EMNLP 2019

Achieving Verified Robustness to Symbol Substitutions via Interval Bound Propagation

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer,
Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, Pushmeet Kohli
DeepMind University College London

https://arxiv.org/pdf/1909.01492.pdf

読み手 小林颯介

Preferred Networks

Interval Bound Propagation [Gowal+18]: https://github.com/deepmind/interval-bound-propagation
Interval Bound Propagation (WIP...): https://github.com/soskek/interval-bound-propagation-chainer

Verify: ある特性, 仕様を満たすことを証明, 保証する

Robustness: 入力が変化しても予測 (分類結果) が変化しない

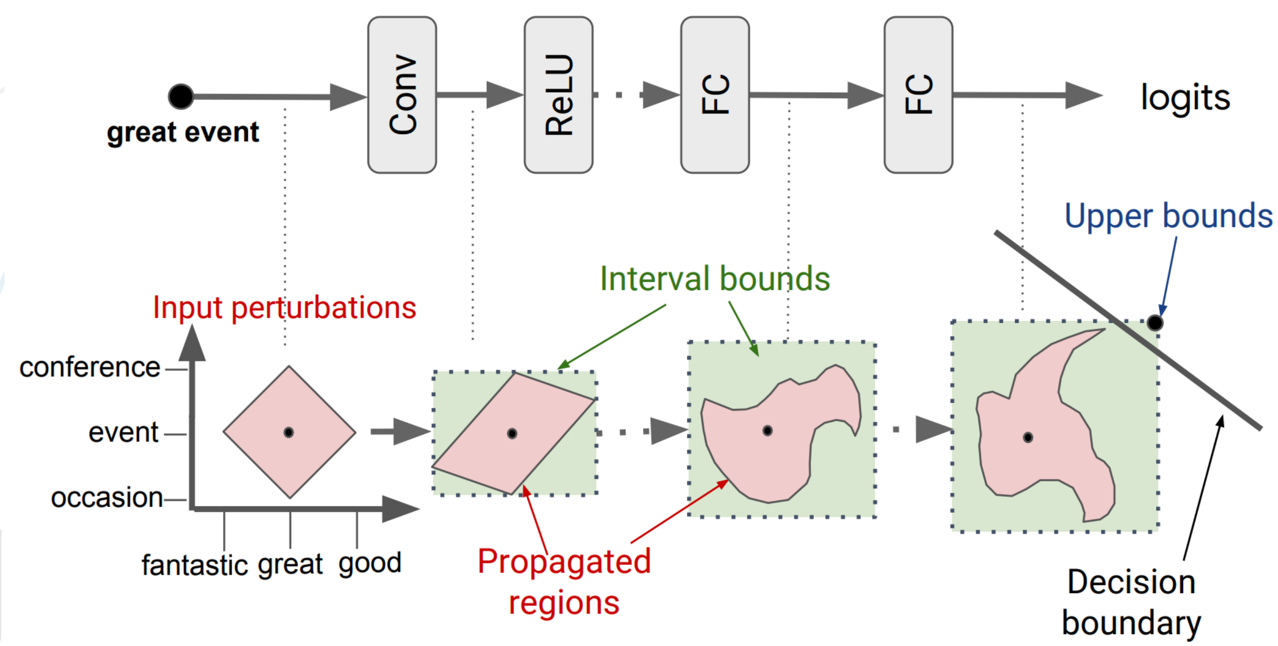
Symbol Substitutions: 文字の置換 or 単語の類義語との置換

EMNLP 2019

Achieving Verified Robustness to Symbol Substitutions via Interval Bound Propagation

