Visual Attention with Sparse and Continuous Transformations

António Farinhas
antonio.farinhas@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Portugal
January 2021

Abstract

Visual attention mechanisms have become an important component of neural network models for Computer Vision applications, allowing them to attend to finite sets of objects or regions and identify relevant features. A key component of attention mechanisms is the differentiable transformation that maps scores representing the importance of each feature into probabilities. The usual choice is the softmax transformation, whose output is strictly dense, assigning a probability mass to every image feature. This density is wasteful, given that non-relevant features are still taken into consideration, making attention models less interpretable. Until now, visual attention has only been applied to discrete domains – this may lead to a lack of focus, where the attention distribution over the image is too scattered. Inspired by the continuous nature of images, we explore continuous-domain alternatives to discrete attention models. We propose solutions that focus on both the continuity and the sparsity of attention distributions, being suitable for selecting compact and sparse regions such as ellipses. The former encourages the selected regions to be contiguous and the latter is able to single out the relevant features, assigning exactly zero probability to irrelevant parts. We use the fact that the Jacobian of these transformations are generalized covariances to derive efficient backpropagation algorithms for both unimodal and multimodal attention distributions. Experiments on Visual Question Answering show that continuous attention models generate smooth attention maps that seem to better relate with human judgment, while achieving improvements in terms of accuracy over grid-based methods trained on the same data. Code is available at `https://github.com/deep-spin/mcan-vqa-continuous-attention`.

1. Introduction

Visual attention mechanisms are an important component of modern Deep Learning models. They appear as a way to mimic the human visual system that selectively attends to the most relevant parts of visual stimuli, being able to process large amounts of information in parallel [16]. In the context of Aerospace Engineering, they have been used to improve the performance of off-road robots [18] and in Earth Observation [10]. Also, intelligent agents are likely to be asked to perform autonomous vision-based tasks such as navigation, aerial mapping and object delivery [4].

A neural network with attention automatically learns the relevance of any element of the input by generating a set of weights and taking them into account while performing the proposed task. Moreover, these models are usually very complex and remain black-box models: humans cannot easily understand their inner decision making process. In addition to boosting the performance of a model, attention mechanisms can provide insights into the model's reasoning behind its prediction [22]. The visualization of attention weights can help us analyze the outputs of a neural network and possibly understand some unpredictable outcomes [6].

A key component of visual attention mechanisms is the differentiable transformation that maps scores representing the importance of each feature into probabilities. The usual choice is the softmax transformation, whose output is **strictly dense**, assigning a probability mass to **every image feature**. This density is wasteful, given that non-relevant features are still taken into consideration, making attention models less interpretable [12]. Furthermore, although image data is naturally continuous, visual attention has only been applied to **discrete domains**. In certain applications this may lead to a **lack of focus**, where the attention distribution over the image is too scattered.

We explore continuous-domain alternatives to discrete attention models. We construct 2D continuous attention mechanisms that are able to increase focus on relevant image regions, leading to more interpretable predictions. Our solutions focus on both the **continuity** and the **sparsity** of attention distributions, being able to select compact and sparse regions in images. The first takes adjacency into account and encourages the selected regions to be contiguous and the second is able to single out the relevant features, assigning exactly zero probability to irrelevant regions.

Summing up our main contributions, we propose a framework for using continuous attention with images and derive efficient algorithms for the evaluation and gradient computation of 2D α-entmax continuous attention mechanisms, for $\alpha \in \{1, 2\}$. Then, we introduce novel multimodal continuous attention mechanisms by using mixtures of unimodal attention densities. Finally, we plug our 2D continuous attention mechanisms in a Visual Question Answering model in order to improve focus and possibly provide better explanations via smoother attention maps. In terms of accuracy, we obtain small improvements over grid-based methods trained on the same data.

Notation. Consider a measure space $(S, \mathcal{A}, \nu)$, where S is a set, $\mathcal{A}$ is a σ-algebra and ν is a measure. We denote the set of ν-absolutely continuous probability measures as $\mathcal{M}_+^1(S)$. From the Radon-Nikodym theorem [8, §31], each element of $\mathcal{M}_+^1(S)$ is identified with a probability density function $p : S \to \mathbb{R}_+$, with $\int_S p(t)\, d\nu(t) = 1$; for convenience, we often drop $d\nu(t)$ from the integral. We denote the measure of A as $|A| = \nu(A) = \int_A 1$ and the support of a density $p \in \mathcal{M}_+^1(S)$ as $\mathrm{supp}(p) = \{t \in S \mid p(t) > 0\}$. Given $\phi : S \to \mathbb{R}^m$, we write expectations as $\mathbb{E}_p[\phi(t)] := \int_S p(t)\, \phi(t)$. Finally, we define $[a]_+ := \max\{a, 0\}$.

2. From discrete to continuous attention in 2D

2.1. Regularized prediction maps

The work by Blondel *et al.* [3] introduced Ω-regularized prediction maps for finite domains. Consider an input vector $x \in \mathcal{X}$ and a parametrized model $f : \mathcal{X} \to \mathbb{R}^{|S|}$, producing a score vector $\theta = f(x) \in \mathbb{R}^{|S|}$. For instance, θ can be label scores computed by a neural network model, f. Assuming a regularization function Ω, with $\mathrm{dom}(\Omega) \subseteq \triangle^{|S|}$, this framework allows us to map vectors $\theta \in \mathbb{R}^{|S|}$ into probability vectors in the simplex $\triangle^{|S|}$. Ω-regularized prediction maps can be extended to arbitrary measure spaces $\mathcal{M}_+^1(S)$, assuming that $\Omega : \mathcal{M}_+^1(S) \to \mathbb{R}$ is a lower semi-continuous, proper and strictly convex function. The **Ω-regularized prediction map** (Ω-RPM) $\hat{p}_\Omega : \mathcal{F} \to \mathcal{M}_+^1(S)$ is defined as:

$$\hat{p}_\Omega[f] = \underset{p \in \mathcal{M}_+^1(S)}{\arg\max}\, \mathbb{E}_p[f(t)] - \Omega(p), \tag{1}$$

where $\mathcal{F}$ is the set of functions for which the maximizer exists and is unique. The regularizer Ω in (1) can be chosen in order to recover transformations such as softmax and sparsemax, when S is finite. For the case where S is continuous, more interesting examples of regularizational functionals are shown in the next subsections.

