Appendix 1: Algorithms and FLOPS calculation

In this section, we announced our way to count the number of FLOPS. For brevity, we denoted M (N) as the number of output (input) capsules gathered from all locations over channels at the output (input) layer. In this way, a *votings* tensor has a size of $Q = M \times N \times D$, where D is the size of a capsule. We present routing algorithms and count their calculations as followed.

Algorithm 1: Attention-based routing (consider only 1 iteration)

Input: Votings v : size $M \times N \times D$ i^{th} $g$lobal capsules: g^i: size $M \times D$ Output $(i+1)^{th}$ global capsules: g^i: size $M \times D$ Activation probability for global capsules: $prob\ size\ M$		
$\forall m, n$	$a_{m,n} = \sum_k^D v_{m,n,k} \times g^i_{m,k}$	(1)
$\forall m, n$	$r_{m,n} = \frac{\exp(a_{m,n})}{\sum_k \exp(a_{k,n})}$	(2)
$\forall m$	$g^{i+1}_{m,:} = \sum_n^N r_{m,n} v_{m,n,:}$	(3)
$\forall m$	$g^{i+1}_{m,:} = squash(g^{i+1}_{m,:})$	(4)
$\forall m$	$prob_m = \lVert g_m \rVert_2$	(5)

The number of FLOPS can be counted by each line

(1): For each m, n, it needs $2 \times D$ calculations to compute the cosine similarity between a global capsule g^i_m and a voting capsule $v_{m,n}$, so there are $M \times N \times 2D$ calculations.

(2): First, the summation $\sum_k \exp(a_{k,n})$ requires $2M$ calculations if we assume that exp () is 1 calculation. Then, the softmax requires $M \times N$ calculations for the divisions. Therefore, there are $2M + M \times N$ calculations in total.

(3): To derive a new global capsule g^{i+1}_m, we sum up all N capsules weighted by the coefficients $r_{m,n}$, and the multiplication between N vectors and a scalar take $N \times D$ calculations. The summation among N capsules with D-dim takes $N \times D$ calculations, so we have $M \times N \times 2D$ calculations in total.

(4): $squash(g_m) = \frac{\lVert g_m \rVert_2^2}{1+\lVert g_m \rVert_2^2} \frac{g_m}{\lVert g_m \rVert_2}$ working on $D - dim$ vector requires $3D$ calculations, $2D$ for computing the norm and D for scalar multiplication, so this step consumes $3 \times M \times D$ calculations

(5): the activation probability of M capsules requires $2 \times M \times D$ calculations for computing M norms.

As a result, one iteration in attention-based routing costs $4 \times M \times N \times D + M \times N + 5 \times M \times D + 2M$ calculations approximately. However, the most increasing term is $4 \times M \times N \times D = 4Q$.

Algorithm 2: Fuzzy-based routing (consider only 1 iteration)

Input: Votings v : size $M \times N \times D$ i^{th} $g$lobal capsules: g^i: size $M \times D$ Output $(i+1)^{th}$ global capsules: g^i: size $M \times D$ Activation probability for global capsules: $prob\ size\ M$		
$\forall m, n$	$d_{m,n} = \left(\lVert v_{m,n,:} - g^i_{m,:} \rVert_2\right)^{\frac{2}{m_f - 1}}$	(1)
$\forall m, n$	$f_{m,n} = \frac{1}{\sum_k^M \frac{d_{m,n}}{d_{k,n}}}$	(2)
$\forall m, n$	$f_{m,n} = \left(f_{m,n}\right)^{m_f}$	(3)
$\forall m, n$	$r_{m,n} = \frac{f_{m,n}}{\sum_k f_{m,k}}$	(4)
$\forall m$	$g^{i+1}_{m,:} = \sum_n^N r_{m,n} v_{m,n,:}$	(5)
$\forall m$	$\sigma_m^2 = \sum_n^N r_{m,n} \lVert v_{m,n,:} - g^{i+1}_{m,:} \rVert_2^2$	(6)
$\forall m$	$prob_m = sigmoid(\lambda(\beta_m - 0.5 * ln(\sigma_m^2))$	(7)

(1): For each m, n, it needs $2 \times D$ calculations to compute the Euclidean distance between a global capsule g^i_m and a voting capsule $v_{m,n}$, so there are $M \times N \times 2D$ calculations. Also, ones need another $M \times N \times D$ for the point-wise power, resulting in $3M \times N \times D$ calculations.

