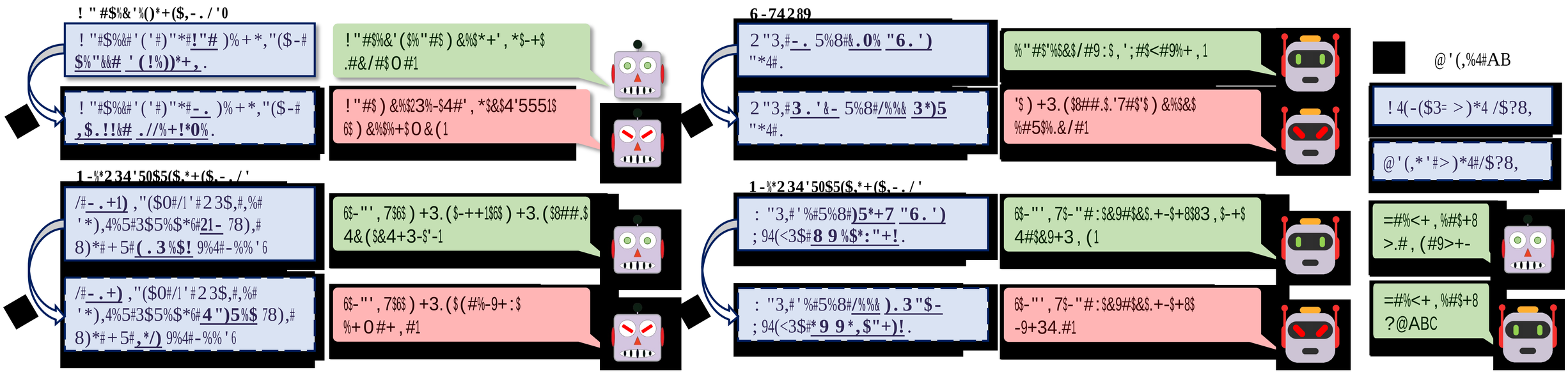
Query-E□cient Black-Box Red Teaming via Bayesian Optimization

Deokjae Lee[1], JunYeong Lee[1], Jung-Woo Ha[2,3], Jin-Hwa Kim[1,2,3], Sang-Woo Lee[2,3,4], Hwaran Lee[2,3], Hyun Oh Song[1]

[1]Seoul National University [2]NAVER AI Lab [3]NAVER Cloud [4]KAIST
{bdbj,hyunoh}@mllab.snu.ac.kr

TL;DR

We propose **Bayesian red teaming** which discovers model failures by choosing or editing user inputs with GP surrogate models.



What is Red Teaming?

- The primary goal of **red teaming** is to identify many diverse positive test cases which lead to model failures.
  For open-domain dialogue task, red teaming aims to discover a set of input utterances that lead to o□ensive responses of the chat-bot.
  For text-to-image generation task, red teaming aims to discover a set of input prompts that generate NSFW images.

Notation

- $G_\theta : U \to O$ is a **victim model** that generates an **output** $o \in O$ for a given **user input** $u \in U$. We assume that G_θ is black-box.
- $R_\varphi : U \times O \to [-1, 1]$ is a **red team classifier** that computes the **red team score** $R_\varphi(u, o)$ representing the o□ensiveness of the output o given a user input u.
- $T \subset U$ is a set of test cases.
- $T^+ := \{t \in T \mid R_\varphi(t, G_\theta(t)) > 0\}$ is a set of positive test cases.
- $B(X)$ is the **-ball** of X, the set of all possible user inputs generated using at most edit operations starting from user inputs in $X \subset U$.
- Self-BLEU$^{(k)}(V) := \mathrm{E}_{W \sim \mathrm{Unif}(\binom{V}{k})}[\text{Self-BLEU}(W)]$ is a modified Self-BLEU metric that measures the **diversity** of a text set.

