IS ATTENTION BETTER THAN MATRIX DECOMPOSITION?

Anonymous authors
Paper under double-blind review

ABSTRACT

As an essential ingredient of modern deep learning, attention mechanism, especially self-attention, plays a vital role in the global correlation discovery. However, is hand-crafted attention irreplaceable when modeling the global context? Our intriguing finding is that self-attention is not better than the matrix decomposition (MD) model developed 20 years ago regarding the performance and computational cost. We show that optimization algorithms can help design global context modules, light, memory-efficient, and more potent than attentions. This paper then proposes a series of Hamburgers based on this technique, in which we employ the optimization algorithms for solving MDs to factorize the input representations into sub-matrices and reconstruct a low-rank embedding. Hamburgers with different MDs can perform favorably against the popular global context module self-attention when carefully coping with gradients back-propagated through MDs. Comprehensive experiments are conducted in the vision tasks where it is crucial to learn long-range dependencies, including semantic segmentation and image generation, demonstrating significant improvements over self-attention and its variants.

1 INTRODUCTION

Since self-attention and transformer (Vaswani et al., 2017) showed significant advantages over recurrent neural networks and convolutional neural networks in capturing long-distance dependencies, attention has been widely adopted by computer vision (Wang et al., 2018; Zhang et al., 2019a) and natural language processing (Devlin et al., 2019) for global information mining. However, is hand-crafted attention irreplaceable when modeling the global context?

This paper focuses on a new approach to design global context modules with human inductive bias easily. The key idea is, *if we formulate the inductive bias into an objective function, the optimization algorithm to minimize the objective function can construct a computational graph, i.e., the architecture we need in the networks*. We particularize this idea by developing a counterpart for the most representative global context module, self-attention. Considering extracting global information in the networks as finding a dictionary and the corresponding codes to capture the inherent correlation, we model the context discovery as low-rank completion of the input tensor. This paper then proposes a global correlation block, Hamburger, by employing matrix decomposition (MD) to factorize the learned representation into sub-matrices so as to reconstruct a low-rank embedding. The iterative optimization algorithm to solve matrix decomposition defines the central computational graph, *i.e.,* Hamburger's architecture.

Our work takes advantage of the matrix decomposition models as the foundation of Hamburger, including Vector Quantization (VQ) (Gray & Neuhoff, 1998), Concept Decomposition (CD) (Dhillon & Modha, 2001), and Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999). Additionally, instead of directly applying Back-Propagation Through Time (BPTT) algorithm (Werbos et al., 1990) to differentiate the iterative optimization, we adopt a truncated BPTT algorithm, *i.e.,* one-step gradient, to back-propagate the gradient effectively. We illustrate the advantages of Hamburger in the fundamental vision tasks where global information has been proven crucial, including semantic segmentation and image generation. The experiments prove that optimization-designed Hamburger can perform competitively with state-of-the-art attention models when avoiding the unstable gradient back-propagated through the iterative computational graph of MD. Hamburger sets new state-of-the-art records on the PASCAL VOC dataset (Everingham et al., 2010) and PASCAL Context

dataset (Mottaghi et al., 2014) for semantic segmentation and surpasses existing attention models for GANs in the challenging image generation at the scale of ImageNet (Deng et al., 2009).

The contributions of this paper are listed as follows:

- We show a white-box and easy-following approach to design global information blocks, i.e., by turning the optimization algorithm that minimizes an objective function reflecting human priors like global correlation into the architecture of the module.
- We propose Hamburger, a light, fast, and memory-efficient global context module based on matrix decomposition, surpassing the most commonly-used attentions on semantic segmentation and image generation by a large margin (1.0 in mIoU(%) on the PASCAL VOC test set and over 5% in FID on ImageNet than current state-of-the-art attentions).
- We figure out that the main obstacle of applying MD in the networks is the unstable gradient back-propagated through the computational graph of its optimization. As a pragmatic solution, the proposed one-step gradient facilitates the training of Hamburger with MDs.

2 Methodology

2.1 Warm Up

Since matrix decomposition is pivotal to the proposed Hamburger, we first review the idea of matrix decomposition. A common view is that matrix decomposition factorizes the observed matrix into a product of several sub-matrices, *e.g.,* Singular Value Decomposition. However, a more illuminating perspective is that, by assuming the generation process, matrix decomposition acts as the inverse of the generation, disassembling the atoms that make up the complex data. From the reconstruction of the original matrices, matrix decomposition recovers the latent structure of observed data.

Suppose that the given data are arranged as the columns of a large matrix $\boldsymbol{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$. A general assumption is that there is a low-dimensional subspace, or a union of multiple subspaces hidden in $\boldsymbol{X}$. That is, there exists a dictionary matrix $\boldsymbol{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_r] \in \mathbb{R}^{d\times r}$ and corresponding codes $\boldsymbol{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_n] \in \mathbb{R}^{r\times n}$ that $\boldsymbol{X}$ can be expressed as

$$\boldsymbol{X} = \overset{\xleftarrow{\quad generation \quad}}{\underset{\xrightarrow[\quad decomposition \quad]{}}{\bar{\boldsymbol{X}} + \boldsymbol{E} = \boldsymbol{D}\boldsymbol{C}}} + \boldsymbol{E}, \tag{1}$$

where $\bar{\boldsymbol{X}} \in \mathbb{R}^{d\times n}$ is the output low-rank reconstruction, and $\boldsymbol{E} \in \mathbb{R}^{d\times n}$ is the noise matrix to be discarded. Here we assume that the recovered matrix $\bar{\boldsymbol{X}}$ has the low-rank property, such that

$$\operatorname{rank}(\bar{\boldsymbol{X}}) \leq \min(\operatorname{rank}(\boldsymbol{D}), \operatorname{rank}(\boldsymbol{C})) \leq r \ll \min(d, n). \tag{2}$$

Different MDs can be derived by assuming structures to matrices $\boldsymbol{D}$, $\boldsymbol{C}$, and $\boldsymbol{E}$ (Kolda & Bader, 2009; Udell et al., 2016). MD is usually formulated as an objective with various constraints and then solved by optimization algorithms, with classic applications to image denoising (Wright et al., 2009; Lu et al., 2014), inpainting (Mairal et al., 2010), and feature extraction (Zhang et al., 2012).

2.2 Proposed Method

We focus on building global context modules for the networks without painstaking hand-crafted design. Before starting our discussion, we review the representative hand-designed context block self-attention pithily.

The attention mechanism aims at finding a group of concepts for further conscious reasoning from massive unconscious context (Xu et al., 2015b; Bengio, 2017; Goyal et al., 2019). As a representative, self-attention (Vaswani et al., 2017) is proposed for learning long-range dependencies in machine translation,

$$\text{Attention}\,(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \operatorname{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)\boldsymbol{V}, \tag{3}$$

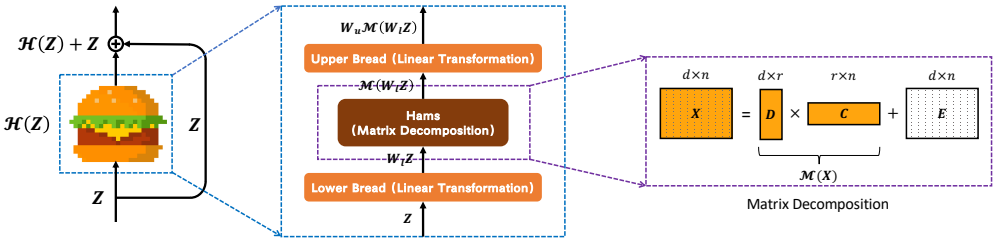

Figure 1: Overview of Hamburger

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n\times d}$ are features projected by linear transformations from the input. Self-attention extracts global information via attending all tokens at a time rather than the typical one-by-one processing of recurrent neural networks.

