We thank the reviewers for acknowledging our contributions and for providing valuable feedback.

To R#1: Transformer latency: For MNIST experiments, TensorFlow reports that the LeNet-5 BNN requires 10,906,677 FLOPS and the WGWIN-GP generator requires 9,292,938 FLOPS. The additional cost of the rejection loop is then ~20.2 million FLOPS. The NVIDIA Titan X (Pascal) is rated at 11.0 TFLOPS, so the latency of rejection is ~0.02 milliseconds on our devices.

To R#3: Additional WGWIN-GP architecture and concatenation details: The architecture of the WGWIN-GP generator closely follows that described in the WGAN-GP paper, adding conditioning via the concatenation method referenced in Section 2.1. The conditioning image is flattened and concatenated with a noise vector and passed to the standard WGAN-GP generator as input. The critic architecture also follows the WGAN-GP paper. We concatenate the corresponding one-hot class label to the generated image and, similarly to [*1], to the output of each hidden convolution layer. We clarified the concatenation process in our paper and have added the missing hidden-layer citation. We also added detailed architecture diagrams for each network to our appendix. Thank you for pointing this out.

To R#3: Bayesian neural network and rejection function interaction: We use Monte Carlo sampling to determine the BNN's predicted class and uncertainty metric. We first sample the model ten times for the given input x_i, effectively ensembling ten different networks. We calculate the mean of the given class probabilities and treat the argmax as the class prediction y_i'. We treat the median of the probabilities for this predicted class as the certainty metric c_i. These two metrics are passed to the rejection function. We did not see a significant difference in WGWIN-GP performance when treating the mean as the certainty metric. Alternative approaches may consider the variance in the predicted class across models. Since uncertainty is represented by the difference in prediction across models, a single network prediction does not provide uncertainty information and thus should not be used alone.

To R#3: Baseline models: We reevaluate the WGWIN-GP using a stronger baseline (expanded LeNet-5 with batch-norm, dropout, and more convolution layers/filters; the exact architecture will be included in the paper) while the rejection function and WGWIN-GP are unchanged. The GWIN continues to have a positive impact. Results are shown in Table 1. CIFAR10 experiments are in progress.

To R#4: Denoising methodologies: A key distinction between our work and the papers mentioned by R#4 is that those papers and their related works focus on using GANs to increase the robustness of a classifier *during training* by generating out-of-distribution training data (similar to hallucination methods in the few-shot learning domain) while our method assumes a *fixed, pretrained* classifier and uses generative methods to translate novel, out-of-distribution examples to the confident distribution *during inference*. Since the GWIN framework learns representations that the classifier labels correctly with high confidence, these generative denoising methods can easily be paired with our framework: a classifier is trained using the aforementioned techniques and the GWIN is then used to transform any novel examples that the new classifier is not entirely robust to. Similarly to DefenseGAN, the flexibility and additive nature of our framework means that we can easily build atop these existing denoising methodologies. Since noise only represents a subset of out-of-distribution observations, we cannot rely entirely on denoising techniques to address classifier robustness. GWINs take a step towards a generalizable, principled framework for "rethinking" uncertain examples and leveraging classifier uncertainty. For completeness, we will add a more detailed Related Works section to the appendix describing how these methodologies address out-of-distribution robustness during the training process.

To R#4: False positive rate: Fig.5 shows the change in rejected sample certainty of the ground truth label for varying rejection thresholds. The WGWIN-GP increases confidence in the correct class more often than not. Depending on the classifier's accuracy on the rejected subset, the threshold can be tuned to maximize the expected number of correct classifications. Typically, a classifier is required to predict so higher accuracy implies fewer misclassifications and therefore fewer false positives. If true rejection is allowed, a model can still use the GWIN to perform a transformation and reject the original observation if the transformed observation is below some arbitrarily high threshold.

Table 1: Test set accuracy for Digits (top) and Fashion (bottom) on rejected observations using GWIN transformation for the given certainty threshold τ averaged over 10 runs. For $\tau = 0$, acc. is 99.2% (Digits) and 91.0% (Fashion).

τ	% Reject	BNN Acc.	BNN+GWIN Acc.	Rejected Acc. Δ	Overall Acc. Δ	% Error Δ
0.70	0.25	46.44 ± 12.30	59.96 ± 7.28	13.53 ± 15.54	0.03 ± 0.04	-3.42 ± 4.15
0.80	0.41	51.79 ± 7.93	66.13 ± 8.39	14.34 ± 11.08	0.06 ± 0.05	-6.39 ± 4.90
0.95	0.81	53.44 ± 3.91	68.60 ± 4.50	15.16 ± 4.85	0.12 ± 0.04	-12.64 ± 3.66
0.99	1.26	60.11 ± 4.53	67.63 ± 3.62	7.52 ± 3.07	0.10 ± 0.04	-9.77 ± 3.69
0.70	2.54	45.24 ± 2.20	57.10 ± 2.75	11.85 ± 3.26	0.30 ± 0.09	-3.17 ± 0.93
0.80	4.04	46.49 ± 2.16	57.49 ± 2.67	11.00 ± 2.60	0.45 ± 0.11	-4.64 ± 1.09
0.95	8.18	52.23 ± 1.44	59.18 ± 1.31	6.95 ± 2.15	0.57 ± 0.19	-5.99 ± 1.93
0.99	12.33	57.04 ± 1.22	61.61 ± 0.86	4.56 ± 1.23	0.56 ± 0.15	-5.88 ± 1.48

[*1] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, arXiv preprint arXiv:1605.05396, 2016.