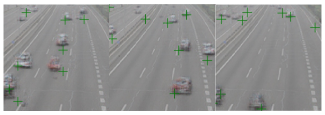
We thank the reviewers for their time and insightful feedback. Most appreciated the novelty of this work and the results.

R1.1: More realistic data. See R3.6. **R1.2: Object appearance is not independent.** Our assumption is that the appearance *latent parameters* are independent (*i.e.*, choice of shape, color, texture and reflectance), not the *rendered appearance*. Interactions such as cast shadows and reflections *can* be captured by the generator network G. While these effects are subtle, some such interactions can be observed, *e.g.*, the shadows in row 2 objects 2&4 in Fig. 4r and row 1 image 3 in Fig. A7 (supp. mat.). **R1.3: Beyond small objects.** It is possible to handle much bigger objects. The only assumption is that objects are *smaller* than the scene, not *much* smaller. This parameter can be controlled by varying H, see also Fig. A7&8. **R1.4: Beyond centroid position.** This is beyond this work, but we are testing this extension by using other composition strategies. **R1.5: More varied interactions.** See R1.2: The generator can account for complex visual interactions. **R1.6: Iterative corrections.** This is possible, but empirically for our data a single feed-forward pass through the interaction module is sufficient. **R1.7: Predicting all at once?** As the primary purpose is regularization, we chose the simpler solution that avoids challenges, such as dealing with order, count, occlusion, *etc*. **R1.8: Didn't really change only one object.** Though very small, this effect is the result of composition at feature level. It could be mitigated by increasing the latent resolution H or with a different composition strategy. **R1.9: Account for intersection.** The model pushes objects apart when appropriate according to the data statistics. In datasets such as CLEVR, the model does allow occlusions (see Fig. A6,7,8) because they occur in the data, while still preventing unrealistic cases such as objects at the same exact positions or implausible configurations. **R1.10: L.218 BlockGAN failure mode.** See R2.3. We also model interaction between *objects and background* (Eq. 1).

R2.1/2/5: Training robustness; Disentanglement: better measures (MIG.) RELATE *always* converged during training and, over 4 runs, $std(FID) < 5$ for *each* dataset of Tab. 2 except Shapestacks (7.9). A better metric than FID to measure disentanglement is possible, but computing the MIG score is not applicable in our case since our model does not feature an inference component. As an alternative, we toggle each object individually, looking at how the generated image changes. We then report the distance between the pixel location corresponding to the maximum image change and the location (scaled θ_i) of the object that was toggled. The median distance is in favor of our model: 17/19/17/23/26 *vs.* BlockGAN's 19/18/272/22/98 for CLEVR5/CLEVR/ShapeStacks/Cars/Balls. Note that Balls has a bigger offset due to bigger latent object size (Tab. A1, H'=8). **R2.3/6 Ablation study inconclusive; BlockGAN fails BallsInBowl, why?** We ran ablations of Tab. 1 *three times* with same hyper-parameters (see A2/A3 in supp. mat.) and found that *only* our model was able to converge. Since BlockGAN does not account for background-object interactions (as we do in Eq. 1, L.134), it cannot place the balls within the bowl unless these are all predicted as part of the same (background) component. Similarly 'w/o pos. reg.' shows that Γ is not sufficient: we need P to force individual objects to appear. Finally 'w/o res.' shows that addition to $\hat{\theta}_k$ forces variance of object position and stabilizes training therefore enabling better disentanglement. **R2.4: Position regularizer vs shift.** It forces the object to appear; a shift alone is not sufficient because the network can still learn to not show individual objects (see R2.3). **R2.7: Position prediction accuracy?** Using the position predictor, RELATE reconstructs position with 11.3 mean pixel errors in Shapestack, which is quite good. **R2.8: Shapestack falling towers.** Proper modelling of falling towers requires modelling the 3D rotations of blocks and their angular dynamics, which the current version of the model does not handle (see conclusions). **R2.9: Physical interpretability?** We do model physically plausible configurations of objects and background. Extending the model to explicitly reason about dynamics is a next step.

R3.1: Value of K As noted in L.140, we do *not* need to specify the value of K, but rather a distribution of values in a reasonable range. In practice, the latter can be a rough guess (see R3.6). **R3.2: Perspective scenes.** CLEVR and RealTraffic are perspective scenes (see also Fig. 5 right: objects are changing sizes when being moved). As noted in L.220, perspective is accounted for when composing with the background. **R3.3/R4.5: Processing details/code.** Most of the requested details are given in the supp. mat. A4. The dataset/code will also be publicly released. **R3.4: L.100. Background vs foreground objects.** If an object (*e.g.*, tree) is fixed, the network can account for the resulting occlusions when composing objects with the background choosing not to render an object based on its position. **R3.5: BlockGAN vs RELATE FID score.** Thanks to our correlation module and position loss, RELATE does not generate unrealistic scenes, such as two objects intersecting or objects not on the ground. It also helps the generator at training time resulting in better FID scores by preventing mode collapse (see BlockGAN ablation). **R3.6: Experiment on GRAM [1]** The GRAM datasets have at most 24k images which is generally not enough to train a GAN and get a robust FID score. Nonetheless, we trained from scratch on MH-30-HD (cropped to 512x512) with number of objects sampled in [1,8]. We obtained 117.8 FID on 400 test images and good object factoring (see figures), quality would undoubtedly improve with more data.



R4.1: Incremental. We introduce two innovations, position correlation Γ and position regularization P, which enable training of our model on more complex data than the original BlockGAN. Empirically, these modifications are *necessary* for it to converge and/or to result in proper disentanglement. **R4.2: L.220.** This line does not state that objects are rendered *in* the background layer, but that they *use* information in this layer for rendering: *i.e.*, the final object appearance depends on its position w.r.t. the background due to the perspective effect. **R4.3: non-object oriented GAN** We trained {DRA,DC}-GAN on 64x64 images. Best FID scores: 80.8/84.4/108./57.2/38.8, with dataset ordering of Tab. 2. **R4.4: Object decomposition score.** We address this issue in R2.1/2/5.