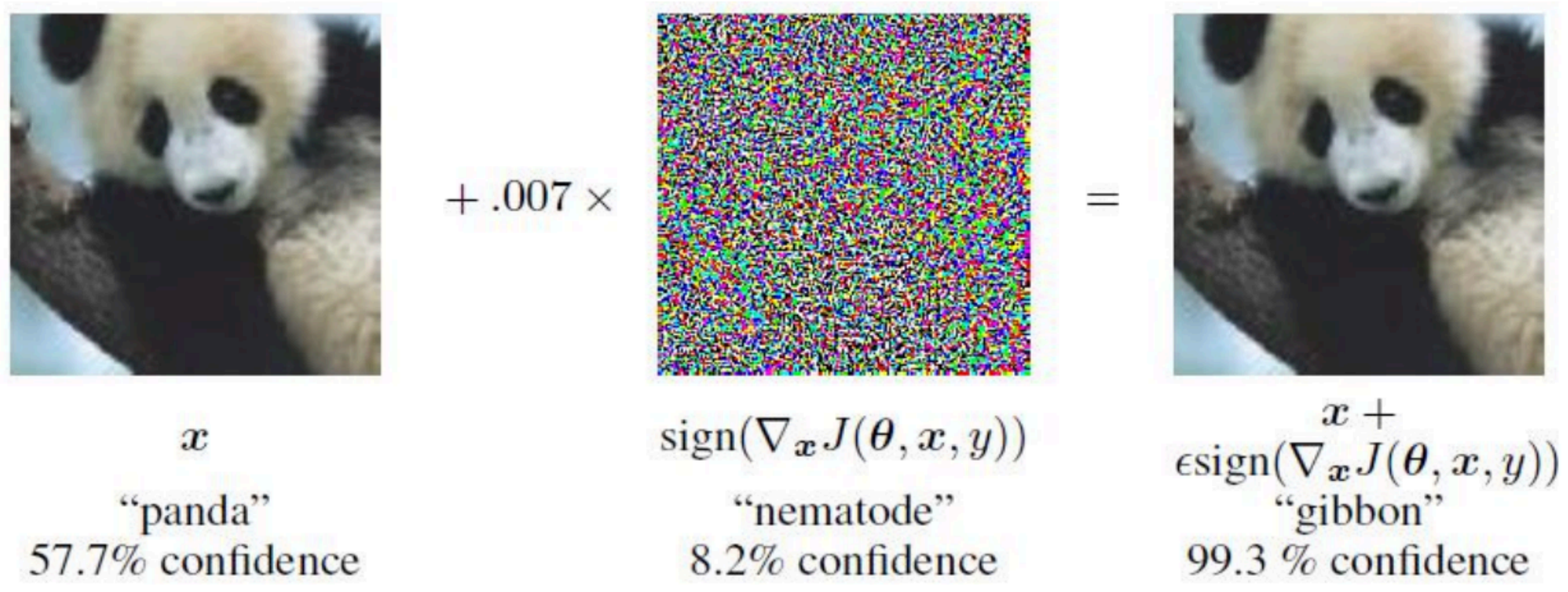
Training Certifiably Robust Neural Networks

Yuhao Mao

14 February 2024

ETH zürich SRILAB

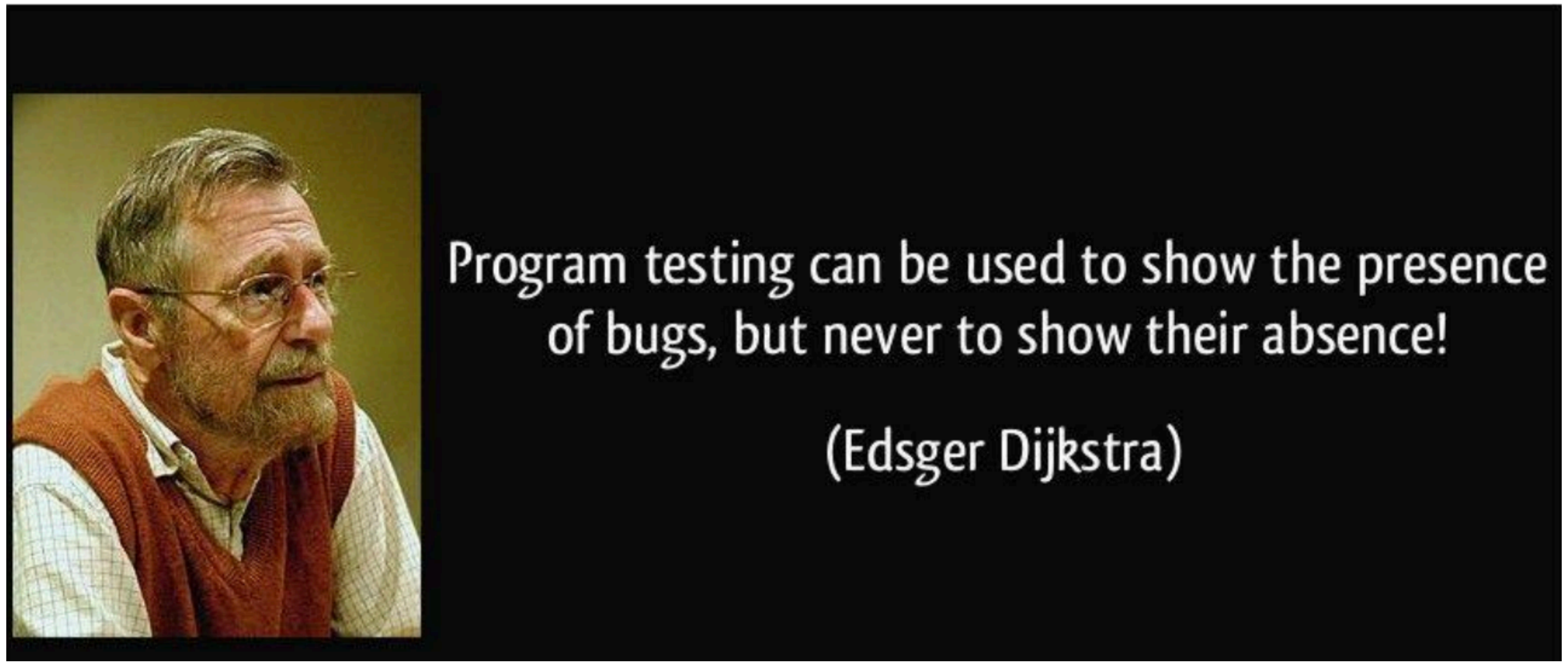
Empirical Robustness



Goodfellow et. al., Explaining and Harnessing Adversarial Examples, ICLR'15

Eykholt et. al., Robust Physical-World Attacks on Deep Learning Visual Classification, CVPR'18

Towards Certified Robustness



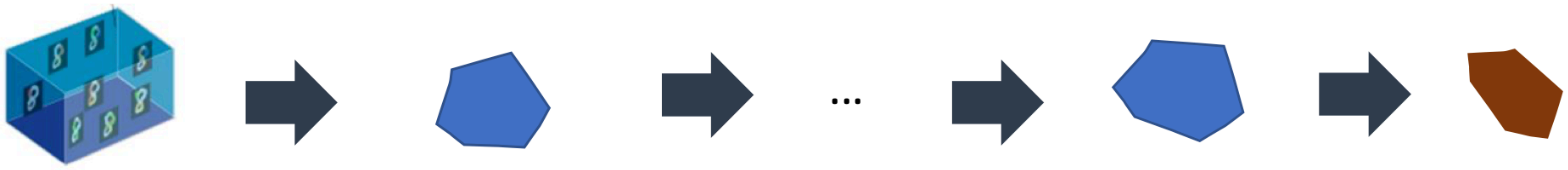
#	paper	model	clean	$APGD_{CE}$	$APGD_{DLR}$	FAB	Square	*AutoAttack*	report.	reduct.
CIFAR-10 - $\epsilon = 8/255$										
1	(Wang et al., 2019)	En_5RN	82.39 (0.14)	**48.81**	49.37	-	78.61	45.56 (0.20)	51.48	-5.9
2	(Yang et al., 2019)	with AT	84.9 (0.6)	**30.1**	31.9	-	-	26.3 (0.85)	52.8	-26.5
3	(Yang et al., 2019)	pure	87.2 (0.3)	**21.5**	24.3	-	-	18.2 (0.82)	40.8	-22.6
4	(Grathwohl et al., 2020)	JEM-10	90.99 (0.03)	**11.69**	15.88	63.07	79.32	9.92 (0.03)	47.6	-37.7
5	(Grathwohl et al., 2020)	JEM-1	92.31 (0.04)	**9.15**	13.85	62.71	79.25	8.15 (0.05)	41.8	-33.6
6	(Grathwohl et al., 2020)	JEM-0	92.82 (0.05)	**7.19**	12.63	66.48	73.12	6.36 (0.06)	19.8	-13.4
CIFAR-10 - $\epsilon = 4/255$										
1	(Grathwohl et al., 2020)	JEM-10	91.03 (0.05)	**49.10**	52.55	78.87	89.32	47.97 (0.05)	72.6	-24.6
2	(Grathwohl et al., 2020)	JEM-1	92.34 (0.04)	**46.08**	49.71	78.93	90.17	45.49 (0.04)	67.1	-21.6
3	(Grathwohl et al., 2020)	JEM-0	92.82 (0.02)	**42.98**	47.74	82.92	89.52	42.55 (0.07)	50.8	-8.2

Croce et. al., Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks, ICML'20

Part 1

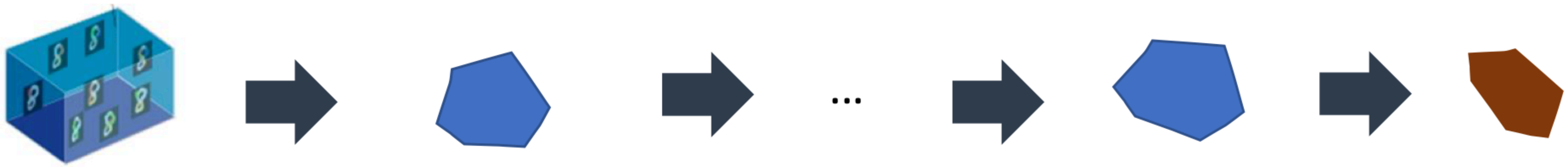
A Quick Start to Neural Network Verification

The concept of Verification



Katz et. al., Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, CAV'17

The concept of Verification



Sound: if verified, then must be correct; if not verified, potentially be correct/incorrect.

Katz et. al., Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, CAV'17

The concept of Verification

Sound: if verified, then must be correct; if not verified, potentially be correct/incorrect.

Complete: if correct, then must be verified.

Katz et. al., Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, CAV'17

The concept of Verification

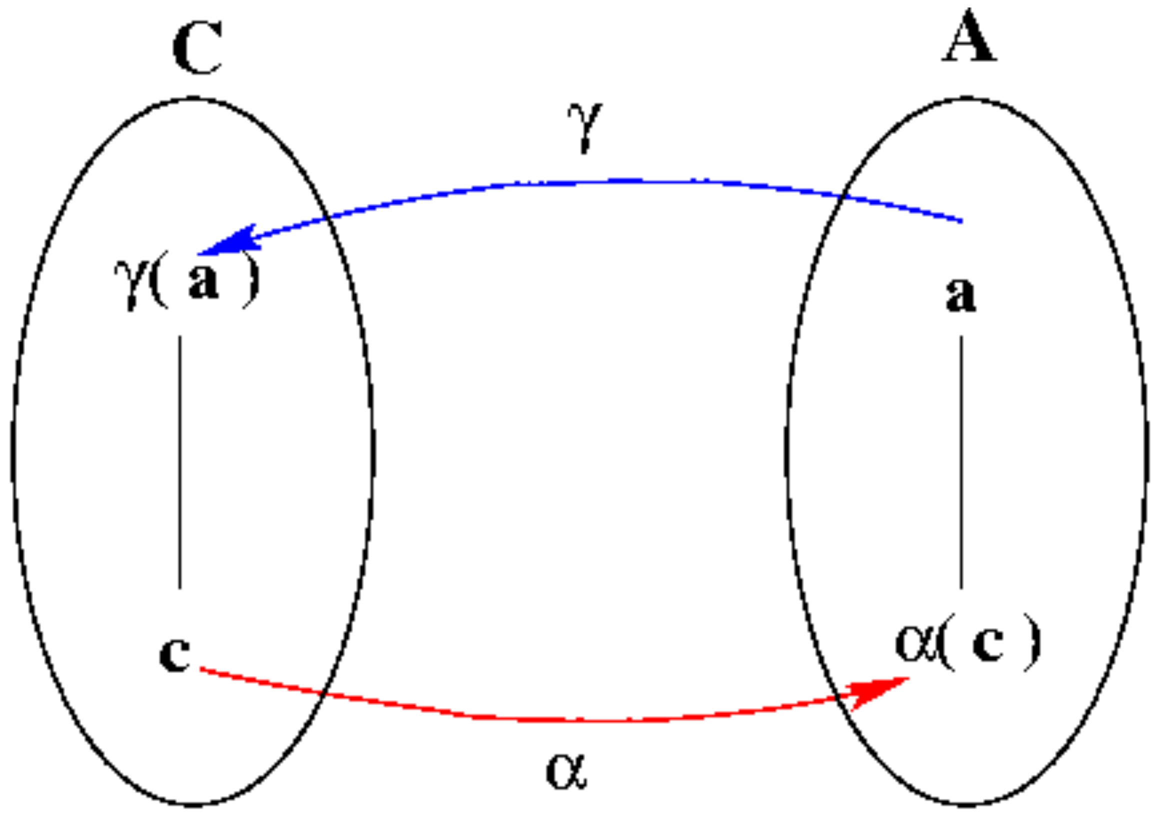
Sound: if verified, then must be correct; if not verified, potentially be correct/incorrect.

Complete: if correct, then must be verified.

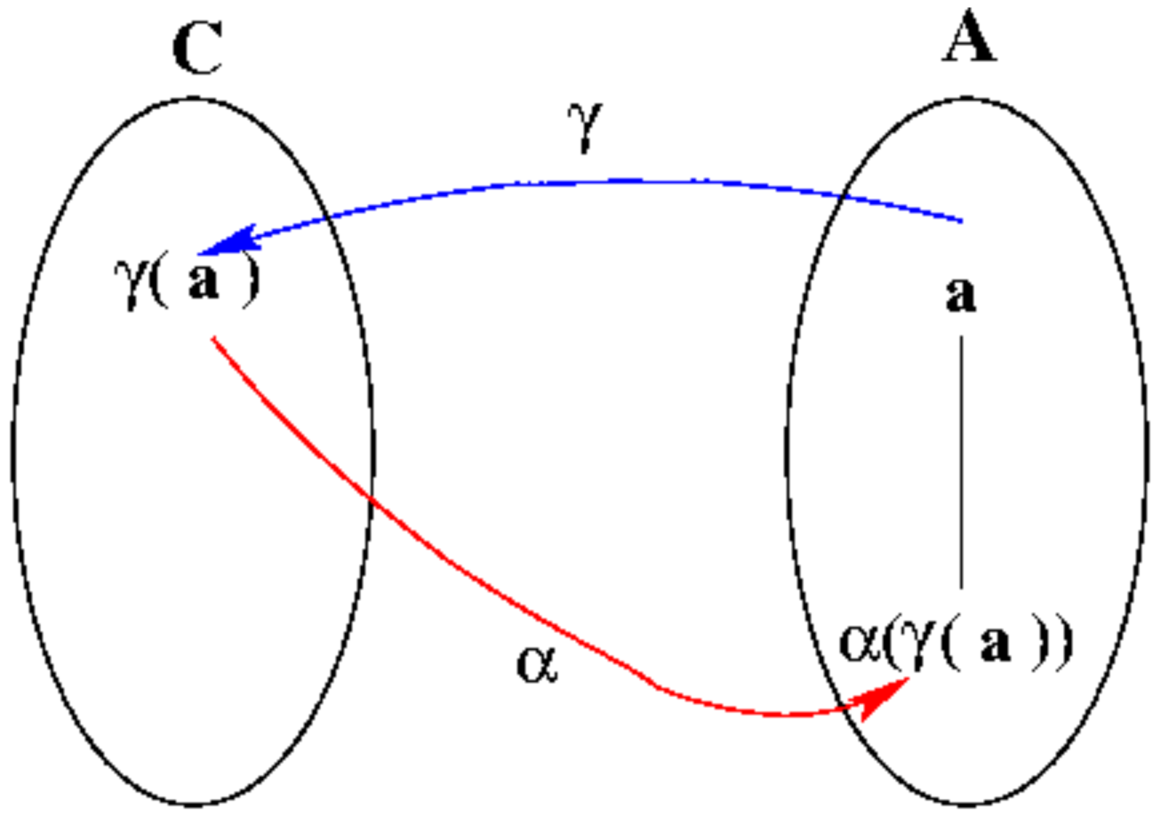
Complete and sound is desirable: but NP-hard in neural network verification.

