We thank the four reviewers (**R1**-**R4**) for their constructive feedback and largely positive comments.

Improving readability. Several reviewers found the paper clear and "exceptionally well written" (**R3**), but they also noted the density of the Methods and (esp. **R1**) felt it was overly formal. We agree that this section should better get across the key ideas, so we will use the extra page allowed in revision to provide higher-level explanations and background for each concept we introduce. To further improve readability, we will also make the following changes:

★ Lines 108-120, the description of PSGs, will be made less formal. The key point is that PSGs are hierarchical graphs with additional structures that connect to visual input. Within-level edges are meant to represent physical connections; parent-to-child edges are meant to represent part-whole relationships; attribute vectors are meant to encode physical properties of elements of the visual input; and spatial registrations map each vertex in the graph to a subset of pixels in the input movie. Because a PSG is hierarchical, the registrations form a hierarchical segmentation of a scene.

★ Lines 121-133 should better address **R1**'s questions about model training. In particular, PSGNets have two types of losses that are *trained jointly*: (1) *rendering losses*, which train the parameters of Graph Vectorization and Feature Extractor modules by backpropagation, because the rendered feature maps are differentiable functions of predicted attribute vectors and image features, respectively; and (2) *perceptual grouping* losses, which train the affinity functions in Graph Pooling modules. Rendering gradients *cannot* flow back into the affinity functions because Label Propagation is not differentiable; perceptual grouping gradients *can* flow into the other modules. To clarify L258 (per **R2**), we will emphasize that depths and normals can *supervise* rendering losses (1) but are never used as input. We describe further experiments without any depths and normals supervision below. Perceptual grouping losses (2) never receive supervision. We will implement **R3**'s idea to make Fig. 1 larger by splitting it into Architecture and Training figures.

★ We can simplify the rest of the Methods as five shorter subsections: (1) ConvRNN Feature Extraction: explain that a ConvRNN, unlike a deep CNN, generates features *from a single layer* with different useful properties after each recurrent pass (see Supplement); (2) Graph Pooling: emphasize ideas of affinity functions and explain why Label Propagation prevents rendering gradients from training this module; (3) Graph Vectorization: Move details of aggregation to Supplement (e.g. L176-184, per **R1**), emphasize taking statistics over segment interiors and boundaries (to encode shape information) and predicting attributes *via* Graph Convolution [Kipf & Welling, *ICLR* 2017]; (4) Rendering: Combine the QTR, QSR, and Losses sections to explain why parameter-free decoding forces node attributes to encode scene properties *explicitly*; discuss (per **R1**) why this leads to interpretable representations, as illustrated by graph editing (Figure 4); (5) Perceptual Grouping: explain why pairwise inductive biases make sense for grouping and how β-VAEs [Higgens *et al.*, *ICLR* 2016] can naturally encode the idea of node co-occurrence and motion-in-concert.

Data to address Reviewer 1. R1 makes an excellent suggestion to compare the MultiDSprites (MDS) and CLEVR6 datasets with our custom datasets. We thus built PSGNetS-RGB, a compact version of the original model that does not use depths/normals supervision. Trained/tested on the Playroom dataset, PSGNetS-RGB achieved scores of (**recall** 0.59, **mIoU** 0.55, **boundary F** 0.57), which are (as expected) somewhat, though not dramatically, lower than the original PSGNetS scores. Without further hyperparameter tuning, PSGNetS-RGB trained/tested on MDS achieved (**r**0.80, **m**0.70, **b**0.72); trained/tested on CLEVR6, it achieved (**r**0.73, **m**0.63, **b**0.67). The higher scores on MDS and CLEVR6 suggest that the Playroom dataset is indeed harder to segment than those used in the literature. Moreover, in running these experiments we found something else worth reporting: the Playroom-trained PSGNetS-RGB achieved *zero-shot transfer performance* of (**r**0.75, **m**0.67, **b**0.71) on MDS and (**r**0.70, **m**0.60, **b**0.66) on CLEVR6, nearly as high as the within-dataset scores. This further indicates that PSGNet inductive biases lead to fairly general object-centric representation learning. We will include these points in the revision, along with MDS/CLEVR6 scores for our implementations of the baseline models (which take longer to train than the allotted Author Response period.)

Data to address Reviewers 2 & 3. The Playroom-trained PSGNetS-RGB model above ablates the three δx input channels, as per **R2**. To measure the effect of ablation, we restored these channels to create PSGNetM-RGB and trained five models with different random seeds. These achieved scores (mean +/- stdev) of (**r**0.65 +/ 0.01, **m**0.59 +/- 0.01, **b**0.60 +/- 0.01), substantially higher than PSGNetS-RGB. This indicates that the δx input channels are useful for segmentation and supports our claim of low performance variability, noted by **R3**. Each model took ~24 hours to train on one Titan Xp GPU; inference takes ~200 ms/image. PSGs are implemented on a GPU by choosing a *maximum* node number per level, then masking out nodes that have no incoming child-to-parent edges (& thus represent nothing.)

Response to Reviewer 4. R4 raised two concerns about (1) the types of edges in the PSG and (2) the inability to group by real-world textures. We want to clarify that these are *not* inherent limitations of our approach, but rather choices we made to limit the scope of the submission. As to (1), PSGs can naturally incorporate additional edge types, for example to represent temporary occlusions, collisions, or support relationships between objects. We use these edge types in ongoing work to predict physical dynamics with PSGs. As to (2), by self-supervising rendered feature maps on higher-order statistics of the input image, such as the covariance of RGB pixels within each node segment, attribute vectors can be trained to explictly encode texture. Perceptual grouping can take advantage of these texture components.