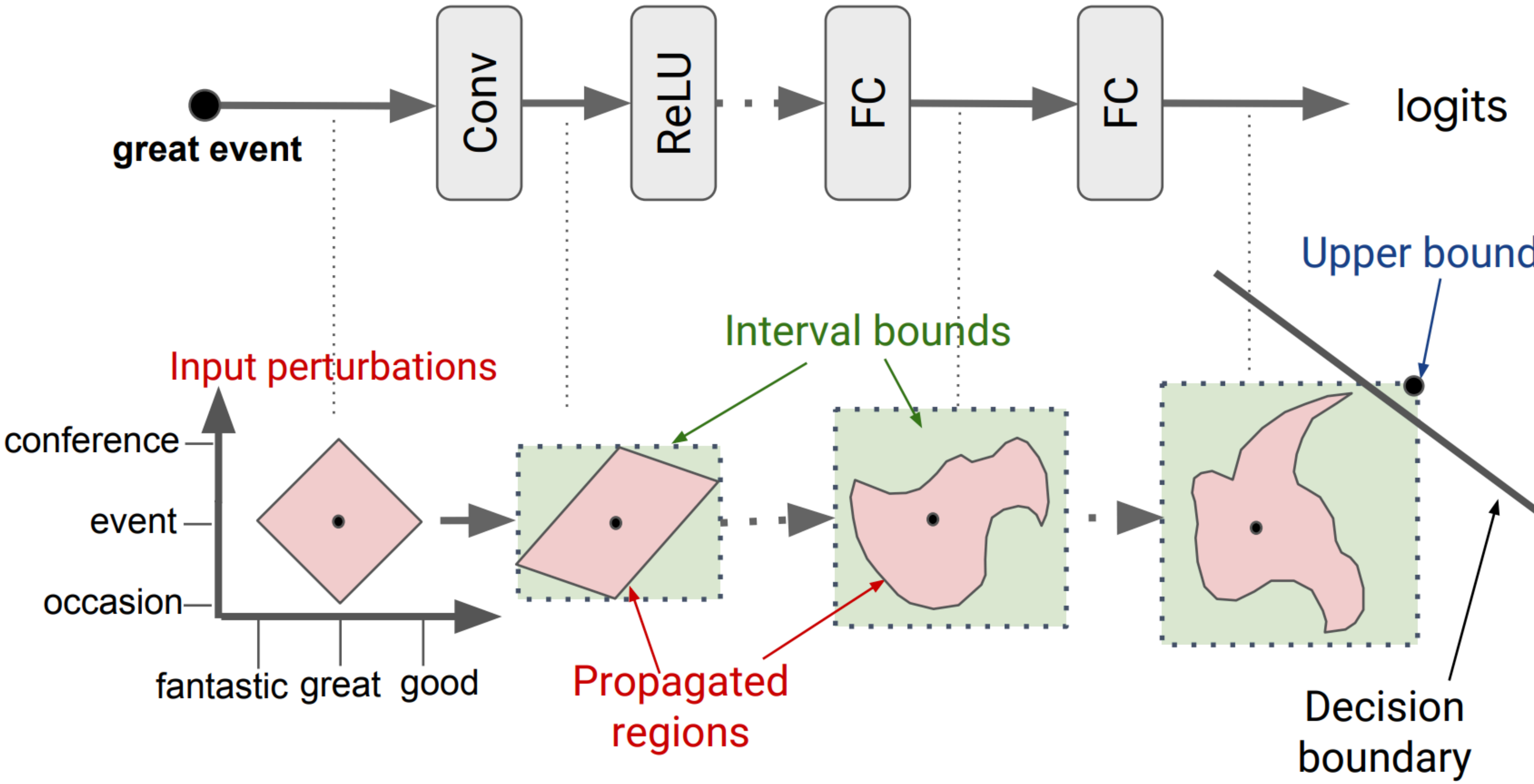
Po-Sen Huang, Robert Stanforth
Dani Yogatama, Sven Gowal, Krishna
DeepMind Univ

Interval Bound Propagation
[Gowal+18]
入力値の (複数パターンの) 変化の
ゆるい上界と下界を
層ごとに伝播させる手法



読み手 小

Preferred Networks


great event
Conv
ReLU
FC
FC
logits
Upper bounds
Interval bounds
Input perturbations
conference
event
occasion
fantastic
great
good
Propagated regions
Decision boundary

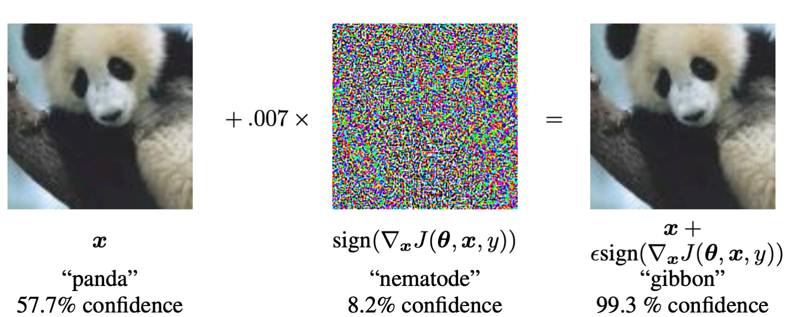
Adversarial Perturbation (敵対的摂動)

- モデルの出力結果が致命的に変わるような入力データ上のわずかな変化
- 典型的には、画像上でのノイズや記号置換をモデルの勾配情報などから見つけ出す

なお、このような (変に) 苦手な事例を Adversarial Example と呼ぶ (作り方, 探し方は問わない)

you ’ ve seen them a million times .
you ’ ve **sern** them a million times .

it ’ s the kind of pigeonhole-resisting romp that hollywood too rarely provides .
it ’ s the kind of pigeonhole-resisting romp that hollywood too rarely **gives** .



[Goodfellow+14]

詳しくは PFN 佐藤元紀氏の「言語処理分野における Adversarial Example」解説へ!!
https://www.ai-gakkai.or.jp/my-bookmark_vol34-no5/