2.2. Choosing the regularization function Ω

The choice of the regularization function Ω leads to different regularized prediction maps – depending on its properties, it can lead to distributions with fixed support within the same family (*e.g.* distributions in the exponential family) (§ 2.2.2) or to alternatives with varying and sparse support, assigning zero probability mass to some entries (§ 2.2.3). We consider generalized negative entropies as regularization functions. Specifically, we consider a generalization of the Shannon's negentropy proposed by Tsallis [20] that uses the notions of *β-exponential* and *β-logarithm* [1]. The **α-Tsallis negentropy** is defined as

$$\Omega_\alpha(p) = \begin{cases} \frac{1}{\alpha(\alpha-1)} \left(\int_S p(t)^\alpha - 1\right), & \alpha \neq 1 \\ \int_S p(t) \log p(t), & \alpha = 1 \end{cases}, \tag{2}$$

where $\Omega_1(p)$ is the Shannon's negentropy and $\Omega_2(p) = \frac{1}{2}\int_S p(t)^2 - \frac{1}{2}$ is known in the literature as Gini-Simpson index.

2.2.1 α-Tsallis negentropy and Ω_α-RPM

Let $\alpha > 0$ and $f \in \mathcal{F}$. [11, Proposition 1] shows that the Ω_α-RPM in (1) can be simply written as

$$\hat{p}_{\Omega_\alpha}[f](t) = \exp_{2-\alpha}(f(t) - A_\alpha(f)), \tag{3}$$

where $A_\alpha : \mathcal{F} \to \mathbb{R}$ is a normalizing function:

$$A_\alpha(f) = \frac{\frac{1}{1-\alpha} + \int_S p_\theta(t)^{2-\alpha} f(t)}{\int_S p_\theta(t)^{2-\alpha}} - \frac{1}{1-\alpha}. \tag{4}$$

Furthermore, it is possible to show (see [11, Proposition 2] or [1, Theorem 5] for a proof) that the normalizing function, A_α (4), is a convex function and its gradient is given by

$$\nabla_\theta A_\alpha(\theta) = \mathbb{E}_{\tilde{p}_\theta^{2-\alpha}}[\phi(t)] = \frac{\int_S p_\theta(t)^{2-\alpha} \phi(t)}{\int_S p_\theta(t)^{2-\alpha}}, \tag{5}$$

where $\tilde{p}^\beta(t) = p(t)^\beta / \|p\|_\beta^\beta$ is the β-escort distribution [20] in which

$$\|p\|_\beta^\beta = \int_S p(t')^\beta d\nu(t') \quad \text{and} \quad \tilde{p}^1(t) = p(t). \tag{6}$$

Figure 1 shows the distributions generated by the Ω_α-RPM for $\alpha \in \{1, 2\}$. The former has full support in $\mathbb{R}^2$ while the latter is able to assign zero probability values.

2.2.2 Shannon's negentropy and Ω_1-RPM

For $\alpha = 1$, $\Omega_1(p)$ is the Shannon's negentropy and the corresponding Ω_1-RPM is:

$$\hat{p}_{\Omega_1}[f](t) = \frac{\exp f(t)}{\int_S \exp(f(t'))d\nu(t')} = \exp(f(t) - A(f)), \tag{7}$$

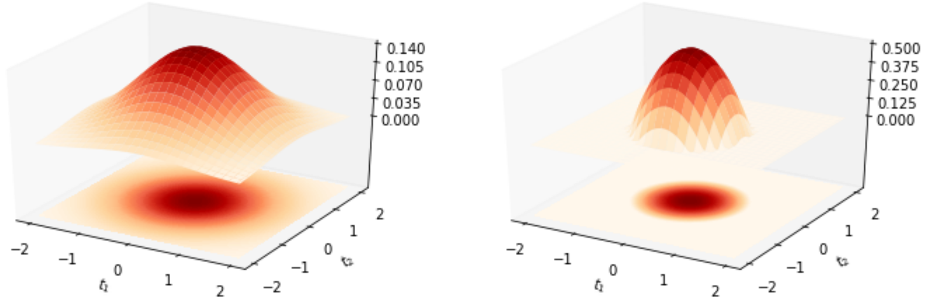

Figure 1. 2D distributions generated by the Ω_α-RPM for $\alpha \in \{1, 2\}$. Left: For $\alpha = 1$, bivariate Gaussian $\mathcal{N}(t; 0, I)$. Right: For $\alpha = 2$, truncated paraboloid $\mathcal{TP}(t; 0, I)$. The *peak* of the density for $\alpha = 1$ ($\mathcal{N}$) is much smaller than for $\alpha = 2$ ($\mathcal{TP}$).

where $A(f) = \log \int_S \exp f(t)$ is the log-partition function (see [11, App. A] for a proof). If S is finite and ν is the *counting measure*, we can write f as a vector in $\mathbb{R}^{|S|}$ and the Ω_1-RPM recovers the softmax transformation,

$$\hat{p}_{\Omega_1}[f] = \operatorname{softmax}(f) = \frac{\exp(f)}{\sum_{k=1}^{|S|} \exp(f_k)} \in \triangle^{|S|}. \tag{8}$$

For continuous domains with $S = \mathbb{R}^N$, ν the *Lebesgue measure*, $\mu \in \mathbb{R}^N$, $\Sigma \in \mathbb{R}^{N \times N} \succ 0$ and choosing $f(t) = -\frac{1}{2}(t-\mu)^\top \Sigma^{-1}(t-\mu)$, the Ω_1-RPM transformation is a multivariate Gaussian,

$$\hat{p}_{\Omega_1}[f] = \mathcal{N}(t; \mu, \Sigma). \tag{9}$$

In particular, this becomes a bivariate Gaussian if $N = 2$.

2.2.3 Gini-Simpson index and Ω_2-RPM

For $\alpha = 2$, $\Omega_2(p)$ is the Gini-Simpson index and the corresponding Ω_2-RPM can be obtained from f by subtracting a constant λ and truncating such that $\int_S \hat{p}_{\Omega_2}[f](t) = 1$:

$$\hat{p}_{\Omega_2}[f](t) = [f(t) - \lambda]_+. \tag{10}$$

For finite S the Ω_2-RPM is the sparsemax transformation. For continuous domains with $S = \mathbb{R}^N$, $\Sigma \in \mathbb{R}^{N \times N}$ positive definite and $f(t) = -\frac{1}{2}(t - \mu)^\top \Sigma^{-1}(t - \mu)$, the Ω_2-RPM transformation is a multivariate truncated paraboloid,

$$\hat{p}_{\Omega_2}[f](t) = \left[-\frac{1}{2}(t - \mu)^\top \Sigma^{-1}(t - \mu) - \lambda \right]_+, \tag{11}$$

with

$$\lambda = -\left(\frac{\Gamma(N/2 + 2)}{\sqrt{\det(2\pi\Sigma)}} \right)^{\frac{2}{2+N}}, \tag{12}$$

where $\Gamma(t)$ is the Gamma function. See [11, Section 2.4] for details.

