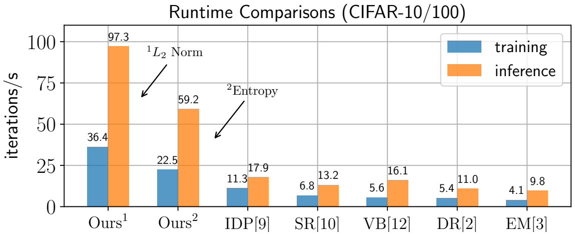
We would like to thank the reviewers for their comments and positive outlook on the paper. We are encouraged that all reviewers find our proposed method, claims and empirical methodology to be correct (R1, R2, R3, R4). R1 found our paper to be well-motivated in tackling a major efficiency drawback of capsule networks, informative and clear, whereas R2 found it to provide an interesting new perspective. R3 acknowledged our theoretical understanding of variational inference and capsule networks. R4 stated that we introduce a novel and significant reformulation of capsule networks which is more coherent and theoretically sound than previous work, achieving exciting state of the art results and enhancing the viewpoint generalizability of capsules by several degrees. We hope our response clarifies all concerns.

[R1] **1. More runtime comparisons.** As requested, we conducted more extensive runtime comparisons with the 5 most prominent and publicly reproducible related works. For fairness, we use the same $\{128, 16, 16, 16, 10\}$ architecture and replace the routing mechanism. We use Pytorch, 2 Titan Xp GPUs and a batch size of 64. As depicted on the right, our method offers considerable speedups over previous works, whilst enhancing performance on pose-aware tasks (see paper).



[R2] **2. Feature occlusion experiments (MultiMNIST).** We thank the reviewer for the valueable suggestion. We empirically demonstrate that, unlike previous methods, modelling uncertainty over part-object connections yields significantly more resilient capsnets under feature occlusion (which is a source of uncertainty). We replicated the experiment setup in [2], and trained our shallow $\{128, 16, 16, 16, 10\}$ model on MultiMNIST by generating occluded digit pairs on the fly. We trained for 300 epochs on $\simeq$ 18M training examples. Table 1 reports both test accuracy and exact match ratio (MR). As shown, our method outperforms previous work by a large margin using fewer parameters.

[R1] [R2] **3. Evaluation on CIFAR-10/100.** Although our work is focused on enhancing capsule network properties in pose-aware tasks, we evaluated our method on CIFAR-10/100 as suggested. We borrow the setup and baselines from [9] and compare with the most prominent previous works which are publicly reproducible (see Table 1). For fair comparisons, we used the shallow model $\{128, 16, 32, 32, 10\}$ described in Section 5, and baseline CNNs of equal depth. By replacing the single `Conv` layer stem with a ResNet-20 backbone we achieve 93.1% (1.92M) on CIFAR-10, and 72.4% (2.01M) on CIFAR-100. With a *thinner* $\{32, 8, 8, 8, 10\}$ model we can achieve 90.5% on CIFAR-10 using only 0.1M parameters.

[R2] [R4] **4. Further details on inference networks $q_\phi(\cdot)$.** This will be rectified in the final version. For clarity, each $q_\phi(\cdot)$ is simply a single layer perceptron with sofplus non-linearities that takes the activations of part capsules $\mathbf{a}_i$ and outputs the parameters $\boldsymbol{\pi}^{(i)}$ of the approximate Dirichlet posterior on the part-object connections. For R2, the number of parameters is kept small both thanks to our choice of Dirichlet prior as discussed in Section 3.3, and our use of fewer capsules than previous work whilst achieving better performance, i.e. at most $\{128, 16, 32, 32, 10\}$.

Table 1: CIFAR10/100 & MultiMNIST.

Method	Test Acc. (# params) CIFAR-10	 CIFAR-100
Baseline CNN	82.2 (2.4M)	51.4 (2.4M)
Baseline CNN [9]	87.1 (18.9M)	62.3 (19M)
Dynamic [2]	84.1 (7.9M)	56.9 (32M)
EM-Routing [3]	82.2 (0.5M)	37.7 (0.5M)
IDP-Attention [9]	85.1 (0.6M)	57.3 (1.5M)
VB-Routing [12]	86.2 (0.4M)	58.4 (0.5M)
Ours	88.3 (0.57M)	63.4 (0.65M)

Method	MultiMNIST (#params) Test Acc. (%)	 Test MR (%)
Baselines [2][9]	91.9 (24.6M)	84.8 (19.6M)
Dynamic [2]	94.8 (8.2M)	-
IDP-Attention [9]	-	91.17 (42M)
Aff-Caps [42]	95.49 (8.2M)	-
Ours	**97.96** (0.23M)	**96.4** (0.23M)

[R3] **5. Explain connection & difference to VB-Routing.** Our method is related to VB-Routing but fundamentally different. In VB-Routing the authors perform closed-form variational-EM updates, which are still iterative and *local*, just like EM-Routing. Therefore, VB-Routing still suffers from the efficiency drawbacks mentioned in Section 1.1. In our case, we perform *global* variational inference of part-object connections in a fully probabilistic capsule network, that is locally non-iterative and is trained end-to-end under a single globally coherent minimum description length objective (Eq. 7). Lastly, the VB-Routing framework does not provide predictive uncertainty estimates, whereas our work is the first to do so in the capsule domain to the best of our knowledge.

[R3] **6. Provide main insights of the method for the field.** As aptly summarised by R4, we provide a more coherent and theoretically sound capsule routing framework by directly optimising an end-to-end MDL objective. Our approach offers a significant speedup over previous methods, provides uncertainty estimates, and is the first non-iterative non-local routing method to enhance capsule network properties such as viewpoint generalisation by several degrees.

[R4] [R3] **7. Improvements to paper clarity & related work.** We thank the reviewers for the constructive feedback, and we agree that the exposition can be difficult to follow. We will make Figures 1 & 4 (R1) more legible, and rearrange the equations. Given the availability of an extra page in the camera-ready version, we will include an algorithm cell and an additional paragraph on related work, incorporating the reference to (Gu, J. and Tresp, V., 2020) mentioned by R3 and prior work on variational inference. Key details from EM-Routing will be added to aid in general understanding (R4). Lastly, the tables will be made clearer, better indicating the differences between methods. For R3, 'Our EM-Routing' simply denotes our implementation of EM-Routing [3], and '$\{32, 8, 8, 8, 5\}$' denotes a variant of our architecture.