Verification

- 特定の機能, 仕様を満たしていることを (現実的に可能なコストで) 検証すること
- プログラムの検証などで発展. 機械学習モデルでの研究は現在発展中

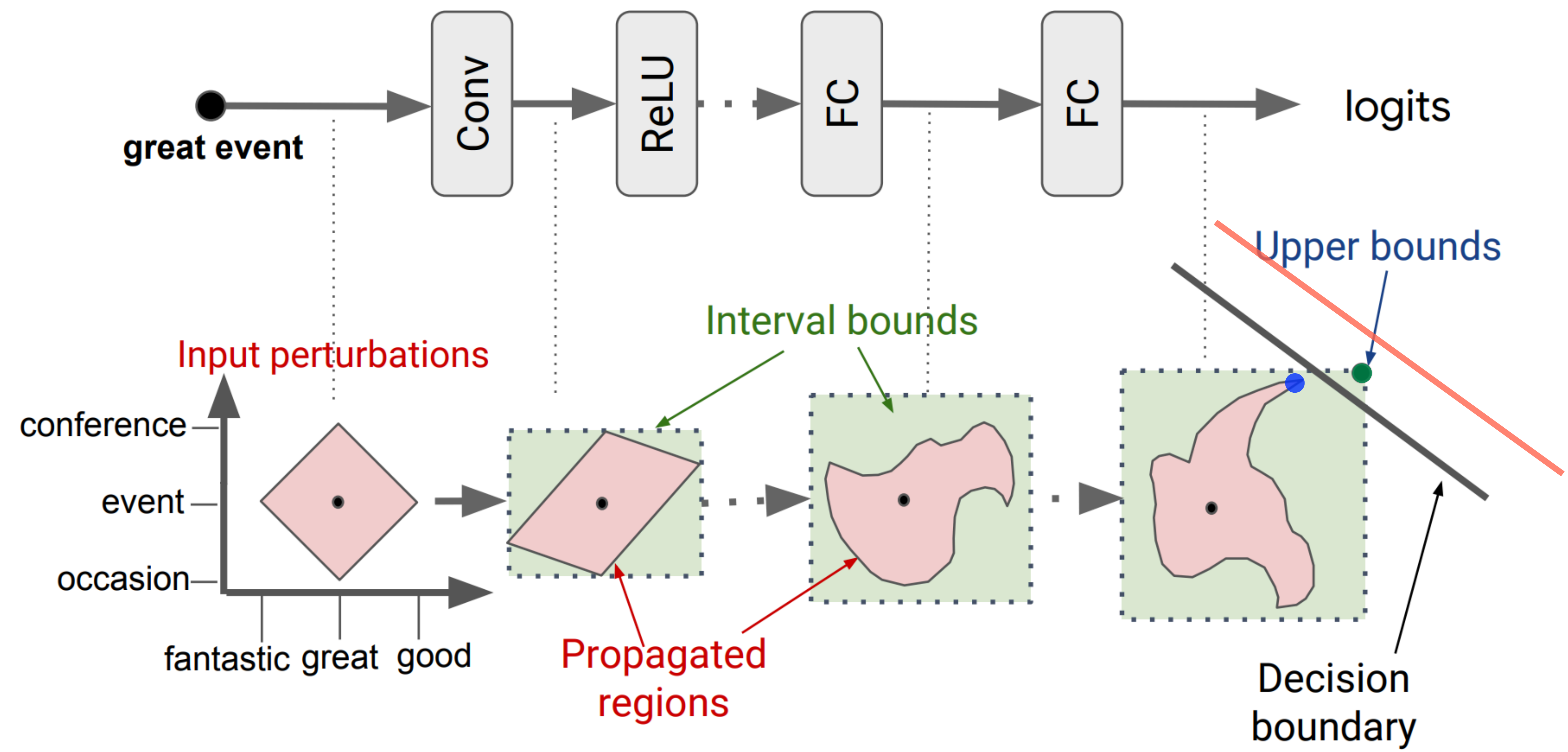
[Slide] Verification of Deep Neural Networks
[Slide] Formal Verification of Deep Neural Networks

- 本論文では
  - 『ある入力の分類結果は、特定の摂動の範囲の入力に対し、同じになる』を保証したい
    - 満たすかテストしたい
    - 満たすように学習したい

Verification

- どう保証できるか？
→ 最悪ケースでも満たせば保証成功
→ 最悪ケースはどう探せばいい？
- Adversarial Example 生成の手法？
しかし 最悪を見つけられるとは限らない
- 総当り？ 場合により可能. しかし大抵高コスト
- 👉「本来の最悪よりも余計に悪いケース」
の中で探しやすいものを代わりに探す
- 「余計に悪いケース」で保証できたら
「本来」も保証できる
(ただし、「余計」が反証されたときに「本来」が反証されるとは限らない)

黒い点: 「ある入力とその出力」
赤い領域: 「摂動時の値域」　黒い枠: 「摂動で最も遠く離れた場合の出力」
緑: 「もっと余計に『摂動で最も遠く離れた場合の出力値域』」
● 緑点: 「余計に離れたときの最悪な出力」
● 青点: 「本来の最悪な出力」
オレンジ太線: (これが分離平面なら保証成功. 余計な最悪の”緑点”は求めやすい)
黒太線: (これが分離平面なら保証成功. だけど本来の最悪の“青点”を求めるのは困難)



論文図より一部加筆

Interval Bound Propagation

- 入力での摂動値の上界 z^ と下界 $\underline{z}$ から伝播
- 単調増加関数ならカンタン
- アフィン変換なら…
  - 中心点 μ を変換
  - 半径 r を変換
  - $\mu \pm r$ で z^ と $\underline{z}$ を計算
- 最終層のlogitで
  正解クラスの下界$\underline{z}_{true}$ と
  誤りクラスの上界$\hat{z}_{false}$を比較
  → $\underline{z}_{true} - \hat{z}_{false} > 0$ なら"保証"
  (実際にはもう少しtightになる計算を考案していますが今回は省略)

$$\underline{z}_k = h_k(\underline{z}_{k-1})$$
$$\overline{z}_k = h_k(\overline{z}_{k-1})$$

