We thank the reviewers for their useful comments. We first clarify the minor confusion raised by **Reviewer 4** about the focus of our approach (discriminative v/s generative). We then address all the individual reviewer recommendations.

Essence of our work: The purpose of our algorithm is to produce undirected graphical models *to perform inference*[1], by *conditioning on any subset* of our random variables. We do *not* want to bake in any information about specific future test inference tasks during training. It is true that when test inference tasks are known in advance, a model trained on a mixture of those may outperform us[2] (**Reviewer 4**) or our model may be incrementally better (**Reviewer 3**). But how well do we perform, compared to a model trained on the mixture, when we are both facing *a completely new task*? Experiment II, our main experiment, now bolstered as described below, shows our superior generalization capabilities to unseen inference tasks. Experiment III touches upon the generative capabilities of our model, such as the ability to produce samples in one shot, only to show the added perks of choosing our method for *inference* in the first place.

Expanding experiment II (reviewer baselines, larger data sets): All of [1],[2],[3] from **Reviewer 2** have now been absorbed into related work. They allow conditioning on arbitrary subsets of variables, like us. However, being purely neural, they require masks defined over the random variables during training, to match query patterns expected in test inference tasks, but we are *completely agnostic to inference* during training. These models fit perfectly into our experiment II setting. In table (a) below, we now use [1] under the `MIX` and `MIX-1` scenarios [3], under model name VAEAC. As expected, `MIX` accuracies are high as the tasks were seen before, but accuracies of `MIX-1` fall drastically, showing the comparative strength of AGM, which generalizes better to unseen tasks. The same is seen across data sets. [1] was shown to be better than [2] in their paper and code for [3] is unavailable. GibbsNet (*with CONV layers*, as requested by **Reviewer 2**) is also added to experiment II as baseline. Although inference-agnostic as us during training, GibbsNet learns a *latent* space and is not resistant to corruption of pixels (c=0.5 task in table (a) below). The VAEAC and GibbsNet baselines compare data-generating approaches to our potential-generating approach (**Reviewer 2**).

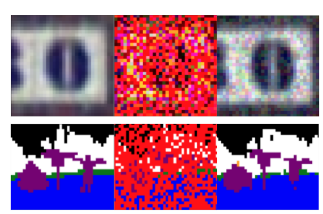
As shown in figure (b) below, experiment II now also includes the larger SVHN (**Reviewer 2**) and Stanford Background Semantic Segmentation (**Reviewer 4**) data sets (+MNIST and Caltech-101). Metrics for each of these data sets are reported in tables like (a). All our experiments show the trends seen in the original paper, but we did enough repeats to *include error bars* (**Reviewer 2**, **Reviewer 4**) with maximum width of ± 0.5.

Other related work: Models [1],[2],[3] mentioned by **Reviewer 4**, now added to our related work section, involve graphical models like us, combined with neural modules, for inference. However, they assume a *fixed set of input and output variables*, solving problems such as image tagging [1],[3] and classification [2], by learning: potentials in [1], and energy functions in [2] and [3]. These methods do *not* solve our problem formulated in our paper introduction, where we do *not* separate purely-input from purely-output variables, and we *permute* the identity of input and output variable indices across data points. As an analogy, *one* of our models should be able to solve image tagging as in [1] or [3], the inverse of that problem, as well as any inpainting pattern on the images. **Reviewer 3** rightly pointed out that the idea of one model producing parameters for another, has its roots in meta-learning. We have consolidated the related work section with: *Meta Networks [Munkhdalai, 2017]* and *Learning feed-forward one-shot learners [Bertinetto 2016]*.

Additional analysis on method: We add time and memory complexity of our method as requested by **Reviewer 2**, relating the complexity of fully-parallelized belief propagation [Bixler, 2018] to edge set cardinalities induced by data, and to the ensemble size used at test time. As requested by **Reviewer 4**, for *every* data set used in experiment I, we now plot how accuracy, and variance of predictions changes with the number of samples (size of ensemble), in the appendix.

MNIST						
		Tested on				
Model	***Trained on***	**f=0.5**	**w=7**	**c=0.5**	**q=1**	**mean**
EGM	**MIX**	93.6 ± 0.2	66.3 ± 0.2	82.6 ± 0.2	86.9 ± 0.3	82.4
	MIX-1	87.4 ± 0.1	64.1 ± 0.3	68.2 ± 0.1	84.2 ± 0.1	76.0
VAEAC	**MIX**	94.2 ± 0.4	72.4 ± 0.4	79.8 ± 0.4	87.9 ± 0.3	83.6
	MIX-1	85.5 ± 0.4	61.2 ± 0.5	65.1 ± 0.4	81.3 ± 0.1	73.3
GibbsNet	-	88.6 ± 0.1	70.5 ± 0.2	68.0 ± 0.1	87.1 ± 0.1	78.6
AGM (Ours)	-	95.5 ± 0.1	72.3 ± 0.1	79.2 ± 0.2	87.4 ± 0.1	83.6

(a) Updated table for experiment II, with baseline models: VAEAC and GibbsNet (with CONV). See footnote 3 for MIX and MIX-1 definitions.



(b) SVHN (top), Stanford Background semantic segmentation (bottom). Per row: Target (image 1), 70% query pixels (red) (image 2), output of AGM (image 3).

[1] We agree with **Reviewer 4** for a title change to 'Training Ensembles of Discrete Undirected Graphical Models Adversarially, for Generalizable Inference', to avoid insinuating that we are learning inference algorithms.

[2] Note to **Reviewer 4**: indeed, our training procedure uses 'unconditioned' samples, but at test time, when answering one query, every graphical model in the ensemble *is conditioned on the same observed data*, as shown in figure 1(b) of the original paper.

[3] In table (a) above, `MIX` is a model trained on the whole mixture of tasks shown horizontally, while `MIX-1` is trained on all tasks but the one it is being tested on, to evaluate generalization to unseen tasks. Task definitions are given in the original paper.