a function of x. For simplicity, we treat x to be fixed and leave the dependency on x implicit in our notation. In order to obtain the standard attention update equation, we constrain the precision parameters to be the same across units: $\alpha_1 = \cdots = \alpha_n = \alpha$, $\beta_1 = \cdots = \beta_n = \beta$, and link the priors of each unit to the lengths of the corresponding key and expected value vectors

$$\pi_{i,j} = \frac{1}{z} e^{\frac{\alpha}{2}\|\xi_j\|^2} e^{\frac{\beta}{2}\|\mu_j\|^2} \tag{10}$$

$$z = \sum_j e^{\frac{\alpha}{2}\|\xi_j\|^2} e^{\frac{\beta}{2}\|\mu_j\|^2}. \tag{11}$$

Assuming $\beta \to 0$ and solving for optimal v^{t+1} in Eq. (6), we obtain the standard attention update (see Appendix B.1)

$$v^{t+1} = \sum_j w_{i,j} \mu_j \tag{12}$$

$$w_{i,j}^t = \frac{e^{\alpha \xi_j^T q}}{\sum_j e^{\alpha \xi_j^T q}}, \tag{13}$$

where each μ_j is the value associated with unit j and v^{t+1} is the output at unit or token i after the attention update. In this case, $w_{i,j}^t$ is no longer a function of t and thus only one EM iteration is needed.

3.4 Offline supervised learning

As is commonly done in standard transformers, the relationship between the input x and the mixture model parameters: $\pi(x), \xi(x), \mu(x)$ can be modeled using a deep network, whose parameters can be trained off-line with task specific supervision.

3.5 Online unsupervised mixture model adaptation

Our framework provides a way to adapt the mixture model parameters based on all the observed input queries prior to doing value inference. This process can be seen as an inference-time adaptation of the model using the additional information contained in the set of queries. We propose an unsupervised Bayesian approach to do this adaptation for the per-unit key vectors $\xi_{1:n} = \{\xi_1, \cdots, \xi_n\}$ and precision parameters $\alpha_{1:n} = \{\alpha_1, \cdots, \alpha_n\}$ given queries $q_{1:n} = \{q_1, \cdots, q_n\}$. For each unit i, the optimal value inference is given by

$$\hat{v}_i = \operatorname*{argmax}_v p_i(v \mid q_{1:n}). \tag{14}$$

Assuming a prior for the key vectors given the observed queries $p(\xi_{1:n}|q_{1:n})$, the likelihood $p_i(v|q_{1:n})$ can be written as

$$p_i(v \mid q_{1:n}) = \int p(\xi_{1:n} \mid q_{1:n}) p_i(v \mid q_i, \xi_{1:n}) d\xi_{1:n} \approx p_i(v \mid q_i, \hat{\xi}_{1:n}), \tag{15}$$

where the expectation over the posterior is approximated by its maximum a posteriori (MAP) value

$$\hat{\xi}_{1:n} = \operatorname*{argmax}_{\xi_{1:n}} p(\xi_{1:n} \mid q_{1:n}) \tag{16}$$

$$\hat{v}_i = \operatorname*{argmax}_v p_i(v \mid q_i, \hat{\xi}_{1:n}). \tag{17}$$

In order to solve (16), we use an iterative EM approach. The initial key parameters $\xi^0_{1:n}$ are provided by the pre-trained model. To avoid overfitting to the current query vectors, we use a Gaussian prior centered on the key parameters provided by the pre-trained network, i.e., $\xi^0_{1:n}$ with a finite precision $\theta_\xi > 0$. The EM update for the key parameters at any iteration t is given by (see Appendix B.2)

$$\xi_k^{t+1} = \frac{\theta_\xi \xi_k^t + \alpha_k \sum_{i=1}^n w_{i,k}^t q_i}{\theta_\xi + \alpha_k \sum_{i=1}^n w_{i,k}^t}. \tag{18}$$

Analogous to the keys, we can also adapt the α_j precision parameters (see Appendix B.3).

