We thank all the reviewers for their words of appreciation, suggestions for improving presentation and insightful control experiments. It has made the work better.

R1 "are the experiments run up to 5 days and then report the best? Is it possible to ... early stopping for NAS?" We keep a search job using four GPUs running until either the best performing models have not changed for a day or the total computation is too much, typically 3-4 days. In the current state, evaluation of a model with final training requires almost as much computation as NAS itself, e.g., CIFAR requires 2 GPU-days final training.

"The parts on multiple workers..." Multiple workers enable the search to reconsider a growth iteration at any of the intermediate models. This reduces the effect of bad early decisions. Furthermore, multiple workers can share intermediate results with each other. Many NAS papers utilize multiple workers like NAS with RL by Zoph et al 2017. Note our reported gpu hours is the sum of all workers' gpu utilization. The ability to use multiple gpus in parallel is very dependent on the search procedure itself. For example DARTS which keeps all possible architectures in a single gpu's memory is hard to parallelize without modifications to the search algorithm itself as in ProxyLessNas.

"..but how the variances for the other methods are not reported" Unfortunately, most other NAS work do not report variance of models or search. We will add the existent ones from AmoebaNet, DARTS, SNAS and PNAS to the paper.

R2 "The connection/differences to NAS methods combining network morphisms with evolutionary algorithms should be discussed in more detail..." We will add details to summarize search methods based on net-morphism, such as `LEMONADE`(Elsken et al. 2018) and Path-level(Cai et al. 2018). Both methods also explore the search space with small and iterative incremental changes. However, they choose the increments based on evolutionary algorithms or REINFORCE, where this work aims to guide the changes with gradient information.

"I propose to include at least the models corresponding to the ones in Table 2 (SNAS, ProxylessNAS) for completeness." In general it is difficult to compare NAS algorithms to each other due to differences in search space, size, quality of starting network, search budget used and whether variances are reported to control for stochasticity during training. We have focused on smaller network regimes to keep experimentation manageable and report results there for a fair comparison. However we will change the limit to 4M to include Path-level and ProxyLessNas. Although please note that Path-level and ProxylessNas start with PyramidNet, which is stronger than the similar starting conditions of NASNet, AmoebaNet, DARTS, SNAS, and ENAS.

Comparison to other supergraph methods The main advantage of Petridish compared to other supergraph-based methods is that Petridish doesn't rely on a good supergraph to be made available (which by itself is a manual design decision) and often is not available on datasets which are not cifar10/100/ImageNet on which considerable prior knowledge via manually designed networks exists which informs the supergraph design. Even where supergraphs are available, Petridish can be viewed as breaking the supergraph optimization into multiple steps as opposed to transforming the search into one giant supergraph optimization like DARTS. We are proving out this aspect of Petridish by running on datasets where prior good supergraphs are not available.

Starting from worse models. The initial model of this work is already one of the simplest in the common search space among the NAS works including DARTS, ENAS, NASNet, and AmoebaNet, and we already know from the micro vs. macro comparison that starting condition is a dominating factor for the search result (paper line 229-235).

R4 "The paper is mostly well written and clear. I am mainly struggling with the iterative process." The cells are incrementally grown for multiple iterations. Each iteration starts with weak-learning with weight sharing (page 4), followed by weak-learner finalization (page 5). Since each weak-learner training with weight sharing does not affect the existing model, we can conduct multiple independent weak-learner trainings simultaneously. In macro search, the layers that are the end of the initial cells grow independently in each iteration at the same time. Cell-search does the same except that we force the same alpha parameters across cells, so that the decision is uniform.

(required) Compare against random growth baseline. **(also requested by R2)** During author response, we constructed 20 models where we grow randomly starting from the initial model. The growth stops when the model computation complexity (in multi-add) is the same or just exceeds the reported model. We train each model 4 times to compute the mean test error rates on CIFAR10. The best mean is 3.03%, and the average mean is $3.32 \pm 0.15\%$, which is close to the random-model performance in DARTS Table 1. In addition, we experiment with replacing feature selection with random choice and leaving all other parts intact, i.e., we keep initialization and finalization of weak learners with parallel workers. The average of mean error rate of the final-trained models is $3.26 \pm 0.04\%$, close to random models.

(required) Compare against enlarged initial models. We created four configurations to enlarge the initial models. The depth are set to 1, 2, 4 and 8 times the depth of the reported models and the number of channels are multiplied by the square root of 8, 4, 2, and 1, so that they have similar complexity as the reported model. These four models have mean error rates ranging from 3.00% to 3.16%, averaged over five instances of final training. In comparison, the mean performance of the reported model of the similar complexity is $2.87 \pm 0.13\%$ error rate.