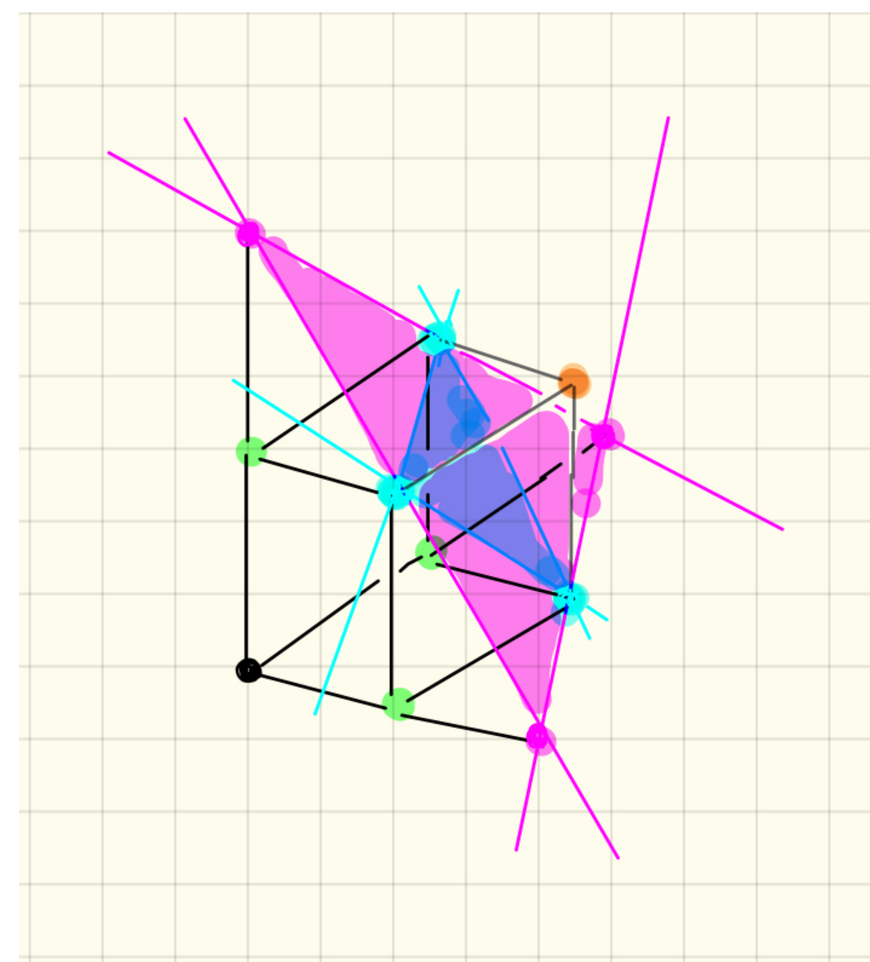
$$\mu_{k-1} = \frac{\overline{z}_{k-1} + \underline{z}_{k-1}}{2}$$
$$r_{k-1} = \frac{\overline{z}_{k-1} - \underline{z}_{k-1}}{2}$$
$$\mu_k = W\mu_{k-1} + b$$
$$r_k = |W| r_{k-1}$$
$$\underline{z}_k = \mu_k - r_k$$
$$\overline{z}_k = \mu_k + r_k$$

Interval Bound Propagation

- 入力での摂動値の上界 z^ と下界 $\underline{z}$ はどう計算？
  - 既存研究: 画像に対して摂動 s.t. $L_\infty \leqq \varepsilon$ 以下
    → 単に $z\hat{}=x+\varepsilon$, $\underline{z}=x-\varepsilon$
  - 今回: 全トークン中の δ 以下が置換
    - 総当りで算出？ → 組み合わせが多ければ無理…
  - 提案: 全組み合わせを覆うような “適切な” 頂点集合による凸包を考え、その頂点集合を使う. それらを変換後、次元ごとにmax, minをとる.

Interval Bound Propagation

- 提案: 全組み合わせを覆うような “適切な” 頂点集合による凸包を考え、その頂点集合を使う
- 理由はさておき
  各トークンを置換した際の点を元のxを中心にして
  δ倍に拡大した点の集合
  と点x をつなぐと “適切な”
  『δ個置換用の凸包』になる



[δ=2, 置換パタン合計$_3C_2$ を図示]
黒点: 元の入力 x
緑点: 各トークンを置換した際の入力 (δ=1)
オレンジ点: 雑にをとった点
水色点: δ=2置換によって取りうる点
ピンク点: δ倍に拡大した緑点. $pink_i = x + \delta * (green_i - x)$
ピンク点とxをつないだ凸包はちゃんと緑点と水色点を覆っている.

iPadを買うとこういう図をつい書いてしまうことが知られています

訓練

- 損失関数
  - 通常のクロスエントロピー と
    上界下界差のlogitでのクロスエントロピー

$$L = \kappa \underbrace{\ell(\mathbf{z}_K, y_{\text{true}})}_{L_{\text{normal}}} + (1-\kappa) \underbrace{\ell(\hat{\mathbf{z}}_K(\delta), y_{\text{true}})}_{L_{\text{spec}}}$$

- カリキュラム
  - κ は1から0.25へと線形にアニーリング
    徐々にverificationの項を強めていく

実験

- データセット
  - SST (Stanford Sentiment Treebank) [2 class]: word, character
  - AG News [4 class]: character　　　　　　　　の計3設定
- 置換
  - Word置換: PPDB類義語へ [Ganitkevitch+2013]
  - Char.置換: キーボードの近傍位置の文字へ (typo)
  - 訓練時の最大置換数δ=3, テスト時は1〜6で検証
- モデル
  - SST Word: 300dim 固定GloVe, 100dim 幅5 CNN, relu, average pooling, linear
  - SST Char.: 150dim random init, …同様
  - AG News Char.: 幅5->10, linear前に2層MLP

