We thank the reviewers (**R4**, **R5**, **R6**) for the useful feedback. Below we address their questions.

(R4, R5, R6) Is StarCaps a capsule network? The table to the right shows the test accuracies of two types of experiments on Small-NORB (novel, familiar viewpoints). ***Type1:*** 3 runs of models `EMCaps` $\{64, 8, 16, 16, 5\}$, StarCaps $\{32, 8, 8, 8, 5\}$, fully trained on familiar views and tested on both novel and familiar views. ***Type2:*** `EMCaps` $\{32, 32, 32, 32, 5\}$, StarCaps $\{32, 32, 16, 16, 5\}$, trained on familiar views and early stopped when test accuracy reached 96.3% (as the CNN model in Table 2 of [`EMCaps`, Hinton et al]). In *Type1*, we notice that StarCaps achieves comparable results (small difference in accuracy) to `EMCaps` both on familiar and novel viewpoints. In *Type2*, on the novel viewpoints, StarCaps performs dramatically better than CNN model (+6.3%) and its accuracy is only slightly lower than `EMCaps` (-0.2%).

	Type1 (low capacity models)		***Type2*** (high capacity models)		
Model	`EMCaps`	StarCaps	CNN	`EMCaps`	StarCaps
#params	68K	73K	4.2M	316K	318K
Familiar	95.66±0.03	95.72±0.02	96.3	96.3	96.3
Novel	86.12±0.05	86.07±0.03	80.0	86.5	86.3

affNIST results: We trained StarCaps $\{32, 8, 16, 16, 10\}$ on MNIST following the data augmentation mentioned by the authors on OpenReview/ICLR18. The test accuracy of StarCaps on affNIST is 93.03% vs. 93.1% for `EMCaps` $\{32, 32, 32, 32, 10\}$. In conclusion, the results show that StarCaps is capable of detecting novel viewpoints similarly to `EMCaps`, retaining capsules properties.

(R4) Performance of StarCaps vs. CNNs Although the main purpose of StarCaps is to alleviate the computational complexity of baseline capsule networks while being able to detect viewpoint variations, StarCaps models achieve accuracies nearly on par with those modern CNN models. On CIFAR10, StarCaps: 91.23%, #params=80K vs. ResNet20: 91.25%, #params=270K vs. ResNet110: 93.57%, #params=1.7M. On CIFAR100, StarCaps: 67.66% vs. ResNet38: 68.54% vs. ResNet110: 71.21%. It is possible that scaling up StarCaps models to match #params in ResNet, would lead to better performance; however this requires further extensive study.

Ablation studies **(a) *ST-Router***: Removing ST-Router leads to lower performance. On MNIST, StarCaps model $\{32, 8, 16, 16, 10\}$ achieves 99.41 w/o ST-Router & 99.59 with ST-Router, while $\{32, 4, 64, 4, 10\}$ achieves 98.37 w/o ST-Router & 99.48 with ST-Router. **(b) *Single parent assumption***: The single parent assumption enforced in `EMCaps`/`DynamicCaps`, while it may allow better encoding of entity representation, it imposes a limitation as it ignores actual/natural use cases for object recognition. We tested StarCaps models designed to force the single parent assumption, the results were comparable to the proposed StarCaps models on MNIST; however on {CIFAR10; CIFAR100} the results were inferior due to varied clutter in backgrounds, $\{89.91; 62.33\}$ vs. StarCaps $=\{91.23; 67.66\}$ & `EMCaps` $=\{88.10; n/a\}$. **(c) *Effect of sharing weights & role of attention***: Experiments with two settings. First, StarCaps with separate weights with Attentions. We didn't notice improvement on MNIST; on CIFAR10 $\{32, 8, 8, 8, 10\}$ achieved 91.31 vs. 91.23. However, the train/test time were significantly higher due to extra matrix multiplications as in `EMCaps`; we couldn't train models on CIFAR100. Second, StarCaps with separate weights w/o Attentions. Experiments on MNIST/CIFAR10 showed very poor performance. In conclusion, the proposed setting of StarCaps provides best results in general, in terms of accuracy and train/test time while preserving capsule properties.

(R5) Visualization of instantiations params We will include visualizations in the supplementary material.

(R5) Vanishing pose In `EMCaps` the initializations of vector parameters (β_a,β_u) control the initial sparsity. In practice, according to our experiments, even with careful initialization of parameters, some `EMCaps` models suffer from unstable performance due to numerical issues with gradients (vanishing gradients). This was confirmed by the authors of `EMCaps` (see OpenReview/ICLR18 comments). We noticed unstable performance in `EMCaps` when a capsule layer has very large #capsules compared to lower/higher capsules, and when multiple adjacent layers have very large #capsules. The routing in StarCaps automatically prunes the unneeded capsules without being sensitive to #capsules/initializations. We will update Table 1, fix the argument in lines (264-280), and add clarifications about the instability issues.

(R6) SmallNORB overall results StarCaps: **S1**=$\{32, 8, 8, 8, 5\}$, $73K$, 98.0%; **S2**=$\{32, 32, 16, 16, 5\}$, $318K$, 98.2%; vs. `EMCaps`: **E1**=$\{64, 8, 16, 16, 5\}$, $68K$, 97.8%, **E2**=$\{32, 32, 32, 32, 5\}$, $316K$, 98.2%

(R6) Related work, capsules introduction, pseudo code We will add pseudo code in the supplementary material, references to (Zhang et al.) and (Yang et al.), and a more detailed introduction to capsule networks.

(R6) Global avg pooling (GAP) In StarCaps, the role of GAP is not routing. We use GAP in Decision-Learner internally in ST-Router ($\mathcal{R}_{ij}$) to rapidly capture confidence maps from an attention matrix A_{ij}. The role of $\mathcal{R}_{ij}$ is to estimate binary connectivity decision signal between two capsules; each $\mathcal{R}_{ij}$ determines the connectivity between a single lower-level capsule and a single higher-level capsule (one-to-one), and not routing between lower and higher capsules (compared to dynamic routing in capsules, or static routing between neurons using max-pooling in CNNs). The name "ST-Router" may raise confusion with "routing" in `EMCaps`/`DynamicCaps`. Each ST-Router acts as a *gating* mechanism between two capsules, whereas the whole set of ST-Routers acts as a *routing* mechanism between all capsules. We will change the name to "ST-Gate". We also use GAP after the final output layer (`ClassCaps`) to calculate the final activation probabilities from the sigmoid of pose matrix, and not for routing between capsules.

(R6) Hard attention StarCaps uses both soft and hard attention modules. The soft attention (Attn-Estimator) estimates soft relevance signal for each higher capsule which is used for scaling the pre-vote. To sparsify the network we use a hard attention module (ST-Router). Each route can be seen as a double-attention (soft & hard) mechanism.