A. Appendix

A.1. Semantics of specifications

We define the semantics of a specification $S = \{(T_1, \delta_1), \ldots, (T_n, \delta_n)\}$ (such that $T_i = (\varphi_i, f_i)$) as follows. Given a string $\mathbf{x} = x_1 \ldots x_m$, a string $\mathbf{y}$ is in the perturbations space $S(\mathbf{x})$ if:

1. there exists matches $\langle (l_1, r_1), j_1 \rangle \ldots \langle (l_k, r_k), j_k \rangle$ (we assume that matches are sorted in ascending order of l_i) such that for every $i \leqslant k$ we have that (l_i, r_i) is a valid match of φ_{j_i} in $\mathbf{x}$;
2. the matches are not overlapping: for every two distinct i_1 and i_2, $r_{i_1} < l_{i_2}$ or $r_{i_2} < l_{i_1}$;
3. the matches respect the δ constraints: for every $j' \leqslant n$, $|\{\langle (l_i, r_i), j_i \rangle \mid j_i = j'\}| \leqslant \delta_{j'}$.
4. the string $\mathbf{y}$ is the result of applying an appropriate transformation to each match: if for every $i \leqslant k$ we have $\mathbf{s}_i \in f_{j_i}(x_{l_i} \ldots x_{r_i})$, then

$$\mathbf{y} = x_1 \ldots x_{l_1 - 1}\, \mathbf{s}_1\, x_{r_1+1} \ldots x_{l_k - 1}\, \mathbf{s}_k\, x_{r_k+1} \ldots x_m.$$

A.2. Proof of Theorem 1

We give the following definition of a convex set:

Definition 1. Convex set: A set $\mathcal{C}$ is **convex** if, for all x and y in $\mathcal{C}$, the line segment connecting x and y is included in $\mathcal{C}$.

Proof. We first state and prove the following lemma.

Lemma 2. Given a set of points $\{p_0, p_1, \ldots, p_t\}$ and a convex set $\mathcal{C}$ such that $\{p_0, p_1, \ldots, p_t\} \subset \mathcal{C}$. These points define a set of vectors $\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2}, \ldots, \overrightarrow{p_0p_t}$. If a vector $\overrightarrow{p_0p}$ can be represented as a sum weighed by α_i:

$$\overrightarrow{p_0p} = \sum_{i=1}^{t} \alpha_i \cdot \overrightarrow{p_0p_i}, \tag{4}$$

where α_i respect to constraints:

$$\sum_{i=1}^{t} \alpha_i \le 1 \wedge \forall 1 \le i \le t.\ \alpha_i \ge 0, \tag{5}$$

then the point p is also in the convex set $\mathcal{C}$.

Proof. We prove this lemma by induction on t,

- Base case: $t = 1$, if $\overrightarrow{p_0p} = \alpha_1 \cdot \overrightarrow{p_0p_1}$ and $0 \le \alpha_1 \le 1$, then p is on the segment p_0p_1. By the definition of the convex set (Definition 1), the segment p_0p_1 is inside the convex, which implies p is inside the convex: $p \in p_0p_1 \subseteq \mathcal{C}$.
- Inductive step: Suppose the lemma holds for $t = r$. If a vector $\overrightarrow{p_0p}$ can be represented as a sum weighed by α_i:

$$\overrightarrow{p_0p} = \sum_{i=1}^{r+1} \alpha_i \cdot \overrightarrow{p_0p_i} \tag{6}$$

where α_i respect to constraints:

$$\sum_{i=1}^{r+1} \alpha_i \le 1, \tag{7}$$

$$\forall 1 \le i \le r+1.\ \alpha_i \ge 0. \tag{8}$$

We divide the sum in Eq 6 into two parts:

$$\overrightarrow{p_0p} = \sum_{i=1}^{r+1} \alpha_i \cdot \overrightarrow{p_0p_i} \tag{9}$$

$$= (\sum_{i=1}^{r} \alpha_i \cdot \overrightarrow{p_0p_i}) + \alpha_{r+1} \cdot \overrightarrow{p_0p_{r+1}} \tag{10}$$

$$= (1 - \alpha_{r+1})\overrightarrow{p_0p'} + \alpha_{r+1} \cdot \overrightarrow{p_0p_{r+1}} \quad \text{, and} \tag{11}$$

$$\overrightarrow{p_0p'} = \sum_{i=1}^{r} \frac{\alpha_i}{1 - \alpha_{r+1}} \cdot \overrightarrow{p_0p_i} \tag{12}$$

