Theoretical Principles of Controllable Generation: Reinforcing the Levenshtein Agent Mitigates the Discrepancy Problem

Anonymous Authors[1]

Abstract

Controllable generation is essentially a multi-objective optimization problem that aims to minimize *"the discrepancy"* between the likelihood and reward objectives. In this paper, we demonstrate theoretical principles of controllable generation such that 1) the objective mismatch in RL and the latent hole in VI can be integrated into a single discrepancy problem (by *Equivalence Theorem*), 2) data perturbation mitigates the discrepancy problem (by *Data Pre-Processing Inequality*), and 3) optimizing MLMs approximates perturbation effects (by *Pseudo-Likelihood Estimation*). Based on these principles, we propose a novel controllable sequence generation method called Levenshetin Agent (LEVA), which does not rely on large pre-trained models, by formulating the controllable generation as an editing process of sequences. The proposed method is evaluated on text style transfer and drug discovery tasks, without relying on large pre-trained models, and the result demonstrates that the proposed method is highly competitive.

1. Introduction

In the midst of a renaissance in artificial intelligence research, the topic of controllable generation is receiving unprecedented attention. Research on controllable generation becomes increasingly important as controllability has recently emerged as a key concept for solving real-world problems such as recommendation (Wang et al., 2020), autonomous driving (Han et al., 2016), and drug discovery (Guimaraes et al., 2017) tasks controlled "as intended."

Reinforcement learning (RL) is one of the main pillars that is probably most related to controllability. Considering large pre-trained models as agents, fine-tuning them with specific reward functions has become a common approach in the study of controllable generation. In this context, controllable generation is essentially a multi-objective optimization problem aiming to find a Pareto solution that simultaneously satisfies multiple objectives, e.g., the likelihood function f_1 and the reward function f_2. A Pareto solution is obtained from the feasible region where the different objectives overlap with each other (see Figure 1), which is why previous studies have mostly relied on large pre-trained models; large models trained on massive datasets can estimate an objective function large enough to cover different objectives. However, we often encounter the cases that the development of a large pre-trained model is challenging. In addition, relying on pre-trained models limits their extension to field data in that pre-training can be done with benchmark datasets in specific domains, e.g., text and image datasets, significantly reducing the applicability to real-world problems; for example, we cannot develop an artificial melody generator by leveraging the capabilities of a model pre-trained on the Wikipedia dataset. To obtain a Pareto solution without relying on large pre-trained models, the feasible region of multi-objectives must be well-specified (see Figure 1(b)). In other words, *"the discrepancy"* between the likelihood distribution and the reward distribution should be minimized over the latent space where each axis g_1 and g_2 represents the simultaneous direction of multi-objectives (see Figure 1(c)).

The discrepancy problem has been separately discovered in several fields and discussed in their respective contexts. In the context of model-based RL, it is highlighted that *"the objective mismatch"* arises from de-coupled optimization in which training an environment model does not strongly correlate with reward maximization, leading the agent to a biased trajectory (Lambert et al., 2020; Eysenbach et al., 2021; Nikishin et al., 2022). That is, the objective mismatch refers to the situation where the likelihood does not match with the reward function. Similarly, in the context of NLP, a typical example of the objective mismatch is a mismatch between the loss function used in training and the metric considered for evaluation (a.k.a. the loss-evaluation mismatch). It happens when the loss function is optimized at the token level, while the evaluation at the sentence level (Shen et al., 2015; Wiseman & Rush, 2016; Elbayad et al., 2018). On the other hand, some studies in the line of variational inference (VI) proposed the hypothesis that a hole is created in the latent space when estimating the posterior distribution, which is called *"the latent hole problem."* The point is that the prior does not match or overlap with the

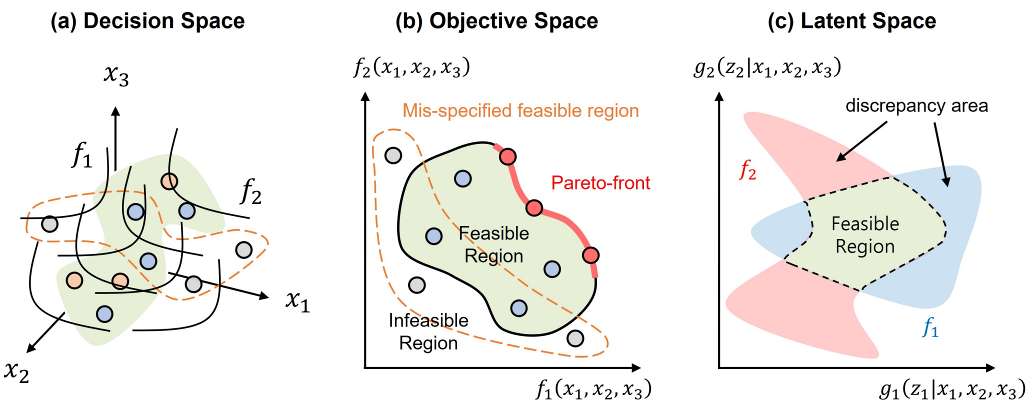

Figure 1. Illustration of the objective mismatch in three domains: (a) variable domain (b) functional domain (c) latent variable domain

aggregated posterior (Rezende & Viola, 2018; Xu et al., 2020; Kalatzis et al., 2020; Aneja et al., 2021).

The two challenges, *the objective mismatch* and *the latent hole problem*, seem to be the same problem or at least related. Intuitively, maximizing the reward under the dynamics of the environment is analogous to estimating the aggregated posterior that obeys the prior. However, we were not sure if the two challenges are actually related. If we find the relation between them, we can better understand the discrepancy problem accompanied by controllable generation. Accordingly, this study aims to analyze their relation in theoretical perspectives and proposes a novel method for the controllable sequence generation[1] that does not rely on large pre-trained generative models. The proposed method is evaluated on the text style transfer task with a benchmark dataset and on the drug discovery task with sample data from a public chemical information database. The result demonstrates that the proposed method is highly competitive. The academic and practical contributions of our work are summarized as follows:

- Controllable generation is defined as a problem of mitigating the discrepancy between the likelihood objective and additional objective such as reward.
- A theoretical analysis is provided, which integrates the objective mismatch and the latent hole problem into a single discrepancy problem.
- Based on the theoretical understanding, a novel method for controllable sequence generation, called LEVenshtein Agent (LEVA), is proposed.
- The capability of the proposed method is evaluated on text style transfer and drug discovery tasks, without relying on large pre-trained models, thereby demonstrating its applicability.

2. Theoretical Analysis

In this section, we provide the theoretical background of our study in detail. Specifically, the goal of this section is to uncover the principles of controllable generation. In this context, the section splits into three parts verifying that: (1) *VI is essentially equivalent to RL in the context of generative model*, (2) *the discrepancy, e.g., among objectives / between prior and posterior, originates from the decrease in mutual information*, and (3) *data pre-processing, e.g., perturbation / augmentation, may increase the model's control capabilities*.

2.1. Preliminaries

Reinforcement learning (RL) is a learning framework that addresses stochastic optimization. Under the RL framework, an agent learns a policy $\pi_\theta(\cdot|s)$ to maximize rewards through sequential and repeated interactions with the environment. The sequence of interactions changes depending on a reward function $R(s, a)$ and the agent is controlled according to this change. When developing deep policy networks, the policy gradient (PG) method is preferred due to its several useful properties, e.g., fast convergence, low bias, stochastic policy, continuous action (Sutton & Barto, 2018):

$$\nabla_\theta J_{\text{PG}}(\theta) = \mathop{\mathbb{E}}_{a\sim\pi_\theta}\left[R(s, a)\nabla_\theta \ln \pi_\theta(a|s)\right] . \qquad (1)$$

Variational inference (VI) is a method widely used to approximate posterior densities. i.e., a conditional density of latent variables given observed variables (Blei et al., 2017). The key idea behind VI is: the inference problem turns

[1] We focus on the sequence generation task for two reasons: (1) Many types of field data have sequence formats, e.g., texts, stock prices, DNA, driving paths, and decision making processes; considering this format increases the generation model's applicability to real-world problems. (2) Sequence generation is challenging with its discrete nature; considering this nature can extend the usable scenarios of the generation model.

into an optimization problem that minimizes the Kullback-Leibler divergence ($\mathcal{D}_{\text{KL}}$) between the parameterized approximate density $q_\theta(z|x)$ and exact posterior $p(z|x)$. However, $\mathcal{D}_{\text{KL}}\left[q_\theta(z|x)||p(z|x)\right]$ is intractable because the evidence $\log p(x)$ is not computable. To address this issue, the primal problem was rewritten into its dual form maximizing the evidence lower bound (ELBO). VI has been successfully introduced as a promising technique for deep generative models (Kingma & Welling, 2013):

$$\underset{\theta,\phi}{\arg\max} \underset{z\sim q_\theta(z|x)}{\mathbb{E}}\left[\ln p_\phi(x|z)\right] - \mathcal{D}_{\text{KL}}\left[q_\theta(z|x)||p(z)\right], \tag{2}$$

where the first term is the reconstruction accuracy and the second term is a penalty on the distributional divergence between $q_\theta(z|x)$ and the prior $p(z)$.

