We thank the reviewers for their time and constructive reviews. We find it encouraging that the ideas we presented were well-received. The comments from the reviewers were very helpful, and we are eager to use the feedback to further clarify the paper and add additional results. [Note: mixup(k) / Bern(k) = mixup / Bernoulli mixing k examples at a time]

R1/R3: **Additional experiments**. We ran experiments on CIFAR10 with $d_h = 256$ and $d_h = 1024$. For CIFAR10 (256), our best result mixup(3) was 0.551 ± 0.006 compared to mixup(2) (0.547 ± 0.007) and our baseline (0.537 ± 0.004), the corresponding quoted result from ACAI is (0.5277 ± 0.0045). For $d_h = 1024$, we get 0.610 ± 0.009 for mixup(2) vs baseline's 0.596 ± 0.001 (due to time constraints of this rebuttal, we were unable to let the experiment fully converge – the quoted ACAI number for this is 0.6399 ± 0.0047). We also ran experiments for $k > 3$ on SVHN for $d_h = 256$ and achieved even better results in this regime, e.g. 0.742 ± 0.021 for mixup(4) vs 0.653 ± 0.014 for mixup(2).

R3: **ACAI implementation fix**. We added in the missing loss term. In short, ACAI outperforms on MNIST/KMNIST, but on SVHN (both d_h = 32 and 256) and CIFAR10 it does not perform well. We will add these new numbers in.

R2: **Disentanglement metric**. We evaluated the disentanglement scores of our methods on the DSprite (see Beta-VAE paper) dataset. We found that Bernoulli(3) significantly outperformed (0.558 ± 0.01) the AE baseline (0.451 ± 0.027) and mixup(3) (0.511 ± 0.049) but not compared to a finely-tuned β-VAE (0.652 ± 0.017). This gives a stronger justification for Bernoulli mixup and its utility in the context of disentanglement.

R1: **"Why is the quality of features measured during training ?"** We followed the ACAI paper, which also did this (i.e see their section 4). Also, as per your suggestion, we will rename ARAE to minimise confusion.

R1: **"Why is one method not consistently better than the other ?"** We performed an analysis comparing the Lipschitz upper bound (see the 'spectral norm' paper from Miyato et al) of our autoencoder for different values of k, and it appears to increase as k gets larger. This may have implications on our results, and will be explored further.

R2: **"The authors claim that a "fundamental difference" between mixup and VAEs is that they impose no constraint, at least not in the probabilistic sense." I disagree with this statement."** What we meant was that, unlike in the VAE setting, we are not explicitly defining a prior function $p(\mathbf{z})$ and enforcing the latent codes to be close to it (i.e. in a KL divergence sense). We agree in the sense that there is definitely a 'prior' induced with mixup (e.g. the independence assumptions with Bernoulli), but it is more implicit than what is done in VAEs. We will re-word this.

R2: **"...it is not clear from the paper whether I'd want to use any form of adversarial mixing to regularize my classifier compared to other..."** Our idea is motivated by a specific problem in generalisation, which is that there may be certain combinations of latent factors that are poorly represented in the training data. Mixing allows us to explore these combinations. Furthermore, the paper 'manifold mixup' addresses this concern and shows that mixup behaves differently to other regularisation schemes, and is competitive with strong baselines (though only uses mixup with $k = 2$). In our work we focus on a wider class of mixing functions in the unsupervised case. We would like to explore these in the context of supervised learning but this is beyond the scope of our work. For more evidence that mixing is desirable – especially in the low data regime, see the 'MixMatch' and 'MixFeat' papers.

R2: **"The interpolation results (Fig 1) are surprisingly poor, with not much semantic interpolation ... ghosting ..."** The supplementary material contains many examples showing semantically meaningful interpolations, on both CelebA and the Zappos shoe datasets. We compare our results to pixel space interpolation and ARAE, and there is relatively less ghosting in the AMR interpolations of our approach.

R2: **"Please explain how the parameters of p = embed(y) are trained."** The mask is sampled using the reparameterisation trick, which means we are able to backprop through the sampling step and back into the embedding function. Its parameters are updated in unison with those of the autoencoder. We will update/clarify these equations.

R2: **"Please make explicit that you used separate validation and test sets to eliminate any doubt that the results are biased."** For each experiment, three different seeds are run, and the best valid accuracy of each seed is taken and averaged. A val set was only used in our case, though this was an oversight. However, risk of overfitting the val set here is minimal since we are only training linear classifiers on top of the autoencoder bottleneck (the classification losses *do not* contribute grads to the autoencoder). If this is concerning, we would be happy to provide held-out test set results.

R3: Addressing **(1)**, our ACAI uses the JSGAN loss, though we also tried LSGAN as per their paper and this did not appear to make a difference. Assuming mixup with $k = 2$ the only difference is that we don't predict mixing coef, and contrary to ACAI our generator *also* enforces reconstructions to look realistic by fooling D (which tries to classify them as fake). Regarding the usefulness of our proposed mixup for $k > 2$, for SVHN (256) and ablations, mixup with $k > 2$ is superior, as well as CIFAR10, and our Bern(3) results on disentanglement. **(2)** typo, this should be 'ACAI mixup(2)'. For **(3)**, for SVHN $d_h = 256$ only the ACAI results quoted from their paper are superior. While this discrepency should not be downplayed, our own implementation of it (which is also shown in the table) does not perform as well, and this has less confounders at play since it is under the same experimental setup as our methods.