A APPENDIX

A.1 RESULTS ON CLEVRER

We report additional results on CLEVRER Yi et al. (2019) dataset. To support the question answering setup, we adopt a straightforward implementation utilizing an off-the-shelf language encoder from CLIP Radford et al. (2021).

CLEVRER has four question types: descriptive, explanatory, predictive, and counterfactual. The descriptive questions have single-word answers. The size of the answer vocabulary is 21. The other three question types are in the form of multiple choices. Each question may have zero, one, or multiple correct answers. We follow the standard practice and formulate the descriptive question answering as a 21-way classification task given questions, and the other question answering tasks as binary classification given question and candidate answer pairs. Each question or answer is encoded into a 512-dim embedding by the CLIP text encoder. We report per question accuracy.

We follow the same video only self-supervised learning process during model pretraining. When finetuning, we first encode each image in a video into slots, and the question (and optionally an answer) corresponding to the same video as token embeddings. The question and answer embeddings are appended to the end of the video slot sequence. The entire sequence of video, question, and answer tokens are fed into the temporal transformer to perform reasoning. During pretraining, we randomly sample 32 frames of 64×64 and pretrain the model for 500 epochs. We use 4 slot tokens, 2 temporal transformer layers, and 768 hidden units. The overall pretraining setup is similar to that of CATER. During finetuning, we randomly sample 64 frames (among 128 total frames for each CLEVRER video) and finetune the model for 200 epochs.

Table 6 shows comparison with previous methods, we observe that our IS-CL framework performs competitively across all question types, especially on predictive and counterfactual questions.

Table 6: Benchmark results on the CLEVRER dataset.

Method	Object-centric	Object superv.	Descriptive	Explanatory	Predictive	Counter.
MAC (V+)	✓	✓	86.4%	22.3%	42.9%	25.1%
NS-DR	✓	✓	88.1%	79.6%	68.7%	42.2%
DCL	✓	✓	90.7%	82.8%	82.0%	46.5%
ALOE	✓	✗	94.0%	96.0%	87.5%	75.6%
IS-CL (ours)	✗	✗	91.6%	95.1%	88.9%	76.0%

A.2 MORE ABLATION STUDY ON SLOTS

Our hypothesis is that the proposed IS-CL framework learns implicit symbolic representation which dynamically routes visual information into the encoded slot tokens of a Transformer encoder. To validate our hypothesis, we perform a "probing" experiment testing the **locality** of information encoded in the slots. We first pretrain the IS-CL framework with K_1 slot tokens. During finetuning, we remove a subset of the pretrained slot tokens and retain only $K_2 < K_1$ slot tokens. If the IS-CL pretraining leads to localized, implicit symbolic representations, then removing a subset of the tokens would lead to a significant drop in performance compared to the original model. Intuitively, it

Table 7: Change of slot tokens as measured on ACRE.

Finetuning with removed slots					Finetuning with all slots				
K_1	K_2	comp	sys	iid	K_1	K_2	comp	sys	iid
4	1	83.52%	73.23%	80.02%	1	1	91.75%	90.34%	90.96%
4	2	81.45%	72.17%	76.11%	2	2	90.82%	88.21%	88.73%
-	-	-	-	-	4	4	93.03%	92.36%	92.13%

should also perform worse than a model pretrained with K_2 tokens, which is pretrained to utilize a smaller set of tokens to encode implicit symbolic representation.

Results are shown in Table 7. We observe that when the slot tokens are removed, the resulting models are outperformed both by finetuning the original model (3rd row, right), and also by models pretrained with the same smaller number of slots. The observations are in line with our hypothesis.