Though self-attention and its variants achieved great success, researchers are confronted with (1) developing new global context modules based on self-attention, typically via hand-crafted engineering, and (2) explaining why current attention models work. This paper bypasses both issues and finds a method to easily design global context modules via a well-defined white-box toolkit. We try to formulate the human inductive bias, like the global context, as an objective function and use the optimization algorithm to solve such a problem to design the module's architecture. The optimization algorithm creates a computational graph, takes some input, and finally outputs the solution. We apply the computational graph of optimization algorithms for the central part of our context module.

Based on this approach, we need to model the networks' global information issue as an optimization problem. Take the convolutional neural networks (CNN) as an example for further discussion. The networks output a tensor $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ after we feed into an image. Since the tensor can be seen as a set of HW C-dimensional hyper-pixels, we unfold the tensor into a matrix $\boldsymbol{X} \in \mathbb{R}^{C\times HW}$. When the module learns the long-range dependencies or the global context, the hidden assumption is that the hyper-pixels are inherently correlated. For the sake of simplicity, we assume that hyper-pixels are linearly dependent, which means that each hyper-pixel in $\boldsymbol{X}$ can be expressed as the linear combination of bases whose elements are typically much less than HW. In the ideal situation, the global information in $\boldsymbol{X}$ can be *low-rank*. However, due to vanilla CNN's poor ability to model the global context (Wang et al., 2018; Zhang et al., 2019a), $\boldsymbol{X}$ is usually corrupted with redundant information or incompleteness. The above analysis suggests that modeling the learned representations' global information can be achieved by completing the low-rank part $\bar{\boldsymbol{X}}$ in $\boldsymbol{X}$ and discarding the noise part $\boldsymbol{E}$ described in Eq. (1), *i.e.,* the classic matrix decomposition models. Using the notion of Sec. 2.1, the general objective function of matrix decomposition is

$$\min_{\boldsymbol{D},\boldsymbol{C}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{D}\boldsymbol{C}) + \mathcal{R}_1(\boldsymbol{D}) + \mathcal{R}_2(\boldsymbol{C}) \tag{4}$$

where $\mathcal{L}$ is the reconstruction loss, $\mathcal{R}_1$ and $\mathcal{R}_2$ are regularization terms for the dictionary $\boldsymbol{D}$ and the codes $\boldsymbol{C}$. Denote the optimization algorithm to minimize Eq. (4) as $\mathcal{M}$. $\mathcal{M}$ is the core architecture we deploy in our global context module.

In the later sections, we introduce our global context block, Hamburger, and then discuss detailed MD models and optimization algorithms for $\mathcal{M}$. Finally, we handle the gradient issue for back-propagation through matrix decomposition.

2.2.1 Hamburger

Hamburger consists of one slice of "ham" (matrix decomposition) and two slices of "bread" (linear transformation). As the name implies, Hamburger first maps the input $\boldsymbol{Z} \in \mathbb{R}^{d_z\times n}$ into feature space with a linear transformation $\boldsymbol{W}_l \in \mathbb{R}^{d\times d_z}$, namely "lower bread", then uses matrix decomposition $\mathcal{M}$ to solve a low-rank signal subspace, corresponding to the "ham", and finally transforms extracted signals into the output with another linear transformation $\boldsymbol{W}_u \in \mathbb{R}^{d_z\times d}$, called "upper bread",

$$\mathcal{H}(\boldsymbol{Z}) = \boldsymbol{W}_u\mathcal{M}(\boldsymbol{W}_l\boldsymbol{Z}), \tag{5}$$

where $\mathcal{M}$ is matrix decomposition to recover the clear latent structure, functioning as a *global nonlinearity*. Detailed architectures of $\mathcal{M}$, *i.e.,* optimization algorithms to factorize $\boldsymbol{X}$, are discussed

in Sec. 2.2.2. Fig. 1 describes the architecture of Hamburger, where it collaborates with neural nets via Batch Normalization (BN) (Ioffe & Szegedy, 2015), a skip connection, and finally outputs $\boldsymbol{Y}$,

$$\boldsymbol{Y} = \boldsymbol{Z} + \mathrm{BN}(\mathcal{H}(\boldsymbol{Z})). \tag{6}$$

2.2.2 HAMS

This section describes the structure of "ham", *i.e.,* $\mathcal{M}$ in Eq. (5). As discussed in the previous section, by formulating MD as an optimization problem, algorithms to solve MD naturally compose $\mathcal{M}$. $\mathcal{M}$ takes the output of "lower bread" as its input and computes a low-rank reconstruction as its output, denoted as $\boldsymbol{X}$ and $\bar{\boldsymbol{X}}$, respectively.

$$\mathcal{M}(\boldsymbol{X}) = \bar{\boldsymbol{X}} = \boldsymbol{D}\boldsymbol{C}. \tag{7}$$

We investigate two MD models for $\mathcal{M}$, Vector Quantization (VQ), and Non-negative Matrix Factorization (NMF) to solve $\boldsymbol{D}$ and $\boldsymbol{C}$ and reconstruct $\bar{\boldsymbol{X}}$, while leaving Concept Decomposition (CD) to Appendix B. The selected MD models are introduced briefly because we endeavor to illustrate the importance of the low-rank inductive bias and the designing method for global context modules rather than any specific MD models. It is preferred to abstract the MD part as a whole, *i.e.,* $\mathcal{M}$ in the context, and focus on how Hamburger can show the superiority in its entirety.

Vector Quantization Vector Quantization (VQ) (Gray & Neuhoff, 1998), a classic data compression algorithm, can be formulated as an optimization problem in term of matrix decomposition:

$$\min_{\boldsymbol{D},\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{C}\|_F \quad \text{s.t. } \mathbf{c}_i \in \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_r\}, \tag{8}$$

where $\boldsymbol{e}_i$ is the canonical basis vector, $\mathbf{e}_i = [0, \cdots, \underset{i\text{th}}{1}, \cdots, 0]^\top$. The solution to minimize the objective in Eq. (8) is K-means (Gray & Neuhoff, 1998). However, to ensure that VQ is differentiable, we replace the hard $\arg\min$ and Euclidean distance with $softmax$ and $cosine$ similarity, leading to Alg. 1, where $cosine(\boldsymbol{D}, \boldsymbol{X})$ is a similarity matrix whose entries satisfy $cosine(\boldsymbol{D}, \boldsymbol{X})_{ij} = \frac{\mathbf{d}_i^\top \mathbf{x}_j}{\|\mathbf{d}\|\|\mathbf{x}\|}$, and $softmax$ is applied column-wise and T is the temperature. Further we can obtain a hard assignment by a one-hot vector when $T \to 0$.