2.3. Building continuous attention mechanisms

Discrete attention. Assume an input image split in L pieces: an image represented by L grid-level or object-level features with a D-dimensional representation each, packed as a *value matrix* $V \in \mathbb{R}^{D \times L}$. A discrete attention mechanism computes a *score vector* $f = (f_1, ..., f_L) \in \mathbb{R}^L$, in which high scores correspond to more relevant parts of the input. Then, a transformation $\rho : \mathbb{R}^L \to \triangle^L$ (usually softmax) is used to map scores into probabilities, *i.e.*, ρ is applied to the score vector to produce a probability vector $p = \rho(f)$. Finally, p is used to compute a *context vector* as a weighted average, $c = Vp \in \mathbb{R}^D$, that is used to produce the network's decision.

2.3.1 The continuous case: score and value function

Instead of assuming a finite set $S = \{1, \ldots, L\}$, we assume a continuous measure space S (the $\mathbb{R}^2$ plane) and represent the image as a *value function* $V : S \subseteq \mathbb{R}^2 \to \mathbb{R}^D$ that maps points in the $\mathbb{R}^2$ plane onto a D-dimensional vector representation. The score vector is replaced by a *score function* $f : S \to \mathbb{R}$ that can be mapped to a probability density $p : S \to \mathbb{R}_+$, with $\int_S p = 1$. Instead of using a discrete transformation for that mapping, we can use the Ω_α-RPM. The output weighted average (context vector) becomes an expectation of the value function with respect to the probability density, $c = \mathbb{E}_p[V(t)] = \int_S p(t)V(t) \in \mathbb{R}^D$.

The score function f and the value function V need to be parametrized: we consider linear parametrizations in terms of a vector of basis functions and a vector of parameters. We can then define $f_\theta(t) = \theta^\top \phi(t)$ and $V_B(t) = B\psi(t)$, where $\phi : S \to \mathbb{R}^M$ and $\psi : S \to \mathbb{R}^N$ are basis functions and $\theta \in \mathbb{R}^M$ and $B \in \mathbb{R}^{D \times N}$ are parameters.

Defining continuous attention mechanism. Consider that $\Omega : \mathcal{M}_+^1(S) \to \mathbb{R}$ is a regularization functional. An attention mechanism is a mapping $\rho : \mathbb{R}^M \to \mathbb{R}^N$ from an input parameter vector $\theta \in \mathbb{R}^M$ to a vector $r \in \mathbb{R}^N$,

$$\rho(\theta) = r = \mathbb{E}_p[\psi(t)], \tag{13}$$

with $p = \hat{p}_\Omega[f_\theta]$ and $f_\theta(t) = \theta^\top \phi(t)$. If $\Omega = \Omega_\alpha$, this is called α-entmax attention, denoted as ρ_α. The values $\alpha = 1$ and $\alpha = 2$ lead to softmax and sparsemax attention, respectively. The context vector can then be computed as $c = Br$, which is equivalent to $c = \mathbb{E}_p[V_B(t)]$.

Defining the value function. An input image is usually represented as a discrete matrix $H \in \mathbb{R}^{D \times L}$ (*e.g.*, a matrix with D channels and L image locations so that each location is represented by a D-dimensional vector). We can use **multivariate ridge regression** to approximate H and obtain a continuous signal, a value mapping $V_B : S \to \mathbb{R}^D$. Given that $V_B(t) = B\psi(t)$, this consists in optimizing over B to minimize the squared loss plus a ridge penalty. Assuming that $t_l = (\frac{l_1}{\sqrt{L}}, \frac{l_2}{\sqrt{L}})$ for $l_1, l_2 \in [0, \sqrt{L}]$ and choosing the columns of the matrix $F \in \mathbb{R}^{N \times L}$ to be the basis vectors $\psi(t_l)$, we obtain

$$\begin{aligned} B^\star &= \underset{B}{\operatorname{argmin}} \|BF - H\|_F^2 + \lambda \|B\|_F^2 \\ &= H \underbrace{F^\top (FF^\top + \lambda I_N)^{-1}}_{G} = HG \end{aligned} \tag{14}$$

where $\|B\|_F = (\sum_{i=1}^m \sum_{j=1}^n B_{ij}^2)^{1/2}$ is the Frobenius norm of B and $G = F^\top (FF^\top + \lambda I_N)^{-1}$ is a $L \times N$ matrix. For given L and N, both F and G depend only on the value of $\psi(t_l)$ and can be obtained offline. We can choose $N << L$, so that the resulting expression for V_B has ND coefficients, much cheaper than the LD coefficients of H.

Gradient backpropagation. To train models with gradient-based optimization the Jacobian of the α-entmax transformation ρ_α is needed. Martins *et al.* [11] proved an expression for evaluating J_{ρ_α}. For $\beta \geq 0$, a generalized β-covariance, $\mathrm{cov}_{p,\beta}[\phi(t), \psi(t)]$, is defined as

$$\|p\|_\beta^\beta \times \left(\mathbb{E}_{\tilde{p}_\beta}\left[\phi(t)\psi(t)^\top\right] - \mathbb{E}_{\tilde{p}_\beta}[\phi(t)]\, \mathbb{E}_{\tilde{p}_\beta}[\psi(t)]^\top\right), \tag{15}$$

that for $\beta = 1$ recovers the usual covariance. The Jacobian of the α-entmax transformation is then

$$J_{\rho_\alpha}(\theta) = \frac{\partial r(\theta)}{\partial \theta} = \mathrm{cov}_{p,2-\alpha}(\phi(t), \psi(t)), \tag{16}$$

with $p = \hat{p}_\Omega[f_\theta]$ and $f_\theta(t) = \theta^\top \phi(t)$. (16) allows efficient gradient backpropagation with continuous attention and will be used in the next section to derive expressions for the gradient computation of 2D α-entmax continuous attention mechanisms for $\alpha \in \{1, 2\}$.

3. 2D continuous attention with Gaussian RBFs

3.1. How can we write the attention density?

A continuous attention mechanism is as a mapping from an input parameter vector θ to a vector $r = \mathbb{E}_p[\psi(t)]$, where $p = \hat{p}_\Omega[f_\theta]$ and $f_\theta(t) = \theta^\top \phi(t)$. If $\Omega = \Omega_\alpha$, $\alpha \in \{1, 2\}$ leads to softmax and sparsemax attention, respectively. Taking $S = \mathbb{R}^2$, $\hat{p}_{\Omega_1}[f_\theta]$ is a bivariate Gaussian $\mathcal{N}(t; \mu, \Sigma)$ and $\hat{p}_{\Omega_2}[f_\theta]$ becomes a bivariate truncated paraboloid $\mathcal{TP}\ (t; \mu, \Sigma)$. In both cases, μ and Σ are related to the canonical parameters by $\theta = [\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1}]$.