Katz et. al., Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, CAV'17

Abstract Interpretation

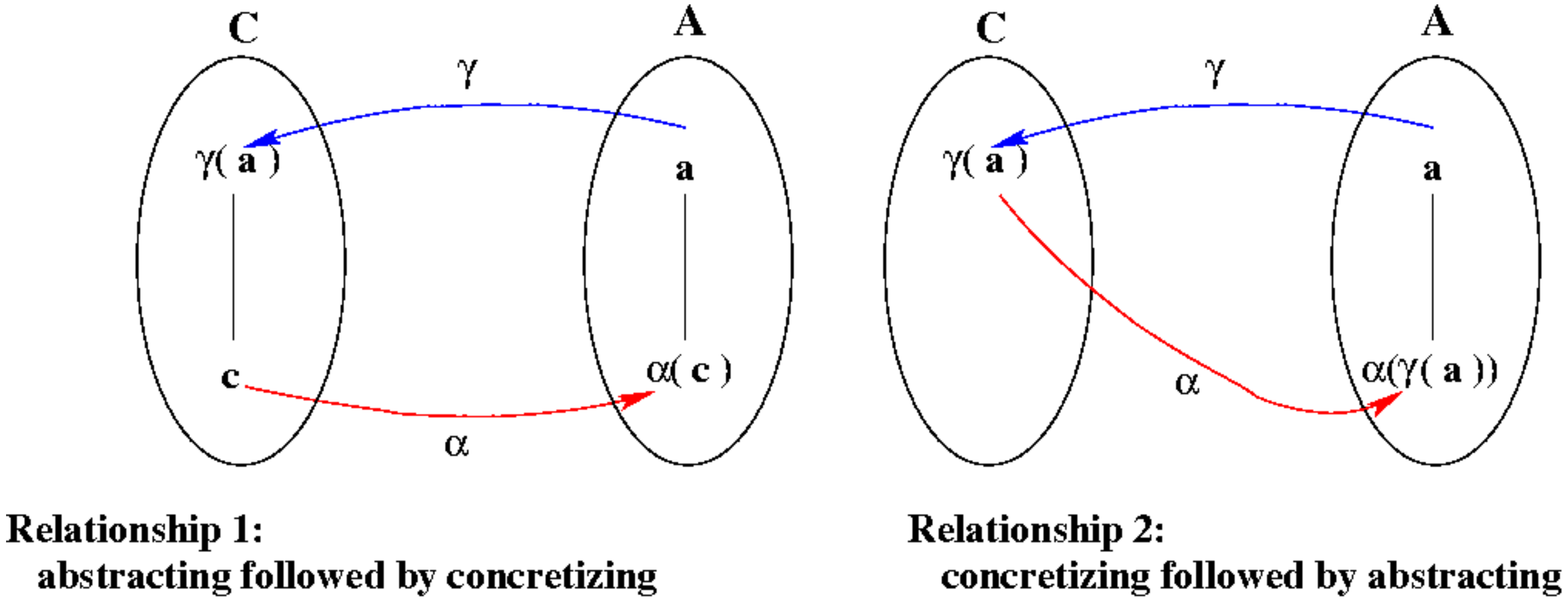


Relationship 1:
abstracting followed by concretizing



Relationship 2:
concretizing followed by abstracting

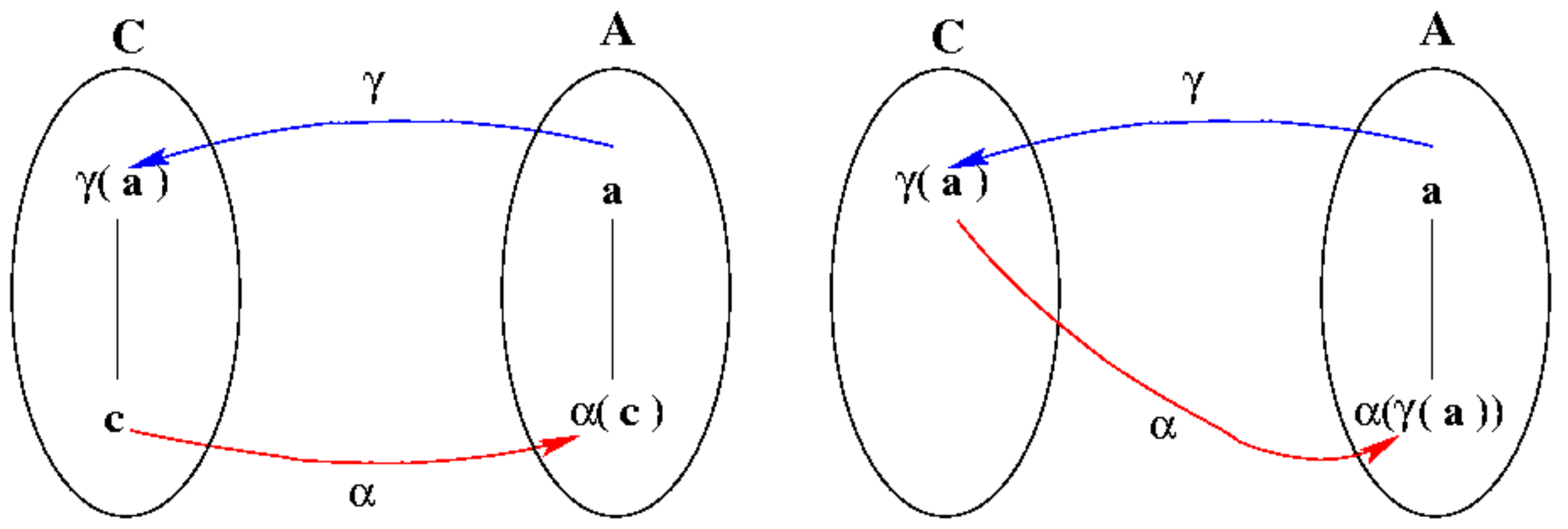
Abstract Interpretation



Poison Test: find a poisonous bottle inside N bottles.

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



Relationship 1:
abstracting followed by concretizing

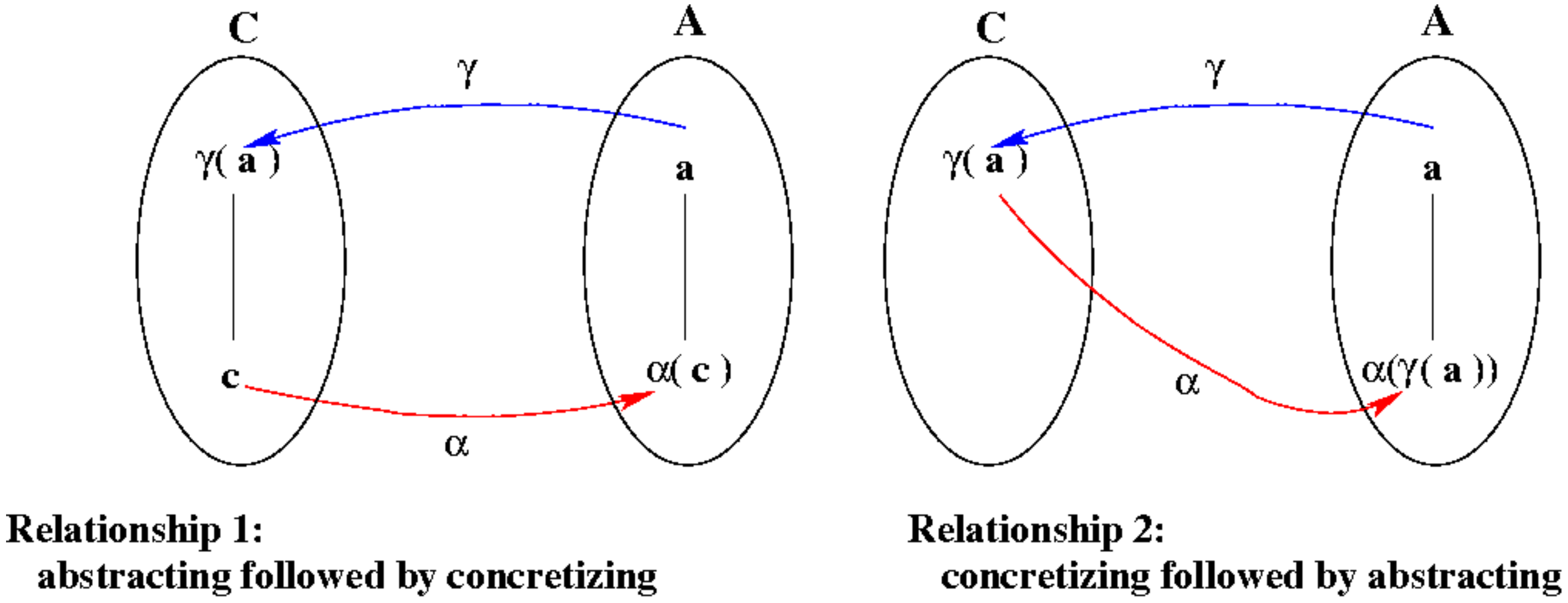
Relationship 2:
concretizing followed by abstracting

Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



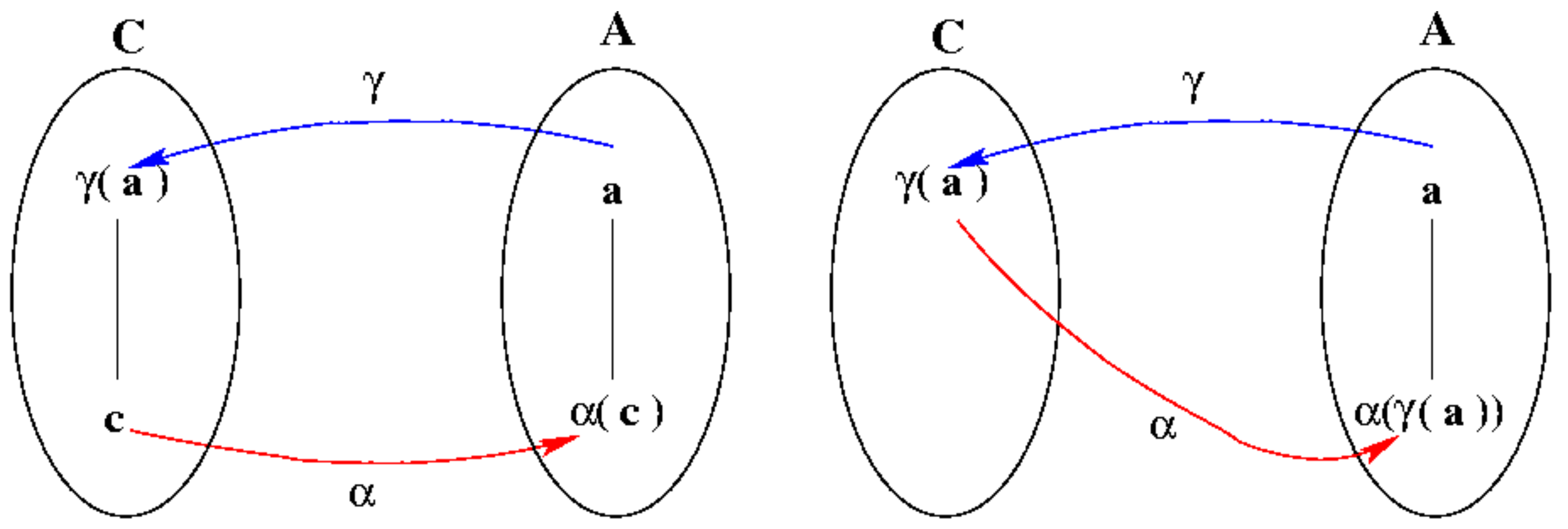
Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:

Positive -> contain poison

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



Relationship 1:
abstracting followed by concretizing

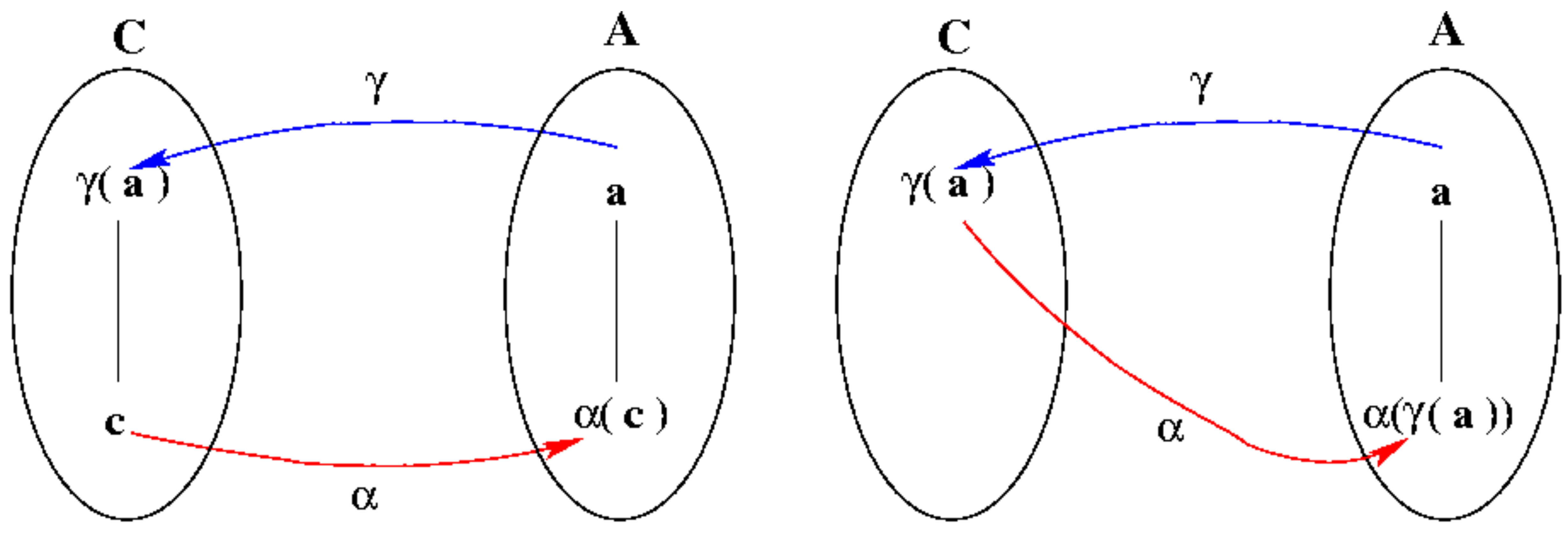
Relationship 2:
concretizing followed by abstracting

Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:
Positive -> contain poison
Negative -> no poison

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



Relationship 1:
abstracting followed by concretizing

Relationship 2:
concretizing followed by abstracting

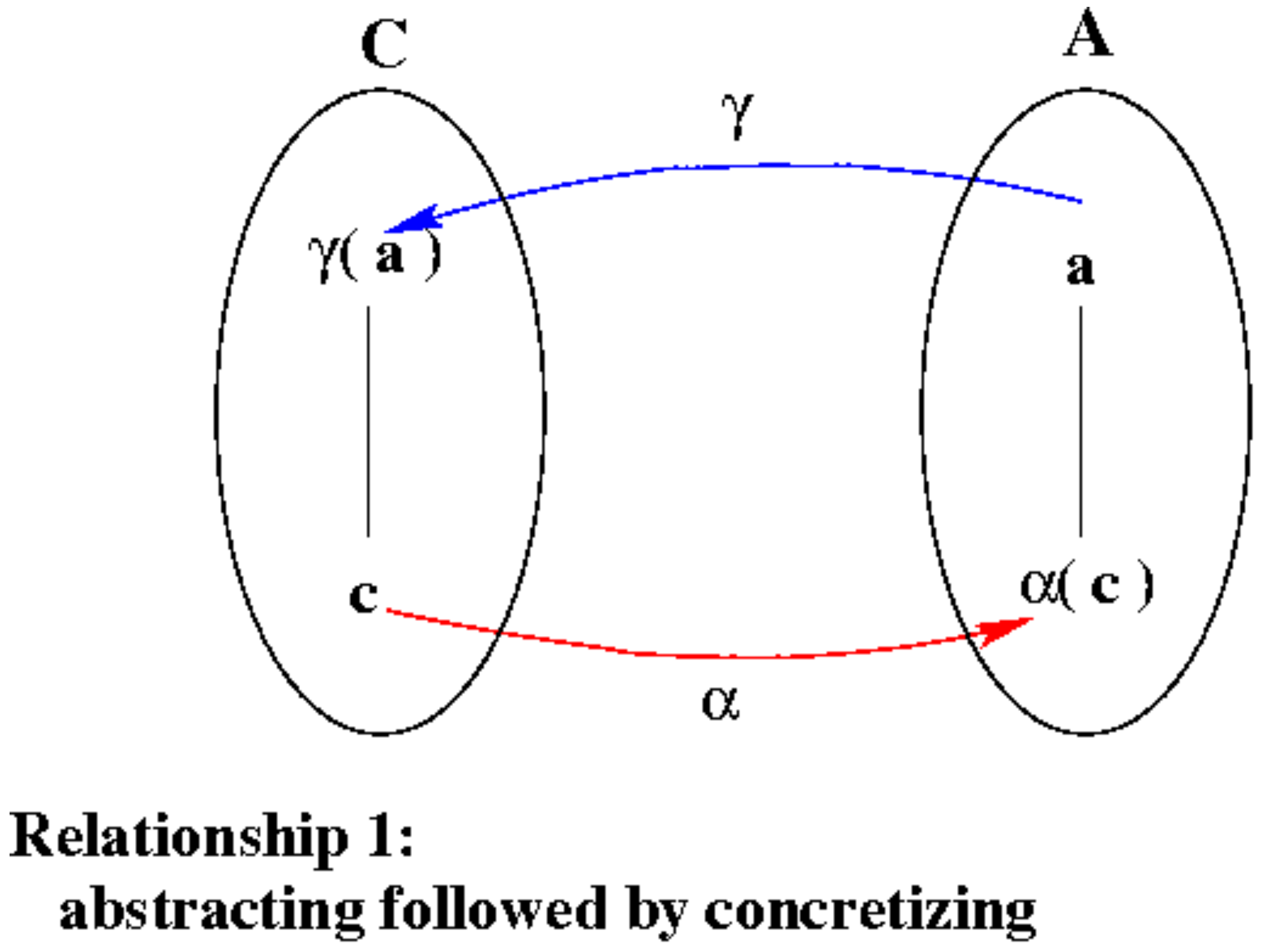
Verify that the following program never throws type error:

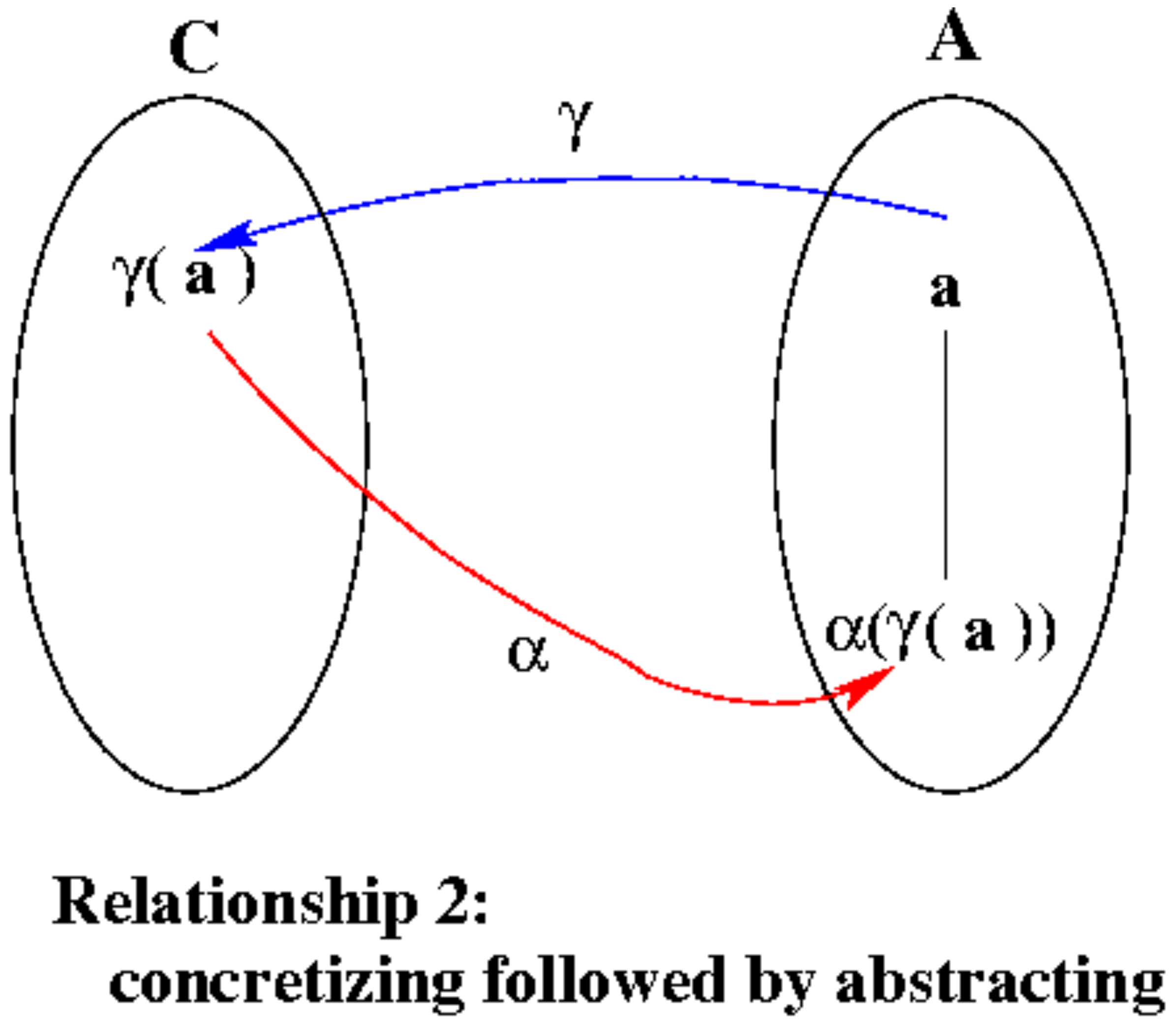
Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:
Positive -> contain poison
Negative -> no poison

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation





Verify that the following program never throws type error:

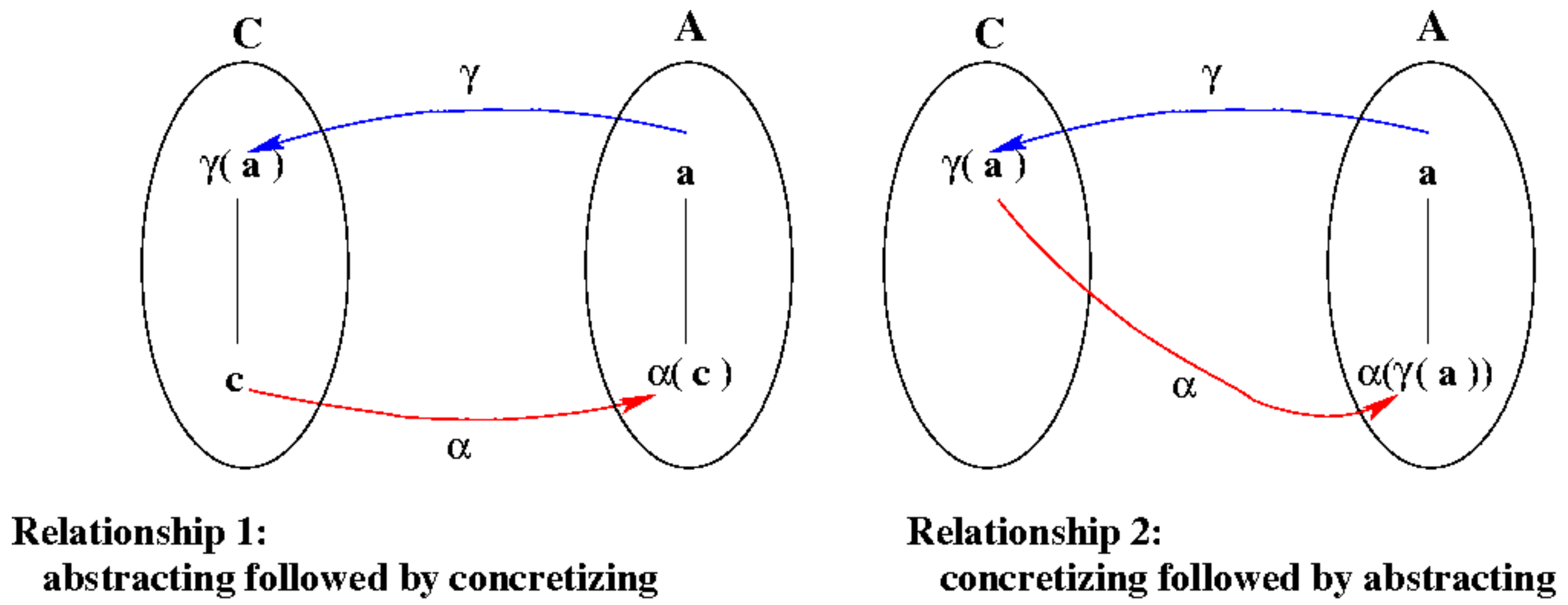
```
int x, y, z;
z = x+y;
```

Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:
Positive -> contain poison
Negative -> no poison

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



Verify that the following program never throws type error:

```
int x, y, z;
z = x+y;
```

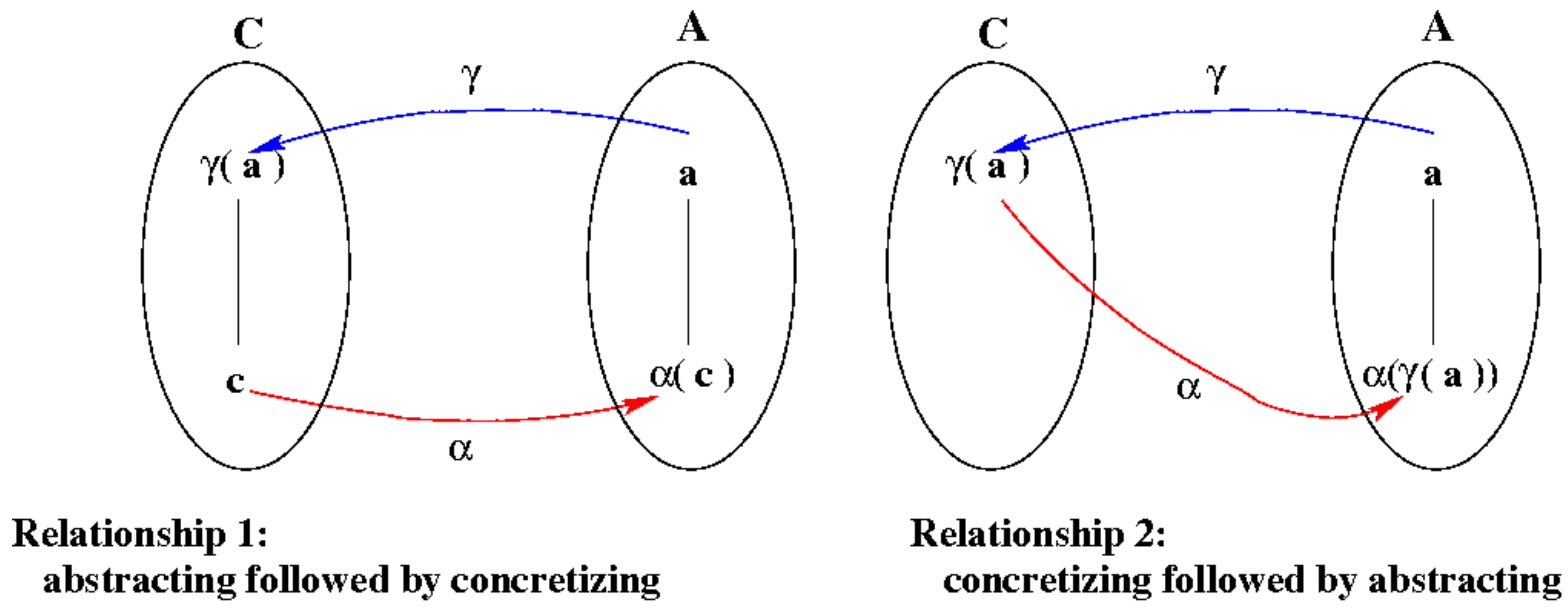
x -> int

Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:
Positive -> contain poison
Negative -> no poison

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



Verify that the following program never throws type error:

```
int x, y, z;
z = x+y;
```

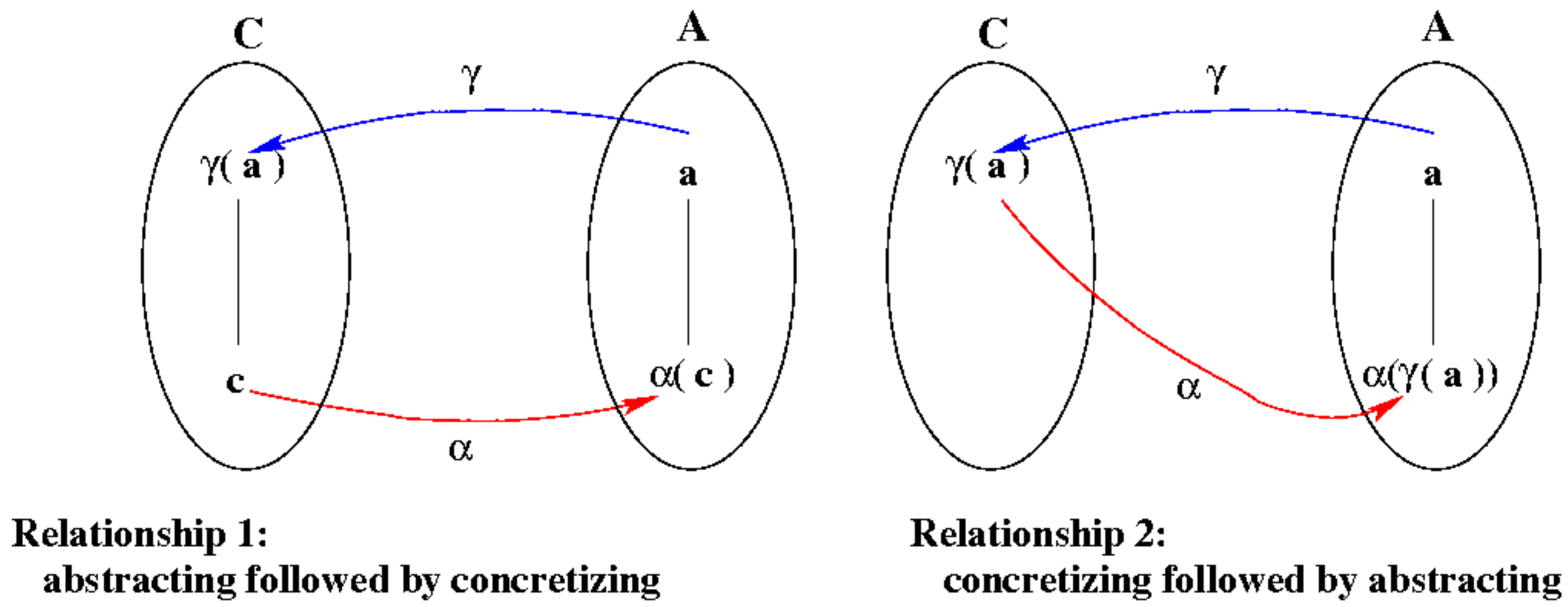
x -> int
y -> int

Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:
Positive -> contain poison
Negative -> no poison

https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation



Verify that the following program never throws type error:

int x, y, z;
z = x+y;

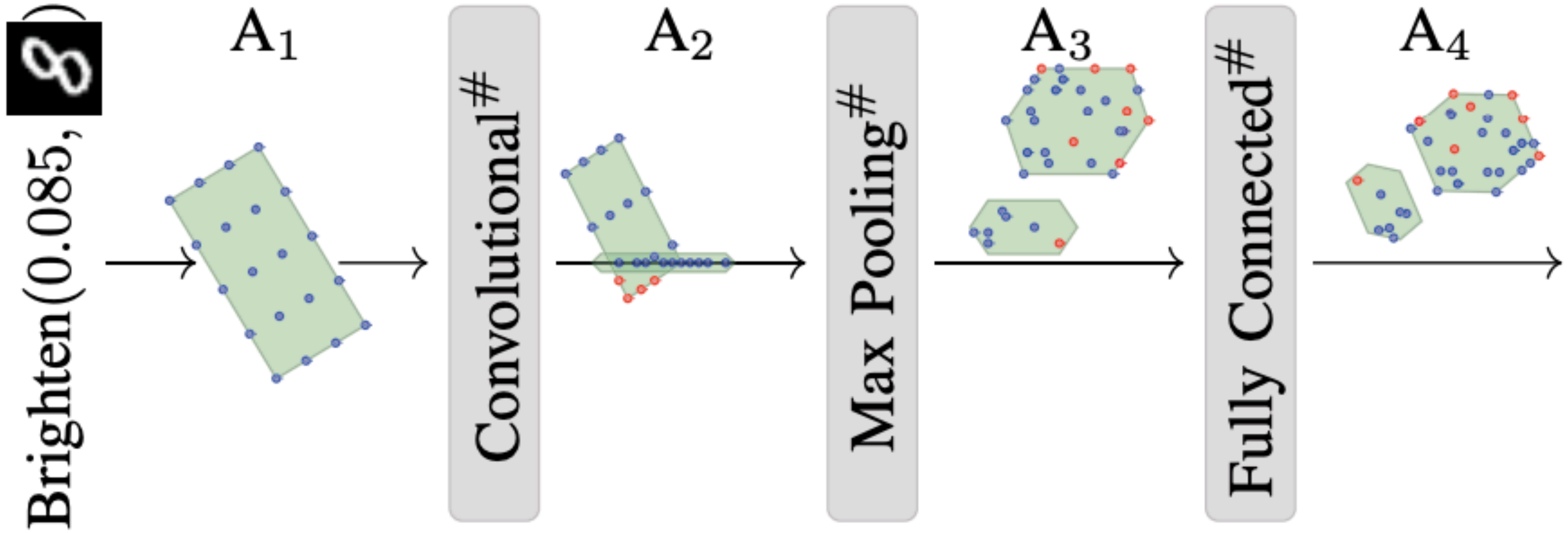
x -> int
y -> int
int + int -> int

Poison Test: find a poisonous bottle inside N bottles.