Bayesian Red Teaming (BRT)

- Black-box red teaming aims to identify many diverse positive test cases in a limited **query budget** N_Q.
- The black-box red-teaming problem can be formulated as

$$\underset{T \subset U}{\text{maximize}} \ |T^+| = \sum_{t \in T} \mathbb{1}[R_\varphi(t, G_\theta(t)) > 0] \quad (1)$$
$$\text{subject to } |T| = N_Q,$$
$$\text{Self-BLEU}^{(k)}(T^+) < D.$$

- We reformulate Eq (1) into the sequence of relaxed optimization problems to construct the test case set $T = \{t_1, \cdots, t_{N_Q}\}$ in a sequential manner:

$$t_{n+1} = \underset{u \in U \setminus T_n}{\text{argmax}} \ L_\lambda(u; T_n) := \underbrace{R_\varphi(u, G_\theta(u))}_{f(u):\text{black-box}} - \lambda \underbrace{\text{Self-BLEU}^{(k)}(\{u\} \cup T_n^+)}_{g(u;T_n):\text{white-box}},$$
grey-box objective

where $T_n = \{t_1, \ldots, t_n\}$ is the set of test cases selected in previous steps.

- For e□ciency, standard BRT (BRT (s)) searches the test case on an existing user input pool $\hat{U}$, *e.g.*, utterances from dialogue datasets or utterances zero-shot generated by LM.
- BRT (s) first evaluates random user inputs for exploration, then repeats the following steps:
  1. Fit GP parameters given evaluation history $D = \{(t_i, f(t_i))\}_{i=1}^n$.
  2. Compute the expected improvement of L_λ based on the posterior.
  3. Evaluate the maximizer $t_{n+1} \in \hat{U}$ of the acquisition function and append the pair $(t_{n+1}, f(t_{n+1}))$ to the evaluation history.
  4. Update the white-box terms $\{g(u; T_{n+1})\}_{u \in \hat{U}}$.

Edit-Based BRT

- Edit-Based BRT (BRT (e)) extends the search space to the -ball of $\hat{U}$, denoted by $B(\hat{U})$.
- Since BRT (e) has a substantially larger search space, it includes an additional GP surrogate model for e□cient exploration.