実験

- 比較訓練法
  - 普通の訓練 [生データでのクロスエントロピーのみ]
  - Random 置換 [生と置換データを1:1で訓練]
  - Adversarial training [同じく1:1で訓練]
    (HotFlip [Ebrahimi+2018] で生成したExampleを混ぜる)
  - Interval Bound Propagation [κ:1-κ, アニーリング]

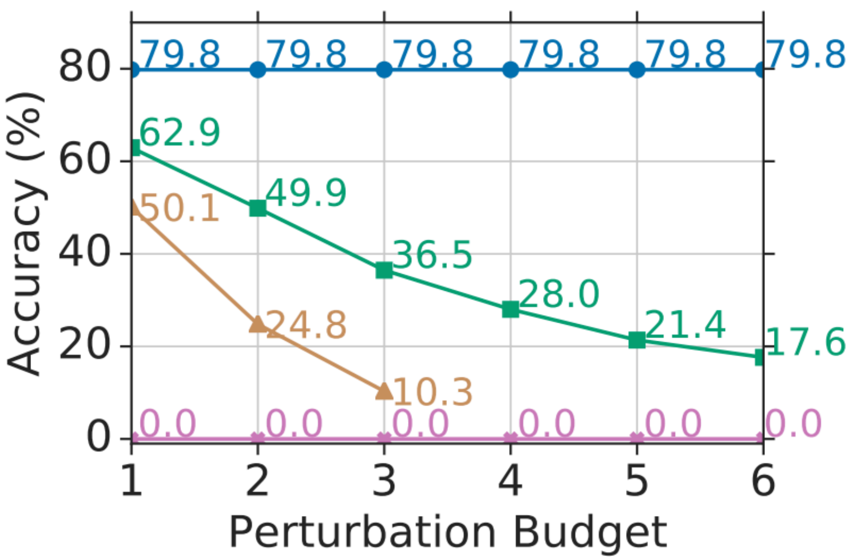
- テスト
  - 普通のAccuracy
  - Adversarial Example Accuracy (同様にHotFlipで生成)
  - 総当りで見つけた最悪置換事例 (Oracle) でのAccuracy
    (組み合わせ数が爆発しているのでできるところまで…)
  - Interval Bound Propagation (IBP)

最大総当りパターン数
(平均ではない)

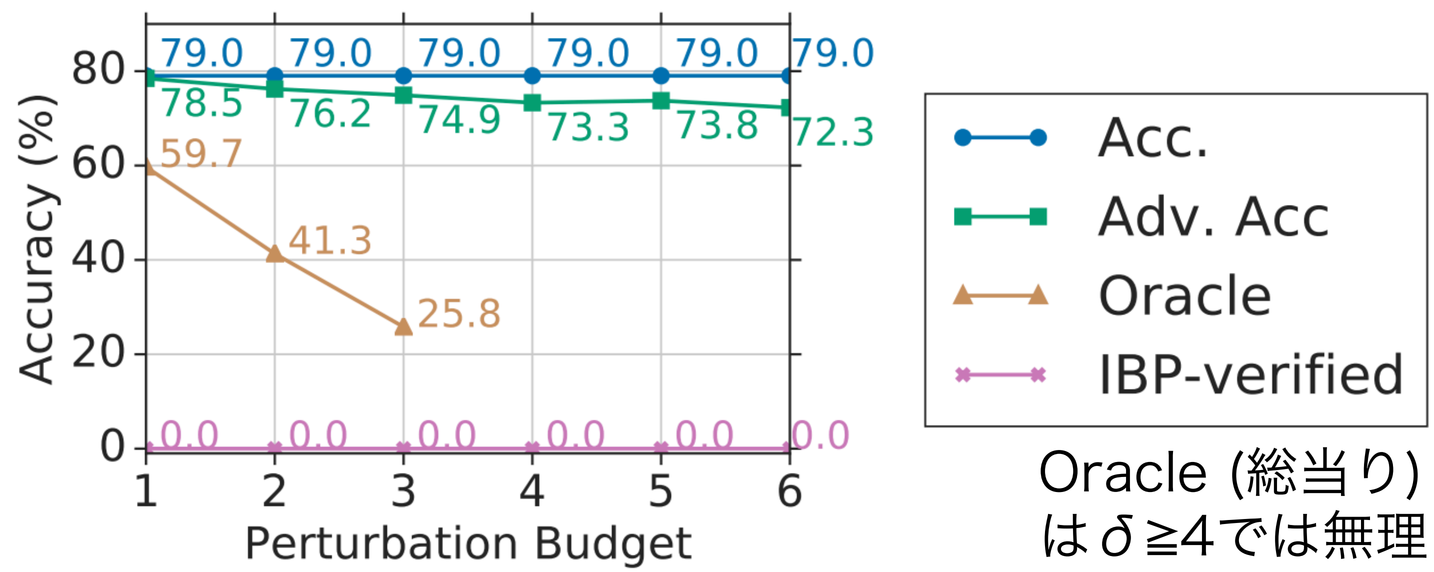
Perturbation radius	δ =1	δ =2	δ =3
SST-word	49	674	5,136
SST-character	206	21,116	1,436,026
AG-character	722	260,282	-

実験結果: SST-char. 置換数δを変えて.