2.2. Equivalence Theorem

We present a theorem that VI is essentially equivalent to RL in the context of the generative model. Specifically, the prior distribution in VI plays a role of reward function in RL, and a single generative model $G_\theta(x)$ becomes an agent trained to approximate a policy that maximizes the ELBO.

Theorem 1. *Let a single generative model $G_\theta(x)$ consist of two parts, i.e., encoder $q_{\theta'}(z|x)$ and decoder $p_{\theta''}(x|z)$, with the aggregated parameter $\theta = \{\theta^{'}, \theta^{''}\}$. Then, the total derivative of its ELBO w.r.t. θ is equivalent to the policy gradient including free dynamics $z \sim p(z)$ as reward.*[2]

Theorem 1 assumes that the stochastic process of the generative model can be decomposed into the encoding and decoding processes. We have separated the two stochastic processes for analytical convenience, but since optimizing a single model is our concern, it is trivial that both processes should be considered simultaneously. Accordingly, if we take the total derivative of the ELBO *w.r.t.* θ, then the total gradient is derived as follows:

$$\frac{\partial G_\theta}{\partial \theta'} + \frac{\partial G_\theta}{\partial \theta''} = \underset{x\sim G_\theta}{\mathbb{E}}\left[\Big(\ln p(z)\Big)\nabla_\theta \ln G_\theta(x)\right] \tag{3}$$

where $\ln p(z)$ is given as the reward function with $p(z)$ as the prior. This provides an intuitive explanation of the equivalence between RL and VI. For example, sampling z in a low-density region, i.e., the smaller the value of $p(z)$, means an agent receives the less amount of reward, i.e., the smaller the value of $\ln p(z)$. See Appendix B.1 for the proof of Theorem 1.

2.3. Decrease in Mutual Information

In the equivalence between RL and VI, we can now expect that *the objective mismatch* and *the latent hole problem* arose for the same reason. It is obvious that the reward (or the prior) $\ln p(z)$ accounts for the additional learning directions that the generative model should adhere to besides the data distribution $p_d(x)$. This seems to create a learning bottleneck, leading to the discrepancy problem. To analyze the origin of such discrepancy, we paid attention to the second term of Equation (2), i.e., $\mathcal{D}_{\text{KL}}\left[q_\theta(z|x)||p(z)\right]$, and discovered that the discrepancy originates from the decrease in mutual information $\mathcal{I}(x;z)$. Specifically, maximizing the ELBO implies that the second term should converge to zero, and by Lemma 2, the second term is reduced to $\mathcal{I}(x;z) - \mathcal{D}_{\text{KL}}\left[p(z)||q(z)\right]$. We do not need to consider $-\mathcal{D}_{\text{KL}}\left[p(z)||q(z)\right]$ because it is only about z so does not have any clues for the discrepancy between $p_d(x)$ and $p(z)$. As a result, when the ELBO is maximized, $\mathcal{I}(x;z)$ disappears giving rise to the discrepancy problem. See Appendix B.2 for the proof of Lemma 2.

Lemma 2 (Hoffman & Johnson (2016))**.** *Given the Kullback-Leibler divergence between the conditional and the marginal distributions of a random variable z, i.e., $\mathcal{D}_{KL}\left[p(z|x)||p(z)\right]$, marginalizing it by $p(x)$ leads to $\mathcal{I}(x;z) - \mathcal{D}_{KL}\left[p(z)||q(z)\right]$.*

2.4. Data Pre-processing Inequality

As demonstrated in the previous section, the discrepancy problem arises when the mutual information goes to zero. This suggests that we can partially prevent the discrepancy problem by keeping the mutual information larger than zero. The mutual information can be rewritten as follows (see Appendix C.1 for the derivation):

$$\mathcal{I}(x;z) = \mathcal{H}(x) - \mathcal{H}(x|z) \tag{4}$$

where $\mathcal{H}(x)$ is the marginal information of data and $\mathcal{H}(x|z)$ is the conditional information of data given the additional learning directions. Note that we cannot address $\mathcal{H}(x|z)$ because for controllable generation it is what the generative model $G_\theta(x)$ is supposed to estimate as a result of the optimization. Accordingly, it is only $\mathcal{H}(x)$ that is of our interest, and we present a theorem related to it. The key idea of Theorem 3 is to introduce an arbitrary distribution $u(\cdot)$ over x and to utilize the non-negativity of Kullback-Leibler divergence (line 1 of the proof). Then, we can obtain information inequality between empirical and arbitrary distributions (line 2-3 of the proof). However, the use of arbitrary distribution over empirical distribution is counter-intuitive, so we relaxed the arbitrary distribution as an empirical distribution over processed data x' by reparameterization trick (line 4 of the proof). As a result, Theorem 3 states information before and after data pre-processing,

[2] Although we referred to $p(z)$ as "free dynamics" as it is described in the field of control theory, it is conceptually the same with "the prior" in VI, except that none of the specific distribution is assumed.

e.g., perturbation/augmentation, holds an inequality (line 5 of the proof); we call it as *data pre-processing inequality* (DPPI).[3]

Theorem 3. *Over data space $x \in \mathcal{X}$, empirical distributions $p_d(x)$ always contain less information than arbitrary distributions $u(x)$ iff $p_d(x) \neq u(x)$.*

Proof.

$$\begin{aligned} &\mathcal{D}_{\mathrm{KL}}\left[p_d(x)||u(x)\right] = \sum_x p_d(x) \ln \frac{p_d(x)}{u(x)} \geq 0 \\ &\Leftrightarrow -\sum_x p_d(x) \ln p_d(x) \leq -\sum_x p_d(x) \ln u(x) \\ &\Leftrightarrow \mathop{\mathbb{E}}_{x \sim p_d}\left[-\ln p_d(x)\right] < \mathop{\mathbb{E}}_{x \sim p_d}\left[-\ln u(x)\right] \\ &\qquad\qquad (\because p_d(x) \neq u(x)) \\ &\Leftrightarrow \mathcal{H}(x) < \mathop{\mathbb{E}}_{x \sim p_d}\left[-\ln p_d(x')\right] \\ &\qquad\qquad (\because x' = x + u,\ u \sim \sigma(u)) \\ &\Leftrightarrow \mathcal{H}(x) < \mathcal{H}(x')\ . \end{aligned}$$

Assume one of the most catastrophic situations where $p(z)$ is exclusively independent of both $p_d(x)$ and $p_d(x')$ so that G_θ cannot capture any dependencies between them, i.e., $\mathcal{H}(x|z) = \mathcal{H}(x'|z) = 0$. In this situation, the pre-processed data x' provide more information than the original data x and $\mathcal{I}(x';z)$ is always larger than zero (see Corollary 4 for the proof). Consequently, when trained on pre-processed data, the generative model is more likely to perform better with controllable generation as it becomes robust to the discrepancy problem. Meanwhile, simple augmentation does not increase $\mathcal{I}(x';z)$, but rather converges it to zero because $\mathcal{H}(x') \to 0$ (see Theorem 5 in Appendix B.3), hence we chose perturbation, not augmentation, as a pre-processing technique. Note that keeping the mutual information positive by perturbation does not violate the ELBO maximization (see Theorem 6 in Appendix B.4)

Corollary 4. *If $\mathcal{H}(x|z) = \mathcal{H}(x'|z) = 0$, then $\mathcal{I}(x';z) > 0$ always holds.*