Randomly mix N/2 bottles and test:
Positive -> contain poison
Negative -> no poison

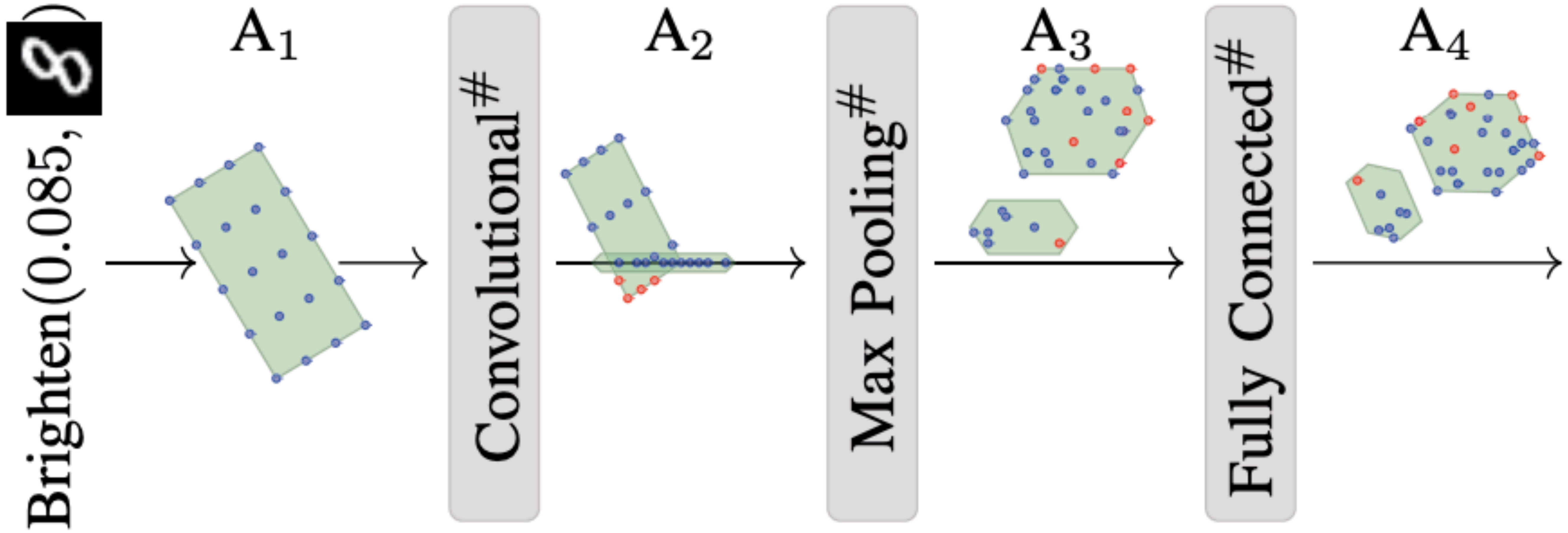
https://pages.cs.wisc.edu/~horwitz/CS704-NOTES/10.ABSTRACT-INTERPRETATION.html

Abstract Interpretation for Neural Net



Gehr et. al., AI[2]: Safety and Robustness Certification of Neural Networks with Abstract Interpretation, SP'18

Abstract Interpretation for Neural Net



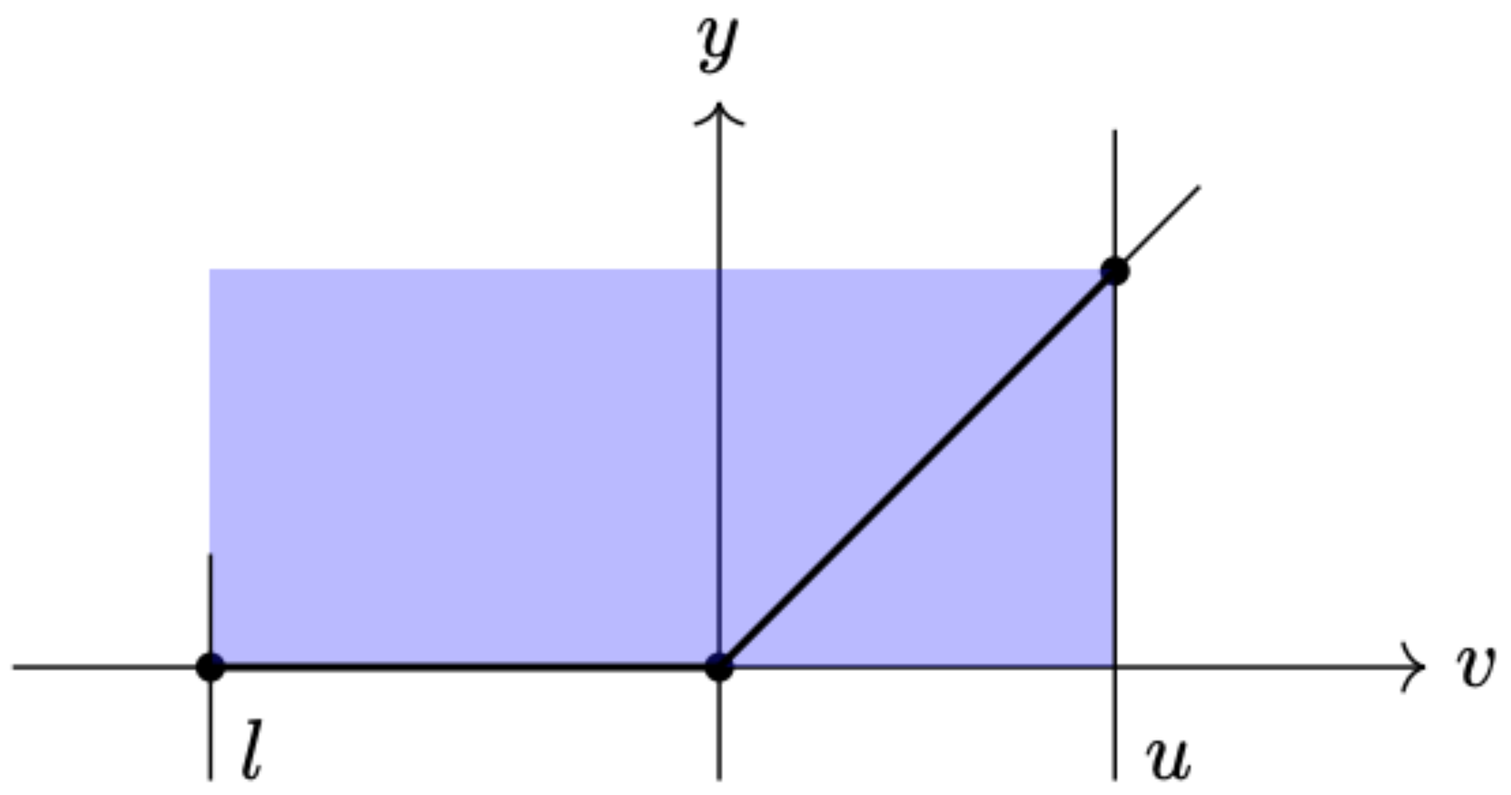
Sound but incomplete

Gehr et. al., AI[2]: Safety and Robustness Certification of Neural Networks with Abstract Interpretation, SP'18

Box Domain

Relax the exact set as a hyper-box (interval).

Imprecise for both linear and ReLU layers.



$$[a, b] + [c, d] = [a + b, c + d]$$

$$[a, b] - [c, d] = [a - d, b - c]$$

$$ReLU([a, b]) = [ReLU(a), ReLU(b)]$$

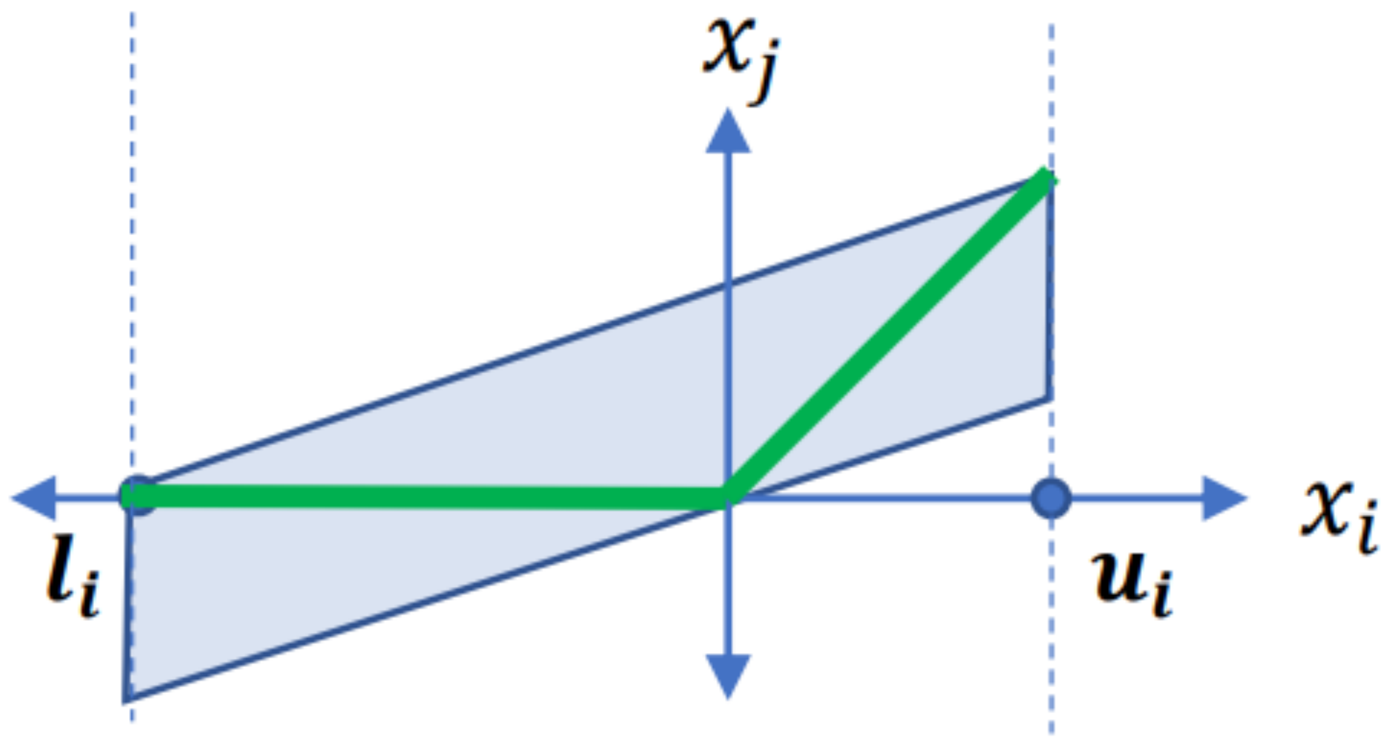
Gehr et. al., AI^2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation, SP'18

Zonotope Domain

Relax the exact set as a zonotope.

Precise for linear but imprecise for ReLU layers.

$$\boldsymbol{a}^\top \boldsymbol{x} + b + \boldsymbol{c}^\top \boldsymbol{e}, \boldsymbol{e} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]$$

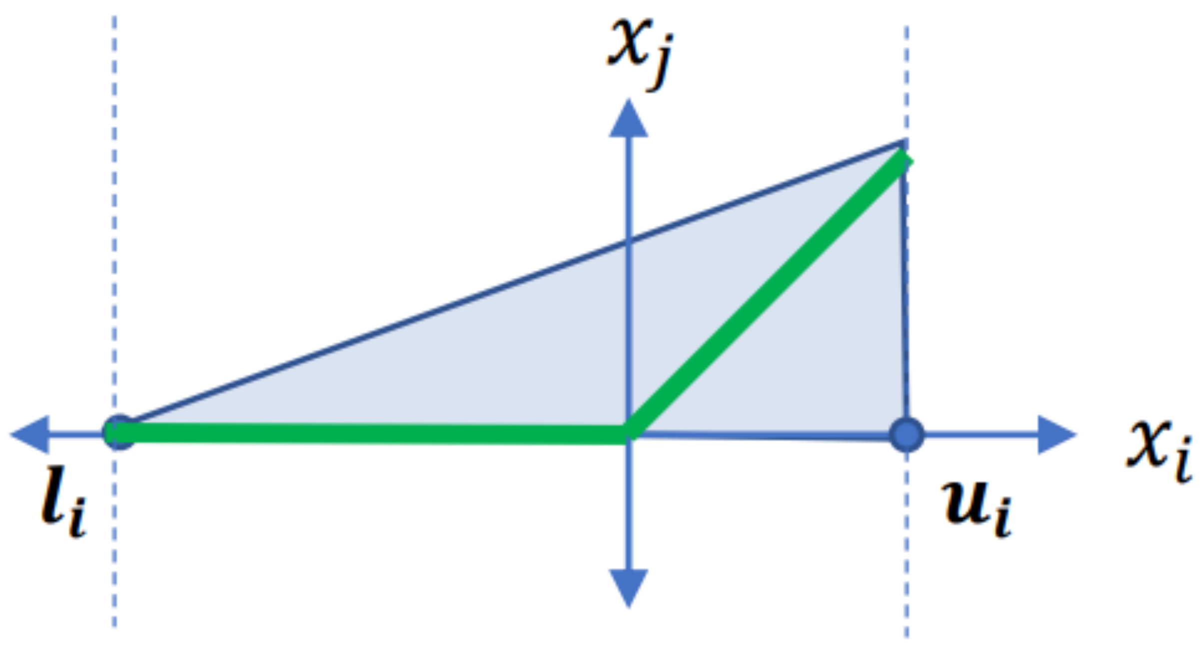


Wong et. al., Provable defenses against adversarial examples via the convex outer adversarial polytope, ICML'18

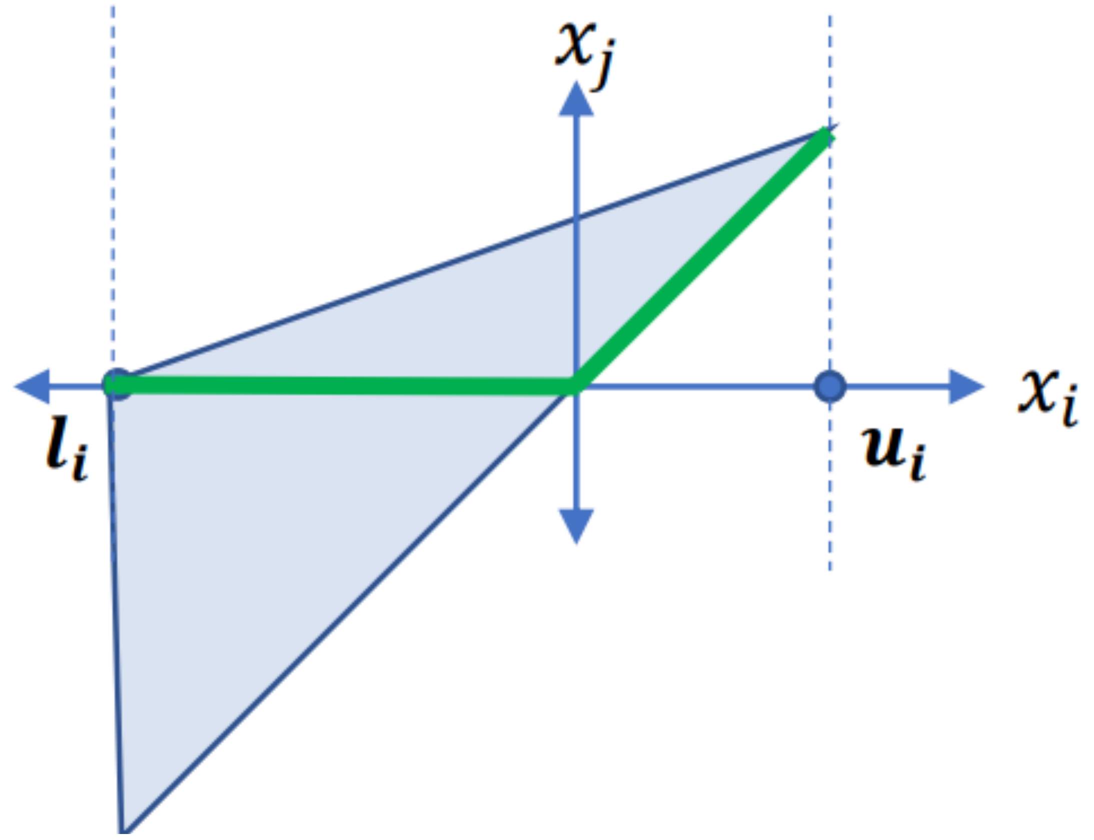
DeepPoly/CROWN Domain

Relax the exact set as linear constraints.