We focus on a combined attention setting where we assume that we have access to L discrete attention weights α_i, where $i \in \{1, ..., L\}$. First, we obtain $p \in \triangle^L$ from discrete attention. For 2D continuous softmax we have $\mu = \mathbb{E}_p[t]$ and $\Sigma = \mathbb{E}_p[tt^\top] - \mu\mu^\top$. This property does not hold for $\mathcal{TP}$ distributions; thus, for 2D continuous sparsemax, we need to find out how to obtain the parameter Σ from the variance. To estimate Σ of a $\mathcal{TP}(t; \mu, \Sigma)$ from discrete attention weights we perform the following steps:

- express the variance as a function of Σ, $f(\Sigma) = \iint \mathcal{TP}(t; \mu, \Sigma) tt^\top dt$;
- from discrete attention, obtain the variance $\mathrm{Var} = \mathbb{E}_p[tt^\top] - \mu\mu^\top$;
- invert f to obtain $\Sigma = f^{-1}(\mathrm{Var})$.

We now put forward a theorem that results from following this approach (the proof in Appendix A.1 of the thesis).

Theorem 1 *Let $\mathcal{TP}\ (t; \mu, \Sigma)$ be a d-dimensional multivariate truncated paraboloid where $t, \mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d} \succ 0$, defined as in [11, Section 2.4]. Let $\lambda = -\left(\frac{\Gamma(d/2+2)}{|2\pi\Sigma|^{1/2}}\right)^{\frac{2}{2+d}}$ be the constant that ensures the distribution normalizes to 1, where $\Gamma(t)$ is the Gamma function. Then, the variance of $\mathcal{TP}$ is related to Σ by*

$$\mathrm{Var}(\Sigma) = f(\Sigma) = -\frac{\lambda\, \Sigma}{\frac{d}{2} + 2}. \tag{17}$$

Example. *For $d = 2$,*

$$\mathrm{Var}(\Sigma) = -\frac{\lambda\Sigma}{3} = \frac{\Sigma}{3\sqrt{\pi}\, |\Sigma|^{1/4}} \tag{18}$$

Using properties of the determinant,

$$|\mathrm{Var}(\Sigma)| = \left|\frac{\Sigma}{3\sqrt{\pi}\, |\Sigma|^{1/4}}\right| = \frac{|\Sigma|^2}{9\pi}. \tag{19}$$

Using (19) to invert (18), we obtain the expression that can be used to compute Σ in a truncated paraboloid density, given the variance Var *computed from discrete attention:*

$$\Sigma = 9\pi |\mathrm{Var}|^{1/2} \mathrm{Var}. \tag{20}$$

Algorithm 1: Continuous softmax attention with $S = \mathbb{R}^D$, $\Omega = \Omega_1$, and Gaussian RBFs.

```
Parameters: Gaussian RBFs $\psi(t) = [\mathcal{N}(t; \mu_j, \Sigma_j)]_{j=1}^N$, basis functions $\phi(t) = [t, \mathrm{vec}(tt^\top)]$, value function $V_B(t) = B\psi(t)$
            with $B \in \mathbb{R}^{D \times N}$, score function $f_\theta(t) = \theta^\top \phi(t)$ with $\theta \in \mathbb{R}^M$

Function Forward($\theta := [\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1}]$):
    $r_j \leftarrow \mathbb{E}_{\hat{p}_\Omega[f_\theta]}[\psi_j(t)] = \mathcal{N}(\mu, \mu_j, \Sigma + \Sigma_j), \quad \forall j \in [N]$        // Eq. 23
    return $c \leftarrow Br$ (context vector)

Function Backward($\frac{\partial \mathcal{L}}{\partial c}, \theta := [\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1}]$):
    for $j \leftarrow 1$ to $N$ do
        $\tilde{s} \leftarrow \mathcal{N}(\mu, \mu_j, \Sigma + \Sigma_j)$
        $\tilde{\Sigma} \leftarrow (\Sigma^{-1} + \Sigma_j^{-1})^{-1}$
        $\tilde{\mu} \leftarrow \tilde{\Sigma}(\Sigma^{-1}\mu + \Sigma_j^{-1}\mu_j)$
        $\frac{\partial r_j}{\partial \theta} \leftarrow \mathrm{cov}_{\hat{p}_\Omega[f_\theta]}(\phi(t), \psi_j(t)) = [\tilde{s}(\tilde{\mu} - \mu); \tilde{s}(\tilde{\Sigma} + \tilde{\mu}\tilde{\mu}^\top - \Sigma - \mu\mu^\top)]$        // Eqs. 24,25
    return $\frac{\partial \mathcal{L}}{\partial \theta} \leftarrow \left(\frac{\partial r}{\partial \theta}\right)^\top B^\top \frac{\partial \mathcal{L}}{\partial c}$
```

3.2. Evaluation and gradient computation

We derive expressions for the evaluation and gradient computation of 2D continuous α-entmax attention mechanisms for $\alpha \in \{1, 2\}$, where $\psi(t)$ are Gaussian RBFs.

3.2.1 2D continuous softmax ($\alpha = 1$)

Let us consider an arbitrary D-Dimensional scenario and take $D = 2$. If $S = \mathbb{R}^D$, for $\phi(t) = [t, tt^\top]$, the distribution $p = \hat{p}_{\Omega_1}[f_\theta]$, with $f_\theta(t) = \theta^\top \phi(t)$, is a multivariate Gaussian and $\theta = [\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1}]$. We derive closed-form expressions for the attention mechanism output $\rho_1(\theta)$ in (13) and its Jacobian $J_{\rho_1}(\theta)$ in (16), when $\psi(t)$ are Gaussian RBFs. Check Alg. 1 for pseudo-code.

Forward pass. Each coordinate of the attention mechanism output becomes the integral of a product of Gaussians,

$$\mathbb{E}[\psi(t)] = \int_{\mathbb{R}^D} \mathcal{N}(t; \mu, \Sigma)\mathcal{N}(t; \mu_j, \Sigma_j). \tag{21}$$

The product of two Gaussians is a scaled Gaussian,

$$\mathcal{N}(t; \mu, \Sigma)\mathcal{N}(t; \mu_j, \Sigma_j) = \tilde{s}\mathcal{N}(t; \tilde{\mu}, \tilde{\Sigma}), \tag{22}$$

where $\tilde{s} = \mathcal{N}(\mu; \mu_j, \Sigma + \Sigma_j)$, $\tilde{\Sigma} = (\Sigma^{-1} + \Sigma_j^{-1})^{-1}$ and $\tilde{\mu} = \tilde{\Sigma}(\Sigma^{-1}\mu + \Sigma_j^{-1}\mu_j)$. So, (21) can be computed as:

$$\mathbb{E}[\psi(t)] = \tilde{s} \int_{\mathbb{R}^D} \mathcal{N}(t; \tilde{\mu}, \tilde{\Sigma}) = \tilde{s} \tag{23}$$