Proof. $\mathcal{I}(x;z) \geq 0$ and $\mathcal{H}(x) < \mathcal{H}(x')$ is trivial. Then ,

$$\mathcal{H}(x') - 0 > \mathcal{H}(x) - 0 \geq \mathcal{H}(x) - \mathcal{H}(x|z) \geq 0\ .$$

$$\therefore\ \mathcal{I}(x';z) > \mathcal{I}(x;z) \geq 0$$

2.5. Pseudo-Likelihood Estimation

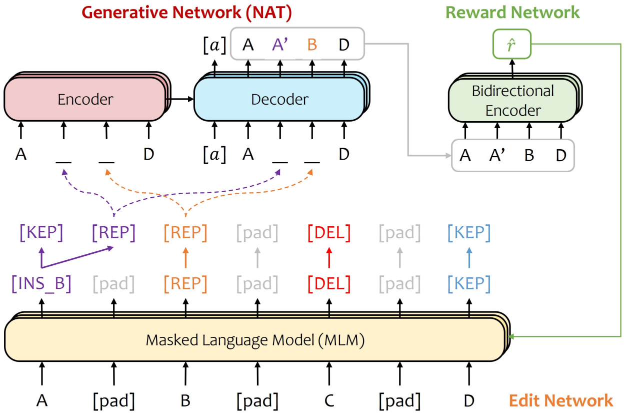
Generative Network (NAT)
Reward Network
Encoder
Decoder
Bidirectional Encoder
$[a]$ A A' B D
$\hat{r}$
A _ _ D
$[a]$ A _ _ D
A A' B D
[KEP] [REP] [REP] [pad] [DEL] [pad] [KEP]
[INS_B] [pad] [REP] [pad] [DEL] [pad] [KEP]
Masked Language Model (MLM)
A [pad] B [pad] C [pad] D
Edit Network

Figure 2. Framework of LEVenshtein Agent (LEVA). The proposed method performs an editing-based controllable generation, consisting of the generative, reward, and edit networks. The generative and reward networks are pre-trained and fixed as the environment to which the edit network interacts as an agent.

Given the perturbation data, it is necessary to define the model with commensurate capabilities on it. To this end, we decided to use the pseudo-likelihood (PL) estimator as a loss function. The PL estimator was first introduced by Besag (1975) to consider the conditional densities of a random variable given the others, $p_\theta(x_i|x_1,...,x_{i-1},x_{i+1},...,x_n) = p_\theta(x_i|x_{-i})$. It is well known that the gradient computation of pseudo-log-likelihood (PLL) approximates the score matching with the Langevin Monte Calro method (Hyvarinen, 2007):

$$\nabla_\theta \mathcal{L}_{\mathrm{PLL}}(\theta) = \mathop{\mathbb{E}}_{x \sim p_d}\left[\nabla_\theta \ln p_\theta(x)\right] - \mathop{\mathbb{E}}_{x' \sim p_{x'}}\left[\nabla_\theta \ln p_\theta\left(x'\right)\right]$$

where a perturbation x' is introduced such that only the i-th feature is slightly changed keeping all the others fixed, $x' = [x_i + \Delta|x_{-i}]$. Hence, we can endow our model with the capability to address perturbation effects by optimizing PLL. Since the objective of masked language models (MLMs), like BERT (Devlin et al., 2018), can be viewed as stochastic maximum pseudo-likelihood estimation (Wang & Cho, 2019; Salazar et al., 2020), we can internalize the perturbation effect by modeling MLMs. That is, the controllable generation is reduced to the problem of finding the proper positions of tokens to be masked and infilling the masked positions with substitute tokens.

3. Proposed Method

We demonstrated that 1) the objective mismatch in RL and the latent hole in VI can be integrated into a single discrepancy problem (by *Equivalence Theorem*), 2) data perturbation mitigates the discrepancy problem (by *Data Pre-Processing Inequality*), and 3) optimizing MLMs approximates perturbation effects (by *Pseudo-Likelihood Estimation*). Then, how can we implement controllable generation with these findings?

Controllable generation can be implemented by reinforcing the generative model according to reward maximization. Since *Equivalence Theorem* presents reward maximization

[3] Do not confuse it with data processing inequality (aka DPI).

as equivalent to ELBO maximization, What we need to do is to maximize ELBO while encoding $q(z|x)$ to adhere to the reward distribution $p(z)$. The simplest way of achieving this strategy is to include reward-related labels and encode them together with data. In our case, we added the attribute token a at the prefix of sequences, which encodes the empirical reward distribution $p_d(a)$, so that the latent space is disentangled to encode the embeddings of likelihood z_x and reward z_a at the same time. However, adding a does not guarantee the generative model to be controlled because $\mathcal{I}(x, [z_x; z_a])$ goes to zero thereby z_a will disappear. According to *Data Pre-processing Inequality* and *Pseudo-Likelihood Estimation*, such discrepancy can be mitigated by optimizing MLMs, so controllable generation should be handled as a learning process of masking some parts of the sample and infilling them with reward-enhancing substitutes.

3.1. Problem Formulation

To address controllable generation by the MLMs' manner, we formulate the controllable sequence generation as an editing process of sequences: deciding whether to keep, delete, insert, or replace a token of specific position t, which is possibly important for control. The editing process is defined by three steps. First, select an operation $o_t \in \mathcal{O}$ that measures the Levenshtein distance (Levenshtein et al., 1966) between an original sequence $x = [x_1, x_2, ..., x_T]$ and an edited sequence $x' = [\hat{x}_t|x_{-t}] = [x_1, .., \hat{x}_t, .., x_T]$. Note that we denote the edit operations, i.e., keep, delete, insert, and replace, by [KEP], [DEL], [INS] and [REP], respectively.[4] Second, confirm which token to mask based on selected operations; only the token with [REP] operation is regarded as “to be masked”. Lastly, the masked tokens, denoted as blank “____”, are filled with candidate tokens $\hat{x}_t$ sampled from the conditional distribution $p(\hat{x}_t|x_{-t}, o_t, a)$. Here, the attribute a indicates a special token that represents the code vector related to reward information. The editing process is realized through the agent's interaction with the environment such that the agent selects an edit operation (blank-setting) and the environment returns a substitute token (blank-infilling).

3.2. Environment Model

The environment consists of two models: the generative model and the reward model. These two models are pre-trained. To avoid any misunderstanding that we use large pre-trained models, please note that our environment models are pre-trained with the available field data for the specific controllable generation problem.

As shown in Figure 2, the generative model p_θ is built based on non-autoregessive transformers (NATs) (Gu et al., 2017; Kasai et al., 2020). The encoder takes an original sequence only, but the decoder receives the original sequence prefixed with an attribute token. The loss function of the generative model is as follows,

$$\mathcal{L}_{\text{gen}}(\theta) = \mathop{\mathbb{E}}_{\substack{o_t \sim p_\psi \\ x \sim p_d}} [\ln p_\theta(\hat{x}_t|x_{-t}, o_t, \bar{a})] \tag{5}$$

where $\bar{a}$ denotes the target attribute; for example, in the sentiment style transfer task, if a is a sentiment token of positive attribute, then $\bar{a}$ should be a negative sentiment token. During pre-training, the blank positions are given by Poisson distribution, i.e., $t \sim \text{Pois}(\lambda)$, and the model is trained to fill them up with the original tokens (see Appendix A for the details). It is worth noting that Equation (5) denotes the blanks as being sampled by the agent model, i.e., $o_t \sim p_\psi$, which is the case of inference. We can express it as the case of pre-training by replacing p_ϕ and o_t with $\text{Pois}(\lambda)$ and t, respectively.

On the other hand, the reward model p_ϕ is defined as a single bidirectional encoder. The structure is equal to the encoder part of Transformer (Vaswani et al., 2017) except that the linear layer to predict rewards is added on the top head of the encoder. The loss function is defined as,

$$\mathcal{L}_{\text{reward}}(\phi) = \mathop{\mathbb{E}}_{x, a \sim p_d} [\ln p_\phi(\hat{r} = a|x)] \tag{6}$$

where x and a are the input sequence and the attribute token, respectively. It is then pre-trained to estimate the probability that a given sequence has a specific attribute.