Precise for linear but imprecise for ReLU layers.



$$ReLU(x) \geq 0, \quad ReLU(x) \leq \frac{u}{u-l}(x-l)$$



$$ReLU(x) \geq x, \quad ReLU(x) \leq \frac{u}{u-l}(x-l)$$

Singh et. al., An Abstract Domain for Certifying Neural Networks, POPL'19

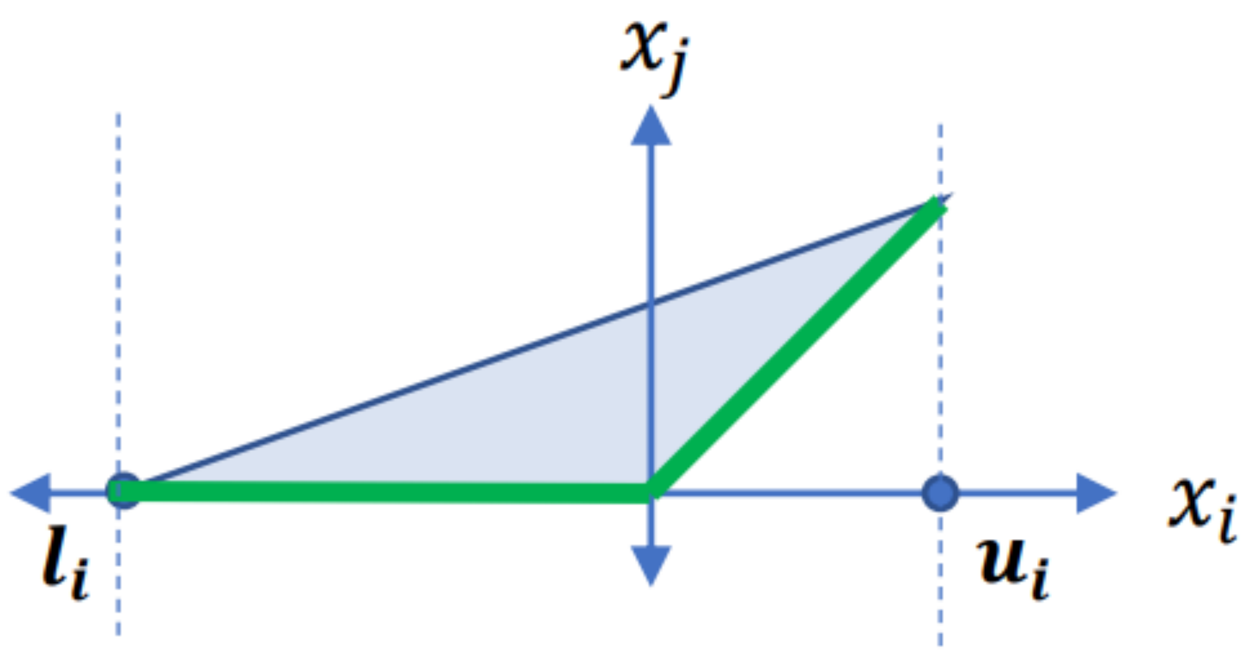
Zhang el. al., Efficient Neural Network Robustness Certification with General Activation Functions, NeurIPS'18

Triangle Domain

Relax the exact set as linear constraints.

Precise for linear but imprecise for ReLU layers.

The most precise convex domain.



$$ReLU(x) \geq 0,$$
$$ReLU(x) \geq x,$$
$$ReLU(x) \leq \frac{u}{u-l}(x-l)\,.$$

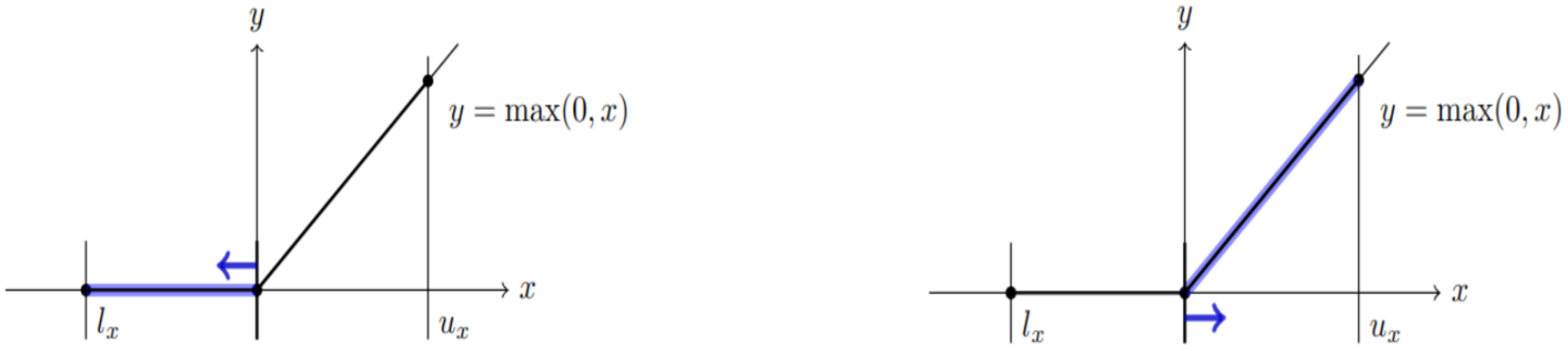
Ehler, Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks, ATVA'17

Complete Verification

Encode the ReLU as a Mixed Integer Linear Programming (MILP).

Complete but NP-hard to solve.

Branch-and-Bound (BaB) for solving.



Bunel et. al., Branch and Bound for Piecewise Linear Neural Network Verification, JMLR'20

Scale of Verification: VNN'22

	Name	Network Type	# Parms	# Neurons	Input Dim	Domain
Complex	Carvana UNet	Complex U-Net	150k - 330 k	275k - 373k	5828	BaB* with DeepPoly
	VGGNet 16	Conv + ReLU + MaxPool	138M	13.6 M	164k	Box + DeepPoly
CNN / ResNet	Cifar Biasfield	Conv + ReLU	363k	45k	16	BaB* with DeepPoly
	Large ResNet	ResNet (Conv + ReLU)	1.3M - 7.9M	55k - 286k	3k-9k	BaB* with DeepPoly
	Collins Rul CNN	Conv + ReLU	60k - 262k	5.5k - 28k	400-800	BaB* with DeepPoly
	oval21	Conv + ReLU	54k - 214k	3.1k - 6.2k	3072	BaB* with DeepPoly
	ResNet A/B	ResNet (Conv + ReLU)	354k	11k	3072	BaB* with DeepPoly
FC	MNIST FC	FC + ReLU	270k - 530k	512 - 1536	784	BaB* with DeepPoly + MILP refinement

* BaB is implemented via KKT

Take-away

Take-away

- Neural network verification is challenging: a general network is NP-hard to verify.

Take-away

- Neural network verification is challenging: a general network is NP-hard to verify.

- Many abstract domains are designed to scale the verification in the cost of completeness.

Take-away

- Neural network verification is challenging: a general network is NP-hard to verify.
- Many abstract domains are designed to scale the verification in the cost of completeness.
- In general, more precise domains require more space and more computation, thus less scalable.

Part 2

Connecting Certified and Adversarial Training

Training for Robustness

Training for Robustness

Expected Loss

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{x}, y)$$

Training for Robustness

Expected Loss

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{x}, y)$$

Training for Robustness

Expected Loss

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{x}, y)$$

Expected Worst-case Loss

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} \left[\max_{\boldsymbol{x}' \in B(\boldsymbol{x},\epsilon)} L\left(\boldsymbol{x}', y\right) \right]$$

Training for Robustness

Expected Loss

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{x}, y)$$

NP-hard

Expected Worst-case Loss

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} \left[\max_{\boldsymbol{x}' \in B(\boldsymbol{x},\epsilon)} L\left(\boldsymbol{x}', y\right) \right]$$

Adversarial Training

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} \left[L\left(\boldsymbol{x}', y\right) \right]$$

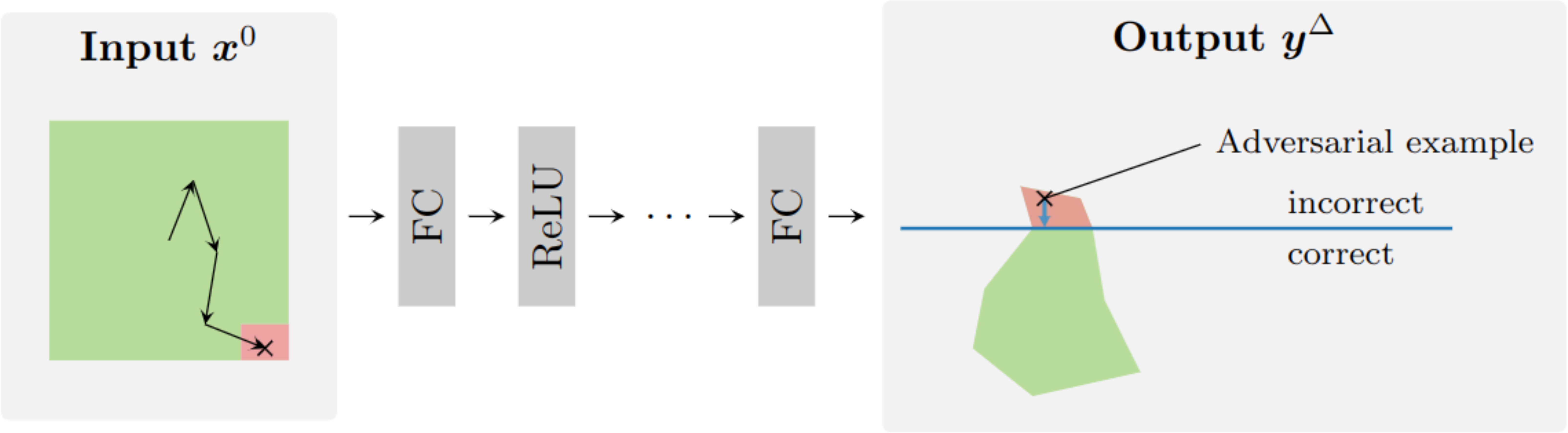
$$\boldsymbol{x}' \in B(\boldsymbol{x}, \epsilon)$$

Adversarial Training

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y}\left[L\left(\boldsymbol{x}', y\right)\right]$$

$$\boldsymbol{x}' \in B(\boldsymbol{x}, \epsilon)$$

PGD



Certified Training

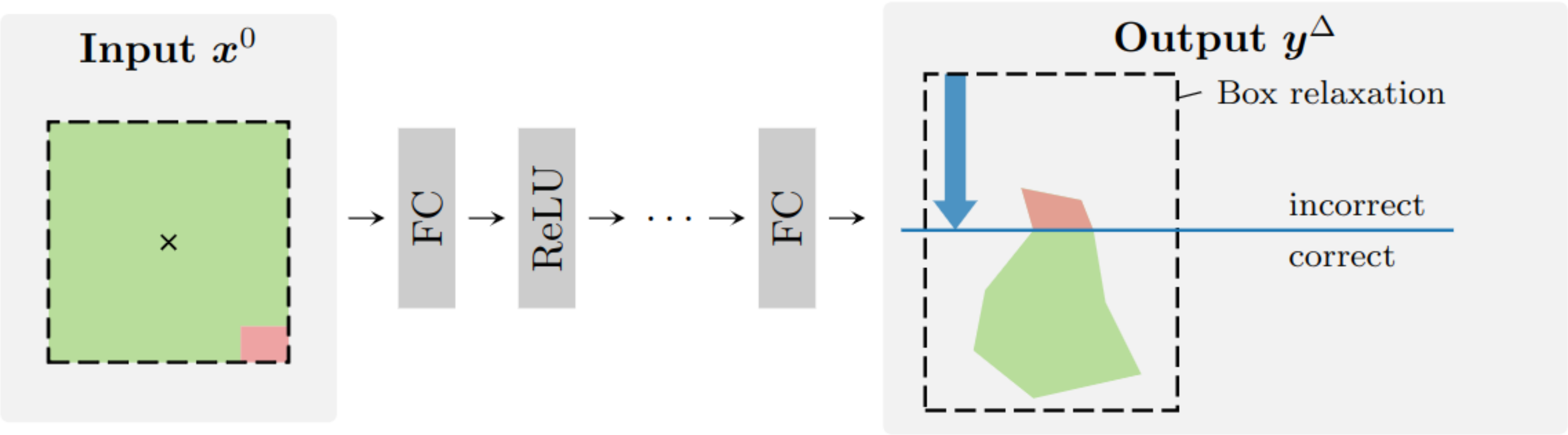
Certified Training

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} \left[L\left(B(\boldsymbol{x}, \epsilon), y \right) \right]$$

Certified Training

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y}\left[L\left(B(\boldsymbol{x},\epsilon),y\right)\right]$$

IBP



Research Question

Gowal et al. "Scalable verified training for provably robust image classification." ICCV 2019.
Mirmann et al. "Differentiable abstract interpretation for provably robust neural networks." ICML 2018.
Shi et al. "Fast certified robust training with short warmup." NeuIPS 2021.

Research Question

- Adversarial training has good empirical robustness, but is hard to certify.

Gowal et al. "Scalable verified training for provably robust image classification." ICCV 2019.
Mirmann et al. "Differentiable abstract interpretation for provably robust neural networks." ICML 2018.
Shi et al. "Fast certified robust training with short warmup." NeuIPS 2021.

Research Question

- Adversarial training has good empirical robustness, but is hard to certify.
- Certified Training (**I**nterval **B**ound **P**ropagation, SOTA in 2021) has good certified robustness, but at the cost of greatly reduced standard accuracy.

Gowal et al. "Scalable verified training for provably robust image classification." ICCV 2019.
Mirmann et al. "Differentiable abstract interpretation for provably robust neural networks." ICML 2018.
Shi et al. "Fast certified robust training with short warmup." NeuIPS 2021.

Research Question

- Adversarial training has good empirical robustness, but is hard to certify.
- Certified Training (**I**nterval **B**ound **P**ropagation, SOTA in 2021) has good certified robustness, but at the cost of greatly reduced standard accuracy.
- Can we combine these two, so that we have both better certified robustness and better standard accuracy than IBP?

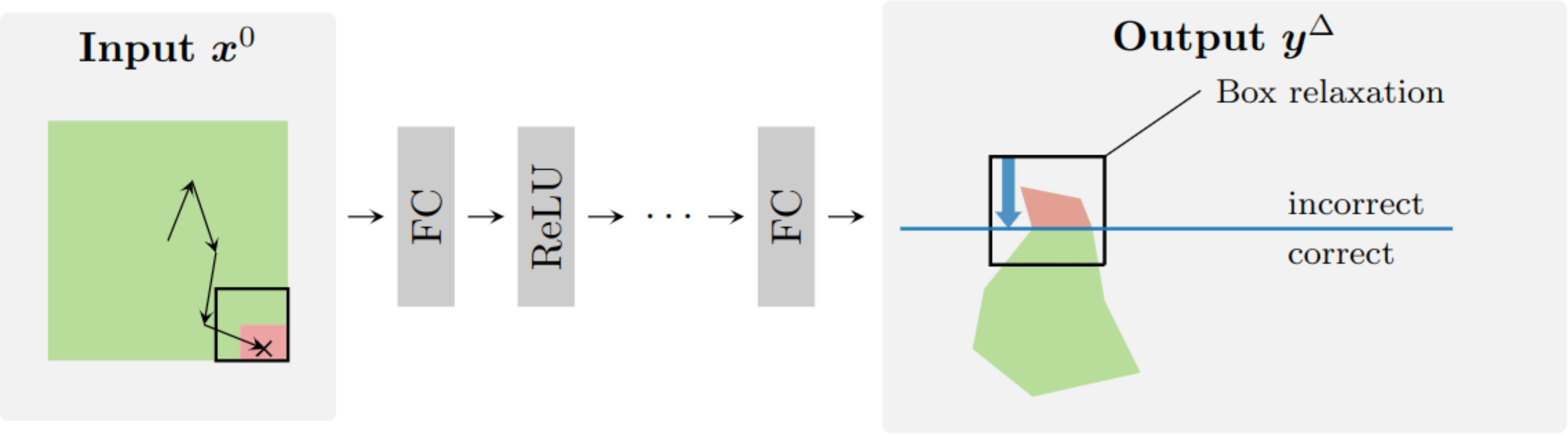
Gowal et al. "Scalable verified training for provably robust image classification." ICCV 2019.
Mirmann et al. "Differentiable abstract interpretation for provably robust neural networks." ICML 2018.
Shi et al. "Fast certified robust training with short warmup." NeuIPS 2021.

