We thank the reviewers for detailed comments and a positive assessment of our work. We will release a modularized implementation of MINs alongside all the tasks to make it easy to use and reproduce the results **(R1)**. We will incorporate experimental details into the main text and substantially revise the experimental section to clearly define the baseline methods and the task setup **(R2, R3)**. We thank **R4** for proposing the interesting idea to extend MINs to optimize multiple objectives, and we will discuss this as potential future work.

R2: It's not obvious that Disc will have any reason to look at y at all: The design of the conditional discriminator is based directly on conditional GANs (cGANs) (Mirza et al. 2014), and should work for the same reason that cGANs work. We will clarify this in the paper. Intuitively, the learning dynamics of a MIN (or a cGAN) would force the generator to first produce valid samples, since the discriminator can otherwise easily distinguish fake and real samples simply based on validity. Once the generator produces valid samples, the discriminator can only win in the GAN game by exploiting the correlation between the conditional value y and the output x, thus forcing the generator to accurately model the conditional distribution $p(x|y)$. We experimented with the stategy R2 mentioned, that pairs real x with incorrect ys as negative examples, similar to Reed et al. 2016 (AC-GAN), but did not observe a benefit, and so we omitted it in the interest of simplicity.

R2, R3: Experimental evaluation details: We will clearly define the task specifications and prior methods in the final version of the paper in an extra page devoted to specifically this, and ensure that all experimental details are provided. We already provide the hyperparameter settings and setup details in App. D.2, but will bring them to the main text and add more details. To clarify some specific details: **(i)** the contextual bandit task requires predicting the right class label (x) of an image (which is the context) given access to (c, x, y) tuples where $y = \mathbb{I}(\text{x is the correct label for image c})$. The evaluation metric is simply the standard test classification accuracy for unseen test images. **(ii)** In stroke width optimization, we define stroke width as the number of pixels with intensity more than a threshold (= 0.2) in the image. An image is invalid if the number of pixels with intensity more than 0.2 is more than 90%, indicating that this image is unlike MNIST digits. **(iii)** We provide **3 tasks** (controller, protein, and bandit) with quantitative and objective evaluation scores. Two of these (protein, bandit) are taken directly from and compare to baselines from prior works.

R2: Intuition for re-weighting distribution: We already provide some intuition in 188-189, and we will elaborate on this further here and in the paper. Our choice of p weighs the higher y values higher (thus reducing "bias" as a result of training on suboptimal y values although the inverse map is queried at only large y values) while also ensuring that the number of points for the chosen large y values is large enough (thus reducing "variance"). The expression for p captures this tradeoff: $\exp(y - y^*)$ captures the bias, and $\frac{N_y}{N_y + K}$ (which becomes 0 for small N_y and tends to 1 otherwise) captures the variance effect, and our choice of p multiplies the two expressions.

R1: Table 2 and new experimental comparisons: We used a NN baseline here, and not a GP. We have the edited the citation to include several other papers that utilize a forward function for optimization. We have also added a comparison to **(1)** forward model ensemble + optimizing mean over the ensemble, and, **(2)** forward model ensemble + variance penalty (that captures uncertainty), and on the controller task, we find that over 4 seeds each, **(1)** attains a mean return of **325.3** and **(2)** attains **165.3**, and MINs outperform both methods attaining **1960.1** mean return (see Table 2, Hopper). We will add a comparison of these forward model variants to all the tasks in the final.

R1: Details about the method. We have included these details in the paper now. We respond here to some questions:
- **Decoupled effects of components / Ablations:** Lines 340-356 in Section 4.1 present ablation studies, where we study the effect of decoupling **(i)** training the inverse map **(ii)** re-weighting and **(iii)** approx-infer. We have now presented these results as a table more effectively in the paper. There is a small drop in performance w/o Approx-Infer (Table 1, Fig 2, lines 340-353). If the reviewer suggests some new ablation studies, we are happy to add those in the final version.
- **Training a GAN:** We trained 4 seeds for the controller, protein, MNIST and bandit tasks and were able to run only 1 seed for face optimization, since it requires about a week to train. We directly used implementations of GANs (details in App. D.2, lines 766-778) that have been previously employed on these domains *without* changing GAN hyperparameters at all. We did observe some mode collapse, but found that the models were still able to produce good solutions – this seems logical, since we are only interested in one mode in optimization (the one near the best solution).
- **Approx-Infer.** As discussed in lines 582-585 (App. A), we found it empirically convenient to use a fixed "penalty" version, with $\lambda_1 = 10$ and $\lambda_2 = 0.5$. We fixed these values across all tasks and added a discussion of this in Sec. 3.2.
- **Practical λ and τ.** We used empirical densities and replace this choice by λ and τ (discussed in App. D.2 (lines 783-792)), i.e., Thm 3.1 provides a functional form of the re-weighting distribution.
- **Exploration.** We did *not* re-initialize the nets after a round as this was computationally and performance-wise better.

R2: Theorem 3.1 and MSE: We have added a discussion of this in the paper now. We chose the gradient $\nabla_\theta \mathcal{L}(\mathcal{D})$, since this gradient is used to update the parameters θ of the inverse map, and we are interested in choosing p such that the expected gradient under p aligns close to the expected gradient on the optimal datapoint(s). The choice of MSE between gradients was convenient and gave us a way to select the re-weighting, but in principle, we could use other metrics, such as, finding the optimal p by minimizing the cosine similarity between these gradients.