3.6 Online value propagation

The proposed model allows for fixing the outputs of a selected subset of units to predefined values and letting them propagate to other units in a principled way. This aspect of our model is of particular interest to interactive semantic segmentation, where a human annotator provides corrections to the output of a semantic segmentation model. In this case, assuming an attention layer at the output of a deep model, the memory units correspond to pixels and the output values correspond to the semantic label for that pixel, *e.g.* foreground or background. Based on the network's prediction, an annotator provides corrections for a subset of the pixels, which are the ground truth for those pixels. These correspond to the fixed predefined values for those units, whose effect is to be propagated to semantically similar pixels globally across the image to make the process more efficient. More formally, suppose the annotator has provided the correct values for the first $s < n$ units. We want for this information to improve the inference about the value for all the other units $i > s$. Within our framework, this inference is given by

$$\hat{v}_i = \operatorname*{argmax}_v p_i(v \mid q_i, q_{1:n}, v_{1:s}), \text{ for } s < i <= n. \tag{19}$$

In order to do this inference, we adopt a Bayesian approach similar to model adaptation of Section 3.5. Let λ represent the set of network parameters, e.g., $\pi, \xi, \mu, \alpha, \beta$. Writing the inference as an expectation over the model posterior $p(\lambda \mid q_{1:n}, v_{1:s})$

$$p_i(v \mid q_{1:n}, v_{1:s}) = \int p(\lambda \mid q_{1:n}, v_{1:s}) p_i(v \mid q_i, \lambda) d\lambda \approx p_i(v \mid q_i, \hat{\lambda}), \tag{20}$$

where we approximate the expectation with its MAP estimate as before

$$\hat{\lambda} = \operatorname*{argmax}_\lambda p(\lambda \mid q_{1:n}, v_{1:s}) \tag{21}$$

$$\hat{v}_i = \operatorname*{argmax}_v p_i(v \mid q_i, \hat{\lambda}). \tag{22}$$

Eq. (21) is solved using EM. Specifically, value propagation across units is achieved by updating the μ_k for each unit k starting with the initial value μ_k^0 provided by the pre-trained model (Section 3.4). Following a similar approach as in Section 3.5, the EM update for μ_k^{t+1} at iteration t is given by

$$\mu_k^{t+1} = \frac{\theta_\mu \mu_k^t + \beta_k \sum_{i=1}^s w_{i,k}^t v_i}{\theta_\mu + \beta_k \sum_{i=1}^s w_{i,k}^t} \tag{23}$$

$$w_{i,k}^t = p_i(u_k \mid q_i, v_i, \mu_{1:n}^t) = \frac{\pi_{i,k}\, p(q_i \mid u_k, \xi_k)\, p(v_i \mid u_k, \mu_k^t)}{\sum_{j=1}^n \pi_{i,k}\, p(q_i \mid u_j, \xi_j)\, p(v_i \mid u_j, \mu_j^t)}, \tag{24}$$

where θ_μ is the precision for the Gaussian prior over each μ_k. See also Appendix B.4.

3.7 Combining offline learning and online adaptation

The inference time adaptation of parameters is differentiable. So it can be included as part of the traditional supervised optimization *e.g.* via stochastic gradient descent and used to learn the parameters of the prior distributions over $\xi, \mu, \alpha, \beta, \pi$.

3.8 Position embeddings

Positional embeddings [53, 47] are useful in attention models to encode the relative or absolute positions of tokens. In computer vision applications, relative position embeddings have been found to be critical to capture the interactions between features based on their pairwise positional relations [43, 49, 55]. We propose to encode relative position embeddings by introducing extra parameters in the per-unit likelihoods of the mixture components and their priors. Let r^q_{j-i}, and r^k_{j-i} denote the relative position embeddings for a query and key interacting at units i and j respectively. The query/key marginal with the position embeddings is given by (see Appendix B.5)

$$p_i(q \mid \xi_j, r^q_{j-i}, u_j) \propto \mathcal{N}(q \mid \xi_j,\, \frac{1}{\alpha_j} I_d)\mathcal{N}(q \mid r^q_{j-i},\, \frac{1}{\alpha_j} I_d) \propto \mathcal{N}(q \mid \frac{\xi_j + r^q_{j-i}}{2},\, \frac{1}{2\alpha_j} I_d), \tag{25}$$

where $\mathcal{N}(a \mid b,\, c)$ is the Gaussian likelihood function over a with mean b and covariance matrix c. I_d is a $d \times d$ identity matrix. The mixture component priors with position embeddings take the form

$$\pi_{i,j} \propto \mathcal{N}(\xi_j \mid r^k_{j-i},\, \frac{1}{\alpha_j} I_d) \exp\left[\frac{\alpha_j}{2}\left(2\|\xi_j\|^2 + \|r^q_{j-i}\|^2 + \|r^k_{j-i}\|^2\right)\right] \exp\left[\frac{\beta_j}{2}\|\mu_j\|^2\right]. \tag{26}$$