Research Question

- Adversarial training has good empirical robustness, but is hard to certify.
- Certified Training (**I**nterval **B**ound **P**ropagation, SOTA in 2021) has good certified robustness, but at the cost of greatly reduced standard accuracy.
- Can we combine these two, so that we have both better certified robustness and better standard accuracy than IBP?
- The answer is YES!

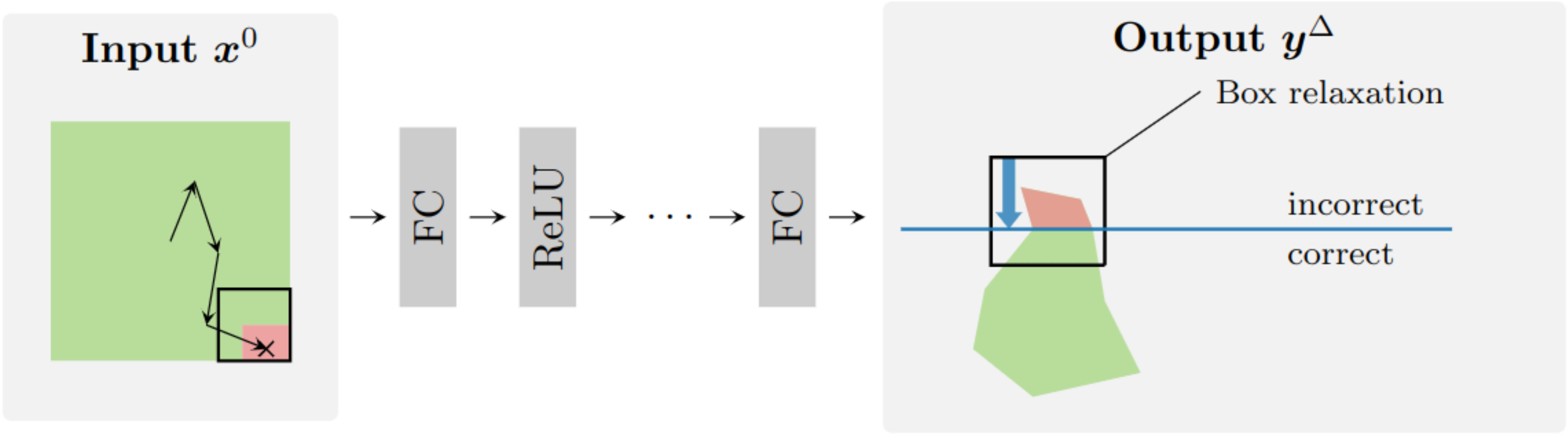
Gowal et al. "Scalable verified training for provably robust image classification." ICCV 2019.
Mirmann et al. "Differentiable abstract interpretation for provably robust neural networks." ICML 2018.
Shi et al. "Fast certified robust training with short warmup." NeuIPS 2021.

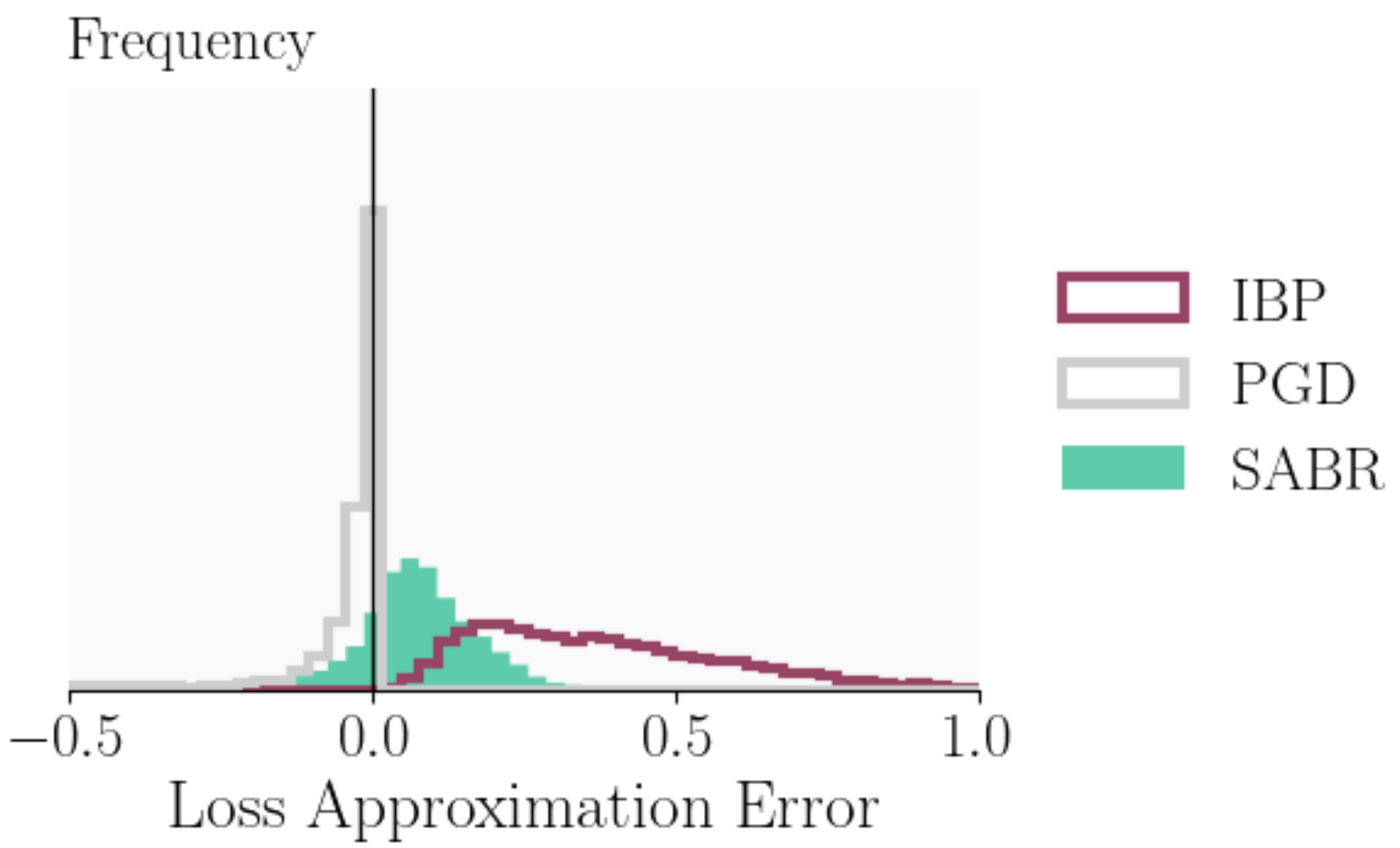
Small Adversarial Bound Regions



Müller et al. "Certified Training: Small Boxes are All You Need." ICLR 2023.

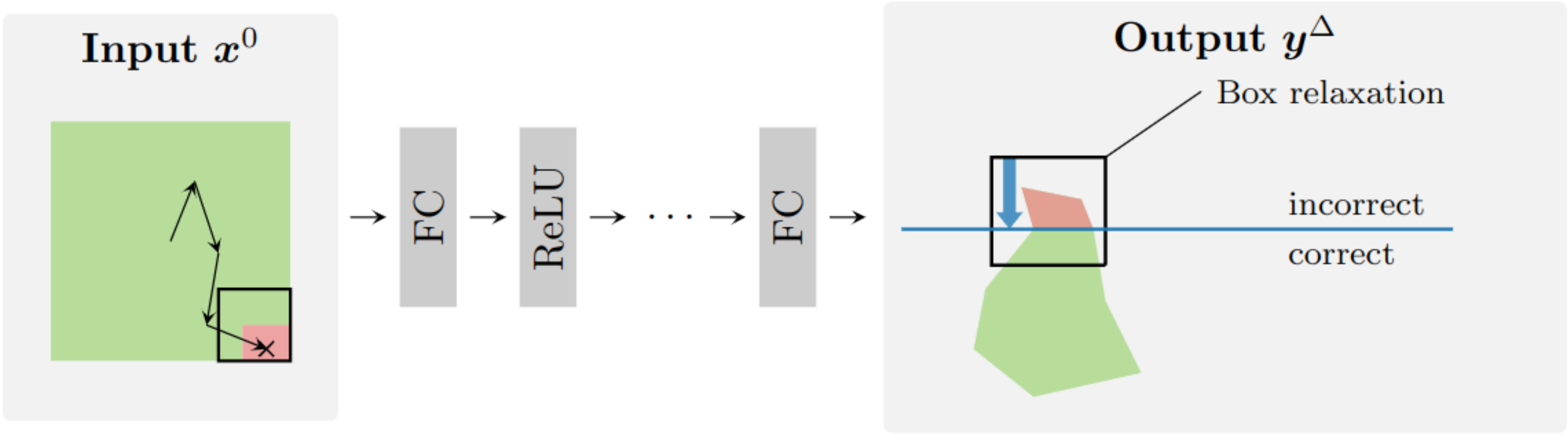
Small Adversarial Bound Regions

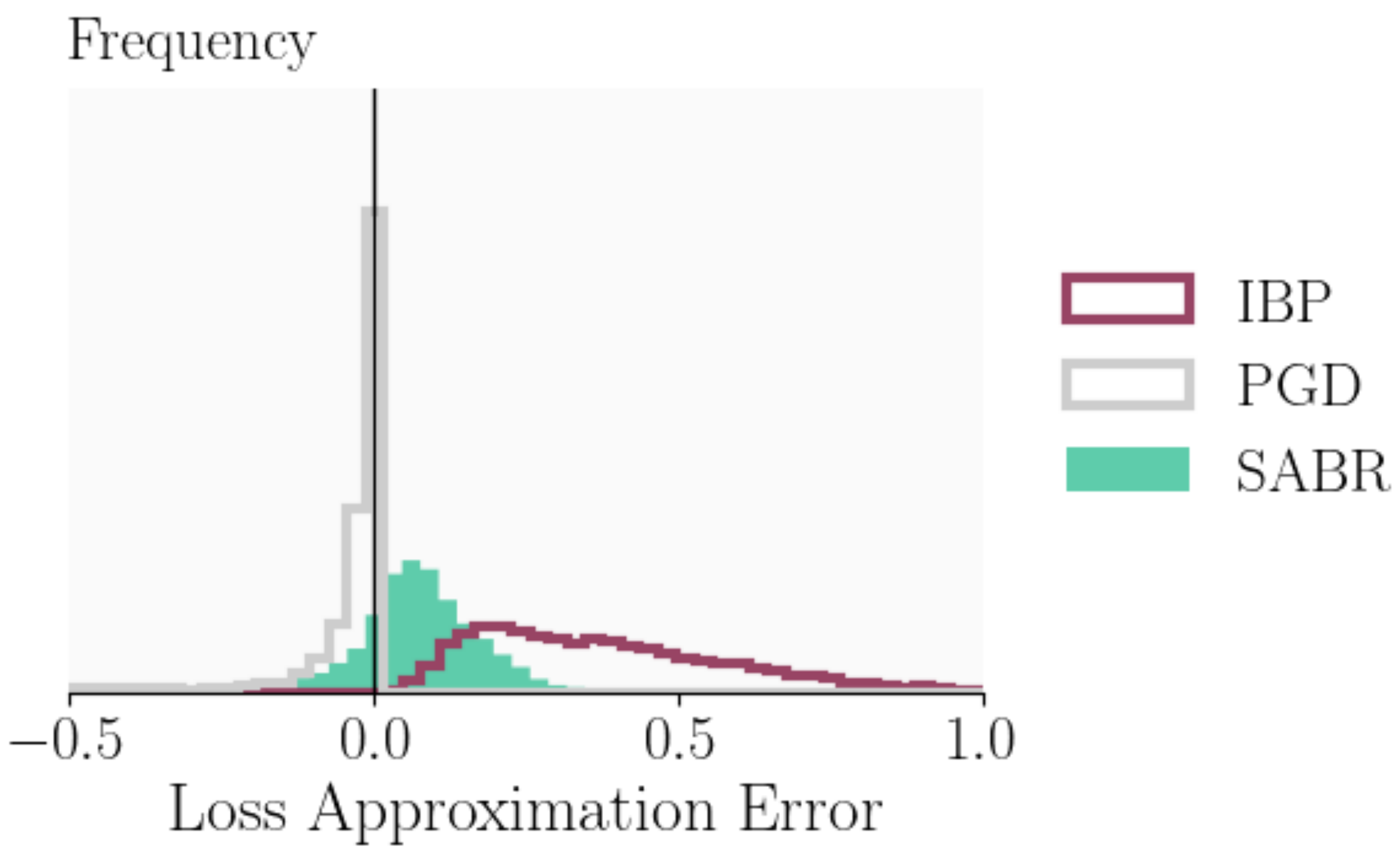


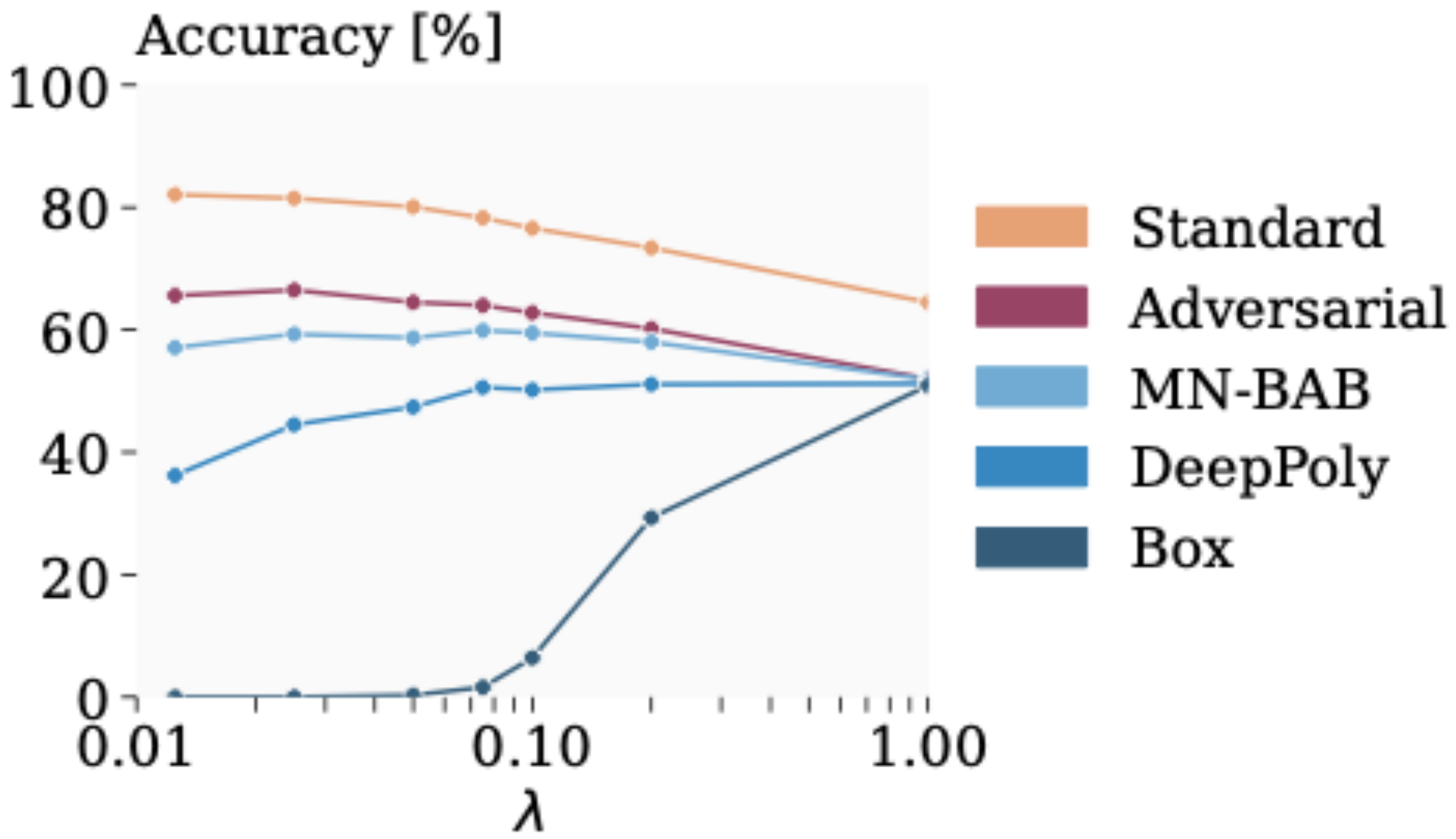


Müller et al. "Certified Training: Small Boxes are All You Need." ICLR 2023.

