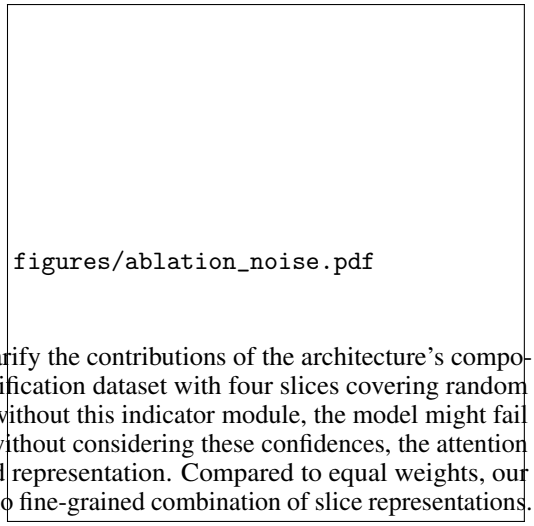
We thank the reviewers for their time and thoughtful feedback. Taking their helpful comments into account, we sought and received additional feedback to extend and clarify the presentation of our work. Since submission, we have also achieved **state-of-the-art scores** on the SuperGLUE benchmark by 2.7 points using the proposed approach.

Ablation studies. [R1, R2, R3] We thank all reviewers for their detailed feedback about the proposed architecture. We have added ablation studies to Section 3.3 to clarify specific components and will include additional experiments and details in the final draft. **Coping with noise**: We test the robustness of our approach on a simple synthetic example: in the Figure to the right, we show noisy SFs (top row: no noise, 40% noise, 80% noise) and the corresponding slice indicator's output as a heatmap (bottom row: darker indicates higher likelihood of slice-membership). We show that the indicator assigns low relative probabilities on noisy (40%) SF samples (bottom middle) and ignores a very noisy (80%) SF, assigning relatively uniform scores to all samples (bottom right).



Architecture ablations: We thank R2 and R3 for suggestions to clarify the contributions of the architecture's components. We perform an ablation study using a synthetic, binary classification dataset with four slices covering random data subsets. We observe that indicator outputs contribute +3.4 F1; without this indicator module, the model might fail to handle noisy SFs. The predictor confidences contribute +4.6 F1; without considering these confidences, the attention mechanism might combine non-expert features into the reweighted representation. Compared to equal weights, our attention mechanism contributes +5.6 F1; without attention, there is no fine-grained combination of slice representations.

Presentation of model architecture. [R2, R3] In response to R2's feedback, we have updated each module with dimension annotations and updated Figure 2 with visual cues to specify where slice labels are used during training (i.e. as *labels* for training indicators and *masks* for training predictors). Following R3's suggestion, we have more clearly expressed the costs (e.g. model size, training time) of each approach, especially MoE, with respect to OURS in Table 1.

We thank R2 for pointing out vague notation in Section 3.2, which we have clarified in the updated draft. In L160, we changed p to w to avoid confusion with probability notations. In L165, $g(P)$ (from Section 3.2(e)) indeed refers to the concatenation of each $g(p_i)$ from Section 3.2(d). Our notation refers to the binary setting, where $c = 1$ such that $P \in \mathbb{R}^{h \times k+1}$. In L166, we clarify that the predictor confidence is computed using the maximum probability over prediction classes. Our experiments are binary classification tasks, for which we use the absolute value of the predictor logit as this confidence. Additionally, R2 is correct that z is not used in these reweighted features, z'. Instead, the base representation is modeled as a trivial slice: p_{BASE}, and a final prediction is made based on the reweighted z'. We clarify that h is a hyperparameter for flexibility in specific applications; for simplicity, we set this to r in experiments.

R1: We agree with R1 that Section 3.3 could be reframed more conservatively. We have updated the title to "Synthetic Experiments" to clarify that our observations are grounded in the synthetic setting. R1 asked about a missing Figure 3c; Figure 3c refers to the right-most graphic in Figure 3, and we have labeled this more explicitly in the updated draft.

R1 asked for clarification about experimental details and provided feedback for the organization of Section 4. In our data splits, we ensure that the proportion of examples belonging to each slice is equivalent across train/valid/test for appropriate evaluation of slices; we did not collect/re-use different data sources. Furthermore, we moved SF implementation/evaluation details and a description of slices from the appendix to the body of the paper. In Section 4.3, we have included an error analysis regarding counter-intuitive trends in slice-specific performance. For example, results on COLA may be explained by limitations in the backbone architecture; we observe low performance on MOE, which has extra capacity, on certain slices (e.g. *ends with adverb* and *has but*) where OURS also underperforms. Following R1's suggestion, we have included detailed descriptions about the baselines. Specifically, we anchor our work in data programming [23], a baseline from weak supervision literature that learns to combine noisy, user-provided heuristics.

R2: We thank R2 for feedback on our empirical evaluation. In experiments, we use strong baselines as backbones: pre-trained BERT++ (proposed by SuperGLUE organizers) for COLA/RTE and pre-trained ResNet for CYDET. We thank R2 for the suggestion to report the evaluation server results; we obtain three new SOTA scores on SuperGLUE tasks: (+3.8/+2.8 Avg. F1/acc. on CB, +2.8 acc. on COPA, +2.5 acc. on WiC).

R3: R3 asked about the weighting of our loss terms. In practice, one may set a hyperparameter for each loss term; to simplify our study, we set all weights to 1. R3 is also correct that "hard" slice features would be more susceptible to noise; we will include this baseline in Table 1. We emphasize, however, that introducing such features would violate the key assumption that slice information/metadata is not available during inference, as discussed in Section 3.1. Finally, R3 asked how many slices our model can support. Our experiments include an average of 10 slices per application, while in an industrial collaboration, we have deployed the *Slice-based Learning* on hundreds of production slices.