- AdvTrain: 普通のAdvには頑健になるが最悪ケース (Oracle) にはまだまだ弱い
- 提案IBP: Oracleにも頑健. ただし通常の性能自体は少し低くなる

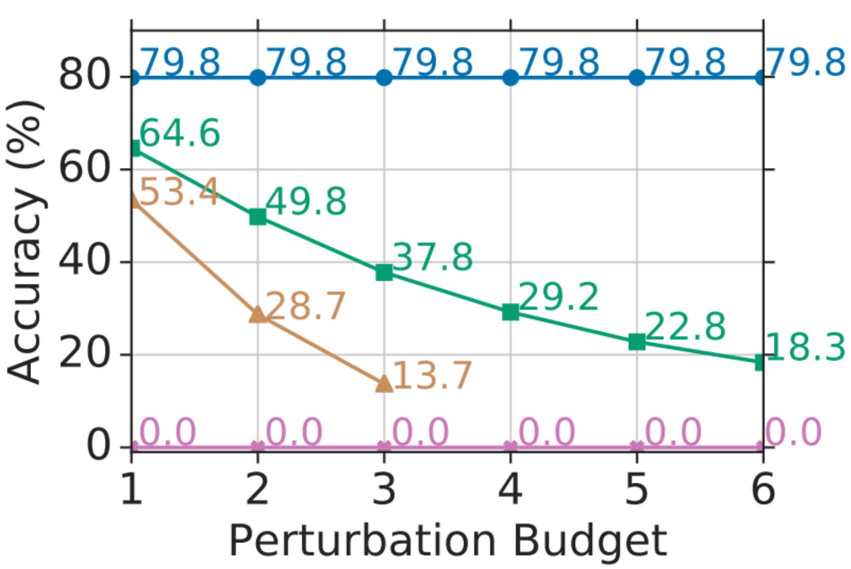


(a) Normal Training

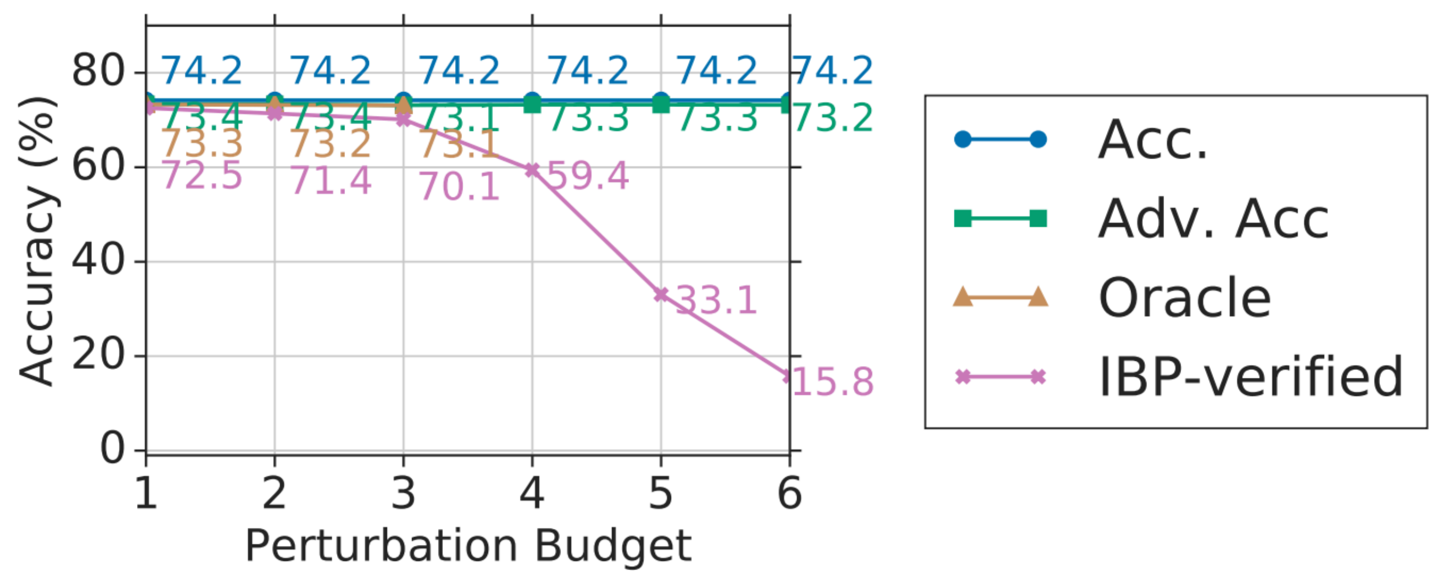


Oracle (総当り) は$\delta \geqq 4$では無理だったので省略

(b) Adversarial Training



(c) Data Augmentation Training



(d) Verifiable Training (IBP)

実験結果: SST-char/word δ=3, AG-char δ=2

- 基本的にデータセットが変わっても同様の傾向
- 置換パターンが多ければ多いほど
  Adversarial Example Accuracy と
  Oracle Accuracy の差が大きい. Advはやはり探索漏れが多い
- でも IBP訓練だとほぼ差がない

