We thank the reviewers for their helpful reviews. We performed all the additional experiments that were suggested: hyperparameter tuning for each individual encoding, (preliminary) experiments on the DARTS search space, and comparison to GraphNN's and VAE's (and we point out to R3 that experiments on NAS-Bench-201 were already in our paper). Please see the details below.

R1 We agree that it would strengthen our experiments to perform hyperparameter tuning on each NAS algorithm for each encoding. We just need to be careful that we do not perform hyperparameter tuning for specific *datasets* (in accordance with NAS best practices (Lindauer and Hutter, 2019; Yang et al. 2020)). Therefore, we perform the hyperparameter search on CIFAR-100 from NAS-Bench-201, and apply the results on NAS-Bench-101. We defined a search region for each hyperparameter of each algorithm, and then for each encoding, we ran 50 iterations of random search on the full hyperparameter space. We choose the configuration that minimizes the validation loss of the NAS algorithm after 200 queries. See the figure below for the results of Reg. Evolution (top left), and Local Search (top middle). Most encodings improved or stayed the same, though a few did slightly worse (because the hyperparameters for those encodings did not generalize from NAS-Bench-201 to NAS-Bench-101). Finally, we agree with the other comments/clarifications and will include them in the final version of the paper.

R2 While we do not have access to a large GPU cluster, we agree that results on the DARTS search space would strengthen the paper. We now provide preliminary results for experiments on the DARTS search space. Before this project, we had already computed a set of 1200 architectures from the DARTS search space trained to 50 epochs, which we now use to test different encodings on a neural predictor. The experimental setup is the same as in our paper, except the architectures were not all drawn i.i.d.—120 were drawn i.i.d. and the rest are mutations. We see similar trends as with the nas-bench-101/201 datasets. See the figure below (top right). Next, we ran an initial experiment testing three different encodings with random search. For each encoding, we trained 100 architectures to 25 epochs. See the figure below (bottom left). Here, the path-based encodings outperformed the adjacency encoding. With more time, we will train the architectures to more epochs. We also did not have time to run multiple trials, so there are no error bars. We will do our best to extend these to a more full set of results for the final version of the paper.

R3 We point out that our paper does include experiments from NAS-Bench-201 (briefly mentioned on lines 247-248, with the details in the appendix). We agree that we should include plots of the regret, and so we add an example below (bottom middle). We will correct all the other clarifications/suggestions you have mentioned. Finally, we agree that our paper would become stronger if we include learned encodings such as GraphNN and VAE. We found an open-source implementation for "Neural Predictor for Neural Architecture Search" and we also used the open-source code from the paper "D-VAE: A Variational Autoencoder for Directed Acyclic Graphs" (which was designed for the ENAS search space). We ran new experiments with these encodings in the same setting as our neural predictor experiment in our submission, and we show the new results in the figure below (bottom right). In fact, the path encoding outperforms the trainable encodings, which was also noticed in prior work, e.g. BANANAS (White et al. 2019). We will add these results to the paper.