We thank all reviewers for their comments. We begin with general responses, followed by specific ones.

**R1,R3,R4,R5: More results on general images.** Our model can also be applied to less constrained images by allowing the user to specify the regular region. It also works for images where not all planes of a box exhibit regular patterns. Fig. 1 (i) shows example results. In (a), we run BPI on a user-specified region (the orange bounding box). BPI outputs a reasonable program even though the left plane is curved. We also show that our model can be applied to a broader class of images than buildings, such as the bamboo forest in (b), and the scene with irregular planes in (c).

**R3,R6: Failure cases.** Failure modes can be roughly divided into three categories, shown in Fig. 1 (ii). First, our model might misdetect vanishing points and wireframe segments, as illustrated in (d), where the model missed the wireframe between the floor and the right plane (detected wireframes shown as red lines). A second type of failure can occur when the image has a solid color plane. Illustrated in (e), our model segments the pure white part of the floor as part of the left/right planes. Finally, the inference might fail due to irregular planes, as shown in (f), where the buildings on the left do not form a rectangular plane. These issues could be mitigated with light user interaction, such as specifying wireframes. We will add more failure modes to the revised paper.

**R3,R4,R5: Contribution and baselines.** Our main contribution is to jointly model 3D structure and program-like, repeated patterns and use the inferred structure for image manipulation. We achieve high-fidelity and regularity-preserving image editing by exploiting these structures and regularities. Our model outperforms existing methods for general image editing (GatedConv) and also models relying on similar assumptions (Huang et al. [2014]).

**R1: Non-lattice patterns.** Non-lattice patterns can be handled the same way as in P3I [20]. Specifically, we can extend our DSL to include other types of patterns (e.g., circular, and radial), and apply the same inference algorithm.

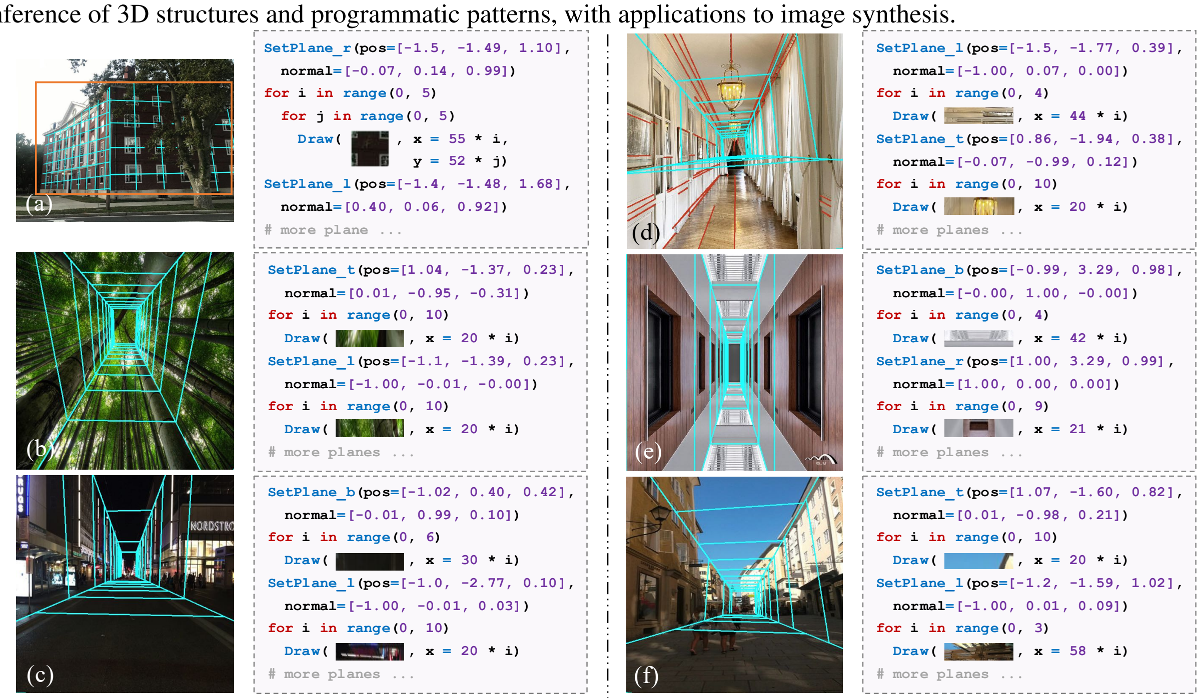
**R1: Search space and speed.** For the corridor dataset, each image contains 1,506 wireframe combinations on average. 46 programs are evaluated on each plane. The full process takes 196s. Note that BPI reduces the search space significantly based on the box prior, so that the search can be done efficiently (23× faster than without the box prior).

**R1: The inferred program for Figure 6 Row 1.** The program for this image can be found in Figure 1 of the main paper. Note that the side planes in this image are not continuous repeating textures due to the lights on the ceiling.

**R3: Camera parameters.** Thanks! We will revise our wording to "works with fixed camera intrinsics and extrinsics". It is worth noting that while it's intrinsically ambiguous to infer camera parameters from a single picture, as they are entangled with the pose of the box or the distance to the object, different sets of camera intrinsics turn out to be equivalent when it comes to image synthesis and manipulation. Even if the assumed camera parameters are different from the ground truth values of the image, it won't affect the produced pictures. We will supply a derivation in the supp.

**R4: PlaneRCNN with 3D prior.** We add an extra experiment for PlaneRCNN with the box prior. Specifically, we post-process the output of PlaneRCNN and seek mutually perpendicular planes, using the surface normals. This improves the segmentation IoU from 0.52 to 0.58, evaluated on the corridor dataset. Our result is 0.84.

**R5: Related work.** We will cite and discuss the referred paper. Unlike the suggested paper, we focus on the *joint* inference of 3D structures and programmatic patterns, with applications to image synthesis.




(i) General Cases (ii) Failure Cases

Figure 1: More image examples and the corresponding programs. We use cyan lines to visualize the lattice structure on each plane.