Backward pass. We have that each row of the Jacobian $J_{\rho_1}(\theta)$ becomes a first or second moment under the resulting Gaussian:

$$\begin{aligned} \mathrm{cov}_{p,1}(t, \psi(t)) &= \mathbb{E}_p[t\psi_j(t)] - \mathbb{E}_p[t]\mathbb{E}_p[\psi_j(t)] \\ &= \tilde{s} \int_{\mathbb{R}^D} t\mathcal{N}(t; \tilde{\mu}, \tilde{\Sigma}) - \tilde{s}\mu \\ &= \tilde{s}(\tilde{\mu} - \mu), \end{aligned} \tag{24}$$

and, noting that $\Sigma = \mathbb{E}[(t - \mu)(t - \mu)^\top] = \mathbb{E}[tt^\top] - \mu\mu^\top$,

$$\begin{aligned} \mathrm{cov}_{p,1}(tt^\top, \psi(t)) &= \mathbb{E}_p[tt^\top\psi_j(t)] - \mathbb{E}_p[tt^\top]\mathbb{E}_p[\psi_j(t)] \\ &= \tilde{s} \int_{\mathbb{R}^D} tt^\top \mathcal{N}(t; \tilde{\mu}, \tilde{\Sigma}) - \tilde{s}(\Sigma + \mu\mu^\top) \\ &= \tilde{s}(\tilde{\Sigma} + \tilde{\mu}\tilde{\mu}^\top - \Sigma - \mu\mu^\top). \end{aligned} \tag{25}$$

3.2.2 2D continuous sparsemax ($\alpha = 2$)

For $\alpha = 2$, we show how to reduce both the forward and the backward passes to expressions including univariate integrals (with closed form-expression in terms of the erf function) over an interval by using the change of variable formula and working with polar coordinates. We prove that for this case both the attention mechanism output (13), and its Jacobian (16) can be written in terms of functions of the form $-\lambda \int_0^{2\pi} \tilde{s}(\theta)F(\theta)$ and can be easily computed using simple 1D numerical integration methods. Alternatively, we could use 2D methods for numerical integration to solve these expressions directly; yet, to obtain good results we would have to use complicated bivariate adaptive methods that take a lot of time to reach convergence and that are not GPU friendly - it would be impracticable to plug these mechanisms in neural networks and train them end-to-end. Nevertheless, we study different ways to compute 2D integrals over ellipses, including the simplistic approach that we use in this work: we show in Section 5 that for VQA, in practice, we can approximate these integrals with naive sums, without compromising the overall performance of the model. We encourage the reader to look at Chapter 4 and Appendix A of the thesis for more information, including detailed derivations of all the expressions we use.

4. Multimodal attention densities

4.1. Mixture models

Unimodal distributions such as Gaussians or $\mathcal{TP}$ have, by its own nature, a single maximum and so cannot model multimodal distributions properly. Hence, mixture models appear as a very important tool to represent arbitrarily complex probability density functions. Formally, if x is a d-dimensional vector representing a data point, a mixture model assigns it the probability

$$p(x|\Theta) = \sum_{k=1}^{K} \pi_k p(x|\theta_k), \tag{26}$$

where the parameters π_k are called *mixing coefficients* and satisfy $\pi_k \geq 0$ for $k = 1, \dots, K$ and $\sum_{k=1}^{K} \pi_k = 1$; each θ_k is the set of parameters defining the k-th component of the mixture; and $\Theta = \{\theta_1, \dots, \theta_K, \pi_1, \dots, \pi_K\}$ is the complete set of parameters needed to specify the mixture. The definition of mixture is completely general regarding the functional form of each component, meaning that $p(x|\theta_k)$ can take different forms. For instance, if $p(x|\theta_k) = \mathcal{N}(x|\mu_k, \Sigma_k)$, this is known as a Gaussian Mixture Model (GMM), whose complete set of parameters includes $\pi = \{\pi_1, ..., \pi_K\}$, $\mu = \{\mu_1, ..., \mu_K\}$ and $\Sigma = \{\Sigma_1, ..., \Sigma_K\}$.

4.2. Multimodal continuous attention

We can extend the framework presented in Sections 2 and 3 to multimodal distributions by considering mixtures of unimodal distributions,

$$p(t) = \sum_{k=1}^{K} \pi_k p_k(t), \tag{27}$$

where each $p_k = \hat{p}_\Omega[f_{\theta_k}]$ is a unimodal distribution (*e.g.*, a Gaussian or a truncated paraboloid) and $\pi \in \Delta^K$ are mixing coefficients defining the weight of each component of the mixture. For instance, in the first example, $p(t)$ becomes a mixture of Gaussians; we discuss later possible methods for obtaining π.

Using the definition of continuous attention mechanism (13) and, from the linearity of expectations, we can compute the output of the multimodal attention mechanism as

$$r = \mathbb{E}_p[\psi(t)] = \sum_{k=1}^{K} \pi_k \underbrace{\mathbb{E}_{p_k}[\psi(t)]}_{r_k} = \sum_{k=1}^{K} \pi_k r_k, \tag{28}$$

where r_k is the output of an individual (unimodal) attention mechanism. The context representation is

$$c = \mathbb{E}_p[B\psi(t)] = Br = \sum_{k=1}^{K} \pi_k \underbrace{Br_k}_{c_k}, \tag{29}$$

where each c_k is the context representation of each individual attention mechanism; that is, c is a mixture of the context representations for each component. The backpropagation step for the multimodal case is simple, since this decomposes into a linear combination of unimodal attention mechanisms, each of which has a simple/closed-form Jacobian.

Note that our construction is not the same as the standard multi-head attention scenario, where the projection matrices learned as parameters are head-specific [21]. On the contrary, we assume that B does not depend on k. From a computational point of view, this property seems appealing – we can compute a single B per example and still obtain a context vector that contains information from different "heads", through different unimodal attention mechanisms.

What if we have access to a set of attention weights? Consider that we are provided with a set of points equally spaced in the unit square and its correspondent discrete attention weights. Intuitively, the higher the attention weight, more important the contribution of that point to the network's decision. For multimodal distributions, we can think of this problem as that of fitting a mixture model to weighted data. In that context, we have to deal with 2 different issues: how to estimate the parameters defining the mixture model (§ 4.3.1) and the number of components (§ 4.3.2).

4.3. The EM algorithm for GMMs

The EM algorithm is the standard method to estimate the parameters defining a mixture model, which converges to a *maximum likelihood* estimate of the mixture parameters (see [13] for a detailed exposition). For $\alpha = 1$, the corresponding unimodal attention density is a Gaussian; we can easily adapt EM to deal with weighted data and obtain the full set of parameters of a Gaussian mixture – defining a multimodal attention density $p(t)$.

4.3.1 Weighted data

Let $X = \{x_1, ..., x_N\}$ be the observed data and $W = \{w_1, ..., w_N\}$ the weights associated with X, where $w_n \geq 0$ is the weight indicating the relevance of observation x_n. We can change the usal EM algorithm for non-weighted data and include the information provided by the weights by changing the way we re-estimate the parameters each iteration. It should go as follows:

1. Initialize the parameters μ_k, Σ_k and π_k and evaluate the initial value of a weighted log likelihood function

$$\sum_{n=1}^{N} w_n \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}, \tag{30}$$

where the log likelihood of each point is multiplied by the correspondent weight.