Algorithm 1 Soft VQ

Input $\boldsymbol{X}$. Initialize $\boldsymbol{D}, \boldsymbol{C}$.
for k from 1 to K **do**
  $\boldsymbol{C} \leftarrow softmax(\frac{1}{T} cosine(\boldsymbol{D}, \boldsymbol{X}))$
  $\boldsymbol{D} \leftarrow \boldsymbol{X}\boldsymbol{C}^\top diag(\boldsymbol{C}\mathbf{1}_n)^{-1}$
end for
Output $\bar{\boldsymbol{X}} = \boldsymbol{D}\boldsymbol{C}$.

Algorithm 2 NMF with MU

Input $\boldsymbol{X}$. Initialize non-negative $\boldsymbol{D}, \boldsymbol{C}$
for k from 1 to K **do**
  $\boldsymbol{C}_{ij} \leftarrow \boldsymbol{C}_{ij} \frac{(\boldsymbol{D}^\top \boldsymbol{X})_{ij}}{(\boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{C})_{ij}}$
  $\boldsymbol{D}_{ij} \leftarrow \boldsymbol{D}_{ij} \frac{(\boldsymbol{X}\boldsymbol{C}^\top)_{ij}}{(\boldsymbol{D}\boldsymbol{C}\boldsymbol{C}^\top)_{ij}}$
end for
Output $\bar{\boldsymbol{X}} = \boldsymbol{D}\boldsymbol{C}$.

Non-negative Matrix Factorization If we impose non-negative constraints on the dictionary $\boldsymbol{D}$ and the codes $\boldsymbol{C}$, it leads to Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999):

$$\min_{\boldsymbol{D},\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{C}\|_F \quad \text{s.t. } \boldsymbol{D}_{ij} \geq 0, \boldsymbol{C}_{jk} \geq 0. \tag{9}$$

To satisfy the non-negative constraints, we add a ReLU non-linearity before putting $\boldsymbol{X}$ into NMF. We apply the Multiplicative Update (MU) rules (Lee & Seung, 2001) in Alg. 2 to solve NMF, which guarantees the convergence.

As white-box global context modules, VQ, CD, and NMF are straightforward and light, showing remarkable efficiency. They are formulated into optimization algorithms that mainly consist of matrix multiplications with the complexity $\mathcal{O}(ndr)$, much cheaper than complexity $\mathcal{O}(n^2 d)$ in self-attention as $r \ll n$. All three MDs are memory-friendly since they avoid generating a large $n \times n$ matrix as an intermediate variable, like the product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ of self-attention in Eq. (3). In the later section, our experiments prove MDs are at least on par with self-attention, though the architectures of $\mathcal{M}$ are created by optimization and look different from classic dot product self-attention.

2.3 ONE-STEP GRADIENT

Since $\mathcal{M}$ involves an optimization algorithm as its computational graph, a crux to fuse it into neural nets is how the iterative algorithm back-propagates gradient. The RNN-like behavior of optimization suggests Back-Propagation Through Time (BPTT) algorithm (Werbos et al., 1990) as the standard choice to differentiate the iterative process. We first review the BPTT algorithm below. However, in practice, the unstable gradient from BPTT does harm Hamburger's performances. Hence we build an abstract model to analyze the drawbacks of BPTT and try to find a pragmatic solution while considering MD's nature as an optimization algorithm. As shown in Fig. 2, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{h}^t$ denote input, output and intermediate result at time step t, respectively, while $\mathcal{F}$ and $\mathcal{G}$ are operators. At each time step, the model receives the same input $\mathbf{x}$ processed by underlying neural networks.

$$\mathbf{h}^{t+1} = \mathcal{F}(\mathbf{h}^t, \mathbf{x}). \tag{10}$$

The intermediate results $\mathbf{h}^i$ are all discarded. Only the output of the last step $\mathbf{h}^t$ is passed through $\mathcal{G}$ for output $\mathbf{y}$,

$$\mathbf{y} = \mathcal{G}(\mathbf{h}^t). \tag{11}$$

In the BPPT algorithm, the gradient from output $\mathbf{y}$ to input $\mathbf{x}$ is given, according to the Chain rule:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \sum_{i=1}^{t} \frac{\partial \mathbf{y}}{\partial \mathbf{h}^t} \left(\prod_{j=i+1}^{t} \frac{\partial \mathbf{h}^j}{\partial \mathbf{h}^{j-1}} \right) \frac{\partial \mathbf{h}^i}{\partial \mathbf{x}}. \tag{12}$$

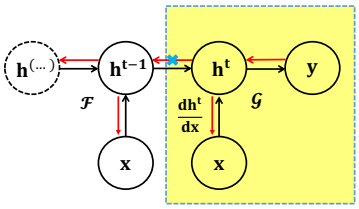

Figure 2: One-step Gradient

A thought experiment is to consider $t \to \infty$, leading to a fully converged result $\mathbf{h}^*$ and infinite terms in Eq. (12). We suppose that both $\mathcal{F}$ and $\mathcal{G}$ are Lipschitz with constants L_h *w.r.t.* $\mathbf{h}$, L_x *w.r.t.* $\mathbf{x}$, and $L_\mathcal{G}$, and $L_h < 1$. Note that these assumptions apply to a large number of optimization or numerical methods. Then we have:

Proposition 1 $\{\mathbf{h}^i\}_t$ has linear convergence.

Proposition 2 $\lim_{t\to\infty} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}^*}(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \mathbf{h}^*})^{-1} \frac{\partial \mathcal{F}}{\partial \mathbf{x}}$.

Proposition 3 $\lim_{t\to\infty} \|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^0}\| = 0$, $\lim_{t\to\infty} \|\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\| \leq \frac{L_\mathcal{G} L_x}{1-L_h}$.

Table 1: One-step Gradient & BPTT

Method	One-step	BPTT
VQ	77.7(77.4)	76.6(76.3)
CD	78.1(77.5)	75.0(74.6)
NMF	78.3(77.8)	77.4(77.0)

It is easy to incur gradient vanishing w.r.t. $\mathbf{h}^0$ when L_h is close to 0 and gradient explosion w.r.t. $\mathbf{x}$ when L_h is close to 1. The Jacobian matrix $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$, moreover, suffers from an ill-conditioned term $(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \mathbf{h}^*})^{-1}$ when the largest eigenvalue of $\frac{\partial \mathcal{F}}{\partial \mathbf{h}}$, *i.e.,* the Lipschitz constant of $\mathcal{F}$ *w.r.t.* $\mathbf{h}$, approaches 1 and its minimal eigenvalue typically stays near 0, thus restricts the capability of the gradient to search the well-generalized solution in the parameter space. The erratic scale and spectrum of the gradient back through the optimization algorithm indicate the infeasibility to apply BPTT to Hamburger directly, corroborated by the experiments in Tab. 1.