3.3. Agent Model

The agent model p_ψ is trained to find some tokens related to reward information. It selects an operation to be performed at each token. For example, if an agent decides to execute an ”insert front” operation ([INS_F]) for one token, that token is to “keep” and its front token is masked as a “replace”able candidate. The agent model is implemented based on the simple MLM strucuture and trained by the following PG method,

$$\begin{aligned}
&\nabla_\psi \mathcal{L}_{\text{edit}}(\psi) \\
&= \mathop{\mathbb{E}}_{\substack{o_t \sim p_\psi \\ x, \bar{a} \sim p_d}} \Big[\hat{r} \nabla_\psi \big(\ln p_\psi(o_t|x_t) - \eta \mathcal{D}_{\text{KL}}[p_{\text{unif}}(o_t)||p_\psi(o_t|x_t)]\big)\Big] \\
&= \mathop{\mathbb{E}}_{\substack{o_t \sim p_\psi \\ x, \bar{a} \sim p_d}} \Big[\hat{r} \big(\nabla_\psi \ln p_\psi(o_t|x_t) + \eta p_{\text{unif}}(o_t) \nabla_\psi \ln p_\psi(o_t|x_t)\big)\Big] \\
&= \mathop{\mathbb{E}}_{\substack{o_t \sim p_\psi \\ x, \bar{a} \sim p_d}} [\hat{r}\,(1 + \eta p_{\text{unif}}(o_t))\, \nabla_\psi \ln p_\psi(o_t|x_t)]
\end{aligned} \tag{7}$$

where $\hat{r} = p_\phi(\hat{r} = \bar{a}|x')$ is the estimated reward for target attribute $\bar{a} \sim p_d$ and the edited sequence $x' = [\hat{x}_t|x_{-t}]$ is given by $\hat{x}_t \sim p_\theta$. Note that we added $\mathcal{D}_{\text{KL}}[p_{\text{unif}}(o_t)||p_\psi(o)]$

[4] We defined three subclasses of insert operation: insert back [INS_B], insert front [INS_F], and insert all [INS_A].

Table 1. Comparative results on ACC, BLEU and PPL. Red-colored values indicate the best score and orange-colored values indicate the second-best score at each metric.

Class	Model		ACC ↑	BLEU ↑	PPL ↓
Non-RL	*CA* (Shen et al., 2017)		0.71	54.06	8.10
Non-RL	*UNMT* (Zhang et al., 2018)		**0.92**	74.34	8.20
RL	*CycleRL* (Xu et al., 2018)		0.50	62.65	8.98
RL	*DualRL* (Luo et al., 2019)		0.84	**81.32**	8.30
Non-RL	*DRG*	*B-GST*	0.83	72.86	**7.97**
	(*Sudhakar et al., 2019*)	*G-GST*	0.75	73.07	8.58
Non-RL	*StyleTransformer*	*Cond-Dis*	0.84	77.78	9.94
	(*Dai et al., 2019*)	*Multi-Dis*	0.79	**83.49**	8.08
RL	*DIRR*	*Direct*	0.85	79.93	8.01
	(*Liu et al., 2020*)	*Cycle*	0.82	75.90	8.49
Non-RL	*DeepLatent* (He et al., 2020)		0.80	68.07	8.24
Non-RL	*Tag & Generate* (Madaan et al., 2020)		0.84	71.60	8.32
Non-RL	*LEWIS* (Reid & Zhong, 2021)		0.81	76.34	8.32
RL	*LEVA (ours)*		**0.86**	69.55	**7.70**

term to penalize the agent p_ψ falling into a local optimum such that all the operations are selected to be [KEP]. $p_{\text{unif}}(\cdot)$ is the probability density function of $U(0, 1)$, and $1 + \eta p_{\text{unif}}(o_t)$ is obtained as an auxiliary reward term. Auxiliary reward forces the policy gradient updated such that the uniform operations are preferred for the exploration of the agent. To prevent the exploration from breaking the training stability, we suggest setting η to a small value (e.g., 0.0005).

3.4. Reward Shaping

Providing the attribute reward $\hat{r}$ alone can degenerate the naturalness of generated sequences. Therefore, it is necessary to design rewards more elaborately to keep the generated sequences natural. As an additional reward, we introduced the [KEP] ratio out of selected edit operations for a given sequence,

$$\hat{r} = p_\phi(\hat{r} = \bar{a} | x') + \frac{1}{T} \sum_{t=0}^{T} \mathbf{1}(o_t = [\text{KEP}]) \ ,$$

where $\mathbf{1}(\cdot)$ is an indicator function that returns 1 only if the operation for the t-th token is selected as [KEP]. By doing so, the agent model p_ψ is reinforced to edit only parts of the sequence to keep it natural and preserve the content of the sequence.

4. Experiment

4.1. Data and Baselines

We evaluate the performance of the proposed method for text style transfer, a well-known benchmark task for controllable sequence generation. Specifically, the experiment was run on the Yelp review dataset, where the training, validation, and test sets are approximately 440k, 4k, and 1k in size, respectively. Reviews with a length of less than 4 have been removed. We compared the performance with 10 baselines. For fair comparison, the baselines were selected to include both Non-RL and RL-based approaches. Especially, Tag & Generate (Madaan et al., 2020) and LEWIS (Reid & Zhong, 2021) are comparable to our method in that they are editing-based methods.

Table 2. Examples of controlled generation provided by the top-3 methods. In spite of a relatively low BLEU score, LEVA seems to successfully control the style of sequence while preserving the content.

Control	**negative** → **positive**
Original	*this place is a terrible place to live !*
UNMT	*this place is definitely fantastic place to live !*
Multi-Dis	*this place is a great place to live !*
DIRR-CA	*this place is a great place to live !*
LEVA (ours)	*this place is a great place to live !*
Control	**positive** → **negative**
Original	*great place to have some fresh and delicious donuts .*
UNMT	*_num_ minutes to have some crap and cold walmart .*
Multi-Dis	*no stars to have some fresh and cold donuts .*
DIRR-CA	*terrible place to have some fresh and bland donuts .*
LEVA (ours)	*worst place to have some coffee and expensive donuts .*

4.2. Metric

For the evaluation metric, we used ACC, BLEU (Papineni et al., 2002) and perplexity (PPL) (Gamallo et al., 2017). ACC stands for the control accuracy, which indicates *"how well the style is controlled."* Typically, ACC is measured using a pre-trained attribute predictor. According to this practice, we also measured ACC using the pre-trained BERT (Sanh et al., 2019) fine-tuned on Yelp data.[5] BLEU is a metric based on N-gram statistics. It computes the ratio of tokens overlapped between the reference and generated sequences, meaning *"how well the content is preserved"* during the generative process. The reference can be either an original sequence in which a style attribute is not transferred (self-BLEU) or a gold sequence in which a style attribute is transferred by a human (gold-BLEU). We used the original sequence as a reference and computed the BLEU score by the nltk package.[6] Lastly, PPL, a metric to quantify *"how natural the generation is."*, is defined as the geometric mean of token likelihoods. It measures how naturally tokens are aligned within a given sequence. Similar to ACC, the PPL score was evaluated using the pre-trained GPT-2 (Radford et al., 2019) that is fine-tuned on Yelp training and validation set with the 5e-3 learning rate.