2. **E step.** Evaluate the responsibilities using the current parameter values
$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}. \quad (31)$$

3. **M step.** Re-estimate the parameters using the current responsibilities
$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} w_n \gamma(z_{nk}) x_n, \quad (32)$$
$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} w_n \gamma(z_{nk})(x_n - \mu_k^{\text{new}})(x_n - \mu_k^{\text{new}})^\top, \quad (33)$$
$$\pi_k^{\text{new}} = N_k, \quad (34)$$
where
$$N_k = \sum_{n=1}^{N} w_n \gamma(z_{nk}). \quad (35)$$

4. Re-evaluate the weighted log likelihood (30) using the current parameter values and check for convergence of either the parameters or the log likelihood. Return to step 2 if the convergence criterion is not satisfied.

If we consider that the weight associated with each observation is the same, *i.e.*, $w_n = \frac{1}{N}$, we recover the usual expressions for the EM algorithm.

4.3.2 Estimating the number of components

The *maximum likelihood* criterion cannot be used to estimate the number of components K in a mixture density. If $\mathcal{M}_k$ is a class composed by all Gaussian mixtures with K components, it is trivial to show that $\mathcal{M}_K \subseteq \mathcal{M}_{K+1}$ and thus the maximized likelihood is a non decreasing function of K, useless as a criterion to estimate K [5].

Several *model selection* methods were proposed to tackle the concern of estimating the number of components of a mixture [14, Chapter 6]. The likelihood function as defined in (30) is of no direct use, since it increases with k. We focus on penalized likelihood methods such as the Bayesian Information Criterion (BIC) [19] or the Minimum Description Length (MDL) [17], where the EM algorithm is used to obtain different parameter estimates for a range of values of k, $\{\hat{\Theta}_k,\ k = k_{min}, \ldots, k_{max}\}$, and the number of components is chosen according to
$$k^\star = \arg\min_k \{\mathcal{C}(\hat{\Theta}_k, k),\ k = k_{min}, \ldots, k_{max}\}, \quad (36)$$
where $\mathcal{C}(\hat{\Theta}_k, k)$ is a model selection criterion that usually has the form
$$\mathcal{C}(\hat{\Theta}_k, k) = -2\ln(X|\hat{\Theta}_k) + \mathcal{P}(k), \quad (37)$$
where $\mathcal{P}(k)$ is an increasing function penalizing higher values of k. We can write
$$\mathcal{P}(k) = \lambda\, k, \quad (38)$$
where $\lambda > 0$ is an hyperparameter obtained using *cross-validation*. The resulting model selection criterion
$$\mathcal{C}(\hat{\Theta}_k, k) = -2\ln(X|\hat{\Theta}_k) + \lambda\, k, \quad (39)$$
will be used in Section 5 to estimate the number of components in a multimodal continuous attention density.

4.3.3 Initialization

The EM algorithm requires an initial choice for the set of parameters $\Theta = \{\pi, \mu, \Sigma\}$. This becomes an issue of the utmost importance because EM is not guaranteed to converge to a global maximizer of the log likelihood function, getting stuck at a local maximizer most of the times, meaning that the final estimate depends on the initialization. A common strategy to alleviate this issue consists of considering several different initializations (*e.g.*, multiple random initializations), run EM that number of times and choose the final estimate that leads to the highest likelihood [13].

5. Applications in VQA

We now plug our 2D continuous attention mechanisms in a VQA model that uses grid features to represent the images. All our models use the same features and were trained only on the train set without data augmentation.

Dataset and architecture. We used the VQA-v2 dataset [7] and adapted the implementation of [23][1]: our architecture is the same except that we represent the images using grid features generated by a ResNet pretrained on ImageNet [9], instead of bounding-box features [2].

5.1. Experiments with 2D continuous attention

Attention model. We consider 3 different attention models: discrete attention, 2D continuous softmax attention and 2D continuous sparsemax attention. The discrete attention model attends over a 14×14 grid. For continuous attention, we normalize the image size into the unit square $[0, 1]^2$ with each coordinate t_l positioned at $(\frac{l_1}{\sqrt{L}}, \frac{l_2}{\sqrt{L}})$ for $l_1, l_2 \in [0, \sqrt{L}]$ creating a meshgrid. We fit a 2D Gaussian ($\alpha = 1$) or truncated paraboloid ($\alpha = 2$) as the attention

[1] https://github.com/MILVLG/mcan-vqa

Table 1. Overall accuracies of different models on the *test-dev* and *test-standard* splits of VQA-v2. For the continuous attention models we used $N \in \{49, 100\}$ Gaussian RBFs $\mathcal{N}(t; \tilde{\mu}, \tilde{\Sigma})$, with $\tilde{\mu}$ linearly spaced in $[0, 1]^2$ and $\tilde{\Sigma} = 0.001 \cdot \mathrm{I}$. See Table 2 for detailed results for $N = 100$.

ATTENTION	N	Test-Dev	Test-Standard
Discrete softmax	-	65.83	66.13
2D continuous softmax	100	**65.96**	**66.27**
2D continuous softmax	49	65.82	66.12
2D continuous sparsemax	100	65.79	66.10
2D continuous sparsemax	49	65.88	66.10

density. We use the mean and variance according to the discrete attention probabilities and obtain μ and Σ with moment matching. We use $N \in \{49, 100\} \ll 14^2$ Gaussian RBFs, with $\tilde{\mu}$ linearly spaced in $[0, 1]^2$ and $\tilde{\Sigma} = 0.001 \cdot \mathrm{I}$. Overall, the number of neural network parameters is the same as in discrete attention.

Results. The results in Table 1 show that the accuracies for all the attention models are similar, with a slight advantage for 2D continuous softmax with $N = 100$ basis functions. Even though we used less basis functions than image regions ($N \ll L$), 2D continuous attention performed as well as (or even better than) discrete attention. Moreover, we can see that we don't need a large number of basis functions to obtain good results given that for $N = 49$ the results are already satisfying: all the results are very similar on the *test-standard* split; on the *test-dev* split, 2D continuous sparsemax performed a bit better than the other variants.

Attention visualization. Figure 2 shows an example where the attention is too scattered in the discrete model, possibly mistaking the lamp with a TV screen; contrarily, our continuous attention models focus on the right region and answer the question correctly, with 2D continuous sparsemax enclosing all the relevant information in its supporting ellipse. By fitting a Gaussian as the attention density (continuous softmax) every region in the image is assigned with some probability mass; by fitting a truncated paraboloid (continuous sparsemax), the attention density becomes sparse, *i.e.*, only the relevant regions of the image are assigned with non-zero probability mass – we found out that this usually leads to more interpretable attention maps. Besides, other examples in the thesis suggest that discrete attention is more diffuse than its continuous counterpart. This might be good for very complex question/image pairs, although continuous attention ellipses are also capable of becoming wide, including different regions of interest.