The analysis inspires us a possible solution. Note that *there are a multiplication of multiple Jacobian matrices* $\frac{\partial \mathbf{h}^j}{\partial \mathbf{h}^{j-1}}$ *and a summation of an infinite series in BPTT algorithm*, leading to uncontrollable scales of gradients. It enlightens us to drop some minor terms in the gradient while preserving its dominant terms to ensure the direction is approximately right. Considering $\{\frac{\partial \mathbf{y}}{\partial \mathbf{h}^t} \left(\prod_{j=i+1}^{t} \frac{\partial \mathbf{h}^j}{\partial \mathbf{h}^{j-1}} \right) \frac{\partial \mathbf{h}^i}{\partial \mathbf{x}}\}_t$ as a series, it makes sense to use the first term of Eq. (12) to represent the gradient if the scale of its terms decays exponentially measured by the operator norm. The first term of the gradient is from the last step of optimization, leading to the one-step gradient,

$$\widehat{\frac{\partial \mathbf{y}}{\partial \mathbf{x}}} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}^t} \frac{\partial \mathbf{h}^t}{\partial \mathbf{x}}. \tag{13}$$

The one-step gradient is a linear approximation of BPTT algorithm when $t \to \infty$. It is easy to implement, requiring a *no_grad* operation in PyTorch (Paszke et al., 2019) or *stop_gradient* operation in TensorFlow (Abadi et al., 2016) and reducing the time and space complexity from $\mathcal{O}(t)$ in BPTT to $\mathcal{O}(1)$. We test adding more terms to the gradient but its performance is worse than using one step. According to experimental results, one-step gradient is acceptable to differentiate MDs.

Table 2: Ablation on components of Hamburger with NMF Ham.

Method	**mIoU**(%)	**Params**
baseline	75.9(75.7)	32.67M
basic	78.3(77.8)	+0.50M
- ham	75.8(75.6)	+0.50M
- upper bread	77.0(76.8)	+0.25M
- lower bread	77.3(77.2)	+0.25M
only ham	77.0(76.8)	+0M

3 EXPERIMENTS

In this section we present experimental results demonstrating the techniques described above. Two vision tasks that benefit a lot from global information and attention mechanism attract us, including semantic segmentation (over 50 papers using attention) and deep generative models like GANs (most state-of-the-art GANs ensemble self-attention since SAGAN (Zhang et al., 2019a)). Both tasks are highly competitive and thus enough for comparing Hamburger with self-attention. Ablation studies show the importance of MD in Hamburger as well as the necessity of the one-step gradient. We emphasize the superiority of Hamburger on modeling global context over self-attention regarding both performance and computational cost.

3.1 ABLATION EXPERIMENTS

We choose to conduct all ablation experiments on the PASCAL VOC dataset (Everingham et al., 2010) for semantic segmentation, and report mIoU of *5 runs* on the validation set in the form of $\mathrm{best}(\mathrm{mean})$. ResNet-50 (He et al., 2016) with output stride 16 is the backbone for all ablation experiments. We employ a 3×3 conv with BN (Ioffe & Szegedy, 2015) and ReLU to reduce channels from 2048 to 512 and then add Hamburger, the same location as popular attentions in semantic segmentation. For detailed training settings, please see Appendix F.1.

Breads and Hams We ablate each part of the Hamburger. Removing MD (ham), causes the most severe decay in performance, attesting the importance of MD. Even if only the parameter-free MD is added (only ham), the performance can visibly improve. Parameterization also helps the Hamburger process the extracted features. Bread, especially upper bread, contributes considerable performance.

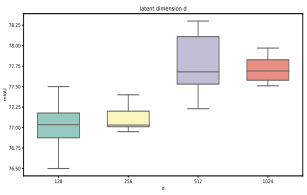

Figure 3: Ablation on d

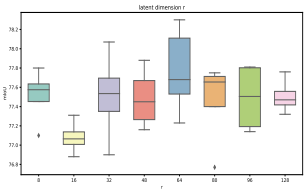

Figure 4: Ablation on r

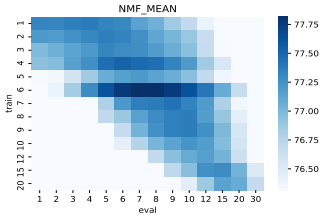

Figure 5: Ablation on K

Latent Dimension d and r It is worth noting that there is no simple linear relation between d and r with performances measured by mIoU, though $d = 8r$ is a satisfactory choice. Experiments show that $r = 8$ performs well, revealing that it can be very cheap for modeling the global context.

Iterations K We test more optimization steps in the evaluation stage. In general, the same K for training and test is recommended. $K = 6$ is enough for CD and NMF, while even $K = 1$ is acceptable for VQ. Typically 3∼6 steps are enough since simple MD's prior is still biased, and full convergence can overfit it. The few iterations are cheap and act as *early stopping*.

3.2 A CLOSE LOOK AT HAMBURGER

To understand the behavior of Hamburger in the networks, we visualize the spectrums of representations before and after Hamburger on the PASCAL VOC validation set. The input and output tensors

are unfolded to $\mathbb{R}^{C\times HW}$. The accumulative ratio of squared largest r singular values over total squared singular values of the unfolded matrix has been shown in Fig. 6. A truncated spectrum is usually observed in classic matrix decomposition models' results due to the low-rank reconstruction. In the networks, Hamburger also promotes energy concentration while preserving informative details via the skip connection. Additionally, we visualize the feature maps before and after Hamburger in Fig. 7. MD helps Hamburger learn interpretable global information by zeroing out uninformative channels, removing irregular noises, and completing details according to the context.

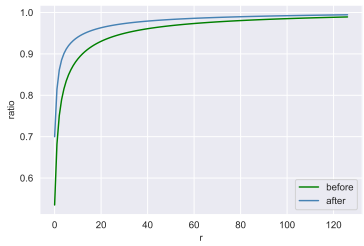

Figure 6: Accumulative Ratio

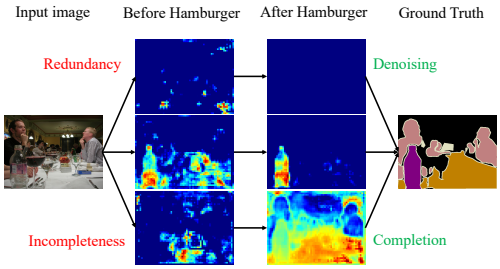

Figure 7: Visualization of feature maps

3.3 A Comparision with Attention

This section shows the advantages of MD-based Hamburger over attention in computational overhead, memory consumption, and inference time. DANet (Fu et al., 2019) is a state-of-the-art application of integrated attention mechanism on semantic segmentation, combining channel-wise and spatial-wise attention and defeating baseline by a large margin. We report the estimated overhead of processing a tensor $\mathcal{Z} \in \mathbb{R}^{512\times 64\times 64}$ in Tab. 3 according to (Molchanov et al., 2016). Excessive memory usage is the key bottleneck of cooperating with attention in real applications. Hence we also provide the GPU load and inference time on NVIDIA TITAN Xp. Hamburger achieves better performance with less computational cost than state-of-the-art attention.

Table 3: Comparisons between Hamburger and Dual Attention.