Small Adversarial Bound Regions



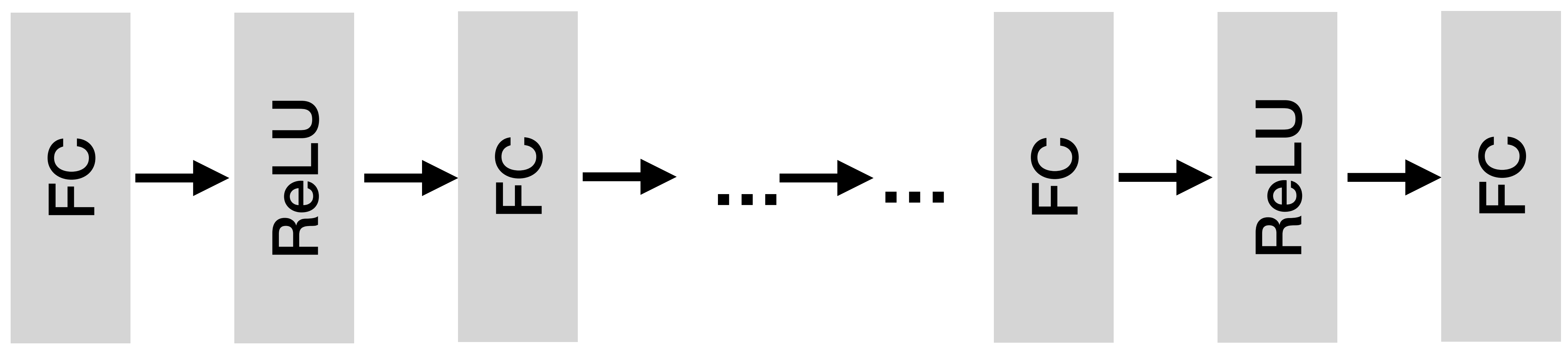




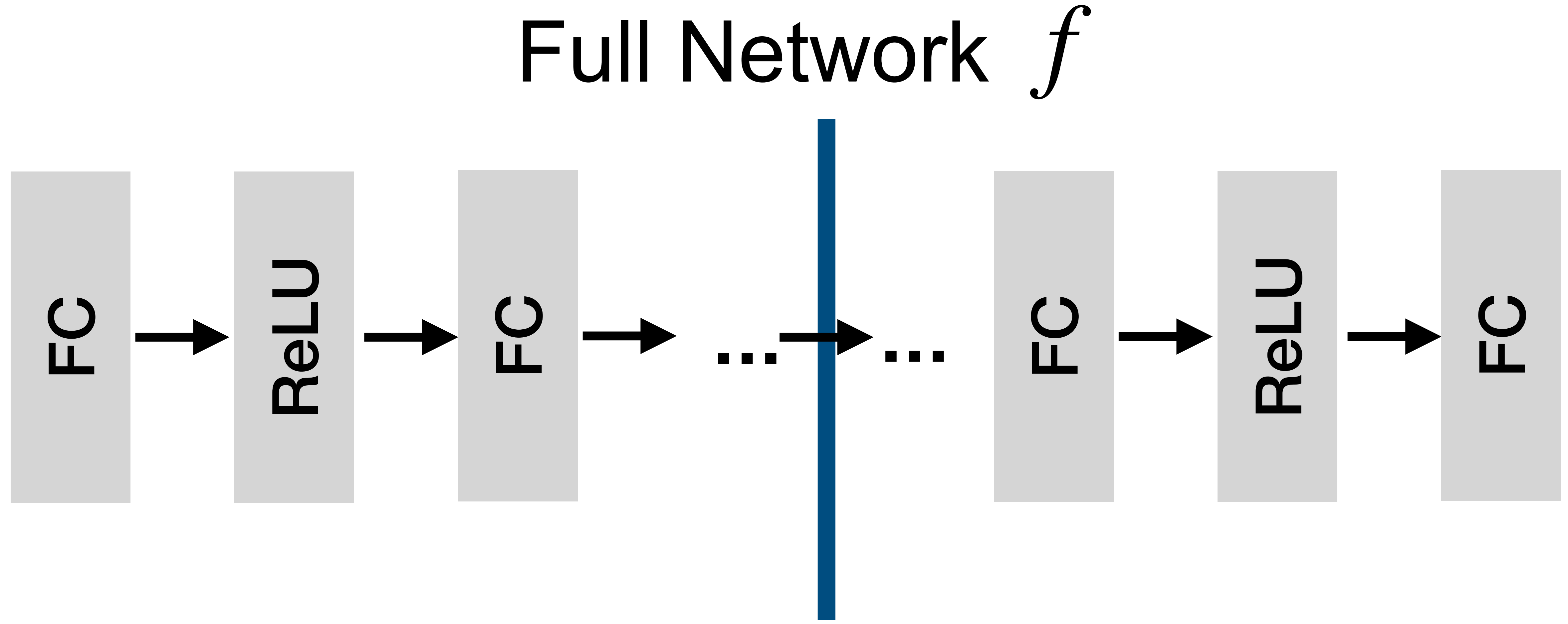
Müller et al. "Certified Training: Small Boxes are All You Need." ICLR 2023.

Training via Adversarially Propagating Subnetworks

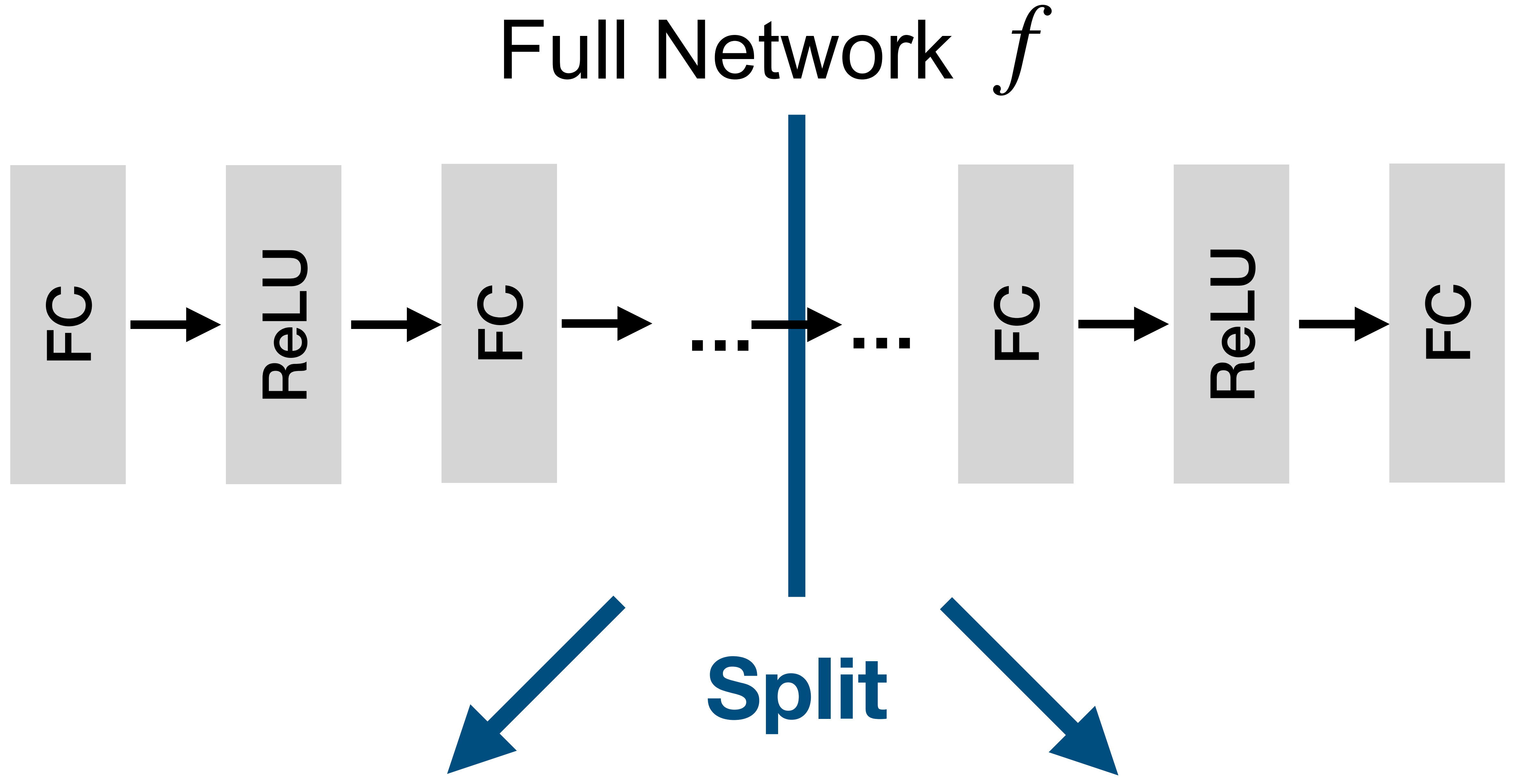
Full Network f



Training via Adversarially Propagating Subnetworks

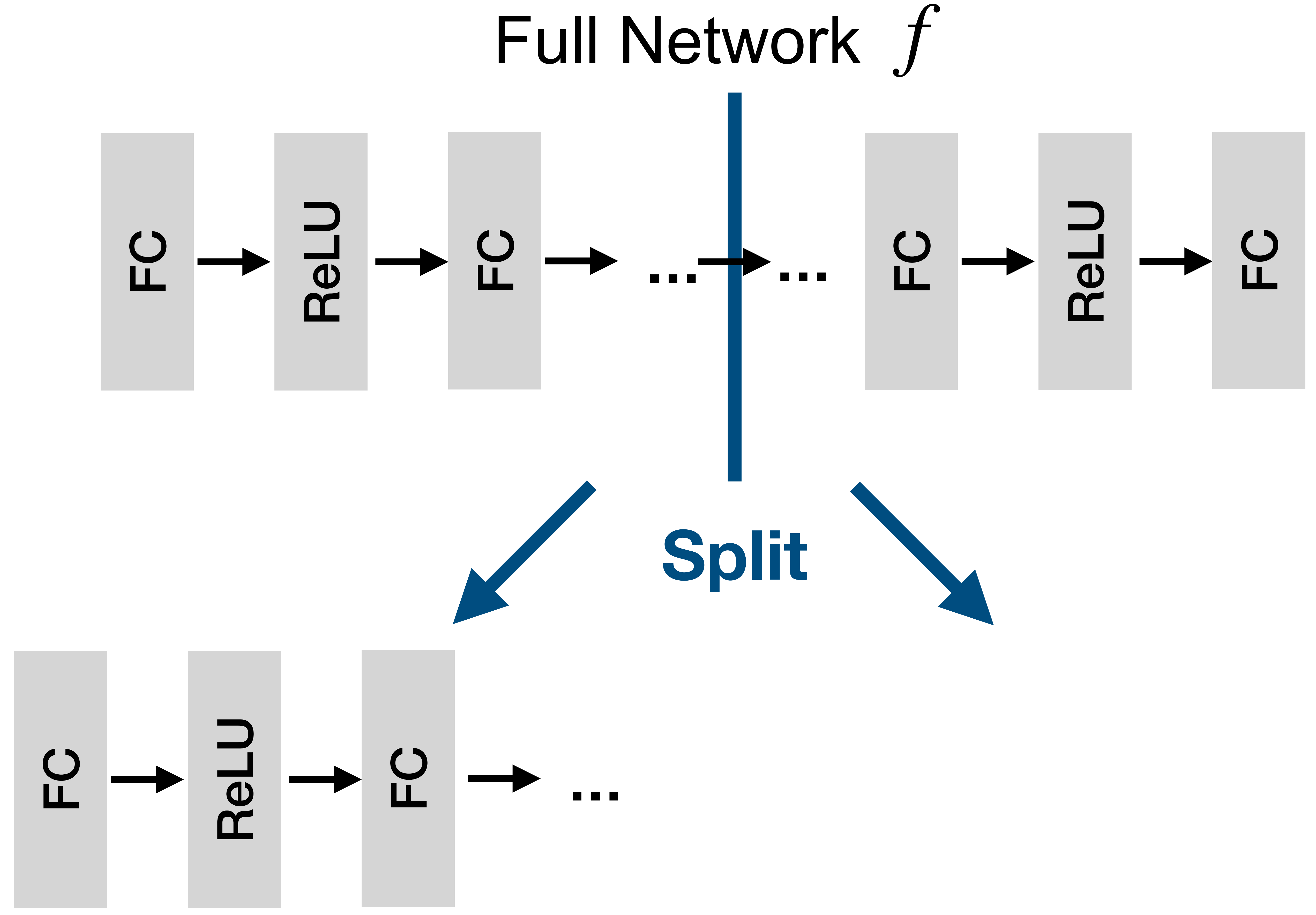


Training via Adversarially Propagating Subnetworks



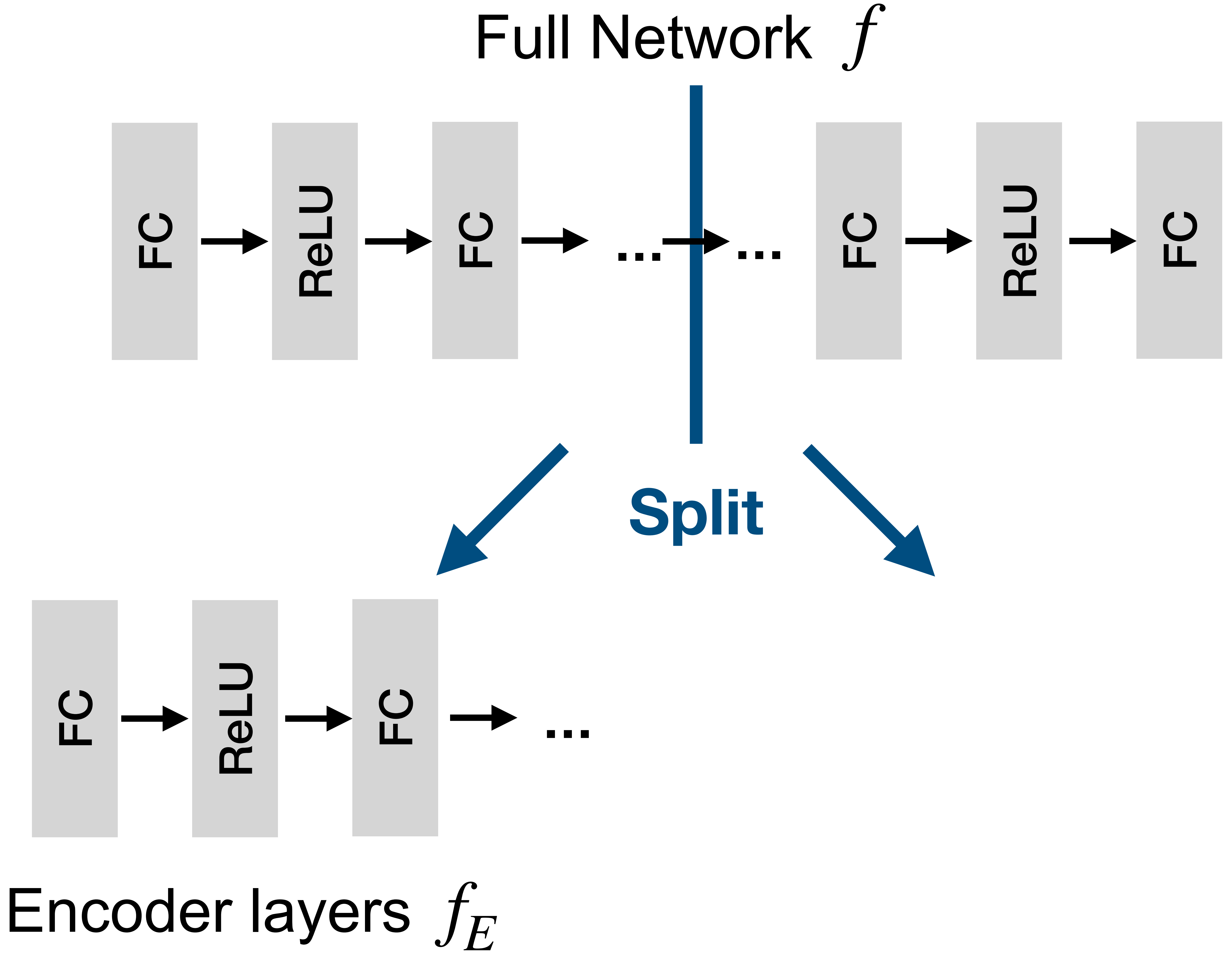
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