4 Experiments

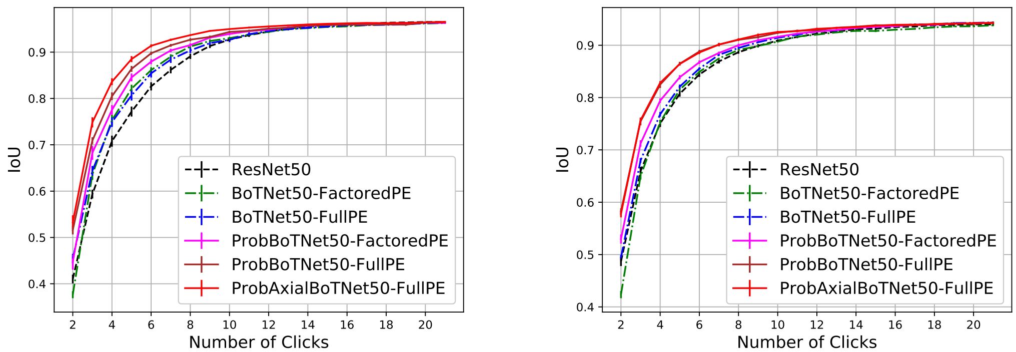
In this section, we report the results of using probabilistic attention at various stages of a deep interactive semantic segmentation network. Specifically, we use it within the BoTNet50 backbone [49] in place of standard attention and also as part of a self-attention based classification head at the network output. We quantify model performance (mean IOU relative to ground truth) as a function of the number of clicks [6] on two widely used public benchmarks for this task: GrabCut[46] and Berkeley [39]. Appendix C provides more details on the interactive segmentation model architectures, training and evaluation protocols.

4.1 Probabilistic attention within a backbone

We adopt the recent work on BoTNet [49] by replacing the convolutional layers with attention layers in the last bottleneck block (c5) of the ResNet50 [24] architecture. Specifically, we use probabilistic attention layers in place of standard attention using either full or axial [55] attention. We experiment with either factored [49] or full relative positional encoding. Factored encoding uses $(H + W)d$ parameters for an image of size (H, W) factoring them along the height and width dimensions, whereas full encoding uses $2(H * W)d - d$ parameters, d per relative offset. Our models are trained on the LVIS [21] dataset at a resolution of 256 pixels. The results are shown in Fig. 1. The results suggest that using probabilistic attention in the BoTNet50 backbone leads to better performance especially for smaller number of clicks. This is true with both full and axial attention BoTNets using probabilistic attention. Using full relative position encoding helps more than using factored encoding perhaps due to the larger number of parameters.

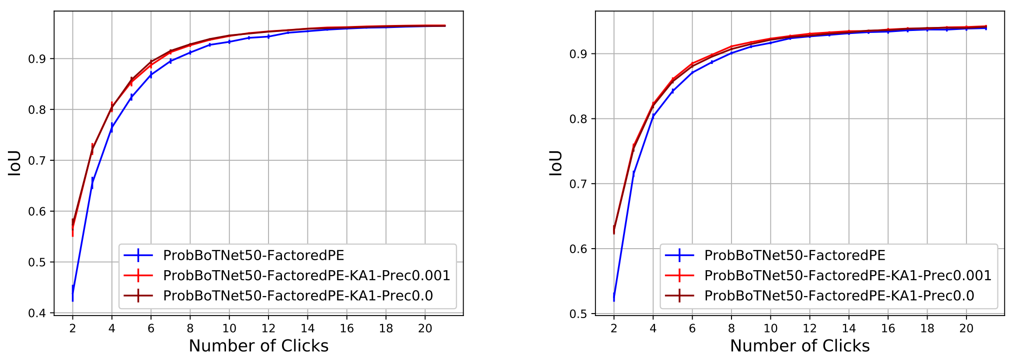
4.2 Key adaptation

We experiment with unsupervised model adaptation as described in Section 3.5 by adapting the keys online (Eq. (18)) based on the observed queries. The degree of adaptation is controlled by the prior precision parameter θ_ξ with lower values leading to a higher degree of adaptation due to the lower weight on the prior keys. Using the probabilistic attention BoTNet50 backbones of the previous section, we experiment with and without key adaptation. With key adaptation, we use two different values of the precision prior, 0.001 and 0, with the latter corresponding to a maximum likelihood update of the keys given observed queries. The results in Fig. 2 show the mean IoU as a function of number of clicks using the ProbBoTNet50-FactoredPE model. We observe that key adaptation leads



(a) GrabCut (b) Berkeley

Figure 1: **Probabilistic attention layers in BoTNet architecture**. Mean IoU as a function of #clicks using different attention layers, position embeddings and full or axial attention in the BoTNet architecture. These are compared against their fully convolutional conterpart ResNet50. The left and the right plots correspond to the GrabCut and Berkeley datasets respectively.