Method	**Params**	**FLOPs**	**Memory**	**GPU Load**		**GPU Time**	
				Train	Infer	Train	Infer
Dual Attention	4.82M	44.4G	123MB	293MB	176MB	10.1ms	7.5ms
Hamburger (CD)	0.50M	7.0G	32MB	46MB	26MB	8.9ms	5.7ms
Hamburger (NMF)	0.50M	7.7G	32MB	50MB	30MB	6.8ms	3.6ms

3.4 Semantic Segmentation

We benchmark Hamburger on the PASCAL VOC dataset (Everingham et al., 2010), and the PASCAL Contest dataset (Mottaghi et al., 2014), against state-of-the-art attentions. We use ResNet-101 (He et al., 2016) as our backbone. The output stride of the backbone is 8. The segmentation head is the same as ablation experiments. NMF is usually better than CD and VQ in ablation studies (see Tab. 1). Therefore, we mainly test NMF in further experiments. We use *HamNet* to represent ResNet with Hamburger in the following section.

Results on the PASCAL VOC test set, and the PASCAL Contest validation set, are illustrated in Tab. 4, and Tab. 5, respectively, without any auxiliary losses (Zhao et al., 2017), OHEM (Shrivastava et al., 2016), or global average pooling (Liu et al., 2015) that can further boost the performances used in the previous work. We mark all attention-based models with * in which diverse attentions compose the segmentation heads. Though semantic segmentation is a saturated task, and most contemporary published works have approximately the same performances or only bring 0.1% to 0.5% improvement over their baselines in mIoU, Hamburger shows considerable improvements over attention models.

Table 4: Comparisons with state-of-the-art on the PASCAL VOC test set w/o COCO pretraining.

Method	mIoU(%)
PSPNet (Zhao et al., 2017)	82.6
DFN* (Yu et al., 2018)	82.7
EncNet (Zhang et al., 2018)	82.9
DANet* (Fu et al., 2019)	82.6
DMNet* (He et al., 2019a)	84.4
APCNet* (He et al., 2019b)	84.2
CFNet* (Zhang et al., 2019b)	84.2
SpyGR* (Li et al., 2020)	84.2
SANet* (Zhong et al., 2020)	83.2
OCR* (Yuan et al., 2020)	84.3
HamNet	**85.4**

Table 5: Results on the PASCAL-Context Val set.

Method	mIoU(%)
PSPNet (Zhao et al., 2017)	47.8
SGR* (Liang et al., 2018)	50.8
EncNet (Zhang et al., 2018)	51.7
DANet* (Fu et al., 2019)	52.6
EMANet* (Li et al., 2019)	53.1
DMNet* (He et al., 2019a)	54.4
APCNet* (He et al., 2019b)	54.7
CFNet* (Zhang et al., 2019b)	54.0
SpyGR* (Li et al., 2020)	52.8
SANet* (Zhong et al., 2020)	53.0
OCR* (Yuan et al., 2020)	54.8
HamNet	**55.2**

3.5 Image Generation

Attention presents as the global context description block in deep generative models like GANs. Most state-of-the-art GANs for conditional image generation integrate self-attention into their architectures since SAGAN (Zhang et al., 2019a), *e.g.,* BigGAN (Brock et al., 2018), S^3GAN (Lučić et al., 2019), and LOGAN (Wu et al., 2019). It is convincing to benchmark MD-based Hamburger in the challenging image generation task on ImageNet (Deng et al., 2009).

Experiments are conducted to compare Hamburger with self-attention on ImageNet 128×128. Self-attention is replaced by Hamburger with NMF ham in both generator and discriminator at feature resolution 32×32, named as HamGAN. HamGAN achieves an appreciable improvement in Fréchet Inception Distance (FID) (Heusel et al., 2017) over SAGAN. Additionally, we compare with a recently developed attention variant Your Local GAN (YLG) (Daras et al., 2020) using their codebase and the same training settings. HamGAN offers over 5% improvement in FID while being over 15% faster for the total *training time* (1.54 iters/sec of HamGAN & 1.31 iters/sec of YLG, averaged from 1000 iterations) on the same training platform TPUv3-8.

Table 6: Results on ImageNet 128×128. * are from Tab. 1 and Tab. 2 of Zhang et al. (2019a).

Method	FID↓
SNGAN-projection*	27.62
SAGAN*	18.28
HamGAN	16.05
YLG	15.94
HamGAN	14.88

4 Conclusion

This paper proposes a method to incorporate human priors into the architecture of global information modules. We formulate the global context as a low-rank completion problem. Inspired by such a low-rank prior, we develop the Hamburger module based on well-studied matrix decomposition models. By formulating matrix decomposition's objective function, the computational graph created by its optimization algorithm naturally defines ham, Hamburger's core architecture. Hamburger learns interpretable global context via denoising and completing its input and improves the spectrum's concentration. It is startling that, when prudently coped with the backward gradient, even simple matrix decomposition proposed 20 years ago is as powerful as self-attention in challenging vision tasks semantic segmentation and image generation, as well as light, fast, and memory-efficient. We plan to extend Hamburger to natural language processing by integrating positional information and designing a decoder like Transformer, build a theoretical foundation for the one-step gradient trick, or find a better method to differentiate MDs and ensemble advanced MDs in the future.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016. 5, 17

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015. 15

Yoshua Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017. 2

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 8

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)*, pp. 801–818, 2018. 17

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017. 15

Giannis Daras, Augustus Odena, Han Zhang, and A. Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14519–14527, 2020. 8, 17

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009. 2, 8, 16, 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)*, pp. 4171–4186, 2019. 1

Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, Jan 2001. ISSN 1573-0565. doi: 10.1023/A:1007612920971. 1, 14

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3225–3233. Curran Associates, Inc., 2016. 15

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010. 1, 6, 7, 16

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154, 2019. 7, 8, 17

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019. 2, 15

R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6): 2325–2383, Oct 1998. doi: 10.1109/18.720541. 1, 4

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6691–6701. Curran Associates, Inc., 2017. 15

Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3561–3571, 2019a. 8

Junjun He, Zhongying Deng, L. Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7511–7520, 2019b. 8

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6, 7, 16, 17

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 8, 17

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *International Conference on Learning Representations*, 2018. 15

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015. 4, 6

Sarthak Jain and Byron C Wallace. Attention is not explanation. 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)*, pp. 3543–3556, 2019. 15

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 17

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3): 455–500, 2009. doi: 10.1137/07070111X. 2

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pp. 8606–8616, 2018. 15

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999. 1, 4

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp (eds.), *Advances in Neural Information Processing Systems 13*, pp. 556–562. MIT Press, 2001. 4

Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9167–9176, 2019. 8, 15, 17

Xia Li, Y. Yang, Qijie Zhao, Tian cheng Shen, Zhouchen Lin, and Hong-Cheu Liu. Spatial pyramid based graph reasoning for semantic segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8947–8956, 2020. 8

Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems*, pp. 1853–1863, 2018. 8, 15

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017. 15

W. Liu, Andrew Rabinovich, and A. Berg. Parsenet: Looking wider to see better. *ArXiv*, abs/1506.04579, 2015. 7

Canyi Lu, Jinhui Tang, Shuicheng Yan, and Zhouchen Lin. Generalized nonconvex nonsmooth low-rank minimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4130–4137, 2014. 2

