We would like to thank all the reviewers for their time and comments, which will definitely help improve the paper. We now provide a point-by-point response to the comments.

[R1, Ablation study from Table 2]: The numbers reported in the table were obtained with the real weights, not the estimated ones (so performance needs to be compared to IWDAN-O, not IWDAN). We wanted to deconfound the effects of weight estimation effects from the improvements provided by the two losses. To confirm things, we have run the ablation with weight estimation. On e.g. Digits, DANN+$\mathcal{L}_{DA}^{w}$ has a perf. of 94.35%, which is lower than IWDAN (94.90%). We will clarify this and add an ablation table to the paper with results using weight estimation.

[R1, Hyperparameter selection]: The values reported in B.7 are the default ones in the implementations of DANN, CDAN and JAN released with the respective papers (the links to the github repos are provided in B.7). We did not perform any search on them, assuming they had already been optimized by the authors of those papers. To ensure a fair comparison and showcase the simplicity of our approach, we simply plugged the weight estimation on top of the baselines with their original hyperparameters, and did not optimize those for IW.

[R1, Performance]: The performance we report is the best test accuracy obtained during training over a fixed number of epochs (we will specify that number in the appendix). We used that value for fairness with respect to baselines (as shown in Fig.2 Left, the performance of DANN decreases as training progresses, due to the inappropriate matching of representations showcased in Th.2.1). Thank you for the various suggestions and missing references/discussion, we will include them in the next iteration.

[R2, Markov chain]: We would like to clarify that the $\hat{Y}$ mentioned by the reviewer corresponds to $h(Z)$, the one-hot classifier, in Line 565, and the $\hat{Y}$ used in Line 565 is the block selection operator defined over $\tilde{Z}$, so the Markov chain still holds and as a result Theorem 2.1 is correct. We thank the reviewer for clarifying the CDAN algorithm, and we are happy to update our discussion about CDAN (consequently also the presentation of Theorem 2.1) in our next iteration.

[R2, Weight distance]: In Fig.2 Right, the weights reported are computed using the confusion matrix and the predictions on the target domain as described in Lemma 3.2. For IWDAN-O, those weights are not used, just computed, and the non-zero distance is because at initialization GLS is not verified. As training progresses, the model becomes closer and closer to verifying GLS, and as such, the distance to the true weights goes to 0. We will clarify this in the text.

[R2, h^* satisfying GLS]: This is because the existence of ground-truth labeling function means that for each X, there will be only one true label Y, hence conditioning on Y essentially partitions the input space of X. As a result, the conditional distribution of $h^*(X) \mid Y = y$ reduces to a point mass (a Dirac distribution) concentrated on y over both domains, trivially satisfying GLS.

[R2, Reweighting in Eq.7]: The w appears in the numerator under the assumption of GLS, we will clarify this in our next iteration. If GLS is not verified, as the reviewer pointed out, there would be no w in the numerator.

[R3, GLS hard to hold in practice]: We agree with the reviewer that GLS is hard to satisfy *exactly* in practice, but the goal of Theorem 3.1 and Theorem 3.4 is to inspire our algorithmic design and neither of these two theorems requires GLS to hold exactly.

[R3, Theorem 3.1]: We respectfully disagree with this comment. As we discuss from Lines 140 to 146, our error decomposition is completely orthogonal (hence not a mere transformation) to the existing one (Theorem 2 of [7]). Essentially, they correspond to two different ways of decomposing a joint distribution, i.e., Ours: $\Pr(X, Y) = \Pr(Y) \cdot \Pr(X \mid Y)$ versus Existing: $\Pr(X, Y) = \Pr(X) \cdot \Pr(Y \mid X)$, hence the core notions of each term are completely different. Furthermore, our claim is also correct. The Δ_{CE} is about the conditional distribution of $\Pr(Z \mid Y)$ whereas the optimal labeling function is $\Pr(Y \mid Z)$.

[R3, Label switching and mode collapse]: The problem we aim to tackle in this paper is orthogonal to those two issues. We do not think it is more important, we simply think that having models robust to mismatched label distributions is important for successful domain adaptation (and in the case of large mismatches, the improvement our algorithms provide is rather significant - Table 3, subsampled datasets). We agree that our work is not a silver bullet to all the problems DA encounters, but we do believe our method can augment most (if not all) algorithms designed to improve performance in DA.

[R5, Relation to [1]]: Thanks for your kind comments. We will add a clearer comparison with [1] in our final version. At a high level, both the method in [1] and ours for importance weight estimation use the core idea of moment matching under GLS. However, the specific algorithms used to obtain these estimations are different. In particular, the method in [1] uses the inversion of the estimated confusion matrix directly, while ours proposes to solve the QP in Eq. (4). This difference is very important in practice: matrix inversion is notoriously unstable and the estimated weight by matrix inversion could be infeasible, i.e., the weight could be negative. Our QP formulation explicitly get rids of both issues hence is more numerically stable and guarantees to return feasible importance weights.