Because from Inequality 7, we know that

$$\sum_{i=1}^{r} \alpha_i \le 1 - \alpha_{r+1},$$

which is equivalent to

$$\sum_{i=1}^{r} \frac{\alpha_i}{1 - \alpha_{r+1}} \le 1.$$

This inequality enables the inductive hypothesis, and we know point p' is in the convex set $\mathcal{C}$. From Eq 11, we know that the point p is on the segment of $p'p_{r+1}$, since both two points p' and p_{r+1} are in the convex set $\mathcal{C}$, then the point p is also inside the convex set $\mathcal{C}$.

□

To prove Theorem 1, we need to show that every perturbed sample $\mathbf{y} \in S(\mathbf{x})$ lies inside the convex hull of $abstract(S, \mathbf{x})$.

We first describe the perturbed sample $\mathbf{y}$. The perturbed sample $\mathbf{y}$ as a string is defined in the semantics of specification S (see the Appendix A.1). In the rest of this proof, we use a function $E : \Sigma^m \mapsto \mathbb{R}^{m \times d}$ mapping from a string with length m to a point in $m \times d$-dimensional space, e.g., $E(\mathbf{y})$ represents the point of the perturbed sample $\mathbf{y}$ in the embedding space. We use $\mathbf{x}_{\langle(l,r),j,\mathbf{s}\rangle}$ to represent the string perturbed by a transformation $T_j = (\varphi_j, f_j)$ such that (l, r) is a valid match of φ_j and $\mathbf{s} \in f_j(x_l, \ldots, x_r)$. Then

$$\mathbf{x}_{\langle(l,r),j,\mathbf{s}\rangle} = x_1 \ldots x_{l-1}\,\mathbf{s}\,x_{r+1} \ldots x_m.$$

We further define $\Delta_{\langle(l,r),j,\mathbf{s}\rangle}$ as the vector $E(\mathbf{x}_{\langle(l,r),j,\mathbf{s}\rangle}) - E(\mathbf{x}) = \overrightarrow{E(\mathbf{x})E(\mathbf{x}_{\langle(l,r),j,\mathbf{s}\rangle})}$:

$$\Delta_{\langle(l,r),j,\mathbf{s}\rangle} = (\underbrace{0,\ldots,0}_{(l-1)\times d}, E(\mathbf{s}) - E(x_l \ldots x_r), \underbrace{0,\ldots,0}_{(m-r)\times d}).$$

A perturbed sample $\mathbf{y}$ defined by matches $\langle(l_1, r_1), j_1\rangle \ldots \langle(l_k, r_k), j_k\rangle$ and for every $i \leqslant k$ we have $\mathbf{s}_i \in f_{j_i}(x_{l_i} \ldots x_{r_i})$, then

$$\mathbf{y} = x_1 \ldots x_{l_1-1}\,\mathbf{s}_1\,x_{r_1+1} \ldots x_{l_k-1}\,\mathbf{s}_k\,x_{r_k+1} \ldots x_m.$$

The matches respect the δ constraints: for every $j' \leq n$, $|\{\langle(l_i, r_i), j_i, \mathbf{s}_i\rangle \mid j_i = j'\}| \leqslant \delta_{j'}$. Thus, the size of the matches k also respect the δ constraints:

$$k = \sum_{j'=1}^{n} |\{\langle(l_i, r_i), j_i, \mathbf{s}_i\rangle \mid j_i = j'\}| \leq \sum_{j'=1}^{n} \delta_{j'}. \quad (13)$$