(a) GrabCut (b) Berkeley

Figure 2: **Unsupervised key adaptation**. Mean IoU vs #clicks with and without key adaptation (KA) on the GrabCut and Berkeley datasets. Probabilistic BoTNets with factored position encodings are evaluated without using KA or using 1 iteration of KA with two different prior precision (Prec.) values of 0.001 or 0.

to higher IoUs without any corrective clicks or using only a few corrective clicks. Specifically there is an absolute improvement of about 10% in mean IoU using key adaptation and without using any corrective clicks. Additional results using ProbBoTNet50-FullPE are shown in the the Appendix D. Using a lower value of prior precision seems beneficial and the extreme case of maximum likelihood adaptation leads to the best performance. Note that this effect has been observed in a previous work [18], where it is perceived as a doubly normalized attention scheme (DNAS). This can be attributed to the unsupervised model adaptation accounting for the small domain shift introduced by models trained on LVIS and evaluated on GrabCut and Berkeley datasets. Without using key adaptation additional user input in the form of corrective clicks is required to account for this shift as can be seen by the asymptotically similar behavior with increasing number of clicks.

4.3 Probabilistic attention as a classifier

Deep architectures [36, 23, 12, 10] for semantic segmentation have a fully connected (1x1 conv) layer as their classification head. We replace this layer with a corrective self-attention [53] module that takes corrective click locations as additional inputs to more effectively propagate corrections as follows.

Corrective self attention. We append the corrective channels to the features of the penultimate decoder layer and feed them as inputs to the value embedding layer (see Fig. 3). The query and the key embeddings do not use the corrective channels as their inputs, which allows attention maps to be computed based only on the semantics captured by the features. However, the weights of the value embedding layer can be trained to output the desired labels at the locations of the corrective clicks. Fig. 3 shows a block diagram of our corrective self-attention (CSA) layer. We choose to use probabilistic self-attention for computing the output values.

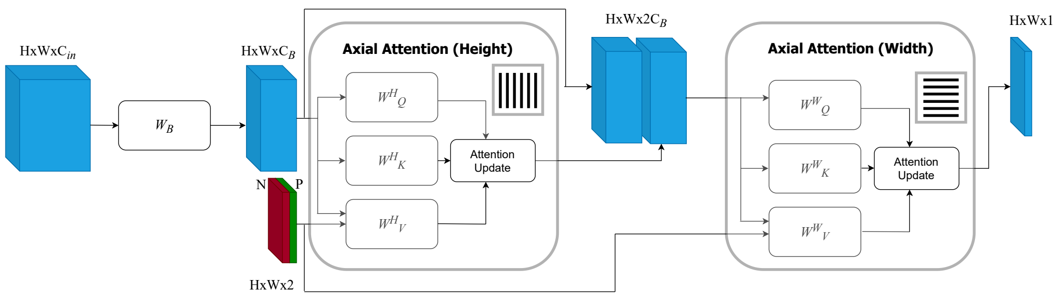


Figure 3: **Corrective self attention layer**. We propose a self-attention based classification head at the output of the decoder to more effectively propagate corrective clicks. C_{in} channels from the penultimate decoder layer are reduced to C_B by a bottleneck layer with weights W_B. These are input to a pair of densely connected Axial Attention [55] modules along height and width dimensions to produce the output logit at full image resolution ($H \times W$). The corrective channels (P and N) are fed only to the value embedding functions (W_V^H and W_V^W) of the attention modules to propagate the corrections more effectively.