5. Results

5.1. Performance Evaluation

We compared the proposed method with four RL-based baselines and eight non-RL baselines. Table 1 shows the

[5] https://huggingface.co/ydshieh/bert-base-uncased-yelp-polarity

[6] https://www.nltk.org/_modules/nltk/translate/bleu_score.html

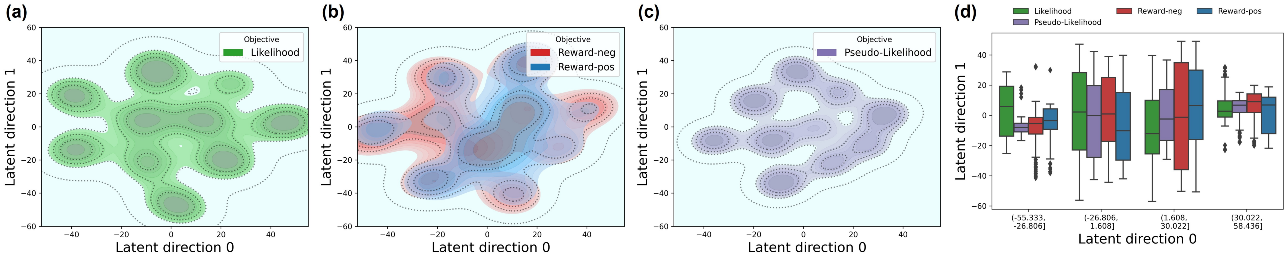

Figure 3. Difference in latent sample distribution by objective functions: (a) likelihood objective, (b) reward objective, (c) pseudo-likelihood objective, (d) a range of occupied areas of latent distribution by interval. For discrepancy mitigation, the latent distribution of the likelihood objective must overlap with the latent distribution of the reward objective. Figure (a) describes the right-downward distribution, whereas figure (c) shows the right-upward distribution that overlaps more with the distribution of figure (b). Note that figure (d) confirms such an overlap trend: The more the bar line of the (pseudo-) likelihood objective is located between two bar lines of the reward objectives, the more the (pseudo-) likelihood objective overlaps with the reward objective in the latent space.

evaluation result on three metrics. It is notable that the proposed method achieved 1st and 2nd top performance in PPL and ACC, respectively. This implies that the proposed method can generate a natural-looking sequence while controlling its style as intended, suggesting LEVA has mitigated the discrepancy between likelihood and reward objectives. In contrast, the proposed method showed relatively low performance with respect to BLEU. This result has left us somewhat puzzled because our method, an editing-based approach, was expected to better preserve contents than the baselines which are of high variance due to the autoregressive LMs. To understand the reason for the relatively low BLEU score, we compared the generated results of baselines. As a result, the baseline that achieved high BLEU and low ACC was the case where the original sequence was copied. On the other hand, if both BLEU and ACC were high, the style transfer was successful, or the model cheated on trick such that original sentence was copied and then one word was added. In summary, the relatively low BLEU score of the proposed model does not appear to be a major problem.

5.2. Evidence of Discrepancy Mitigation

In the previous section, we proved that the proposed method can control the style attributes of sentences. However, it is still unclear whether the proposed method is working as expected from a theoretical perspective. In other words, we need to make sure if the objective mismatch is actually mitigated. To find the evidence that the objective mismatch problem was alleviated in our experiment, we checked how much the sample distribution of the two objective functions overlapped in the latent space. Figure 3(a)-(c) describes the distribution of the sample in the latent space according to the likelihood, reward, and pseudo-likelihood objective, respectively. To obtain Figure 3(a) and (c), we first input both complete and masked sequences into LEVA and retrieved the corresponding latent samples. Then, the latent samples of each sequence are mapped into the respective 2-d space using the T-SNE reduction (Van der Maaten & Hinton, 2008). Similarly, Figure 3(b) was obtained by mapping the positive and negative sequences into the 2-d identical single space. Figure 3(d) shows that the latent distribution of the pseudo-likelihood objective overlaps more with the reward function than with the likelihood objective, meaning that the latent overlap between the likelihood function and the reward function has increased. Therefore, we can conclude that there exists a clear mitigation of the likelihood-reward discrepancy.

6. Applicable Scenarios: Drug generation for repositioning

LEVA does not rely on large pre-trained language models. Thus, it can be applied to controllable sequence generation tasks in any domain. Therefore, we explored its applicability in the field of drug discovery by using LEVA as a molecular editor. Specifically, based on successful cases of drug repositioning (see Table 3), we conducted a pilot study to generate candidate drugs by controlling base compounds to have an effective molecular structure for target diseases. Figure 4 illustrates the LEVA customized for molecular editing. The protein encoder embeds protein sequences related to a particular disease, while the compound decoder is trained to generate compound sequences of the candidate drug for the target disease. To this end, the drug-disease pairs of repositioned indications are required as in Table 3. We manually collected the data from PubChem DB[7] (See Table 4 in Appendix D for the descriptive statistics). Similar to the text style transfer case, the attribute tokens of the original indication are replaced by those of the secondary indication during inference. That is, the molecular editor generates a new candidate compound targeting a specific indication given as an attribute token, leveraging both the protein se-

[7] https://pubchem.ncbi.nlm.nih.gov/

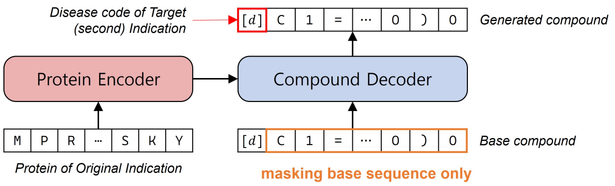

Figure 4. Single-sided masking of LEVA for drug discovery. Everything is the same with Figure 2 except that the encoder and decoder receive the different sequences, i.e., target proteins and drug compounds, and only the sequence of the decoder is masked. Note that the attribute token $[d]$ denotes an indication label of target disease.

Table 3. Successful examples of drug repositioning. The original indication refers to the target disease for which the drug was initially designed, and the secondary indication refers to the actual disease for which the drug is repositioned.

Drug Name	Original Indication	Secondary Indication
Allopurinol	Neoplasm	Parasitic Disease
Doxycycline	Bacterial Infection and Mycoses	Stomatognathic Disease
Mazindol	Stomatognathic Disease	Congenital, Hereditary and Neonatal Disease
Topiramate	Nervous System Disease	Stomatognathic Disease
Valproic acid	Nervous System Disease	Congenital, Hereditary and Neonatal Disease

quence of its original indication and the corresponding base compound sequence. Figure 5 illustrates the base compound and the synthetic compound edited by LEVA for the sake of drug repositioning.

7. Discussion

Despite LEVA outperforming most of the baselines, we observed three issues that are worth being discussed. The first issue is *the offset effect of greedy sampling*. Let's say we have a sentence *"the server was not kind."*, and both *"not"* and *"kind"* are masked. What would happen if we input the masked sentence into the generative model for converting it to a positive sentence? The greedy sampling may select *"very"* for *"not"* and *"unkind"* for *"kind"* because each selection alone will change the sentiment style for sure. As a result, we obtain the new sentence *"the server was very unkind."*, and it is the opposite result of what is expected. The second issue is *the hallucinatory infilling*. The generative model fills in the blanks based on unmasked tokens, and mostly the infilled token is determined by the neighbor tokens around the blank. In other words, the neighbor tokens of the blank are the context that gives a hint for blank-infilling. Accordingly, the blank-infilling is vulnerable to consecutive masked tokens, i.e., span-masking, thus creating the hallucinatory context to address span-masking. The third issue is *the spurious rewards.* When the reward model estimates a reward of a sequence, it does not care whether the sequence is natural-looking or whether tokens are aligned in a possible way. It only cares about the reward estimation itself, so it spuriously predicts a high reward for the sequence that does not make sense. Such spurious reward signals are critical to the edit model. We suggest these three issues for future research topics of controllable sequence generation, especially when using an editing-based approach.

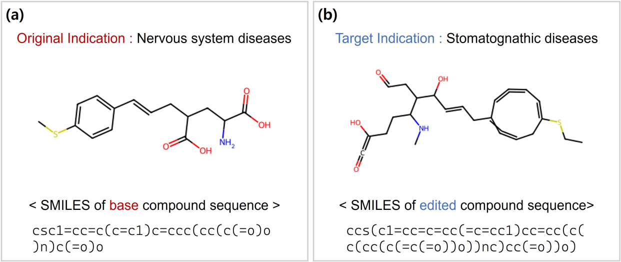
(a)
Original Indication : Nervous system diseases
S
O
OH
O
OH
NH2
< SMILES of base compound sequence >
csc1=cc=c(c=c1)c=ccc(cc(c(=o)o
)n)c(=o)o
(b)
Target Indication : Stomatognathic diseases
O
OH
HO
NH
C
O
S
< SMILES of edited compound sequence>
ccs(c1=cc=c=cc(=c=cc1)cc=cc(c(
c(cc(c(=c(=o))o))nc)cc(=o))o)

Figure 5. Result of drug generation for repositioning. (a) is the base compound effective for the original indication. (b) is the synthetic compound generated by LEVA, expected to be effective for the target indication.