In the embedding space,

$$\begin{aligned}\overrightarrow{E(\mathbf{x})E(\mathbf{y})} = (&\underbrace{0,\ldots,0}_{(l_1-1)\times d}, E(\mathbf{s}_1) - E(x_{l_1} \ldots x_{r_1}),\\ &0,\ldots,0, E(\mathbf{s}_k) - E(x_{l_k} \ldots x_{r_k}), \underbrace{0,\ldots,0}_{(m-r_k)\times d}).\end{aligned}$$

Thus, we can represent $\overrightarrow{E(\mathbf{x})E(\mathbf{y})}$ using $\Delta_{\langle(l,r),j,\mathbf{s}\rangle}$:

$$\overrightarrow{E(\mathbf{x})E(\mathbf{y})} = \sum_{i=1}^{k} \Delta_{\langle(l_i,r_i),j_i,\mathbf{s}_i\rangle}. \quad (14)$$

We then describe the convex hull of $abstract(S, \mathbf{x})$. The convex hull of $abstract(S, \mathbf{x})$ is constructed by a set of points $E(\mathbf{x})$ and $E(\mathbf{v}_{\langle(l,r),i,\mathbf{s}\rangle})$, where points $E(\mathbf{v}_{\langle(l,r),i,\mathbf{s}\rangle})$ are computed by:

$$E(\mathbf{v}_{\langle(l,r),j,\mathbf{s}\rangle}) \triangleq E(\mathbf{x}) + (\sum_{i=1}^{n} \delta_i)(E(\mathbf{x}_{\langle(l,r),j,\mathbf{s}\rangle}) - E(\mathbf{x})).$$

Alternatively, using the definition of $\Delta_{\langle(l,r),j,\mathbf{s}\rangle}$, we get

$$\overrightarrow{E(\mathbf{x})E(\mathbf{v}_{\langle(l,r),j,\mathbf{s}\rangle})} = (\sum_{i=1}^{n} \delta_i)\Delta_{\langle(l,r),j,\mathbf{s}\rangle}. \quad (15)$$

We then prove the Theorem 1. To prove $E(\mathbf{y})$ lies in the convex hull of $abstract(S, \mathbf{x})$, we need to apply Lemma 2. Notice that a convex hull by definition is also a convex set. Because from Eq 14, we have

$$\begin{aligned}\overrightarrow{E(\mathbf{x})E(\mathbf{y})} &= \sum_{i=1}^{k} \Delta_{\langle(l_i,r_i),j_i,\mathbf{s}_i\rangle}\\ &= \frac{1}{\sum_{i=1}^{n} \delta_i} \sum_{i=1}^{k} (\sum_{i'=1}^{n} \delta_{i'})\Delta_{\langle(l_i,r_i),j_i,\mathbf{s}_i\rangle}.\end{aligned}$$

We can use Eq 15 into the above equation, and have

$$\begin{aligned}&= \frac{1}{\sum_{i=1}^{n} \delta_i} \sum_{i=1}^{k} \overrightarrow{E(\mathbf{x})E(\mathbf{v}_{\langle(l_i,r_i),j_i,\mathbf{s}_i\rangle})}\\ &= \sum_{i=1}^{k} (\frac{1}{\sum_{i=1}^{n} \delta_i}) \cdot \overrightarrow{E(\mathbf{x})E(\mathbf{v}_{\langle(l_i,r_i),j_i,\mathbf{s}_i\rangle})}.\end{aligned}$$

To apply Lemma 2, we set

$$\alpha_i = \frac{1}{\sum_{j=1}^{n} \delta_j}.$$

Using Inequality 13 on

$$\alpha_i = \frac{1}{\sum_{j=1}^{n} \delta_j} \geq 0, \quad (16)$$

we get

$$\sum_{i=1}^{k} \alpha_i = \sum_{i=1}^{k} \frac{1}{\sum_{j=1}^{n} \delta_j} = \frac{k}{\sum_{j=1}^{n} \delta_j} \leq 1. \quad (17)$$

The constraints in Inequality 16 and Inequality 17 enable Lemma 2, and by applying Lemma 2, we know that point $E(\mathbf{y})$ is inside the convex hull of $abstract(S, \mathbf{x})$. □

A.3. Details of Experiment Setup

For AG dataset, we trained a smaller character-level model than the one used in Huang et al. (2019). We followed the setup of the previous work: use lower-case letters only and truncate the inputs to have at most 300 characters. The model consists of an embedding layer of dimension 64, a 1-D convolution layer with 64 kernels of size 10, a ReLU layer, a 1-D average pooling layer of size 10, and two fully-connected layers with ReLUs of size 64, and a linear layer. We randomly initialized the character embedding and updated it during training.