(2): First, the summation $\sum_k^M \frac{1}{d_{k,n}}$ requires $2M$ calculations, so the term $\frac{1}{\sum_k\left(\frac{d_{m,n}}{d_{k,n}}\right)}$ consumes $2M+2$ calculations. Totally, ones need $(2M+2)\times N$ to derive $M\times N$ $f_{m,n}$.
(3): The point-wise power requires $M\times N$ calculations
(4): First, the summation $\sum_k^N f_{m,k}$ requires N calculations, and the quotient $\frac{f_{m,n}}{\sum_k f_{m,k}}$ needs $M\times N$ calculations, so there are $N+M\times N$ calculations this step
(5): As the algorithm above, it uses $M\times N\times 2D$ calculations to derive new global capsules
(6): This step is the most consumption in the algorithm when it uses $M\times N\times 2D$ calculations to compute the distances and $2M\times N$ more calculations to perform the summation; thus there are $M\times N\times 2D+2M\times N$ calculations.
(7): there are 5 operations in total applied on m scalars, so it costs $5M$ calculations.
As a result, one iteration in fuzzy-based routing costs $7M\times N\times D+6M\times N+2N+5M$ calculations approximately. However, the most increasing term is $7\times M\times N\times D=7Q$.

Algorithm 3: EM-based routing (consider only 1 iteration)

Input: Votings v : size $M\times N\times D$ Activation probability at the previous iteration a: size M Routing coefficients at the previous iteration r: size $M\times N$ Output Capsules at a higher level μ: size $M\times D$ Activation probability at a higher level a: size M	
$\boldsymbol{M-step}$	
$\forall m,n \quad r_{m,n}=r_{m,n}\times a_n$	(1)
$\forall m,n \quad r_{m,n}=\frac{r_{m,n}}{\sum_k^N r_{m,k}}$	(2)
$\forall m \quad \mu_{m,:}=\sum_n^N r_{m,n}v_{m,n,:}$	(3)
$\forall m \quad \sigma_{m,d}^2=\frac{1}{N}\sum_n^N r_{m,n}\times\left(v_{m,n,d}-\mu_{m,d}\right)^2$	(4)
$\forall m \quad cost_{m,:}=\left(\beta_u+log\left(\sigma_{m,:}\right)\right)\sum_n^N r_{m,n}$	(5)
$\forall m \quad a_m=sigmoid\left(\lambda\left(\beta_a-\sum_d^D cost_{m,d}\right)\right)$	(6)
$\boldsymbol{E-step}$	
$\forall m,n \quad p_{m,n}=\frac{1}{\sqrt{\prod_d^D 2\pi\sigma_{m,d}^2}}\exp\left(-\frac{\sum_d^D\left(v_{m,n,d}-\mu_{m,d}\right)^2}{2\left(\sigma_{m,d}\right)^2}\right)$	(7)
$\forall n \quad r_{m,n}=\frac{a_m p_{m,n}}{\sum_k^M a_k p_{k,n}}$	(8)

(1): This step obviously requires $M\times N$ calculations since $r_{m,n}$ and a_n are both scalars
(2): The summation $\sum_k^N r_{m,k}$ takes N calculations, and the quotient $\frac{r_{m,n}}{\sum_k^N r_{m,k}}$ takes 1 calculation, resulting in $M\times(N+1)$ calculations in total.
(3): As the algorithms above, this step requires $M\times N\times 2D$ calculations
(4): For each m,d, ones need $3N$ calculations to compute the summation, so there are $M\times D\times 3N$ calculations.
(5): First, the summation $\sum_n^N r_{m,n}$ costs N calculations, then $3D$ calculations are needed to perform operations on the $D-dim$ vector σ_m. Therefore, this step uses $M\times(N+3D)$ operations.
(6): Likewise, the $\sum_d^D cost_{m,d}$ requires D calculations, and ones need 3 more calculations to compute a_m. Since we need to compute m activation probabilities, this step requires $M\times(N+3)$ calculations.
(7): Firstly, the term $\frac{\sum_d^D\left(v_{m,n,d}-\mu_{m,d}\right)^2}{2\left(\sigma_{m,d}\right)^2}$ uses $3D+1$ calculations. Secondly, the terms $\frac{1}{\sqrt{\prod_d^D 2\pi\sigma_{m,d}^2}}$ requires $2D+2$ calculations. Totally, this step takes $M\times N\times(5D+3)$ calculations.
(8): The summation $\sum_k^M a_k p_{k,n}$ uses $2M$ calculations and the quotient $\frac{a_m p_{m,n}}{\sum_k^M a_k p_{k,n}}$ takes 2 more calculations, resulting in $N\times(2M+2)$.
As a result, one iteration in EM-based routing costs $10M\times N\times D+8M\times N+3M\times D+3M+2N$ calculations approximately. However, the most increasing term is $10\times M\times N\times D=10Q$.