

1 We thank all the reviewers for their careful reviews. All minor issues will be fixed in the text and more insight into
2 knowledge compilation will be given. Detailed replies follow.

3 **Common concerns.** *Novelty wrt Xu et al 2018:* We: 1) Studied the limitations of GANs in constrained settings;
4 2) Showed how CANs and the SL follow naturally from introducing constraints into the discriminator of GANs; 3)
5 Extended this setup to conditional CANs; 4) Provided a practical implementation with an effective approximate solution
6 for constraints that cannot be evaluated efficiently; 5) Evaluated the utility of the proposed techniques on representative
7 tasks of practical interest. None of these follow trivially from Xu et al. *Knowledge compilation:* Roughly speaking, KC
8 leverages distributivity to rewrite the polynomial in Eq. 9 so to speed up evaluation. This is achieved by identifying
9 shared sub-computations and compactly representing the factorized problem using a DAG. Target representations for
10 the DAG (OBDDs, DNNFs, ...) differ in succinctness and enable different polytime (in the size of the DAG) operations.
11 We reuse existing KC techniques, in particular by compiling the polynomial in Eq. 9 into a Sentential Decision Diagram
12 (SDD) that enables efficient WMC and therefore exact probabilistic computation. *SL decreases the validity of molecules:*
13 In molecule generation, validity is encouraged using the reward-based approach of MolGANs, not the SL. The SL is
14 used instead to improve diversity (using cond. constraints that encourage generating different mixtures of chemical
15 elements). As a result diversity increases (91% to 98.98%) and so does uniqueness (2.4 to 2.5). Validity drops a little
16 (97.4% to 96.6%) because the SL pushes a competing objective. This is expected. We will clarify this in the discussion.

17 **Reviewer 1.** *Unnecessary theory:* Cor. 1 and Prop. 1 serve to position CANs relative to GANs. We will trim the
18 most obvious steps. *Evaluating the SL:* The SL measures the mass allocated by P_g to configurations that violate the
19 constraints and it can be evaluated *exactly* and *efficiently* using KC. No sampling is needed. The cost of KC, which is
20 performed once before training, can be reduced by simplifying the space of binary variables using ϕ . *Link to IPMs:*
21 Thank you, we were not aware of this link. We will cite the relevant papers. *Evidence for conditional CANs:* The
22 increase in diversity in molecule generation is obtained by turning on/off generation of specific atoms (e.g., chlorine)
23 using cond. CANs. This serves as validation for the approach. We will add per-atom statistics in the Supp. Mat.
24 *Schedule of λ :* In level generation, λ is increased as explained in lines 236 and 263. In molecule generation, $\lambda = 0.9$
25 as the SL is not used for validity, so a constant value is enough in practice. Will report in the text. *5 repeats:* Good
26 point. The values we reported are the average over 5 runs for mario and 8 for molecules. The unreported std. devs. are
27 small, around 2–3% for avg. validity of 96.6%. We did not report std. devs. as we couldn't get them for CMA-ES
28 due to time constraints. We will amend this. *Relation to extra papers [1]-[3]:* Thanks for pointing out the works on
29 resampling strategies. We will mention them clarifying the differences wrt to CANs. In particular: CANs are not
30 learned using maximum likelihood, so [1] does not apply; [3] assumes absolutely continuous distributions and corrects
31 the predictions using pointwise estimates, while CANs focus on discrete probabilistic models and use the SL, which is
32 neither an estimate nor pointwise. [2] is more closely related: in a sense, CAN generators act like proposal distributions
33 for rejection sampling wrt the constraint, which is however given and fixed.

34 **Reviewer 2.** Please see the common concerns above.

35 **Reviewer 3.** *Why Eq. 9?:* Eq 9 shows that $P_g(\psi)$ is a polynomial and thus differentiable, enabling end-to-end training.
36 The negative logarithm appears in Eq. 7, we amended the text. *Relation to EBGANs:* True, constraints can be introduced
37 as energy functions into EBGANs, but they must be made differentiable first. There are many ways to do so (e.g.,
38 using fuzzy logic, as discussed in lines 133–140). However, only the SL is probabilistically sound and semantic (i.e.,
39 its value does not depend on the encoding of the constraint, see Figure 1). Also, the SL can be evaluated exactly; no
40 sampling is required. *Relation to [15]:* This work is complementary to ours as it implements a constraint learning
41 component (using inverse reinforcement learning) rather than a constraint enforcing loss term, and it is useful when
42 constraints are unknown or cannot be explicitly encoded (see our discussion in related work). Indeed we showed in the
43 molecule generation experiment how to combine the advantages of constraint learning (using reinforcement learning)
44 with constraint enforcement.

45 **Reviewer 4.** *Learning from invalid examples:* Good point. Experiments with noisy data will be carried out in a future
46 study. These are not strictly necessary to highlight the benefits of CANs. *Non-approximable constraints:* Correct,
47 some constraints are too hard for both KC and for neural net approximation. However, this issue is not specific to
48 CANs and indeed it affects all approaches to learning under constraints (generative or not). *Generated objects must
49 be filtered:* Precisely. So long as the constraints are Boolean (or discrete), checking validity of a configuration is
50 polynomial, so this is not an issue. We will clarify this. *Embedding function is unclear:* The embedding function ϕ
51 extracts Boolean features/variables to which the constraints are then applied. In many cases ϕ is the identity map. We
52 will clarify this. *More baselines:* For level generation, we compared against GAN + CMA-ES which was the SotA
53 when we ran the experiments (and until very recently). MolGAN is also a very respectable method. Still, our goal is not
54 to challenge the SotA but rather to show that CANs lead to clear improvements in many settings and for many tasks
55 (e.g., increasing validity, improving diversity). Also, we did not tweak CANs for specific domains, although this would
56 improve performance. We defer a detailed study of CANs to specific domains (especially level design) to future work.