We thank the reviewers for their insightful comments – we will make sure that all of them are adequately addressed in the revised paper, including: missing references, relevant ablations, more NAS results, more latency measurements, discussion about limitations and design choices – we wholeheartedly agree that these will significantly strengthen the paper. *We will release all the source code and data*. Meanwhile, please find our detailed responses below.

Novelty and contribution: Both latency and accuracy are studied in this paper as they are critical for hardware-aware NAS. We show how improvements in predicting either of them impacts the outcome of the search, and how imprecise prediction of one metric can limit potential gains from improving the other – something that has not been systematically studied before in the context of NAS. Building on top of that, our second contribution is to show an efficient way of improving estimations of both accuracy and latency by using two versions of our GCN-based predictor, tailored for their respective metrics, resulting in a new SOTA on common NAS benchmarks. We want to emphasize that even though accuracy and latency prediction can be studied independently, they are both important in order to improve HW-aware NAS – we will improve writing in order to make the connection clearer (**R3**). • *Key differences from Shi et al. and D-VAE (**R2**):* There are substantial differences between our work and Shi et al. The key idea of Shi et al. is predicting accuracy with a GCN. On the contrary, our work covers latency prediction, binary relation prediction (for accuracy), and combining both for multi-objective NAS. Additionally, our work on the binary relation learning and iterative data selection is a novel solution to the open problem raised by Shi et al. (but was not addressed in their work) that the ranking of models is more important than their absolute accuracy. D-VAE is complementary to our work as we could use D-VAE in place of GCN; studying whether D-VAE further improves the results is an interesting open problem.

Relation learning – symmetry and cycles: We use the binary predictor as a drop-in replacement for the comparison operator in the standard Python sorting function. In the case of symmetric and/or non-transitive relation, the final order of elements will depend on their initial order and the implementation of the sorting algorithm. • *Symmetry (**R1**):* (To avoid confusion we will talk about anti-symmetry *of the relation* which is related to symmetry of the predictor). To encourage anti-symmetry, we include both pairs of model architectures, i.e. (m1,m2) and (m2,m1), in the training set and verify that the resulting relation is highly anti-symmetric, i.e., (p1-p2)(p1'-p2')>0 with probability 0.98, as measured on 1000 randomly sampled points from NAS-Bench-201. • *Cycles (**R2**):* Based on the comments, we ran an experiment to identify cycles and found that for a sample of 1000 models there are more than 10 million cycles (but only 31 for 300 models), suggesting that the number of cycles can grow exponentially. Even though we handle cycles randomly, we are not concerned with them due to our strong empirical results. However, we do agree that fully exploring their impact is an important research question.

Bigger search space / unseen macro architecture: Regarding search space size and scalability: even though we primarily evaluated our approach on NAS-Bench-201, it performed very well when ported to NAS-Bench-101 (which is an order of magnitude larger). Following the suggestion of **R2**, we are currently evaluating our methodology on the DARTS search space and will include the results in the final paper.

Layer-wise latency predictor is also trained with end-to-end latency: The layer-wise predictor is calibrated with the end-to-end latency by a scaling factor, i.e. exactly the same number of training examples – *(model, end-to-end latency)* pairs – are used to train both predictors, which addresses (**R3**)'s concern on fairness. We will clarify the training part in the revision. In latency-constrained NAS, the architectures discovered by any predictor that exceed the constraint are discarded for a fair comparison (**R3**).

GCN design decisions, typos, clarifications (R1): We ran ablation studies based on the comments. • *Normalizing adjacency matrix:* We have tried different normalizations and the best result is achieved without normalization. *Softmax and sigmoid* activations lead to similar results. *Global node and flow direction:* Both forward propagation (with a global sink) and backward propagation (with a global source) result in similar performance. • *LatBench with quantization:* Thanks for the great suggestion. We already support INT8 models on EdgeTPU and Snapdragon DSP, and will release LatBench with FP32/FP16/INT8 on supported platforms by November. • *Typos, clarifications:* We will correct all typos (especially, $\sigma(AH^lW)$) and add details on every term (e.g., A_{ij}). Shaded regions in figures mark interquartile range.

Comparison to other NAS, other questions (R3): • We use Aging Evolution (AE) as the major baseline as it is shown to perform best in the NAS-Bench-201 paper. We are up to 3x more sample efficient than the current best (line 237 and Fig. 6 in the paper). Also, we beat all other SOTA methods on NAS-Bench-101. • Fig. 6 compares BRP-NAS with (AE+layer-wise) as AE and layer-wise are SOTA for NAS and latency predictions, respectively. Indeed the performance of AE is improved by our predictor, which is highlighted in Fig. 2. • Our result in Table 3 is averaged from 32 runs. • The BRP with iterative training has a lower Spearman-ρ than the plain BRP as we are not concerned about the ranking of low performing models and focused on high performing ones at the expense of global ranking quality (see line 242-245 in the paper, and S2.1 Observation 3 in the SM).

Applying our approach to NAS w/o hardware constraints (R4): We already have results on this in Section 4. Fig. 6 (left) and Table 3 show exactly that BRP-NAS outperforms SOTA in unconstrained settings, e.g. accuracy prediction of BRP-NAS vs AE. Generalization of our approach to broader settings (eg. energy or memory usage aware optimization) is a fascinating future direction of research.