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019. 15

Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *International Conference on Machine Learning*, pp. 4183–4192, 2019. 8

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015. 15

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010. 2

Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *ArXiv*, abs/1802.05637, 2018. 16, 17

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018. 17

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pp. 2204–2212, 2014. 15

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016. 7, 17

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 891–898, 2014. 2, 7, 16

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2249–2255, 2016. 15

Niki Parmar, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, pp. 68–80, 2019. 15

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019. 5

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017. 15

Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2931–2951, 2019. 15

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016. 7

Yunzhe Tao, Qi Sun, Qiang Du, and Wei Liu. Nonlocal neural networks, nonlocal diffusion and nonlocal modeling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 496–506. Curran Associates, Inc., 2018. 15

Madeleine Udell, Corinne Horn, Reza Zadeh, and Stephen Boyd. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2016. ISSN 1935-8237. doi: 10.1561/2200000055. 2

Sjoerd Van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353*, 2018. 15

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. 1, 2, 15

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*, pp. 7794–7803, 2018. 1, 3, 15

Paul J Werbos et al. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. 1, 5

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, 2019. 15

John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pp. 2080–2088, 2009. 2

Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy P. Lillicrap. Logan: Latent optimisation for generative adversarial networks. *ArXiv*, abs/1912.00953, 2019. 8

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 *International conference on machine learning*, pp. 2048–2057, 2015a. 15

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 *International conference on machine learning*, pp. 2048–2057, 2015b. 2

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1857–1866, 2018. 8

Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 17

Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2020. 8, 17

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *ICML*, 2019a. 1, 3, 6, 8, 15, 16, 17

Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7151–7160, 2018. 8

Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 548–557, 2019b. 8, 17

Songyang Zhang, Shipeng Yan, and Xuming He. LatentGNN: Learning efficient non-local relations for visual recognition. In *Proceedings of the 36th International Conference on Machine Learning(ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7374–7383, Long Beach, California, USA, 09–15 Jun 2019c. PMLR. 15

Zhengdong Zhang, Arvind Ganesh, Xiao Liang, and Yi Ma. Tilt: Transform invariant low-rank textures. *International journal of computer vision*, 99(1):1–24, 2012. 2

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017. 7, 8, 17

Zilong Zhong, Z. Lin, Rene Bidart, Xiaodan Hu, Ibrahim Ben Daya, J. Li, and A. Wong. Squeeze-and-attention networks for semantic segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13062–13071, 2020. 8

A TABLE OF NOTION

Table 7: Summary of notations in this paper

β	A scalar.
$\mathbf{x}$	A vector.
$\boldsymbol{X}$	A matrix.
$\mathcal{Z}$	A tensor.
$\mathbf{1}_n$	A vector whose n elements are all 1.
$\mathbf{x}_i$	i-th column of matrix $\boldsymbol{X}$.
$\mathbf{h}^t$	Vector $\mathbf{h}$ at time step t.
$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	Jacobian matrix of $\mathbf{y}$ w.r.t. $\mathbf{x}$.
$\Vert\boldsymbol{X}\Vert$	Operator norm.
$\Vert\boldsymbol{X}\Vert_F$	Frobenius norm.
$diag$	Map a vector to a diagonal matrix.
$cosine$	Cosine similarity used in Alg. 1.
$softmax$	Column-wise softmax function.
$normalize$	Column-wise normalization by L_2 norm.

B HAMS

Additionally, we introduce another type of ham adopted by Hamburger, Concept Decomposition.

Concept Decomposition We first enhance Concept Decomposition (Dhillon & Modha, 2001) to the following form:

$$\begin{aligned} &\min_{\boldsymbol{D},\boldsymbol{C}} \Vert\boldsymbol{X} - \boldsymbol{D}\boldsymbol{C}\Vert_F^2 + \beta\Vert\boldsymbol{C}\Vert_F^2 \\ &\text{s.t. } \boldsymbol{D} \in \arg\max_{\boldsymbol{D}} \mathcal{Q}\left(\boldsymbol{D}, \boldsymbol{X}\right). \end{aligned} \tag{14}$$

This problem has a closed solution *w.r.t.* $\boldsymbol{C}$ under a given $\boldsymbol{D}$, *i.e.,* $\boldsymbol{C} = (\boldsymbol{D}^\top\boldsymbol{D} + \beta\boldsymbol{I})^{-1}\boldsymbol{D}^\top\boldsymbol{X}$. Since $\boldsymbol{D}^\top\boldsymbol{D} + \beta\boldsymbol{I}$ is a positive definite matrix with a regularized conditional number, the inverse can be more numerically stable than the original one where a semi-positive definite matrix $\boldsymbol{D}^\top\boldsymbol{D}$ is given under $\beta = 0$. In practice, 0.01 or 0.1 makes no difference for β.

Algorithm 3 Soft CD

Input $\boldsymbol{X}$. Initialize $\boldsymbol{D}$, $\boldsymbol{C}$
for k from 1 to K **do**
  $\boldsymbol{C} \leftarrow softmax(\frac{1}{T}cosine(\boldsymbol{D}, \boldsymbol{X}))$
  $\boldsymbol{D} \leftarrow normalize(\boldsymbol{X}\boldsymbol{C}^\top)$
end for
$\boldsymbol{C} \leftarrow (\boldsymbol{D}^\top\boldsymbol{D} + \beta\boldsymbol{I})^{-1}\boldsymbol{D}^\top\boldsymbol{X}$
Output $\bar{\boldsymbol{X}} = \boldsymbol{D}\boldsymbol{C}$.

The dictionary in CD is given by spherical K-means (Dhillon & Modha, 2001) with objective $\mathcal{Q}\left(\boldsymbol{D}, \boldsymbol{X}\right)$, as mentioned in Eq. (14).

$$\begin{aligned} \arg\max_{\boldsymbol{D},\{\pi_j\}_r} \quad & \textstyle\sum_{j=1}^r \sum_{\mathbf{x}\in\pi_j} cosine\left(\mathbf{x}, \mathbf{d}_j\right) \\ \text{s.t.} \quad & \Vert\mathbf{d}_j\Vert = 1. \end{aligned} \tag{15}$$

The same strategy as VQ is adopted to make the whole algorithm differentiable, however, in which each column of $\boldsymbol{D}$ is normalized to be a unit vector and thus differs from VQ.

C Related Work

The last five years have witnessed a roaring success of attention mechanisms (Bahdanau et al., 2015; Mnih et al., 2014; Xu et al., 2015a; Luong et al., 2015) in deep learning. Roughly speaking, the attention mechanism is a term of adaptively generating the targets' weights to be attended according to the requests. Its architectures are diverse, and the most well-known one is dot product self-attention (Vaswani et al., 2017). The attention mechanism has a wide range of applications, from the single source (Lin et al., 2017) to multi-source inputs (Luong et al., 2015; Parikh et al., 2016), from global information discovery (Wang et al., 2018; Zhang et al., 2019a) to local feature extraction (Dai et al., 2017; Parmar et al., 2019).