For SST2 dataset, we trained the same word-level model as the one used in Huang et al. (2019). The model consists of an embedding layer of dimension 300, a 1-D convolution layer with 100 kernels of size 5, a ReLU layer, a 1-D average pooling layer of size 5, and a linear layer. We used the pre-trained Glove embedding (Pennington et al., 2014) with dimension 300 and fixed it during training.

For SST2 dataset, we trained the same character-level model as the one used in Huang et al. (2019). The model consists of an embedding layer of dimension 150, a 1-D convolution layer with 100 kernels of size 5, a ReLU layer, a 1-D average pooling layer of size 5, and a linear layer. We randomly initialized the character embedding and updated it during training.

For all models, we used Adam (Kingma & Ba, 2015) with a learning rate of 0.001 for optimization and applied early stopping policy with patience 5.

A.3.1. PERTURBATIONS

We provide the details of the string transformations we used:

- T_{SubAdj}, T_{InsAdj}: We allow each character substituting to one of its adjacent characters on the QWERTY keyboard.
- $T_{DelStop}$: We choose $\{$*and*, *the*, *a*, *to*, *of*$\}$ as our stop words set.
- T_{SubSyn}: We use the synonyms provided by PPDB (Pavlick et al., 2015). We allow each word substituting to its closest synonym when their part-of-speech tags are also matched.

A.3.2. BASELINE

Random augmentation performs adversarial training using a weak adversary that simply picks a random perturbed sample from the perturbation space. For a specification $S = \{(T_1, \delta_1), \ldots, (T_n, \delta_n)\}$, we produce $\mathbf{z}$ by uniformly sampling one string $\mathbf{z}_1$ from a string transformation (T_1, δ_1) and passing it to the next transformation (T_2, δ_2), where we then sample a new string $\mathbf{z}_2$, and so on until we have exhausted all transformations. The objective function is the following:

$$\underset{\theta}{\operatorname{argmin}} \underset{(\mathbf{x},y)\sim\mathcal{D}}{\mathbb{E}} \left(\mathcal{L}(x, y, \theta) + \max_{\mathbf{z}\in R(\mathbf{x})} \mathcal{L}(\mathbf{z}, y, \theta)\right) \quad (18)$$

HotFlip augmentation performs adversarial training using the HotFlip (Ebrahimi et al., 2018) attack to find $\mathbf{z}$ and solve the inner maximization problem. The objective function is the same as Eq 18.

A3T adopts a curriculum-based training method (Huang et al., 2019; Gowal et al., 2019) that uses a hyperparameter λ to weigh between normal loss and maximization objective in Eq. (2). We linearly increase the hyperparameter λ during training.