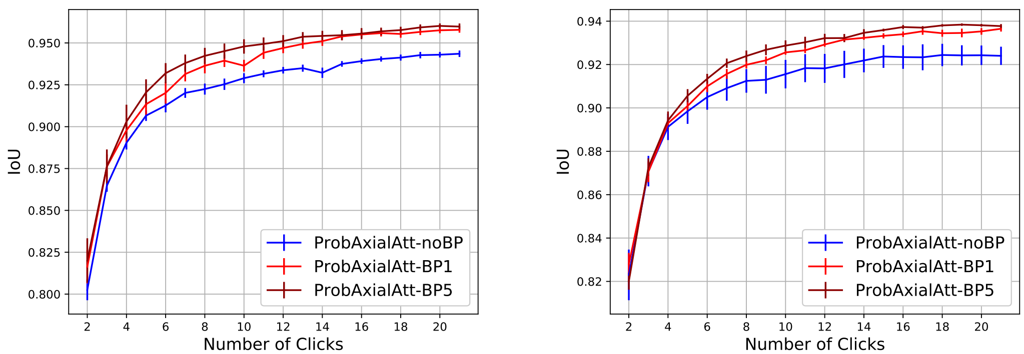
The CSA layer is used at the network output by up-sampling the final feature map of the decoder to the input image resolution and appending positive and negative corrective channels containing only the click locations. The local context size for Axial attention modules is chosen to be 64 pixels. The output of the axial attention block is passed through a sigmoid to estimate the pixelwise probabilities of the object mask.

4.4 Value propagation

We demonstrate the effect of propagating annotator feedback across pixels using online value propagation as described in Section 3.6. We use the CSA layer described above in place of the 1x1 conv classifier head of the HRNetV2+OCR [67] architecture pre-trained on Imagenet classification and fine-tuned for interactive segmentation on SBD [22] at a resolution of 256 pixels. Note that value propagation requires learning one additional parameter per output class (2 in our case) to estimate the fixed logit that corresponds to the annotator feedback for that class at the corrective locations. These are learnt as part of network training using gradient descent. Following standard protocols, we test on the GrabCut [46] and Berkeley [39] datasets. Fig. 4 shows the effect of using different number of value propagation iterations (1 and 5) within the probabilistic attention layer. Clearly, value propagation leads to more effective propagation of labeler feedback relative to not using it. For this experiment, we do value propagation only in the output width block of the CSA layer (Fig. 3) as we found that doing so in both the height and width layers did not work so well in our experiments. We hypothesize that this is probably due to the difficulty in learning the high dimensional fixed parameters corresponding to annotator feedback in the height block of the axial attention layer.

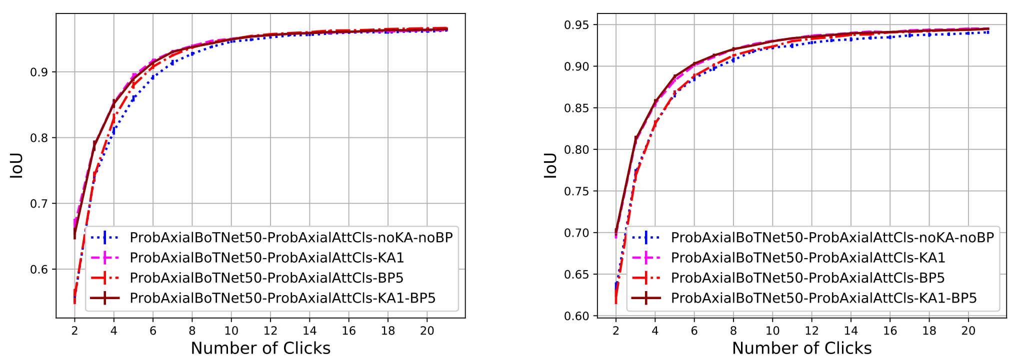
4.5 Combining key adaptation and value propagation

In this section, we experiment with combining key adaptation and value propagation in a single model. For this experiment we use a BoTNet50 architecture with a corrective self-attention classification head at a resolution of 256 pixels. We use axial attention in both the backbone and the classification head with probabilistic self attention updates. The model is trained on LVIS and evaluated on GrabCut and Berkeley datasets. Fig. 5 shows the effect of using either key adaptation or value propagation or



(a) GrabCut (b) Berkeley

Figure 4: **Effect of value propagation using Axial attention**. Mean IoU vs #clicks with and without value propagation using an axial attention based probabilistic CSA layer at the output.We use 1 (BP1) and 5 (BP5) iterations of value propagation at the CSA layer and test on the GrabCut (left) and Berkeley (right) datasets.



(a) GrabCut (b) Berkeley

Figure 5: **Effect of combining key adaptation and value propagation**. Mean IoU vs. #clicks using one or both of key adaptation and value propagation in a single model. Key adaptation is run for 1 iteration (KA1) and value propagation for 5 iterations (BP5) when either or both are used.

both relative to not using them. As observed separately in the previous plots, key adaptation helps in the small #clicks regime whereas value propagation shows greater benefits with increasing #clicks. Using the two jointly allows the model to respond quickly to annotator feedback in both the regimes.