Previous researchers attempt to explain the effectiveness of attention mechanisms from numerous aspects. Capturing long-range dependencies (Wang et al., 2018), sequentially decomposing visual scenes (Eslami et al., 2016; Kosiorek et al., 2018), and inferring relationships between the part and the whole (Sabour et al., 2017; Hinton et al., 2018), simulating interactions between objects (Greff et al., 2017; Van Steenkiste et al., 2018), and learning the dynamics of environments (Goyal et al., 2019) are often considered as the underlying mechanisms of attention.

One common idea from biology is that attention simulates the emergence of concerns in many unconscious contexts (Xu et al., 2015a). A more detailed mathematical description contains many angles. Some work tries to interpret the attention mechanism by visualizing or attacking attention weights (Serrano & Smith, 2019; Jain & Wallace, 2019; Wiegreffe & Pinter, 2019), while others formulate attention into non-local operation (Wang et al., 2018) or diffusion models (Tao et al., 2018; Lu et al., 2019) or build attention-like models via Expectation Maximization (Greff et al., 2017; Hinton et al., 2018; Li et al., 2019) or Variational Inference (Eslami et al., 2016) on a mixture model. A connection between transformer and graph neural network is discussed as well (Liang et al., 2018; Zhang et al., 2019c). Overall, discussions towards attention are still far from reaching agreements or consistent conclusions.

D Proof of Propositions

We investigate an abstract RNN model inspired by numerical methods to understand the drawbacks of BPTT algorithm in differentiating the optimization algorithm of MDs, $\mathcal{M}$. We show the propositions in Sec. 2.3 to illustrate the unstable gradient from $\mathcal{M}$ when using BPTT algorithm, considering MDs' nature as optimization algorithms.

Proposition 1 The iterations of $\mathcal{F}$ have linear convergence.

Proof. It is obvious that $\mathcal{F}$ is a contraction mapping *w.r.t.* $\mathbf{h}$ under arbitrary given $\mathbf{x}$. We can then conclude $\{\mathbf{h}^t\}$ is a Cauthy sequence and $\mathcal{F}(*, \mathbf{x})$ admits a unique fixed point $\mathbf{h}^*$ due to Banach Fixed Point Theorem.

$$\begin{aligned} \|\mathbf{h}^{t+1} - \mathbf{h}^*\| &= \|\mathcal{F}(\mathbf{h}^t, \mathbf{x}) - \mathcal{F}(\mathbf{h}^*, \mathbf{x})\| \\ &\leq L_h \|\mathbf{h}^t - \mathbf{h}^*\| \end{aligned} \tag{16}$$

Eq. (16) shows the linear convergence.

Proposition 2 $\lim\limits_{t\to\infty} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}^*}(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \mathbf{h}^*})^{-1}\frac{\partial \mathcal{F}}{\partial \mathbf{x}}$.

Proof. Note that $\mathcal{F}(*, \mathbf{x})$ admits a unique fixed point $\mathbf{h}^*$ under arbitrary given $\mathbf{x}$, *i.e.,*

$$\mathbf{h}^* = \mathcal{F}(\mathbf{h}^*, \mathbf{x}) \quad \Longrightarrow \quad \mathbf{h}^* - \mathcal{F}(\mathbf{h}^*, \mathbf{x}) = \mathbf{0} \tag{17}$$

By differentiating the above equation, we can obtain

$$(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \mathbf{h}^*})\frac{\partial \mathbf{h}^*}{\partial \mathbf{x}} = \frac{\partial \mathcal{F}}{\partial \mathbf{x}} \tag{18}$$

The Jacobian matrix $\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \mathbf{h}^*}$ is invertible, which implies the existence of the implicit function $\mathbf{h}^*(\mathbf{x})$. Immediately, we have

$$\lim_{t\to\infty} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}^*}\frac{\partial \mathbf{h}^*}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}^*}(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \mathbf{h}^*})^{-1}\frac{\partial \mathcal{F}}{\partial \mathbf{x}}, \tag{19}$$

which completes the proof.

Proposition 3 $\lim_{t\to\infty} \|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^0}\| = 0$, $\lim_{t\to\infty} \|\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\| \leq \frac{L_{\mathcal{G}} L_x}{1-L_h}$.

Proof.

$$\|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^0}\| = \|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^t} \prod_{i=1}^{t} \frac{\partial \mathbf{h}^i}{\partial \mathbf{h}^{i-1}}\| \leq \|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^t}\| \prod_{i=1}^{t} \|\frac{\partial \mathbf{h}^i}{\partial \mathbf{h}^{i-1}}\| \leq L_{\mathcal{G}} L_h^t \tag{20}$$

Then we have:

$$\lim_{t\to\infty} \|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^0}\| = 0. \tag{21}$$

$$\begin{aligned}
\|\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\| &= \|\sum_{i=0}^{t} \frac{\partial \mathbf{y}}{\partial \mathbf{h}^t} \prod_{j=i+1}^{t} \frac{\partial \mathbf{h}^j}{\partial \mathbf{h}^{j-1}} \frac{\partial \mathbf{h}^i}{\partial \mathbf{x}}\| \\
&\leq \sum_{i=0}^{t} \|\frac{\partial \mathbf{y}}{\partial \mathbf{h}^t}\| \prod_{j=i+1}^{t} \|\frac{\partial \mathbf{h}^j}{\partial \mathbf{h}^{j-1}}\| \|\frac{\partial \mathbf{h}^i}{\partial \mathbf{x}}\| \\
&\leq L_{\mathcal{G}} (\sum_{i=0}^{t-1} L_h^i) L_x \\
&= \frac{L_{\mathcal{G}} L_x (1 - L_h^t)}{1 - L_h}
\end{aligned} \tag{22}$$

Then we have:

$$\lim_{t\to\infty} \|\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\| \leq \frac{L_{\mathcal{G}} L_x}{1 - L_h}. \tag{23}$$

E Datasets

PASCAL VOC The PASCAL VOC dataset (Everingham et al., 2010) is a widely used dataset in both semantic segmentation and detection. For segmentation, it contains 10,582 images for training, 1,449 images for validation and 1,456 images for testing. PASCAL VOC dataset involves 20 foreground object classes and a background class for segmentation and detection.

PASCAL Context The PASCAL Context dataset (Mottaghi et al., 2014) is a challenging dataset in semantic segmentation, which provides detailed labels and involves 59 foreground object classes and a background class for segmentation. It consists of 4,998 and 5,105 images in training and validation set, respectively.

ILSVRC 2012 The ILSVRC 2012 (ImageNet) (Deng et al., 2009) dataset contains 1.3M training samples and 50k test images, categorized into 1000 object classes. We resize images to resolution 128×128, as done in SNGAN with projection (Miyato & Koyama, 2018) and SAGAN (Zhang et al., 2019a).

F Details of Experiments

F.1 Abalation Experiments

We use ResNet-50 (He et al., 2016) with the output stride 16 as the backbone. The backbone is pre-trained on ImageNet (Deng et al., 2009). We apply a poly-learning rate policy under batch size 12 and 30k iterations (about 35 epochs) for fast experiments (less than 12H using 1 NVIDIA TITAN Xp GPU). The initial learning rate is set to 0.009, multiplied by $(1 - \frac{iter}{iter_{max}})^{0.9}$ after each iteration, with momentum 0.9 and weight decay 0.0001. Hyperparameters of Hamburger are the same as Appendix F.3.

