R1 raises a concern about the engineering relevance of our work, saying there is "no direct conclusion on how to improve program induction models". While our focus is cognitive modeling, we do also hope to guide research in program induction. This can come in at least two ways. (1) our results show that an approach integrating library learning (via symbolic compression) and learned search control (via a neural network) is a promising way to describe human learning. This motivates further exploring these approaches in program induction in AI. (2) We discovered that a learned syntactic simplicity bias over the space of programs (library learning) does not suffice for inferring human-like motor plans. Instead, a hybrid model combining this simplicity bias with a bias toward efficient motor plans was needed. We suspect this finding points toward a more general insight, that program induction for planning problems would be improved by adding inductive biases in favor of short *and* efficient programs. Here efficiency could be AI-specific (e.g computability, ergonomics), or could be based on human motor constraints, to generate human-like behavior. We will revise our paper to highlight these points.

R1 asks why efficiency biases (described in Lines 94 - 106) were not applied to the PI model. We did not bias the PI model because it acts as an "efficiency lesioned" comparison to the Hybrid model (which we did bias).

R1 raises a concern about how best to compare human and model behavior, given that part of the training of the Hybrid model was supervised by motor data, while humans were not. The supervision by motor data was used to infer efficiency constraints that humans naturally bring to the task. From a human's perspective, these biases are already present, so there is no need to further "train" on these biases. Comparing different models, trained with and without efficiency biases, with humans thus highlights the importance of capturing this human prior knowledge in the model.

R1 asks about consistency of models trained with different random seeds. We have observed the results of learning to be relatively stable over random seeds. But we thank the reviewer for raising this important point. We can prepare supplemental figures showing results from several other random seeds.

R1 asks for illustrations of libraries post-training. Figure 6B,D show subsets of the updated libraries after training, as well as sampled programs from those libraries, but we agree that this core aspect of our model deserves further illustration. We can prepare supplemental figures showing a wider range of learned library routines.

R1 asks which models are shown in Figure 6. All of the model results are for the Hybrid model, except when noted otherwise: "Baseline" in Panels C and D; "Null", "PI" and "MC" in Panel E. We thank the reviewer and can update the figure legend accordingly. We also thank the reviewer for bringing to our attention the error in line 182.

R2 is concerned that this work is incremental relative to DreamCoder/EC2 and Lake et al. 2015. While we build on code of the former and ideas of the latter, there are qualitative differences. DreamCoder/EC2 neither compare model with human behavior nor learn efficient plans, and these issues are intertwined: as mentioned earlier, we discovered that efficiency biases are needed in tandem with simplicity biases to best account for human behavior, which may have repercussions both in our computational understanding of this behavior, and in how we design program inductors for planning problems. Learning in Lake et al did not implement library learning via symbolic compression (therefore lacking higher-order abstractions in drawing/handwriting) and did not use learned search control (via a neural network). Our results highlight the importance of these two features for modeling human learning.

R2 raises a possible confusion about our inference algorithm. To be clear, the model does not search randomly for programs until finding one that works, but instead performs a neurally-guided systematic search. The quotation "programs are sampled from a generative model..." refers to how training data are generated for this neural guidance. Our library learning is similar to "chunking", with chunks discovered via a refactoring step, and does indeed become "more skilled over time" as R2 asks: at the start of learning, none of the training images can be drawn (0/72 combining both training sets) but at the end state, almost all of them can be (68/72).

R4 points out that this work is just a first step in capturing rapid learning of motor plan generalizations, as highlighted in Figure 6E by the gap between (model vs human) and (human vs human) agreement. Indeed, we only attempt to capture a subset of the myriad experiences that influence learned structure (and via only a few dozen training stimuli). We hope that future work helps close this modeling gap.

As R4 notices, we used pre-defined starting primitives (circles, lines, repetitions, etc.), which are relatively simple yet plausibly accessible to humans even before they begin the study. Our approach can in principle generalize; e.g., to model learning at a longer timescale (e.g., developmental), we could start with a spartan yet highly generic basis, and learn to draw from a broader training curriculum. This would be exciting future work, although with less relevance for the behavioral experiment designed here for studying rapid single-session learning of higher-order structure. We thank R4 for this possible future direction.

We thank R4 for suggesting improving readability of the section "Reweighting motor trajectories by motor cost". We will update this section with the suggested modifications: (1) clarifying that "t draws I" means the trajectory t (a sequence of strokes) produces the image I; (2) moving relevant parts of Suppl. 2.2 to this methods section.