We thank all reviewers for their constructive feedback, and provide responses to specific concerns:

[R1: Maximum likelihood (ML) VS RL objectives] The fundamental difference is between "many to one" and "one to many" settings. A ML objective is good choice for prediction tasks where the goal is to match each input to its exact output. This includes models which predict properties from molecules, since each molecule has a well-determined property. However, for the inverse problem the data severely underspecifies the mapping, since any given property combination has many diverse molecules which match; nearly all of these are *not* present in the training data. As the ML objective is only measured on the training pairs, any output that is different from the training data target is penalized. Therefore, a model which generates novel molecules with the correct properties would be penalized by ML training, as it does not produce the exact training pairs; but, it would have a high reward and thus be encouraged by an RL objective.

[R2: Plausibility of the generated molecules, GA mutation] Thanks for your insightful comments on plausibility; we include here additional results on molecule quality. Indeed, QM9 does not represent any real molecule distribution — we do note that all the molecules we show have 9 heavy atoms, consistent with the training dataset. Figure 1 shows example generations from the model trained on the ChEMBL. We then run a series of quality filters from Brown et al. [2019], which aim to detect those which are *"potentially unstable, reactive, laborious to synthesize, or simply unpleasant"*. Of our valid generated molecules, we find **71.3% pass the quality filters**, nearly the same success rate as the test set molecules themselves, **72.2%**; if we were to normalize as in Table 1 of Bradshaw et al. [2019], our performance of $\approx$ **98.7**% outperforms nearly all approaches considered. We agree that GA mutations are a more relevant way to edit molecules, but this still suffers from the same fundamental problem that local changes can have large effects on the properties.

[R3: Molecule generation baselines] Most other work for molecule generation cannot do so in one step, instead using the model as part of an iterative optimization procedure (e.g. RL or Bayesian optimization). The most competitive model we are aware of is Kang and Cho [2018], which can indeed do direct conditional generation. In Table 2 we compare our results using the ChEMBL-trained model on the task considered in their paper, generating conditioned on a single target property. Despite our model not being tailored for this task, we perform similarly well or better in terms of property accuracy, and furthermore, we do so *far faster* — their model employs a beam search decoder averaging 4.5 *seconds* per molecule, with ours requiring 6 *milliseconds*. Additionally, **their model has only 10% uniqueness** of the generated molecules, in comparison with **81% for us**; a massive increase in diversity.

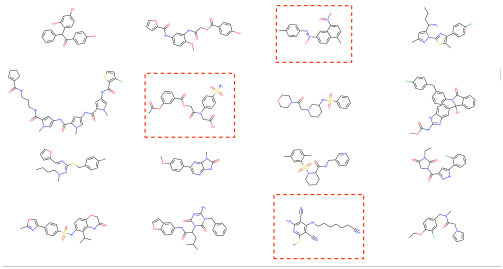

Figure 1: Example generated molecules from the ChEMBL model; red indicates failing the quality filters.

Target	Kang & Cho	Ours
MolWt=250	250.3±6.7	253.8 ±11.8
MolWt = 350	349.6±7.3	351.7 ±12.5
MolWt = 450	449.6±8.9	450.9±13.2
Logp = 1.5	1.539±0.301	1.571±0.371
Logp = 3	2.984±0.295	3.034± 0.348
Logp = 4.5	4.350±0.309	4.499±0.338
QED= 0.5	0.527±0.094	0.502±0.079
QED = 0.7	0.719±0.088	0.691 ±0.063
QED = 0.9	0.840± 0.070	0.882± 0.044

Table 2: Comparing with the the strongest baseline on conditional molecule generation task

[R2, R4: RL baseline] We actually did run an RL baseline (Eq. 2) on QM9 — we mention this in the main paper (lines 239-241), and the (generally uncompetitive) RL results are in Appendix Tables 1.4 & 1.5. As expected, the novelty of the generated molecules increased greatly as the model was able to simulate novel molecules during training, but the validity and conditional generation performance were far worse. Training such an objective even for QM9 was quite costly (6 times slower than our approach), and for ChEMBL did not converge in a reasonable time. While this might be addressable by more advanced control variate schemes, the main motivation of this paper is to take advantage of the RL objective while avoiding expensive computational costs and unstable training.

[R3, R4: Contributions, NLP] As noted conditional generation tasks often occur in NLP, but the setting is usually much more constrained: typically, the "properties" are *individual binary attributes* such as sentiment (positive or negative), or discrete styles ("romantic", "humorous", etc). Conditioning on a small number of binary labels means most methods for style transfer in NLP use auxiliary classifiers as a training signal, and are not easily adapted to larger numbers of real-valued attributes.

[R3: Relation to RAML] We discuss the work of Norouzi et al. [2016] in detail in Section 3.3. There are a number of differences, from motivation (we target an RL objective directly) to implementation (we resample potential training sequences). They also do not use the entropy term in training, only to motivate derivations.