F.2 A Comparison with Attention Mechanism

Dual Attention (Fu et al., 2019) is from the official implementation. We report FLOPs according to (Molchanov et al., 2016). Real-time overhead is measured by NVML on NVIDIA TITAN Xp GPU with a input tensor $\mathcal{Z} \in \mathbb{R}^{512\times64\times64}$. Inference times are averaged results from 10 repeats of 10000 runs.

F.3 Semantic Segmentation

Architectures We use ResNet-101 (He et al., 2016) with the ouptput strid 8 as our backbone. We adopt dilated convolution (Chen et al., 2018) to preserve more detail spatial information and enlarge receptive field as done in the backbone of state-of-the-art attention models (Fu et al., 2019; Li et al., 2019; Zhang et al., 2019b). We employ a 3×3 convolution layer with BN and ReLU to reduce channels from 2048 to 512 and then add Hamburger on the top of the backbone. Note that the input of Hamburger is a tensor $\mathcal{Z} \in \mathbb{R}^{C\times H\times W}$. We unfold $\mathcal{Z}$ to a matrix $\boldsymbol{Z} \in \mathbb{R}^{C\times HW}$ and set $d_z = C$ and $n = HW$ for Hamburger. Latent dimension d and r, *i.e.,* the column vectors' dimension of the input matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ to $\mathcal{M}$ and the number of atoms in the dictionary $\boldsymbol{D} \in \mathbb{R}^{r\times d}$, are set to 512 and 64. The iterations of MD's optimization algorithm, K, are set to 6. Non-negative Matrix Factorization (NMF) is our default ham for semantic segmentation.

Data augmentation In the training stage, we apply random left-right flipping, random scaling (from 0.5 to 2), and cropping to augment the training data. Images are resized to 513×513 for the PASCAL VOC dataset and the PASCAL Context dataset. In the test stage, the multi-scale and flipping strategy is applied as other state-of-the-art attention-based models (Fu et al., 2019; Yuan & Wang, 2018; Yuan et al., 2020).

Optimization We use mini-batch SGD with momentum 0.9 to train HamNet. Synchronized Batch Normalization is adopted in experiments on semantic segmentation. All backbones are fine-tuned from ImageNet (Deng et al., 2009) pre-training. Following previous works (Zhao et al., 2017; Chen et al., 2018), we apply a poly-learning rate policy. The initial learning rate is multiplied by $(1 - \frac{iter}{iter_{max}})^{0.9}$. For the PASCAL VOC dataset, learning rate, weight decay, batch size, iterations are set to 0.009, 0.0001, 16, and 60k, respectively. We fine-tune HamNet on the PASCAL VOC trainval set with the learning rate down to a tenth. The learning rate, weight decay, batch size, iterations are 0.002, 0.0001, 16, and 25k for the PASCAL-Context dataset.

F.4 Image Generation

We use the official GAN codebase[1] from Tensorflow (Abadi et al., 2016) and TF-GAN to train HamGAN and evaluate FID.

Architectures Experiments on ImageNet are conducted using the same architecture as SAGAN (Zhang et al., 2019a), and YLG (Daras et al., 2020), including Spectral Normalization (Miyato et al., 2018) in both the generator and the discriminator, conditional Batch Normalization in the generator, and class projection in the discriminator (Miyato & Koyama, 2018). Hamburger with NMF ham is placed at feature resolution 32×32 in both the generator and the discriminator where self-attention can obtain the best FID according to Zhang et al. (2019a). We use $d = 8r$ for Hamburger, and d is the same as the input channels, while the optimization steps K are 6. Restricted to expenditures of training GANs on ImageNet, d, r, and K are decided according to the ablation experiments on semantic segmentation without new ablation experiments.

Optimization For all models, we use Adam (Kingma & Ba, 2015) optimizer with TTUR (Heusel et al., 2017). HamGAN employs the same training settings as SAGAN (Miyato et al., 2018) and YLG (Daras et al., 2020), respectively.

Evaluation metrics The quality of images generated by GANs are evaluated by Fréchet Inception Distance (FID) (Heusel et al., 2017). Lower FID indicates that the model can generate higher-fidelity images. In our experiments, 50k images are sampled from the generator to compute FID.

[1] https://github.com/tensorflow/gan

G FURTHER RESULTS FROM ABLATION EXPERIMENTS

Initialization We test four types of initialization for the dictionary $\boldsymbol{D}$, including fixed initialization, learned initialization, random initialization, and warm start with online update. Usually random initialization is the best choice that means we can sample each entry of $\boldsymbol{D}$ from a given distribution like $\text{Uniform}(0, 1)$ for NMF as the initialization of the optimization algorithm $\mathcal{M}$. Learned initialization is always the worst one since BPTT algorithm is employed to learn the initialization that the gradient from $\mathcal{M}$ may impede the training of the backbone, instead of one-step gradient. Warm start benefits MD with unit vectors in the dictionary $\boldsymbol{D}$ like CD.

Table 8: Ablation on initializations.

Init	**NMF**	**CD**	**VQ**
fixed	77.4(77.3)	77.7(77.4)	77.3(76.9)
learned	76.8(76.5)	75.0(73.7)	75.9(75.8)
random	**78.3(77.8)**	77.9(77.3)	77.7(**77.4**)
online	77.8(77.5)	**78.1(77.5)**	**78.0**(77.2)

Temperature T As we have claimed, when T approaches 0, we can get a solution close to the original problem in both VQ and CD. In the experiments of VQ and CD, a lower temperature T is more recommended to solve a better D for MD.

Table 9: Influence of temperature T with CD ham.

Temperature T	**mIoU**(%)
1	77.1(77.0)
0.1	78.2(77.5)
0.01	78.1(77.5)

Iterations K We take the iterations K of optimization algorithms $\mathcal{M}$ for all three MD models, NMF, CD, and VQ, into our consideration. More iterations and even fully converged results for $\mathcal{M}$ are tested in the evaluation stage but worse than little optimization steps. The smaller K, ranging from 1 to 8, can be treated as *early stopping* for the optimization algorithm $\mathcal{M}$, obtaining satisfactory performances. For a detailed visualization, see Fig. 8, Fig. 9, Fig. 10.

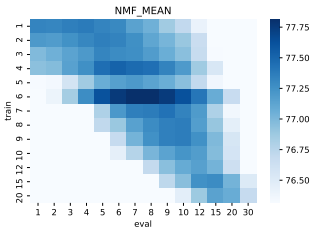

Figure 8: Impacts of K on NMF

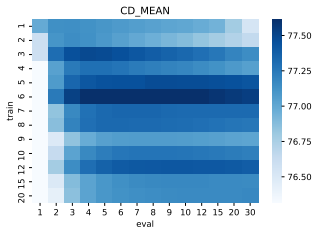

Figure 9: Impacts of K on CD

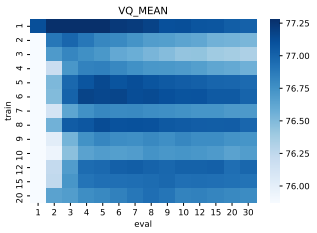

Figure 10: Impacts of K on VQ