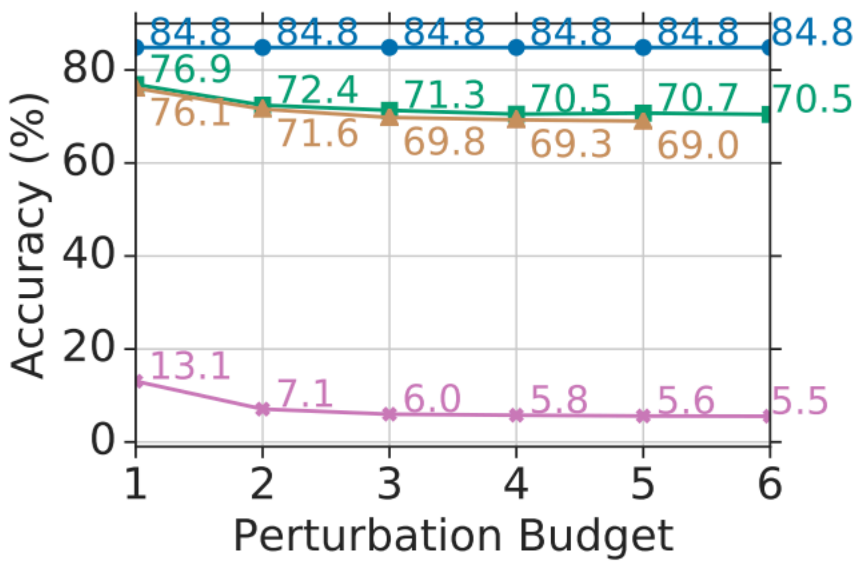
最大置換パターン数: 1,436,026 5,136 260,282

	SST-Char-Level				SST-Word-Level				AG-Char-Level			
Training	**Acc.**	**Adv. Acc.**		**Oracle**	**Acc.**	**Adv. Acc.**		**Oracle**	**Acc.**	**Adv. Acc.**		**Oracle**
Normal	79.8	36.5	-26.2	10.3	84.8	71.3	-1.5	69.8	89.5	75.4	-10.3	65.1
Adversarial	79.0	**74.9**	-49.1	25.8	**85.0**	76.8	-2.2	74.6	**90.5**	85.5	-3.9	81.6
Data aug.	79.8	37.8	-24.1	13.7	85.4	72.7	-1.1	71.6	88.4	77.5	-5.5	72.0
Verifiable (IBP)	74.2	73.1	±0	**73.1**	81.7	**77.2**	-0.7	**76.5**	87.6	**87.1**	±0	**87.1**

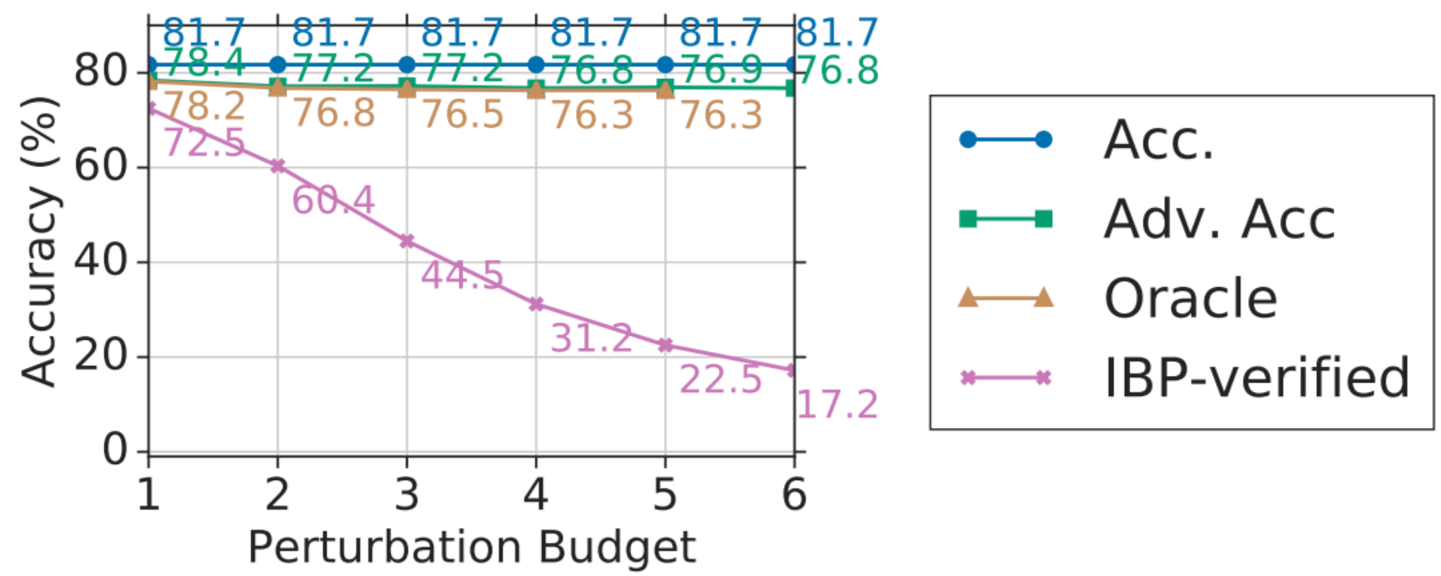
Perturbation radius	δ =1	δ =2	δ =3
SST-word	49	674	5,136
SST-character	206	21,116	1,436,026
AG-character	722	260,282	-