8. Concluding Remarks

In this study, we regarded the controllable generation task as a multi-objective optimization problem, and tried to answer the fundamental question, *"How to mitigate the discrepancy between two different objectives?"* To mitigate the discrepancy, the mutual information between the likelihood and reward distributions must be larger than zero, and we achieved it using a novel method called LEVA. Meanwhile, the necessary and sufficient conditions of recent NLG models in the spotlight, such as ChatGPT, can be summarized as (1) utilizing pre-trained large language models (LLMs); (2) applying reinforcement learning (RL) to precisely control the LLMs according to specific objectives, e.g., suppression of hate speech. However, achieving the condition is very challenging in that the meeting of LLMs with RL triggers astronomical training costs unaffordable in a general research environment. Moreover, it is not possible to prepare a large model pre-trained on all types of sequence data, hence recent NLG models are applicable only to the text domain. In this context, our model can be the best solution. Since LEVA is an MLM that does not rely on large pre-trained models, its training efficiency is comparatively higher than a typical LLM and its applicability is limitless.

References

Aneja, J., Schwing, A., Kautz, J., and Vahdat, A. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing* *systems*, 34:480–493, 2021.

Besag, J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195, 1975.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Dai, N., Liang, J., Qiu, X., and Huang, X. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621*, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Elbayad, M., Besacier, L., and Verbeek, J. Token-level and sequence-level loss smoothing for rnn language models. *arXiv preprint arXiv:1805.05062*, 2018.

Eysenbach, B., Khazatsky, A., Levine, S., and Salakhutdinov, R. Mismatched no more: Joint model-policy optimization for model-based rl. *arXiv preprint arXiv:2110.02758*, 2021.

Gamallo, P., Campos, J. R. P., and Alegria, I. A perplexity-based method for similar languages discrimination. In *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects (VarDial)*, pp. 109–114, 2017.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C., and Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

Han, M., Senellart, P., Bressan, S., and Wu, H. Routing an autonomous taxi with reinforcement learning. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 2421–2424, 2016.

He, J., Wang, X., Neubig, G., and Berg-Kirkpatrick, T. A probabilistic formulation of unsupervised text style transfer. *arXiv preprint arXiv:2002.03912*, 2020.

Hoffman, M. D. and Johnson, M. J. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

Hyvarinen, A. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on neural networks*, 18(5):1529–1531, 2007.

Kalatzis, D., Eklund, D., Arvanitidis, G., and Hauberg, S. Variational autoencoders with riemannian brownian motion priors. In *International Conference on Machine Learning*, pp. 5053–5066. PMLR, 2020.

Kasai, J., Cross, J., Ghazvininejad, M., and Gu, J. Non-autoregressive machine translation with disentangled context transformer. In *International conference on machine learning*, pp. 5144–5155. PMLR, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lambert, N., Amos, B., Yadan, O., and Calandra, R. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

Levenshtein, V. I. et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pp. 707–710. Soviet Union, 1966.

Liu, Y., Neubig, G., and Wieting, J. On learning text style transfer with direct rewards. *arXiv preprint arXiv:2010.12771*, 2020.

Luo, F., Li, P., Zhou, J., Yang, P., Chang, B., Sui, Z., and Sun, X. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*, 2019.

Madaan, A., Setlur, A., Parekh, T., Poczos, B., Neubig, G., Yang, Y., Salakhutdinov, R., Black, A. W., and Prabhumoye, S. Politeness transfer: A tag and generate approach. *arXiv preprint arXiv:2004.14257*, 2020.

Nikishin, E., Abachi, R., Agarwal, R., and Bacon, P.-L. Control-oriented model-based reinforcement learning with implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7886–7894, 2022.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Reid, M. and Zhong, V. Lewis: Levenshtein editing for unsupervised text style transfer. *arXiv preprint arXiv:2105.08206*, 2021.

Rezende, D. J. and Viola, F. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Salazar, J., Liang, D., Nguyen, T. Q., and Kirchhoff, K. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2699–2712, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.240. URL https://aclanthology.org/2020.acl-main.240.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*, 2015.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems*, 30, 2017.

Sudhakar, A., Upadhyay, B., and Maheswaran, A. Transforming delete, retrieve, generate approach for controlled text style transfer. *arXiv preprint arXiv:1908.09368*, 2019.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, A. and Cho, K. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2304. URL https://aclanthology.org/W19-2304.

Wang, P., Fan, Y., Xia, L., Zhao, W. X., Niu, S., and Huang, J. Kerl: A knowledge-guided reinforcement learning model for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 209–218, 2020.

Wiseman, S. and Rush, A. M. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016.

Xu, J., Sun, X., Zeng, Q., Ren, X., Zhang, X., Wang, H., and Li, W. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. *arXiv preprint arXiv:1805.05181*, 2018.

Xu, P., Cheung, J. C. K., and Cao, Y. On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning*, pp. 10534–10543. PMLR, 2020.

Zhang, Z., Ren, S., Liu, S., Wang, J., Chen, P., Li, M., Zhou, M., and Chen, E. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018.

A. Pre-trained Models

As described in Section 3.2, the two components of the environment, i.e., the generative model p_θ and the reward model p_ϕ, are pre-trained. Since the pre-training of the reward model is defined the same as its training stage, here we only address the pre-training of the generative model which has a different scheme between the pre-training and the training stages.

The generative model is defined as a masked language model to fill in the blanks, the positions t of masked tokens x_t, provided by the edit model. Accordingly, the generative model must be pre-trained to predict which tokens to locate at masked positions. To do this, we need to provide the generative model with some randomly masked tokens and have it predict the label tokens of masked positions based on unmasked tokens x_{-t} and the true attribute a,

$$\mathcal{L}_{\text{pre-gen}}(\theta) = \mathop{\mathbb{E}}_{\substack{k\sim\text{Pois}(\lambda)\\ t\sim\text{Unif}[0,L;\ n]\\ x\sim p_d}} \left[\ln p_\theta(\hat{x}_t = x_t | x_{-t}, a)\right].$$

Specifically, the random masking was determined by the Poisson distribution $k \sim \text{Pois}(\lambda)$ with λ as an average sequence length $\times$ 0.15 $\left(= \frac{0.15}{N}\sum_{i=1}^{N} L_i\right)$ and by the uniform distribution $t \sim \text{Unif}[0, L;\ k]$ with k trials; given the number of masked tokens k is determined by the Poisson distribution, then the positions of masked tokens t are determined by the uniform distribution within the range from initial position 0 to end position L. In addition, we implemented two different masking strategies: token masking and span masking. Token masking is simple masking that allows the discontinuous masked positions, while span masking forces the masking over the consecutive tokens. For example, let us assume that the number of and the positions of masked tokens are given as $n = 2$ and $t = 2, 7$, respectively. In this case, token masking makes the second and the seventh tokens, i.e., x_2, x_7, blank by masking over them, but span masking puts the masks on six consecutive tokens from the second to seventh positions, i.e., $x_2, x_3, x_4, x_5, x_6, x_7$.

B. Proofs

B.1. Proof of Theorem 1

This section aims to prove that reward maximization is equivalent to ELBO maximization if the prior distribution is regarded as the reward distribution $z \sim p(z)$. Let us first define a single model $G_\theta(\cdot)$ whose the encoder $q_{\theta'}$ predicts the latent variables z of as similar distribution as the prior distribution and the decoder $p_{\theta''}$ generates the outputs $\hat{x}$ that reconstruct the original inputs x from the latent variables. On the lens of variational inference, encoder parameters θ' can be estimated by minimizing the divergence between the empirical posterior $q_{\theta'}(z|x)$ and the ground-truth posterior distribution $p(z|x)$,

$$\mathcal{D}_{\text{KL}}\left[q_{\theta'}(z|x)||p(z|x)\right] = \mathop{\mathbb{E}}_{z\sim q_{\theta'}}\left[\ln\left(\frac{q_{\theta'}(z|x)}{p_{\theta''}(x|z)}\right) - \ln p(z)\right] + \ln p(x). \tag{8}$$

