We thank the reviewers for the valuable feedback and address specific comments below.

Clarification of the Gibbs sampling procedure: It is not the corruption distribution itself that ultimately generates new, realistic objects; rather, it is the repeated application of the corruption and reconstruction distributions in succession. Running the corrupter repeatedly (with no reconstruction) leads to samples that severely diverge from the data distribution (see Appendix G). Intuitively, the corrupter executes a few random modifications to the current object, "nudging" it off the true data manifold (but still respecting the validity constraints). In turn, the learned reconstructor is trained to undo the corruptions, pushing it back to the data manifold. Ergodic theory states that when a Markov chain is constructed by alternatingly sampling from the conditionals corresponding to a joint distribution, e.g., $p(x \mid \tilde{x})$ and $p(\tilde{x} \mid x)$ for the joint $p(x, \tilde{x})$, asymptotic samples from the chain will be from the marginal distributions. In our approach, we leverage this theory (after Bengio et al., 2013) to construct a generative model from the combination of the fixed corrupter and learned reconstructor. We plan to expand Section 2.1 with additional explanation to make the paper more self-contained.

Model hyperparameters (sampler initialization, corruption distribution): Due to space constraints, some of the details of the model hyperparameters, including chain initialization, were included in Appendix D. For both molecules and Laman graphs, each chain is initialized with a random sample from the training set. For each of the experiments, 10 chains are run for 2K transitions each, resulting in 20K samples. Although the samples are not i.i.d., no burn-in or thinning is used. We find that the initial state has minimal effect on the resulting chain-specific KS distances. Similarly, we find (see Appendix A) that for a range of corruption distributions, the performance is not substantially affected (although extreme parameters do affect performance).

Domain-specific legal moves: The legal inductive moves approximately correspond to a domain-specific graph-rewrite grammar. Defining such moves does require some domain expertise. While a dataset is not strictly required to define the inductive moves, it can help. For the molecular domain, the moves are based on a primitive vocabulary of bonds and rings obtained via tree decomposition of the training set (similar to Jin et al., 2018). For Laman graphs, we leverage the Henneberg operations from rigidity theory without reference to data. The proposed method is applicable to domains where inductive moves can be specified that preserve some notion of validity. For example, for generating source code, grammar parse trees can be perturbed with insertions and deletions of production rules, altering samples while respecting syntactic validity. The more prior knowledge (e.g., hard constraints) that is encoded in the inductive moves, the less the model has to learn. We plan to apply the method to additional domains including constructive solid geometry and program source code in future work.

Additional experimental results (Guacamol benchmarks, SMILES LSTM): We ran the new Guacamol distribution-learning benchmarks after training our model on the ChEMBL dataset. Using the same hyperparameters as for the ZINC model, we obtain validity: 1.0, uniqueness: 0.933, novelty: 0.942, KL divergence: 0.771, FCD: 0.058. Note that these are preliminary results and the FCD score is not directly applicable to our samples. This is due to inherent autocorrelation in the generated chains which is not taken into account by the FCD computation. The autocorrelation may be addressed by standard techniques, e.g., thinning, but there was not time to evaluate this for the author response. We plan to include updated Guacamol results in the paper. We also trained a SMILES LSTM (using the referenced implementation) on the ZINC dataset. Bootstrapped mean (and std) KS distances are as follows: QED: 0.022 (0.003), SA: 0.051 (0.004), logP: 0.052 (0.004). The LSTM is effective at matching the ZINC statistics, producing a much better-matched SA distribution than the other methods. However, the LSTM has limited ability to incorporate structural constraints, e.g., enforcing the presence of a particular substructure.

Application to structured object optimization: The proposed method naturally lends itself to substructure-conditional generation ("autocomplete"), which is relevant to a host of design and engineering disciplines. For example, many classes of drugs, e.g., benzodiazepines, are defined by the presence of a core chemical substructure with some desired properties. By masking the inductive moves executed during transitions, the Markov chain can respect this hard constraint. Virtual screening then allows these samples to be efficiently searched, with optimal ones serving as candidates for additional testing.

Clarification of Figure 1: The message passing referred to in Figure 1 is described further in Appendix B and C. The attention mentioned in Figure 1 refers to the location-specific embeddings computed for the possible insertion and deletion moves. We inadvertently did not specify this and plan to update the main text accordingly.