実験結果: SST-word. 単語ベクトル(入力空間)を変える

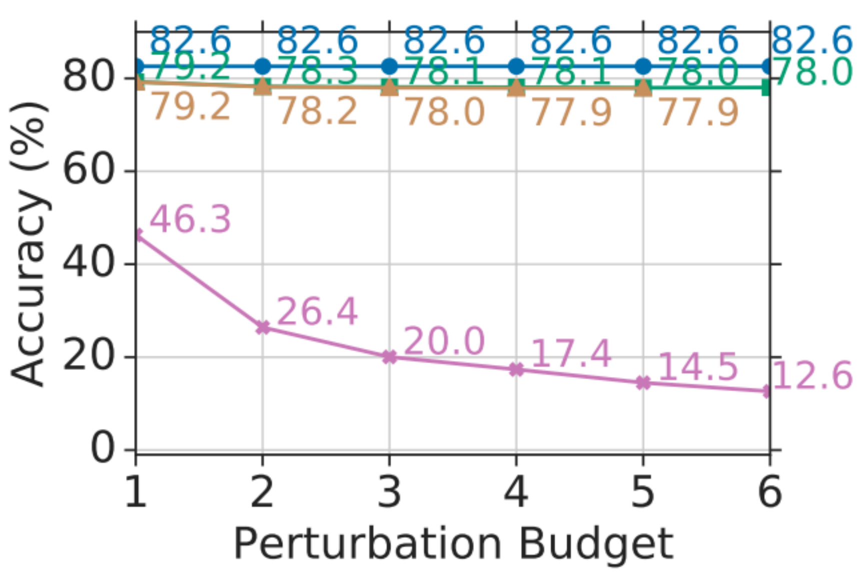
- GloVe v.s. CF: 類義語同士を近づけたGloVe [Mrkšić+16]
- IBPがOracleに対してtightになっている
  → 総当りせずともIBPでそこそこ効率よく検証テストができる
  (少なくともIBPで訓練した場合は)



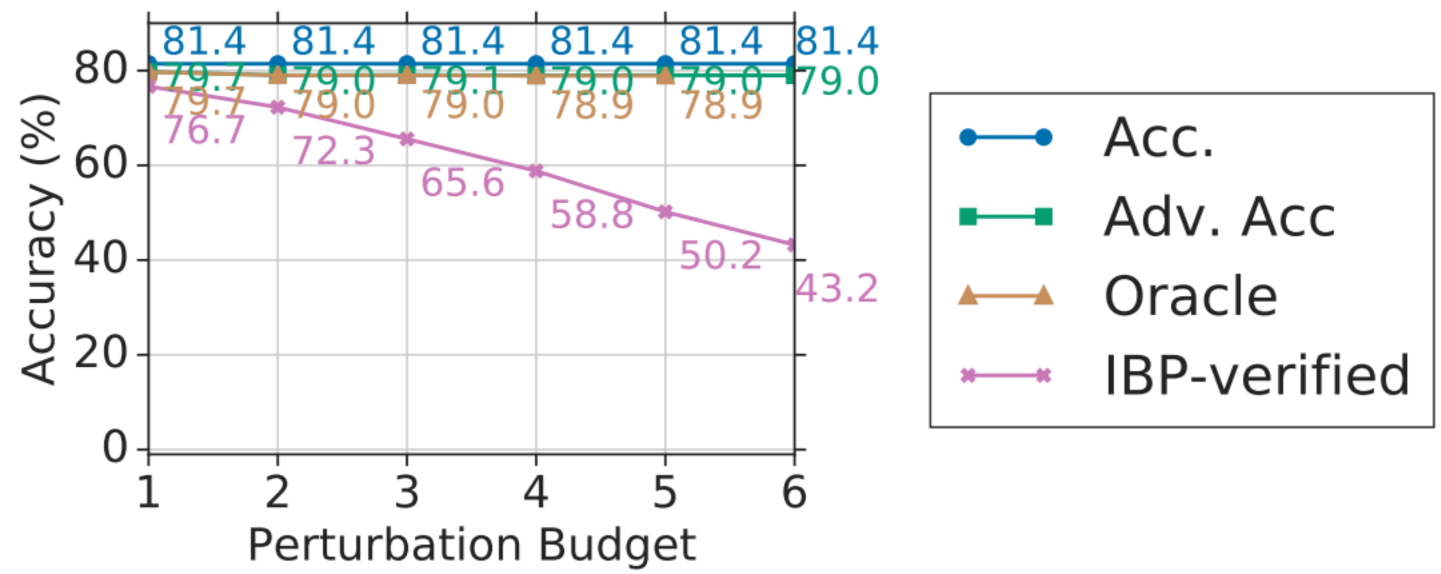
(a) Normal Training *(GloVe)*



(b) Verifiable Training (IBP) *(GloVe)*



(c) Normal Training *(CF)*



(d) Verifiable Training (IBP) *(CF)*

まとめ
Achieving Verified Robustness to Symbol Substitutions via Interval Bound Propagation

- Interval Bound Propagation [Gowal+18]: 入力値変化の上界と下界を順に伝播させる 軽量な 手法 (ほぼforwardが2回増えるだけ)
  - を　可変個の記号置換での変化に適用し、効率的な入力点集合を提案
- テキスト分類において類義語置換とタイポ文字置換に対するモデルの予測頑健性を改善
- 総当り最悪ケースとAdvExmp.のギャップを指摘

補足: できる・できない

- 間違い
  - 使用範囲内なら全摂動に対して予測が同じだと保証 → No
    訓練セットで見たデータなら…? → No
  - あるデータに対する予測が、それをIBPで計算した場合には変わってしまったので、予測を変えてしまう摂動が仕様範囲内に存在する → No
- 正解
  - 未知データの観測前は何も保証できない
  - あるデータに対する予測が、それをIBPで計算した場合にも同じならば、仕様範囲の全摂動に対して予測が同じと保証される