However, it is impossible to calculate $p(z|x)$ because we do not know any moment or shape of $p(z|x)$ and even the relevant data are not given. Accordingly, we need to convert the Equation (8) into the form that is easier to calculate as follows:

$$\begin{aligned}
\ln p(x) &= \mathop{\mathbb{E}}_{z\sim q_{\theta'}}\left[\ln p(x,z) - \ln q_{\theta'}(z|x)\right] + \mathcal{D}_{\text{KL}}\left[q_{\theta'}(z|x)||p(z|x)\right] \\
&\geq \mathop{\mathbb{E}}_{z\sim q_{\theta'}}\left[\ln p(x,z) - \ln q_{\theta'}(z|x)\right] = \mathop{\mathbb{E}}_{z\sim q_{\theta'}}\left[\ln p(x|z) + \ln p(z) - \ln q_{\theta'}(z|x)\right] \\
&= \mathop{\mathbb{E}}_{z\sim q_{\theta'}}\left[\ln p_{\theta''}(x|z)\right] - \mathcal{D}_{\text{KL}}\left[q_{\theta'}(z|x)||p(z)\right]
\end{aligned} \tag{9}$$

which is what we call the Evidence Lower BOund (ELBO). Note that maximizing $\ln p(x)$ is the dual problem of minimizing $\mathcal{D}_{\text{KL}}[q_{\theta'}||p(z|x)]$. Meanwhile, we defined a "single" generative model $G_\theta(\cdot)$ that consists of $q_{\theta'}$ and $p_{\theta''}$. Accordingly, we have to optimize both θ' and θ'' at the same time and the equation below defines the simultaneous optimization of ELBO:

$$\mathop{\arg\max}_{\theta',\ \theta''}\ \mathcal{L}(\theta', \theta'') = \mathop{\mathbb{E}}_{z\sim q_{\theta'}}\left[\ln p_{\theta''}(x|z)\right] - \mathcal{D}_{\text{KL}}\left[q_{\theta'}(z|x)||p(z)\right]. \tag{10}$$

If we want to find an optimal solution x^* under two objectives, what we need to do is 1) taking a partial derivative of each objective with respect to x, 2) summing them up into a single total derivative, and 3) setting the total derivative equal to zero. Likewise, an optimal parameter θ^* can be obtained by taking a partial derivative of encoder $q_{\theta'}$ and decoder $p_{\theta''}$ with respect to θ' and θ'', respectively. Specifically, the partial derivatives of $\mathcal{L}(\theta', \theta'')$ are derived as the two identities: one for encoder $q_{\theta'}(z|x)$,

$$
\begin{aligned}
\nabla_{\theta'} \mathcal{L}\left(\theta', \theta''\right) &= \nabla_{\theta'} \left(\int_z q_{\theta'}(z|x) \ln p_{\theta''}(x|z) - q_{\theta'}(z|x) \ln q_{\theta'}(z|x) + q_{\theta'}(z|x) \ln p(z)\; dz \right) \\
&= \int_z \nabla_{\theta'} q_{\theta'}(z|x) \ln p_{\theta''}(x|z) - \nabla_{\theta'} q_{\theta'}(z|x) \ln q_{\theta'}(z|x) - q_{\theta'}(z|x) \nabla_{\theta'} \ln q_{\theta'}(z|x) + \nabla_{\theta'} q_{\theta'}(z|x) \ln p(z)\; dz \\
&= \int_z \nabla_{\theta'} q_{\theta'}(z|x) \Big(\ln p_{\theta''}(x|z) - \ln q_{\theta'}(z|x) - 1 + \ln p(z) \Big)\; dz \\
&= \int_z q_{\theta'}(z|x) \nabla_{\theta'} \ln q_{\theta'}(z|x) \left(\ln p_{\theta''}(x|z) - 1 - \left(\ln \frac{q_{\theta'}(z|x)}{p(z)} \right) \right)\; dz \\
&= \mathop{\mathbb{E}}_{z \sim q_{\theta'}} \left[\nabla_{\theta'} \ln q_{\theta'}(z|x) \left(\ln p_{\theta''}(x|z) - 1 - \left(\ln \frac{q_{\theta'}(z|x)}{p(z)} \right) \right) \right], \qquad (11)
\end{aligned}
$$

and the other for decoder $p_{\theta''}(x|z)$,

$$
\begin{aligned}
\nabla_{\theta''} \mathcal{L}\left(\theta', \theta''\right) &= \nabla_{\theta''} \left(\int_z q_{\theta'}(z|x) \ln p_{\theta''}(x|z) - q_{\theta'}(z|x) \ln q_{\theta'}(z|x) + q_{\theta'}(z|x) \ln p(z)\; dz \right) \\
&= \int_z q_{\theta'}(z|x) \nabla_{\theta''} \ln p_{\theta''}(x|z) dz \\
&= \mathop{\mathbb{E}}_{z \sim q_{\theta'}} \left[\nabla_{\theta''} \ln p_{\theta''}(x|z) \right]. \qquad (12)
\end{aligned}
$$

When a single model $G_\theta(x) \overset{\text{set}}{=} q_{\theta'}(z|x) = p_{\theta''}(x|z)$ and a single parameter $\theta \overset{\text{set}}{=} \theta' = \theta''$ are assumed, the total derivative is obtained by the sum of partial derivatives as follows:

$$
\begin{aligned}
\nabla_{\theta'} \mathcal{L} + \nabla_{\theta''} \mathcal{L} &= \mathop{\mathbb{E}}_{z \sim q_{\theta'}} \left[\nabla_{\theta'} \ln q_{\theta'}(z|x) \left(\ln p_{\theta''}(x|z) - 1 - \left(\ln \frac{q_{\theta'}(z|x)}{p(z)} \right) \right) \right] + \mathop{\mathbb{E}}_{z \sim q_{\theta'}} \left[\nabla_{\theta''} \ln p_{\theta''}(x|z) \right] \\
&= \mathop{\mathbb{E}}_{x \sim G_\theta} \left[\nabla_\theta \ln G_\theta(x) \Big(\Big(\ln G_\theta(x) - 1 - \ln G_\theta(x) + \ln p(z) \Big) + 1 \Big) \right] \\
&= \mathop{\mathbb{E}}_{x \sim G_\theta} \left[\Big(\ln p(z) \Big) \nabla_\theta \ln G_\theta(x) \right]. \qquad (13)
\end{aligned}
$$

The final line of Equation (13) is equivalent to the formula of policy gradient such that G_θ is the policy π_θ and $\ln p(z)$ is the reward function $R(\cdot)$. That is, maximizing ELBO with respect to θ within a single model framework leads to the policy gradient that maximizes the reward $\ln p(z)$ defined by the prior distribution $z \sim p(z)$. This implies that we can use ELBO maximizer as a surrogate objective when implementing or analyzing the policy-based RL.

B.2. Proof of Lemma 2

Hoffman & Johnson (2016) rewrote ELBO in various ways to highlight the role of encoded data distribution $z \sim q(z)$ and proved that to improve priors is as much important as to improve variational bounds. Specifically, they focused on the second term of ELBO (see Equation (10)),

$$
\mathcal{D}_{\text{KL}} \left[p(z|x) || p(z) \right] = \int_z p(z|x) \ln \left(\frac{p(z|x)}{p(z)} \right) dz,
$$

and marginalize it with respect to the data distribution $x \sim p(x)$. As a result, the second term of ELBO is rewritten as

$$
\begin{aligned}
\mathop{\mathbb{E}}_{x\sim p_d}\left[\mathcal{D}_{\text{KL}}\left[p(z|x)||p(z)\right]\right] &= \int_x p(x) \int_z p(z|x) \ln\left(\frac{p(z|x)}{p(z)}\right) dzdx \\
&= \int_x p(x) \int_z p(z|x) \ln\left(\frac{p(z|x)}{q(z)} \times \frac{q(z)}{p(z)}\right) dzdx \\
&= \int_x p(x) \int_z p(z|x) \ln \frac{p(z|x)}{q(z)} dxdz + \int_x p(x) \int_z p(z|x) \ln \frac{q(z)}{p(z)} dzdx \\
&= \int_{x,z} p(x,z) \ln \frac{p(x,z)}{p(x)\, q(z)} dxdz + \int_z p(z) \ln \frac{q(z)}{p(z)} dz \quad \left(\because\ p(z) = \int_x p(x)p(z|x)dx\right) \\
&= \mathcal{I}(x;z) - \mathcal{D}_{\text{KL}}\left[p(z)||q(z)\right] \\
&= \mathcal{I}(x;z) - p^{\text{avg}}(z) + \mathop{\mathbb{E}}_{z\sim p(z)}[q(z)] \ \le\ \mathcal{I}(x;z) + \mathop{\mathbb{E}}_{z\sim p(z)}[q(z)] \\
&\approx \mathcal{I}(x;z) \qquad (\because\ \text{variational inference does not explicitly model } q(z))\ ,
\end{aligned}
$$

where $\mathcal{I}(x;z)$ is the mutual information between $p(x)$ and $p(z)$, and $\mathbb{E}_{z\sim p(z)}\left[p(z)\right] = \mathbb{E}_{z\sim q(z|x)}\left[\left(\frac{p(z)}{q(z|x)}\right) q(z)\right]$ is an importance sampling that scores how likely the encoder distribution $q(z)$ follows the prior distribution $p(z)$ when z is sampled from the conditional encoder distribution $q(z|x)$. The importance sampling part can be ignored because $q(z)$ is not explicitly modeled by variational inference. Accordingly, to maximize ELBO, i.e., to minimize the second term of ELBO, it is trivial $\mathcal{I}(x;z)$ must be zero.