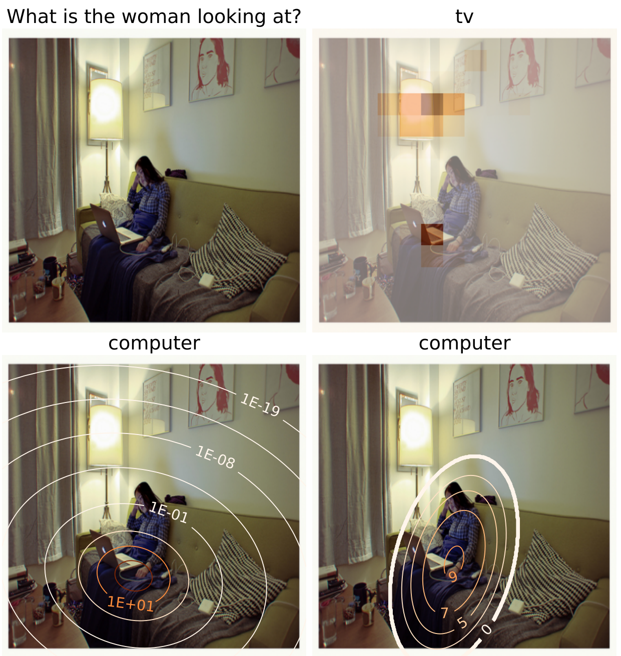

Figure 2. Examples of attention maps for VQA. Top left: original image. Top right: discrete attention. Bottom: 2D continuous softmax (left) and 2D continuous sparsemax (right, where the outer ellipse encloses all probability mass).

5.2. Experiments with multimodal continuous attention (MCA)

We consider 2 different scenarios. First, we choose a fixed number K and assume that each attention density can be modeled as a K-component multimodal distribution (we refer to this attention model as **K-MCA**, hereinafter). Second, we use a model trained with unimodal continuous attention and, at test time, consider multimodal distributions, using the model selection criterion (39) to choose the optimum number of components from a set of possible choices. We refer to the latter as **test-MCA**.

K-MCA. We consider multimodal attention densities with $K \in \{2, 4\}$ components. Instead of initializing the parameters $\Theta = \{\mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K, \pi_1, \dots, \pi_K\}$ randomly, we split the image in K regions and obtain $\{\mu_1, \dots, \mu_K\}$ and $\{\Sigma_1, \dots, \Sigma_K\}$ with moment matching according to the discrete attention weights in the corresponding region. The initial mixing coefficients $\{\pi_1, \dots, \pi_K\}$ can be obtained according to the probability mass in the corresponding region. Then, we run the EM algorithm for weighted data proposed in Subsection 4.3.1 to obtain the final estimates for Θ. Instead of evaluating the log likelihood function (30) each iteration to check for convergence, we re-estimate the parameters of the mixture model for a fixed number of iterations, which can be considered an extra hyperparameter. According to (29), we com-

Table 2. Accuracies of different models on the *test-dev* and *test-standard* splits of VQA-v2. For the continuous attention models we used 100 Gaussian RBFs $\mathcal{N}(t; \tilde{\mu}, \tilde{\Sigma})$, with $\tilde{\mu}$ linearly spaced in $[0,1]^2$ and $\tilde{\Sigma} = 0.001 \cdot \mathrm{I}$. We used 2,5,20,20,20 iterations for the MCA models.

ATTENTION	λ	Test-Dev				Test-Standard			
		Yes/No	Number	Other	Overall	Yes/No	Number	Other	Overall
Discrete	-	**83.40**	43.59	55.91	65.83	83.47	42.99	**56.33**	66.13
2D continuous softmax	-	**83.40**	**44.80**	55.88	65.96	**83.79**	44.33	56.04	**66.27**
2D continuous sparsemax	-	83.10	44.12	55.95	65.79	83.38	43.91	56.14	66.10
2-MCA	-	83.35	44.28	**56.07**	**65.97**	83.59	43.65	56.24	66.21
4-MCA	-	83.39	43.52	55.96	65.85	83.72	43.47	56.03	66.14
test-MCA	10	83.30	44.60	55.81	65.86	83.76	44.08	56.00	66.21
test-MCA	100	83.35	44.75	55.86	65.92	83.77	44.28	56.02	66.25
test-MCA	500	83.39	44.75	55.87	65.95	**83.79**	**44.36**	56.04	**66.27**

pute K individual 2D continuous softmax attention mechanisms in order to obtain obtain the context representations $\{c_1, ..., c_K\}$. The final context is a mixture of the context representations for each component.

test-MCA. We use a model trained with unimodal continuous attention (2D continuous softmax with $N = 100$, from Table 1) and, at test time, consider multimodal distributions. We consider models with a number of components in the range $K \in \{1, \ldots, 5\}$. For $K = 1$ we use the same setup as in the previous section. For $K > 1$ we follow the procedure described in Section 4.3.3 and consider 3 random initializations for each K, $i \in \{1, 2, 3\}$. We use the EM algorithm for weighted data to obtain different parameter estimates $\hat{\Theta}_{Ki}$. We consider $1 + 3 \times 4 = 13$ estimates $\hat{\Theta} \in \{\hat{\Theta}_1, \hat{\Theta}_{21}, \hat{\Theta}_{22}, \hat{\Theta}_{23}, \ldots, \hat{\Theta}_{51}, \hat{\Theta}_{52}, \hat{\Theta}_{53}\}$, and choose the model that minimizes the model selection criterion (39). If the optimum number of components $K^\star > 1$, the final context is computed as $c = \sum_{k=1}^{K^\star} \pi_k \, c_k$, where each c_k is obtained through an individual 2D continuous softmax attention mechanism.

Results. The results in Table 2 show the accuracies for all attention models. Again, the results are very similar, suggesting that there is no clear gain in terms of accuracy when using MCA models to answer the questions in the VQA-v2 dataset. Note, however, that these models also use less basis functions than image regions ($N \ll L = 14 \times 14$). Although we can now identify multiple regions of interest in images (attention focuses), in terms of accuracy for VQA that seems not to be a big advantage. When using test-MCA, an optimum value of components $K^\star$ is chosen to answer each question; from Table 2 it is possible to see that the overall accuracy also increases with these, suggesting that a small K is better for the model's accuracy.