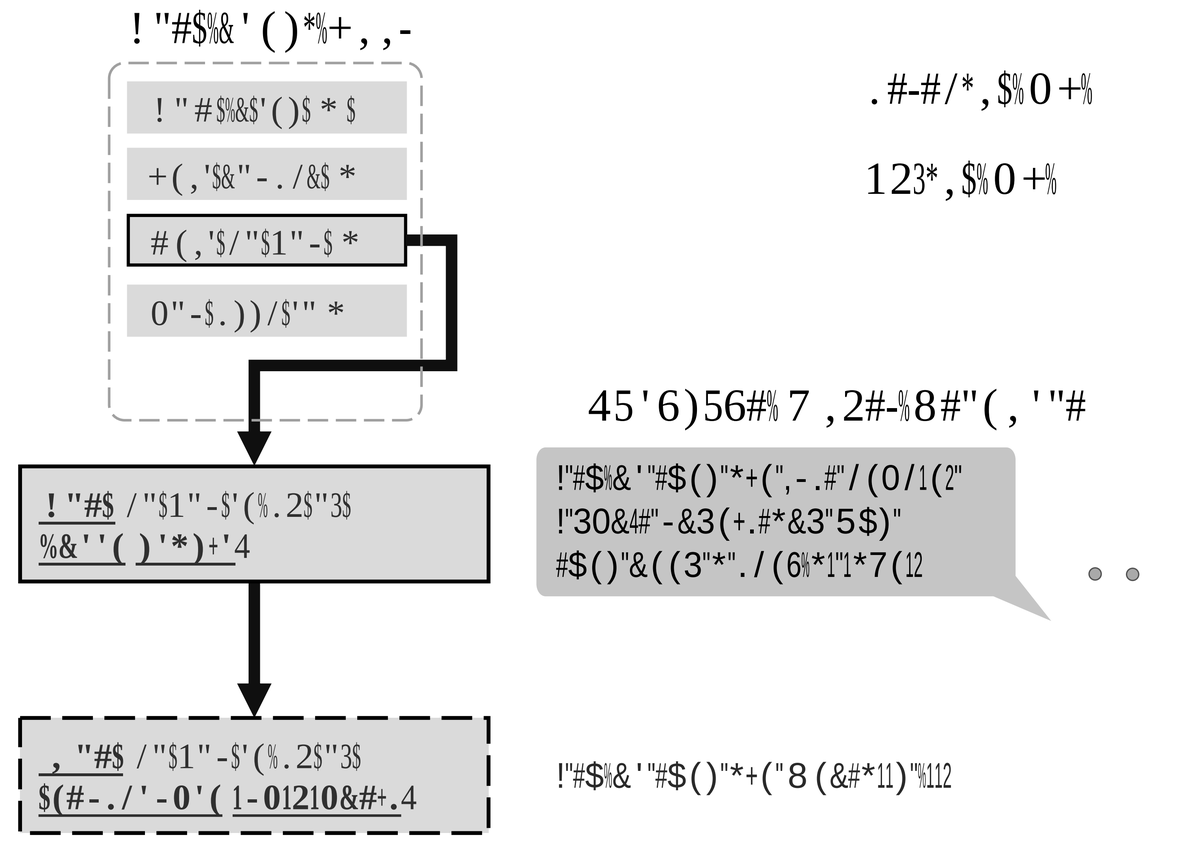


Figure: Illustration of edit-based BRT. Edit-based BRT generates test cases by selecting and editing user inputs in the pool. Here, our edit-based BRT is applied to BlenderBot-3B (BB-3B) using the user input from Bot Adversarial Dialogue.

- BRT (e) employs two GP surrogate models, **selector GP** and **editor GP**:
  Selector GP approximates the maximum value of the function f over the set of edited user inputs $B(\{u\})$, denoted as $\max_{u \in B(\{u\})} f(u)$, for $u \in \hat{U}$.
  Editor GP directly approximates the function value $f(u)$ for $u \in B(\hat{U})$.
- BRT (e) divides the acquisition maximization process into two stages.
  1. Select user input t to be edited with selector GP.
  2. Edit the selected user input $t \in \hat{U}$ with editor GP to obtain $t^{\text{edit}} \in B(\{t\})$.

Results

Method	Bloom ZS RSR % (↑)	Bloom ZS Self-BLEU$^{(k)}$ (↓)	ConvAI2 RSR	ConvAI2 Self-BLEU$^{(k)}$
Rand	0.8 (0.04)	51.6 (0.35)	1.1 (0.07)	34.6 (0.38)
BRT (s)	**10.3** (0.02)	**50.8** (0.06)	**4.3** (0.03)	**33.7** (0.37)
SFS (OPT-1.3B)	7.4 (0.13)	49.6 (0.08)	13.1 (0.26)	42.7 (0.20)
SL (OPT-1.3B)	12.0 (0.07)	58.9 (0.25)	16.4 (0.27)	46.6 (0.26)
BRT (e)	**39.1** (0.53)	**48.6** (0.09)	**44.0** (0.36)	**33.8** (0.14)

Table: Red teaming results of the open-domain dialogue task against BB-3B (N_Q = 20,000).

Method	RSR (↑)	Self-BLEU$^{(k)}$ (↓)
SFS (OPT-1.3B)	6.52 (0.03)	55.18 (0.33)
SL (OPT-1.3B)	47.87 (0.32)	71.13 (0.10)
BRT (e)	**71.34** (0.54)	**52.48** (0.32)

Table: Red teaming results against Stable Di□usion v1.4 against OPT-66B ZS (N_Q = 5,000).

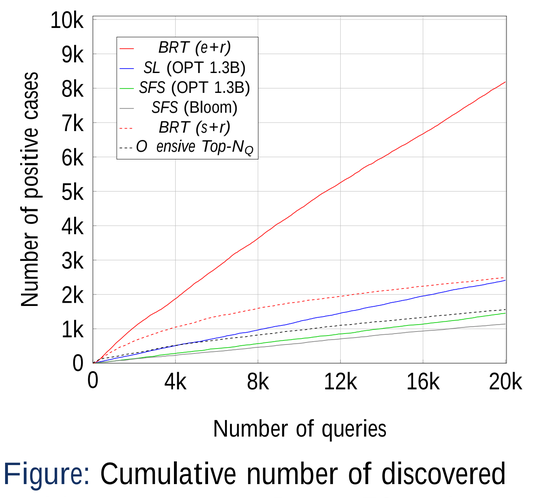


Figure: Cumulative number of discovered positive test cases on Bloom ZS against BB-3B.