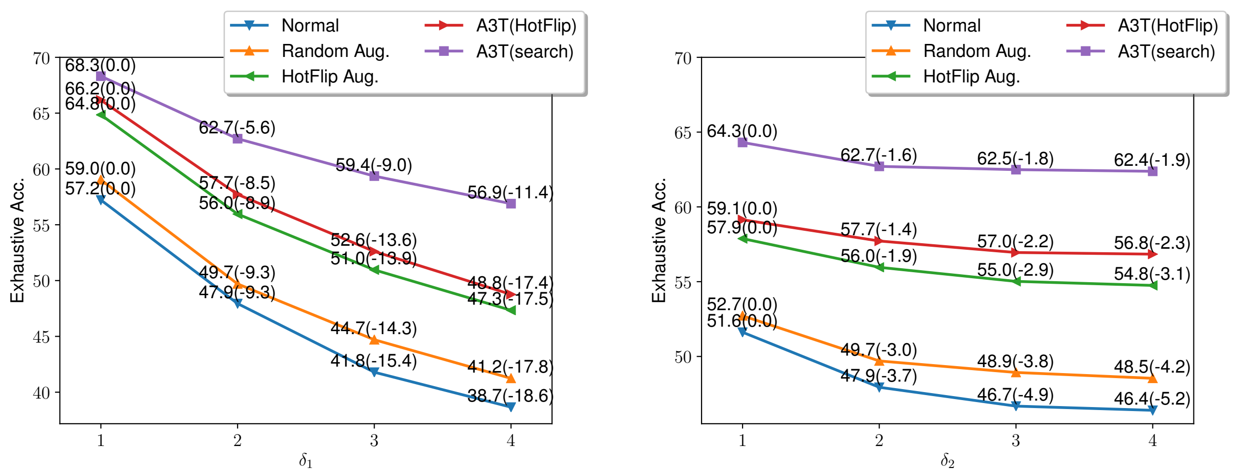
$$\underset{\theta}{\operatorname{argmin}} \underset{(\mathbf{x},y)\sim\mathcal{D}}{\mathbb{E}} \Big((1-\lambda)\mathcal{L}(x, y, \theta) + \lambda \max_{\mathbf{z}\in augment_k(S_{aug},\mathbf{x})} \mathcal{L}(abstract(S_{abs}\mathbf{z}), y, \theta)\Big).$$

Also, we set k in $augment_k$ to 2, which means we select 2 perturbed samples to abstract.

A.3.3. EVALUATION RESULTS

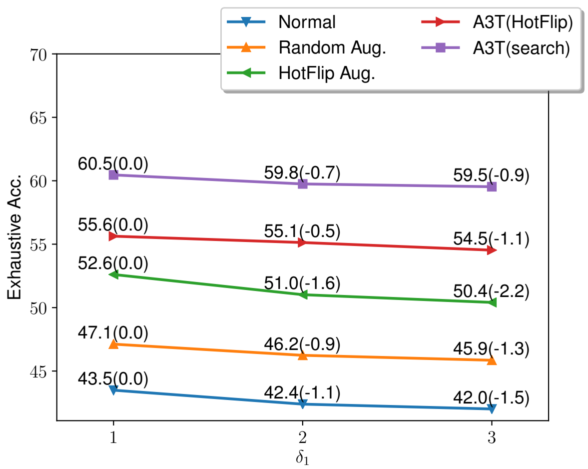
RQ2: Effects of size of the perturbation space In Figure 4, we fix the word-level model A3T (search) trained on $\{(T_{Dup}, 2), (T_{SubSyn}, 2)\}$. Then, we test this model's exhaustive accuracy on $\{(T_{Dup}, \delta_1), (T_{SubSyn}, 2)\}$ (Figure 4(a)) and $\{(T_{Dup}, 2), (T_{SubSyn}, \delta_2)\}$ (Figure 4(b)), where we vary the parameters δ_1 and δ_2 between 1 and 4, increasing the size of the perturbation space. The exhaustive accuracy of A3T(HotFlip) and A3T(search) decreases by 17.4% and 11.4%, respectively, when increasing δ_1 from 1 to 4, and decreases by 2.3% and 1.9%, respectively, when increasing δ_2 from 1 to 4. All other techniques result in larger decreases in exhaustive accuracy ($\geq$17.5% in $\{(T_{Dup}, \delta_1), (T_{SubSyn}, 2)\}$ and $\geq$3.1% in $\{(T_{Dup}, 2), (T_{SubSyn}, \delta_2)\}$).

