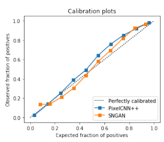

We thank all reviewers for their helpful and detailed comments! We have addressed the issue of dual submission in detail in the rebuttal of paper #6290. As R1 notes and we further elucidate, the problem setting, algorithm specifics, and use-case scenarios of the two papers are different and independent – **model bias** of a **pretrained model** for downstream **Monte Carlo evaluation** here vs. **data bias** during **weakly-supervised learning** for **fair data generation** in #6290.

• **R1:** *Support of the generative distribution p_θ and p.* Our meta-algorithm takes as input a learned model p_θ and p so satisfying the support assumption is tied to the *training* of p_θ (which we do not consider in this work). Nevertheless for a likelihood-based model, the support assumption can be empirically verified via evaluating p_θ on held-out data. The assumption holds true for most variants of VAEs, flows, and autoregressive which have full support by design. We also consider a more general case where we have only sample access to both p_θ and p, where estimating the support is a computationally hard problem (related to estimating the entropy of arbitrary distribution via samples).

To address issues related to the estimation of importance weights via a learned classifier, tricks such as perturbations via small, random Gaussian noise (which has full support), regularization (dropout, early stopping etc.) during training (L306-307), as well as post-processing schemes (L135-143) can be applied. Empirically, we find self-normalization along with early stopping during training (based on validation data) to be sufficient for ensuring good downstream performance for various generative models (GANs, autoregressive models) and modalities considered in this work.

• **R1:** *Defining and measuring the bias introduced by p_θ.* In this work, bias is defined w.r.t. any function f defined over the data domain. Given p_{data}, p_θ and f, the bias is defined as the difference in the expected value of f with respect to p_{data} and p_θ (Footnote 1, Page 1). When p_{data} and p_θ are not known directly, the bias can be estimated empirically via Monte Carlo using a sufficiently large number of samples from p_θ and p_{data} e.g., as shown in Table 1 and Appendix B.

• **R1:** *High variance in importance reweighting.* As with other applications of importance weighting, the extent of and solutions to the high variance issue are empirically motivated. They could introduce a bias (e.g., clipping) but reduce variance more favorably in the tradeoff. In our setting, the primary limitation was that the estimated importance weights could all be small due to artifacts in the generations that were easy to detect via the binary classifier. While we found self-normalization to be most effective, we note in L142 that schemes for post-processing importance weights could be potentially combined, e.g., self-normalized weights could be clipped when variance is a larger issue.

• **R1:** *Choosing the clipping threshold β.* We consider β as a validation hyperparameter with values in {0.001, 0.01, 0.01, 1} chosen to maximally reduce the bias in Monte Carlo evaluation of a downstream function of interest.

• **R2:** *Intuition and guidelines for design choices in L135-143.* Self-normalization is applied only for the generated samples (i.e., those that contribute to bias in Monte Carlo evaluation). Like with other applications, the usage is empirically driven. Generative models tend to produce artifacts that are easy to detect via classifiers and hence, the estimated importance weights are very small ($\ll 1$). In all our experiments, self-normalization was essential to circumvent this issue (see expts. in Tables 4, 5 in Appendix where self-normalization leads to a 53% improvement in mean squared error over vanilla importance weighting). It is hyperparameter free and easy to apply. If variance is high, the range of the weights can be restricted via clipping or flattening with hyperparamters β, α tuned on validation set.

• **R2:** *Data split for reference scores in L168.* Yes, the split is 50-50.

• **R2:** *Running procedure in [45] for long.* Yes, ignoring the high computational requirements of [45] and the fact that the upper bound for rejection sampling is a heuristic estimate, the procedure in [45] could achieve the same effect as the proposed importance weighting approach.

• **R2, R3:** *Calibration.* We believe the default calibration behavior is largely due to the fact that our **binary** classifiers distinguishing real and fake data do not require very complex neural networks architectures and training tricks that lead to miscalibration for **multi-class classification**. As shown by Niculescu-Mizil & Caruana (2005), shallow networks are well-calibrated and Guo et al. (2017) further argue that a major reason for miscalibration is the use of a softmax loss typical for multi-class problems. Top-left figure shows example calibration curves for the experiment in 5.1.

• **R3:** *Interaction of post-hoc normalization schemes with calibration.* While calibration is necessary for a sound density ratio estimation procedure, the utility of the derived importance weights for downstream tasks depends on the underlying expectation of interest. These expectations are evaluated with finite samples and hence, the asymptotic properties of importance weighting (e.g., unbiasedness) are traded off for improved downstream performance using self-normalization and other post-processing schemes.

• **R3:** *Domain adaptation.* We clarify that we are considering the task of multi-class classification and not domain adaptation (L179-181). As we note in L182-183, the Omniglot dataset is a particularly relevant test bed for data augmentation since there are a large number of classes and a few number of training examples per class. We will consider other related scenarios in future version!

• **R3:** D_{g} + LFIW *vs.* D_{g}. Note that this experiment does not only involve Monte Carlo evaluation of a supervised loss but also optimization via gradient methods. In the absence of real data D_{cl}, the classifier training is dominated by D_{g} and correcting the bias in the dataset via LFIW towards an unseen dataset (D_{cl}) can potentially have limited gains.

• **R3:** *Modes getting closer in Fig 1.* As modes get closer, the importance weights will approach 1 (and the class probabilities will approach 0.5) since the mismatch in generative model and data distributions will accordingly decrease.