5 Conclusion

We provide a probabilistic interpretation of the attention mechanism in transformers as a generative mixture model over queries and values parameterized through keys. Using our framework, the attention update is maximum a posteriori inference over values given queries. Specifically, the standard dot-product attention is a special case assuming Gaussians for the mixture likelihoods and a few other constraints. Using Bayesian inference, our interpretation allows for online update of the mixture model parameters as well as the propagation of a set of fixed values specified by an external agent. Although we demonstrate the utility of these aspects on the problem of interactive segmentation, the proposed model is generic and can be extended to other domains with suitable distributional forms for the mixture likelihoods.

Acknowledgments and Disclosure of Funding

We would like to thank Russ Webb, Joshua Susskind, Parisa Assar, and numerous other colleagues for their helpful discussions, draft and code reviews.

References

[1] D. Acuna, H. Ling, A. Kar, and S. Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018.

[2] E. Agustsson, J. R. Uijlings, and V. Ferrari. Interactive full image segmentation by considering all regions jointly. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11614–11623, 2019.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409.0473, 09 2014.

[4] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. Variational attention for sequence-to-sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1672–1682, August 2018.

[5] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[6] R. Benenson, S. Popov, and V. Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 60–65, 2005.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[9] L. Castrejón, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a polygon-rnn. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4485–4493, 2017.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Murphy Kevin 0002, and L. Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 834–848, 2018.

[11] Liang-Chieh Chen, G. Papandreou, Florian Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] 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)*, September 2018.

[13] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, July 2019.

[14] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

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

[16] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention. In *NeurIPS*, pages 9735–9747, 2018.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[18] Nan Ding, Xinjie Fan, Zhenzhong Lan, Dale Schuurmans, and Radu Soricut. Attention that does not explain away, 2020.

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

[20] Alexandre X. Falcao, Jayaram K. Udupa, Supun Samarasekera, Shoba Sharma, Bruce Elliot Hirsch, and Roberto de A. Lotufo. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60(4):233 – 260, 1998.

[21] Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation, 2019.

[22] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

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

[25] Jay Heo, Junhyeon Park, Hyewon Jeong, Kwang Joon Kim, Juho Lee, Eunho Yang, and Sung Ju Hwang. Cost-effective interactive attention learning with neural attention processes. In *ICML*, 2020.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[27] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[28] Y. Hu, A. Soltoggio, R. Lock, and S. Carter. A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks*, 109:31–42, 2019.

[29] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[30] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[31] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. Continuous adaptation for interactive object segmentation by learning from corrections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[32] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[33] Z. Li, Q. Chen, and V. Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018.

[34] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[36] 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 (CVPR)*, June 2015.

[37] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In *British Machine Vision Conference (BMVC)*, 2018.

[38] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[40] Eric N. Mortensen and William A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349 – 384, 1998.

[41] S. W. Oh, J. Y. Lee, N. Xu, and S. J. Kim. Space-time memory networks for video object segmentation with user guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[42] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Hierarchical self-attention network for action localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[43] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models, 2019.

[44] H. Ramadan, Chaymae Lachqar, and H. Tairi. A survey of recent interactive image segmentation methods. *Computational Visual Media*, pages 1 – 30, 2020.

[45] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021.

[46] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" – interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, pages 309–314, 2004.

[47] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, June 2018.

[48] K. Sofiiuk, I. Petrov, O. Barinova, and A. Konushin. F-brs: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, 2020.

[49] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition, 2021.

[50] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[51] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[52] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones, 2021.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, 2017.

[54] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[55] 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 (ECCV)*, 2020.

[56] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[57] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[58] Zian Wang, David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Object instance annotation with deep extreme level set evolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[59] Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*, 2019.

[60] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2048–2057, 07–09 Jul 2015.

[61] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[62] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[63] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[64] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *ECCV*, 2020.

[65] H. Yu, Y. Zhou, H. Qian, M. Xian, and S. Wang. Loosecut: Interactive image segmentation with loosely bounded boxes. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3335–3339, 2017.

[66] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[67] Y. Yuan, X. Chen, and J. Wang. Object-contextual representations for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2019.

[68] S. Zhang, J. H. Liew, Y. Wei, S. Wei, and Y. Zhao. Interactive object segmentation with inside-outside guidance. In *CVPR*, 2020.

[69] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[70] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.