B.3. Proof of Theorem 3

In our definition, perturbation is a process of distorting a part of each sample, while augmentation is a process of adding new samples according to the generative model G_θ that approximates empirical data distribution p_d. A new data point x' can be obtained from either perturbation $x' = x + u,\ u \sim p(u)$ or augmentation by a generative model $x' \sim G_\theta(x)$. Without loss of generality, we can state that $G_\theta(x)$ approximates both $p_d(x)$ and $p_d(x')$ allowing for the commutative property: $G_\theta(x) \approx p_d(x)$, $G_\theta(x) \approx p_d(x')$, therefore $p_d(x) = p_d(x')$. Then, given that empirical data distribution $p_d(x)$ is set to $1/N$ with infinite augmentation $N \to \infty$, marginal information $\mathcal{H}(x')$ after data augmentation always converges to zero.

Theorem 5. *If $p_d(x) \overset{set}{=} \frac{1}{N}$, then $\lim\limits_{N\to\infty} \mathcal{H}(x') \to 0$ always holds.*

Proof. $\mathcal{H}(x') = -\sum_{x'} p_d(x') \ln p_d(x') \approx \sum_{x} G_\theta(x) \ln G_\theta(x) \quad (\because\ G_\theta(x) \approx p_d(x'))$ □

$$
\begin{aligned}
&\approx \sum_x p_d(x) \ln p_d(x) = -\frac{\ln N}{N} \quad \left(\because\ G_\theta(x) \approx p_d(x) \ \text{ and } \ p_d(x) = \frac{1}{N}\right) \\
&\Rightarrow \lim_{N\to\infty} -\frac{\ln N}{N} \longrightarrow 0\ .
\end{aligned}
$$

B.4. Proof of Theorem 4

As stated in Section 2, mutual information is minimized to zero due to ELBO maximization and we proposed to use data perturbation to keep the mutual information larger than zero. Although we proved that perturbation guarantees a positive value of mutual information, it raises the question of whether perturbation violates the ELBO maximization or not. As demonstrated in Lemma 2, the mutual information is obtained by marginalizing the second term of ELBO w.r.t. data distribution $x \sim p(x)$,

$$
\mathop{\mathbb{E}}_{x\sim p_d}[\mathcal{D}_{\text{KL}}[p(z|x)||p(z)]] = \int_x p(x)\mathcal{D}_{\text{KL}}[p(z|x)||p(z)]\ dx \approx \mathcal{I}(x;z)\ . \tag{14}
$$

Accordingly, we can prove the non-existence of violation by confirming that Equation (14) is preserved with $p(x)$ defined over the perturbed data distribution. Since we defined perturbation as $x' = x + u$, the perturbed data distribution can be modeled as,

$$p(x'|x,u) = \frac{p(x',x,u)}{p(x,u)} = \frac{p(x',x,u)}{p(x)p(u|x)} \;.$$

This can be rearranged with respect to $p(x)$ which is defined over perturbed data distribution $x' \sim p(x'|x,u)$,

$$p(x) = \frac{p(x',x,u)}{p(x'|x,u)p(u|x)} = \frac{p(x'|x,u)p(x,u)}{p(x'|x,u)p(u|x)} = \frac{p(x,u)}{p(u|x)} \;.$$

where two $p(x'|x,u)$ terms are erased out. As a result, we can implicitly marginalize ELBO w.r.t. $p(x'|x,u)$ by replacing $p(x)$ with $p(x,u)/p(u|x)$ in Equation (14). Meanwhile, it is trivial that data distribution $p(x)$ and (arbitrary) perturbing distribution $p(u)$ are mutually independent, i.e., $p(x,u) = p(x)p(u)$. With the leverage of implicit perturbation and mutual independence, we can prove that considering perturbed data distribution does not make any changes, that is, does not violate ELBO maximization as shown below.

Theorem 6. *Suppose $p(x'|x,u)$ as perturbed data distribution, and $p(x)$ and $p(u)$ are mutually independent, then $\mathbb{E}_{x\sim p_d}[\mathcal{D}_{KL}[p(z|x)||p(z)]]$ is always preserved.*

Proof.

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim p_d}[D_{\mathrm{KL}}[p(z|x)||p(z)]] &= \int_x p(x)\mathcal{D}_{\mathrm{KL}}[p(z|x)||p(z)]\;dx \\
&= \int_x \frac{p(x',x,u)}{p(x'|x,u)p(u|x)}\mathcal{D}_{\mathrm{KL}}[p(z|x)||p(z)]\;dx \\
&\qquad\qquad \left(\because \text{ implicit perturbation : } p(x) = \frac{p(x',x,u)}{p(x'|x,u)p(u|x)}\right) \\
&= \int_x \frac{p(x,u)}{p(u|x)}\mathcal{D}_{\mathrm{KL}}[p(z|x)||p(z)]\;dx \\
&\qquad\qquad \left(\because \text{ Bayes rule : } \frac{p(x',x,u)}{p(x'|x,u)p(u|x)} = \frac{p(x,u)}{p(u|x)}\right) \\
&= \int_x \frac{p(x)p(u)}{p(u)}\mathcal{D}_{\mathrm{KL}}[p(z|x)||p(z)]\;dx \\
&\qquad\qquad (\because \text{ mutual independence : } p(x,u) = p(x)p(u)\;) \\
&= \int_x p(x)\mathcal{D}_{\mathrm{KL}}[p(z|x)||p(z)]\;dx \\
&= \mathop{\mathbb{E}}_{x\sim p_d}[D_{\mathrm{KL}}[p(z|x)||p(z)]]
\end{aligned}$$

□

C. Derivations

C.1. Derivation of Equation (4)

$$\begin{aligned}
\mathcal{I}(x;z) &= \int_{x,z} p(x,z)\ln\frac{p(x,z)}{p(x)\;q(z)}dxdz = \int_{x,z} p(z)\;p(x|z)\ln p(x|z)dxdz - \int_{x,z} p(x,z)\ln p(x)dxdz \\
&= \int_x p(x|z)\ln p(x|z)dx - \int_x p(x)\ln p(x)dx \\
&= \mathcal{H}(x) - \mathcal{H}(x|z)
\end{aligned}$$

D. Application Study

The table below describes the data statistics used to train LEVA in the drug generation scenario. They were manually collected from PubChem DB. The total number of drug-disease pairs is 27k, and the average sequence lengths of proteins and compounds are up to 1.3k and 120, respectively. As shown in the table, there are a total of six indications and the

Table 4. Descriptive statistics of disease-drug-target dataset.

Disease Name	Count	Label	Avg Sequence Length	
			Drug Compound	Target Protein
Bacterial Infection and Mycoses	12	0	117.7	168.0
Congenital, Hereditary and Neonatal Disease	4948	1	53.7	488.0
Neoplasm	644	2	42.6	1333.0
Nervous System Disease	1540	3	48.7	899.2
Parasitic Disease	3	4	38.0	220.7
Stomatognathic Disease	20570	5	53.3	339.0

samples are highly imbalanced. For example, the samples of the stomatognathic disease account for almost 74% of all samples. Because of such a high imbalance in data distribution, the drugs are generated successfully only when the target indication was provided by stomatognathic disease. This is why figure 5 in the main body of the paper describes the case when stomatognathic disease was given for the target indication.