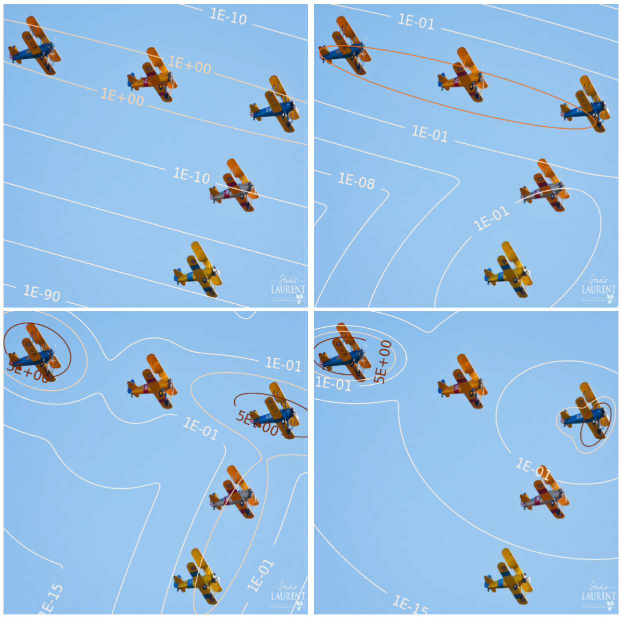

Figure 3. Attention maps generated when answering the question: **How many planes have blue as their main body colour?** Top left: 2D continuous softmax ($N = 100$). Top right: MCA-2. Bottom left: MCA-4. Bottom right: MCA-test ($\lambda = 500$).

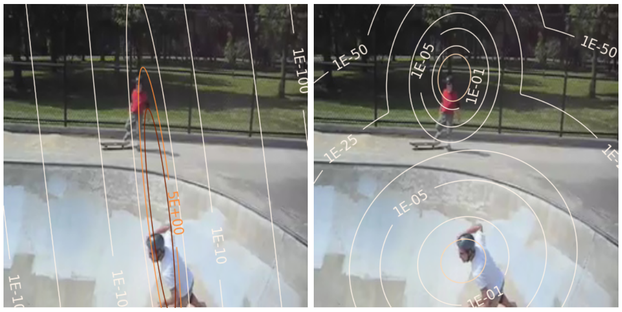

Figure 4. Attention maps generated when answering the question: **How many people are wearing helmets?** Left: 2D continuous softmax ($N = 100$). Right: MCA-test ($\lambda = 500$).

Attention visualization. Figure 3 shows an example in which the models are asked how many planes have blue as their main body colour. When using 2D continuous softmax, the attention density is a Gaussian and the region of interest is correctly identified. However, due to its unimodal nature, it attributes a lot of probability mass to the yellow plane's position – a region in between the 2 blue planes (the leftmost and the rightmost ones). Although a similar situation appears to happen when using MCA-2, the difference in the values of the contours shows that the attention density is more spread across the 5 planes instead of being concentrated in the yellow one. Both MCA-4 and MCA-test are not only capable of identifying 2 blue planes but also to "isolate" its positions, suggesting that these models generate more flexible attention distributions, while enjoying the advantages of modeling attention as a continuous function.

Although the ellipses generated by unimodal continuous attention models are able to become as wide as necessary, that can result in a less interpretable attention map. Multimodal continuous attention solves this problem (Figure 4).

6. Conclusions

We presented continuous-domain alternatives do discrete attention models that focus on the continuity and sparsity of attention distributions. We constructed 2D continuous α-entmax attention mechanisms; for $\alpha = 1$ the attention density is a Gaussian and, for $\alpha = 2$, it becomes a truncated paraboloid distribution with sparse support. We derived their Jacobians, allowing for efficient forward and backward propagation (Section 3). As a natural follow-up, we proposed multimodal continuous attention by using mixtures of unimodal attention densities (Section 4). These attention mechanisms enjoy some of the properties of their unimodal counterparts, while they are able to generate more flexible attention maps and thus can model more complex attention distributions. Finally, we performed experiments on VQA with promising results (Section 5). Continuous attention allowed for obtaining smooth and interpretable attention maps that are more difficult to generate with discrete attention models.

There are many avenues for future research. First, we did not explore other vision tasks that could benefit more from our MCA models than VQA does. For instance, the state-of-the-art on visual counting tasks is an attention-based model that uses grid features and therefore could work with MCA [15]. Another possible way of future research lies in considering mixtures of sparse family distributions (*e.g.*, truncated paraboloids). Moreover, if the mixing coefficients in (27) are obtained from a discrete transformation that produces sparse results (*e.g.*, sparsemax), the number of components in a mixture may vary (K becomes a maximum number of components), possibly making the training of MCA models with varying number of modes easier.

References

[1] Shun-ichi Amari and Atsumi Ohara. Geometry of q-exponential family of probability distributions. *Entropy*, 13(6):1170–1185, 2011. 2

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018. 7

[3] Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning Classifiers with Fenchel-Young Losses: Generalized Entropies, Margins, and Algorithms. 89, 2018. 2

[4] Steven L. Brunton, J. Nathan Kutz, Krithika Manohar, Aleksandr Y. Aravkin, Kristi Morgansen, Jennifer Klemisch, Nicholas Goebel, James Buttrick, Jeffrey Poskin, Agnes Blom-Schieber, Thomas Hogan, and Darren McDonald. Data-driven aerospace engineering: Reframing the industry with machine learning, 2020. 1

[5] Mário A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 24:381–396, 2000. 7

[6] A. Galassi, M. Lippi, and P. Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–18, 2020. 1

[7] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[8] Paul R Halmos. *Measure Theory*, volume 18. Springer, 2013. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778, 2016. 7

[10] Thorsten Hoeser and Claudia Kuenzer. Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. *Remote Sensing*, 12(10):1667, May 2020. 1

[11] André F. T. Martins, António Farinhas, Marcos Treviso, Vlad Niculae, Mário A. T. Figueiredo, and Pedro M. Q. Aguiar. Sparse and continuous attention mechanisms. In *Proc. NeurIPS*, 2020. 2, 3, 4

[12] Pedro Henrique Martins, Vlad Niculae, Zita Marinho, and André Martins. Sparse and structured visual attention, 2020. 1

[13] Geoffrey J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley New York, 1997. 6, 7

[14] G. J. McLachlan and D. Peel. *Finite mixture models*. Wiley Series in Probability and Statistics, New York, 2000. 7

[15] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. Revisiting modulated convolutions for visual counting and beyond. *arXiv preprint arXiv:2004.11883*, 2020. 10

[16] Ronald A Rensink. The Dynamic Representation of Scenes. *Visual Cognition*, 7(1-3):17–42, 2000. 1

[17] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific Publishing Co., Inc., USA, 1989. 7

[18] P. Santana, L. Correia, M. Guedeszy, and J. Baratay. Visual attention and swarm cognition towards fast and robust off-road robots. In *2011 IEEE International Symposium on Industrial Electronics*, pages 2255–2260, 2011. 1

[19] Gideon Schwarz. Estimating the Dimension of a Model. *Annals Statist.*, 6:461–464, 1978. 7

[20] Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52:479–487, 07 1988. 2

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz 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, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. 6

[22] 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)*, pages 11–20, Hong Kong, China, 2019. Association for Computational Linguistics. 1

[23] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:6274–6283, 2019. 7