In Figure 5, we fix the word-level model A3T (search) trained on $\{(T_{DelStop}, 2), (T_{Dup}, 2), (T_{SubSyn}, 2)\}$. Then, we test this model's exhaustive accuracy on $\{(T_{DelStop}, \delta_1), (T_{Dup}, 2), (T_{SubSyn}, 2)\}$ (Figure 5(a)), $\{(T_{DelStop}, 2), (T_{Dup}, \delta_2), (T_{SubSyn}, 2)\}$ (Figure 5(b)), and $\{(T_{DelStop}, 2), (T_{Dup}, 2), (T_{SubSyn}, \delta_3)\}$ (Figure 5(c)), where we vary the parameters δ_1, δ_2 and δ_3 between 1 and 3, increasing the size of the perturbation space. The exhaustive accuracy of A3T(HotFlip) and A3T(search) decreases by 1.1% and 0.9%, respectively, when increasing δ_1 from 1 to 3, decreases by 12.9% and 6.9%, respectively, when increasing δ_2 from 1 to 3, and decreases by 1.4% and 0.9%, respectively, when increasing δ_3 from 1 to 3. All other techniques result in larger decreases in exhaustive accuracy ($\geq$2.2% in $\{(T_{DelStop}, \delta_1), (T_{Dup}, 2), (T_{SubSyn}, 2)\}$, $\geq$13.0% in $\{(T_{DelStop}, 2), (T_{Dup}, \delta_2), (T_{SubSyn}, 2)\}$, and $\geq$2.8% in $\{(T_{DelStop}, 2), (T_{Dup}, 2), (T_{SubSyn}, \delta_3)\}$).

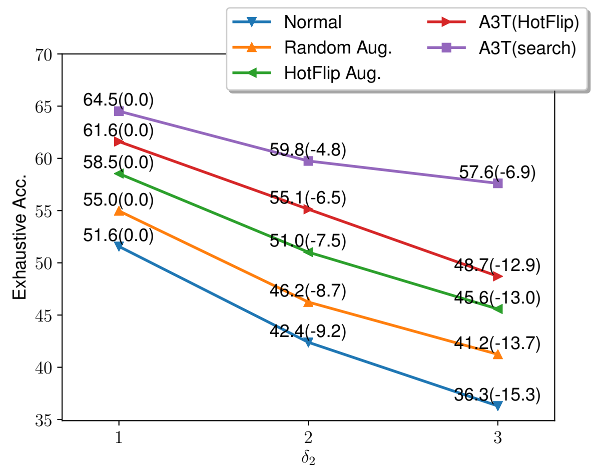


(a) $\{(T_{Dup}, \delta_1), (T_{SubSyn}, 2)\}$ (b) $\{(T_{Dup}, 2), (T_{SubSyn}, \delta_2)\}$

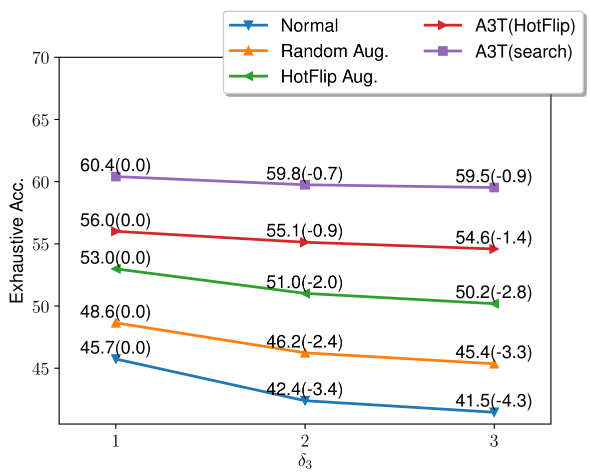
Figure 4. The exhaustive accuracy of $\{(T_{Dup}, \delta_1), (T_{SubSyn}, \delta_2)\}$, varying the parameters δ_1 (left) and δ_2 (right) between 1 and 4.



(a) $\{(T_{DelStop}, \delta_1), (T_{Dup}, 2), (T_{SubSyn}, 2)\}$



(b) $\{(T_{DelStop}, 2), (T_{Dup}, \delta_2), (T_{SubSyn}, 2)\}$



(c) $\{(T_{DelStop}, 2), (T_{Dup}, 2), (T_{SubSyn}, \delta_3)\}$

Figure 5. The exhaustive accuracy of $\{(T_{DelStop}, \delta_1), (T_{Dup}, \delta_2), (T_{SubSyn}, \delta_3)\}$, varying the parameters δ_1 (left), δ_2 (middle), and δ_3 (right) between 1 and 3.