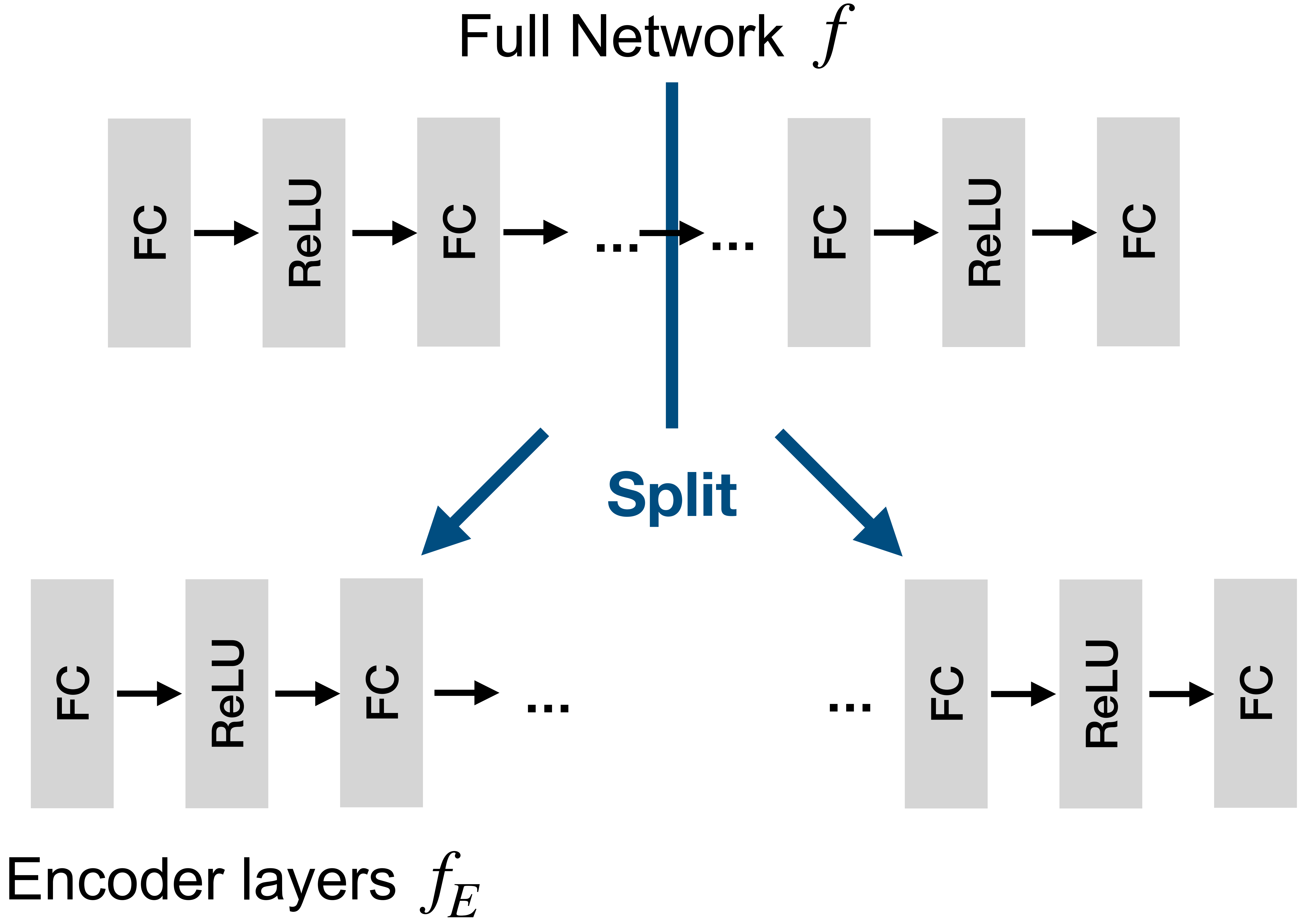
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

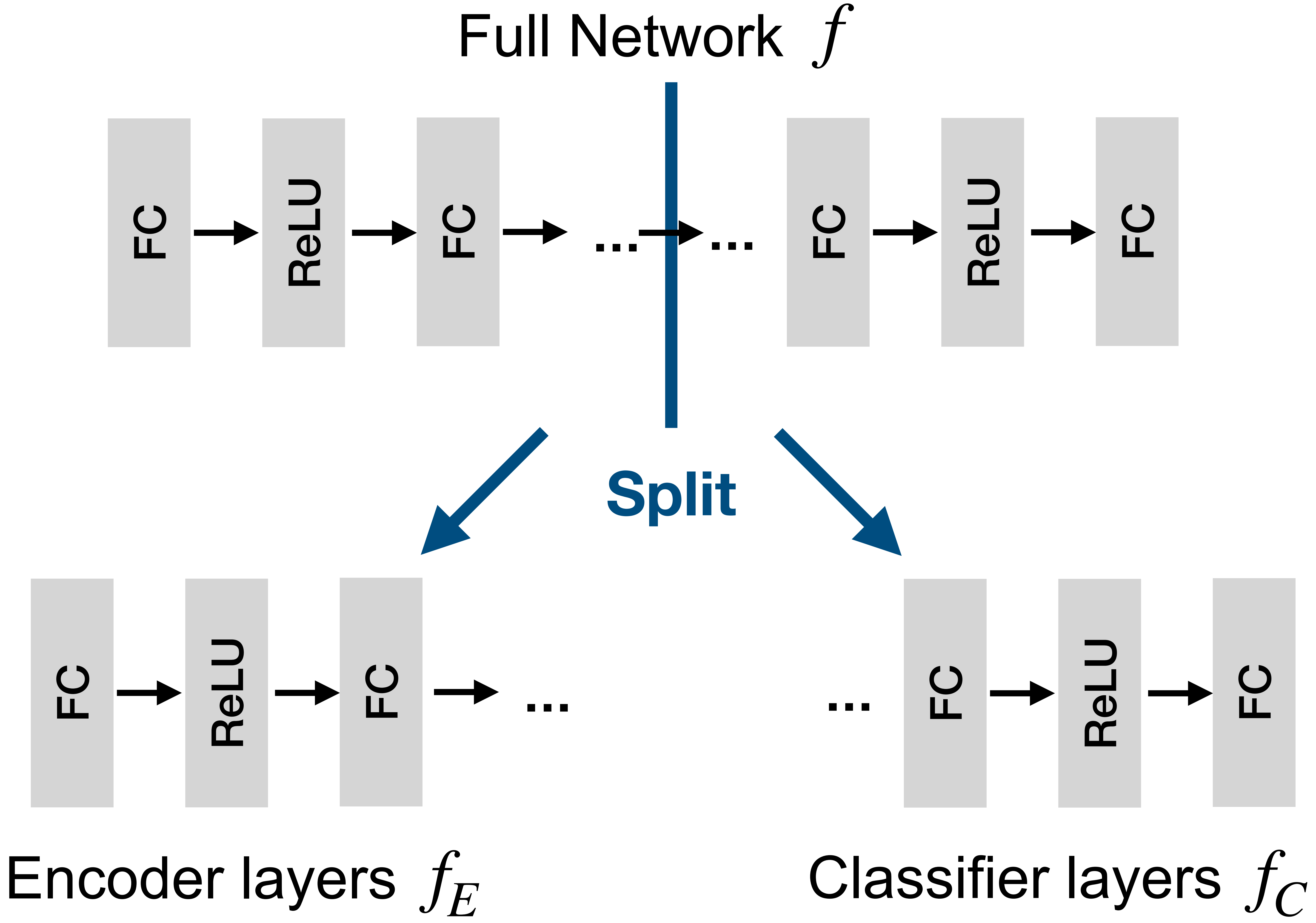


Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

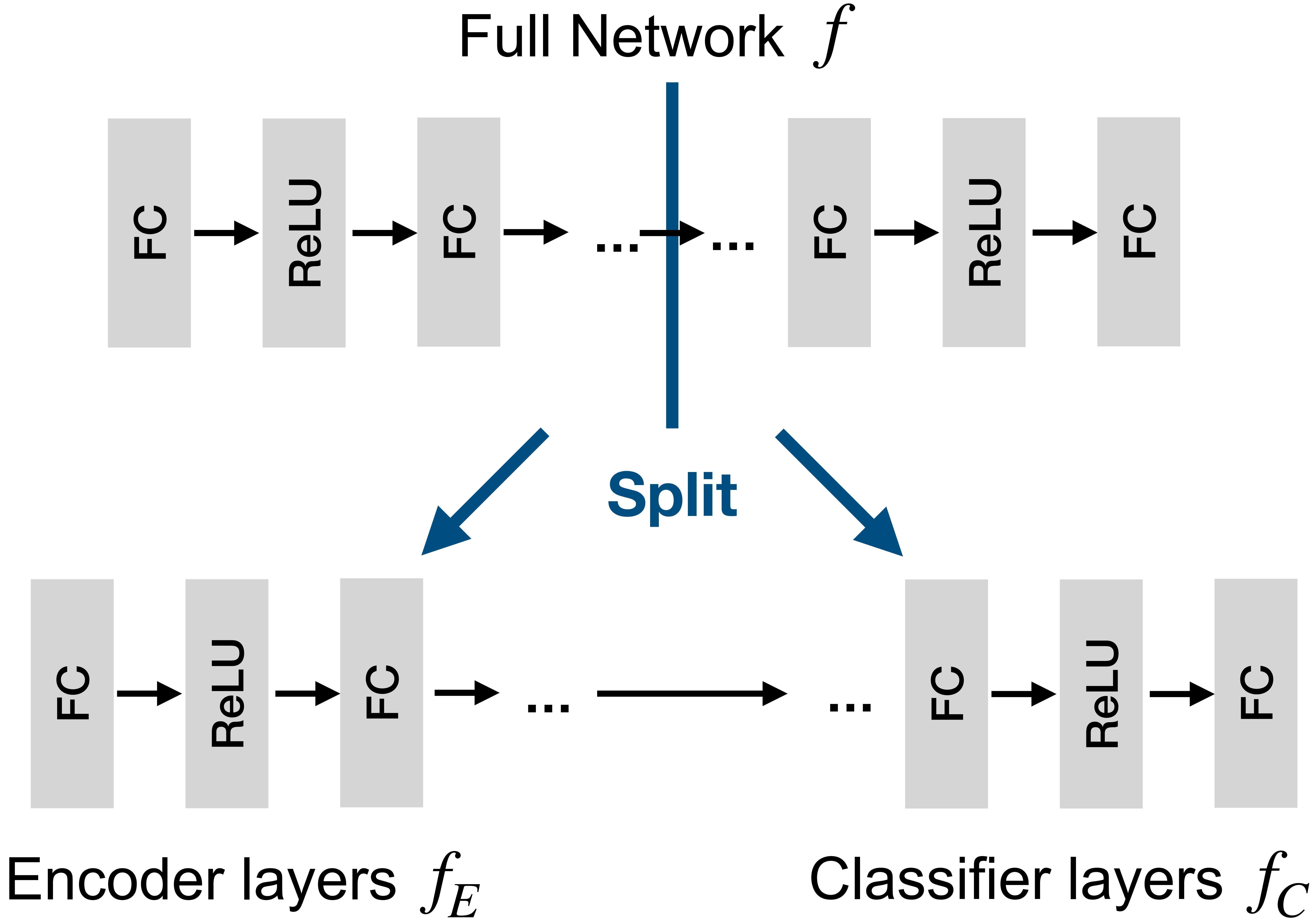


Training via Adversarially Propagating Subnetworks



Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

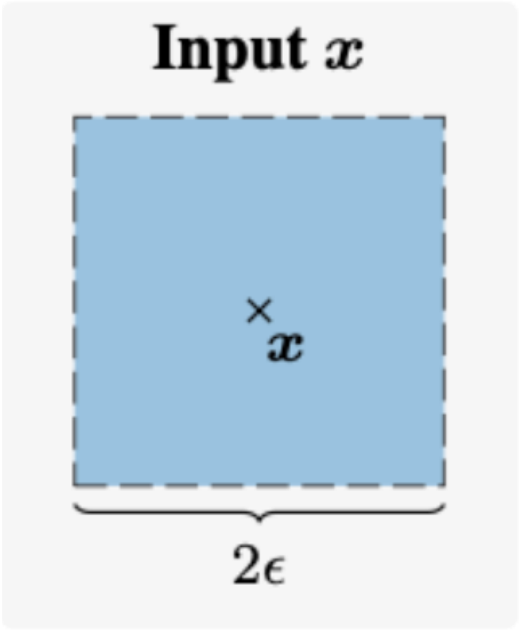


Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

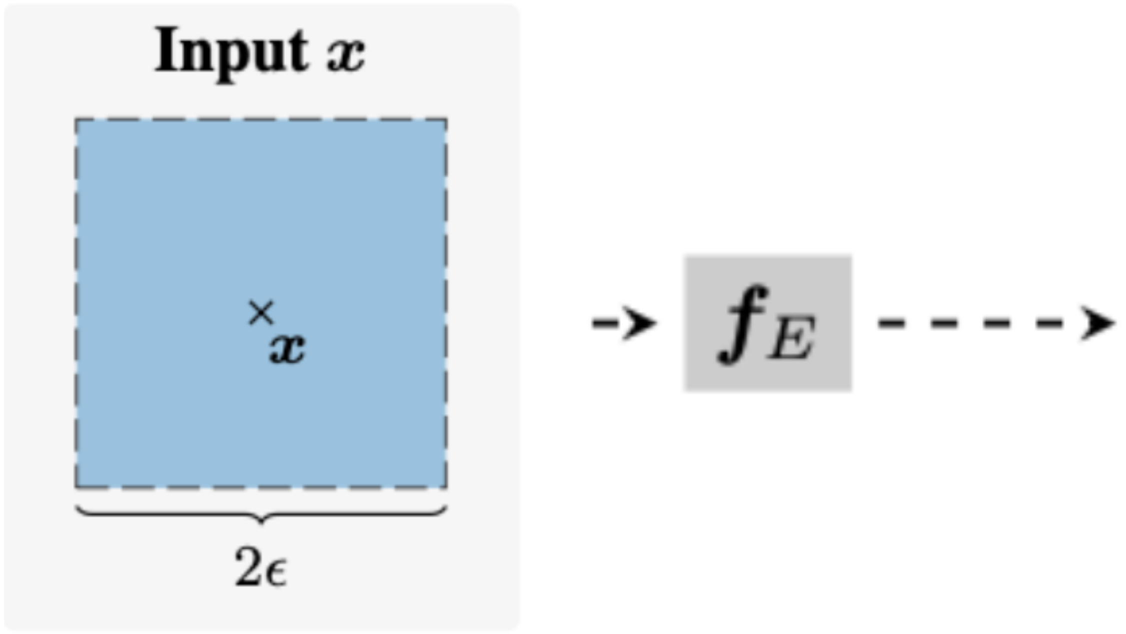
Training via Adversarially Propagating Subnetworks

Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

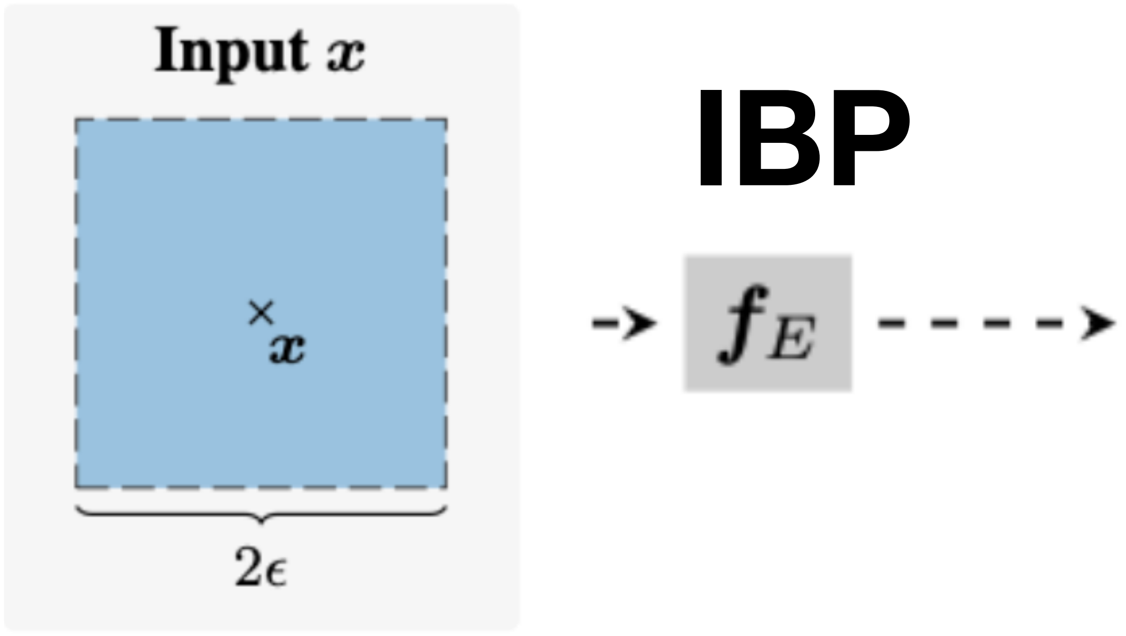


Training via Adversarially Propagating Subnetworks



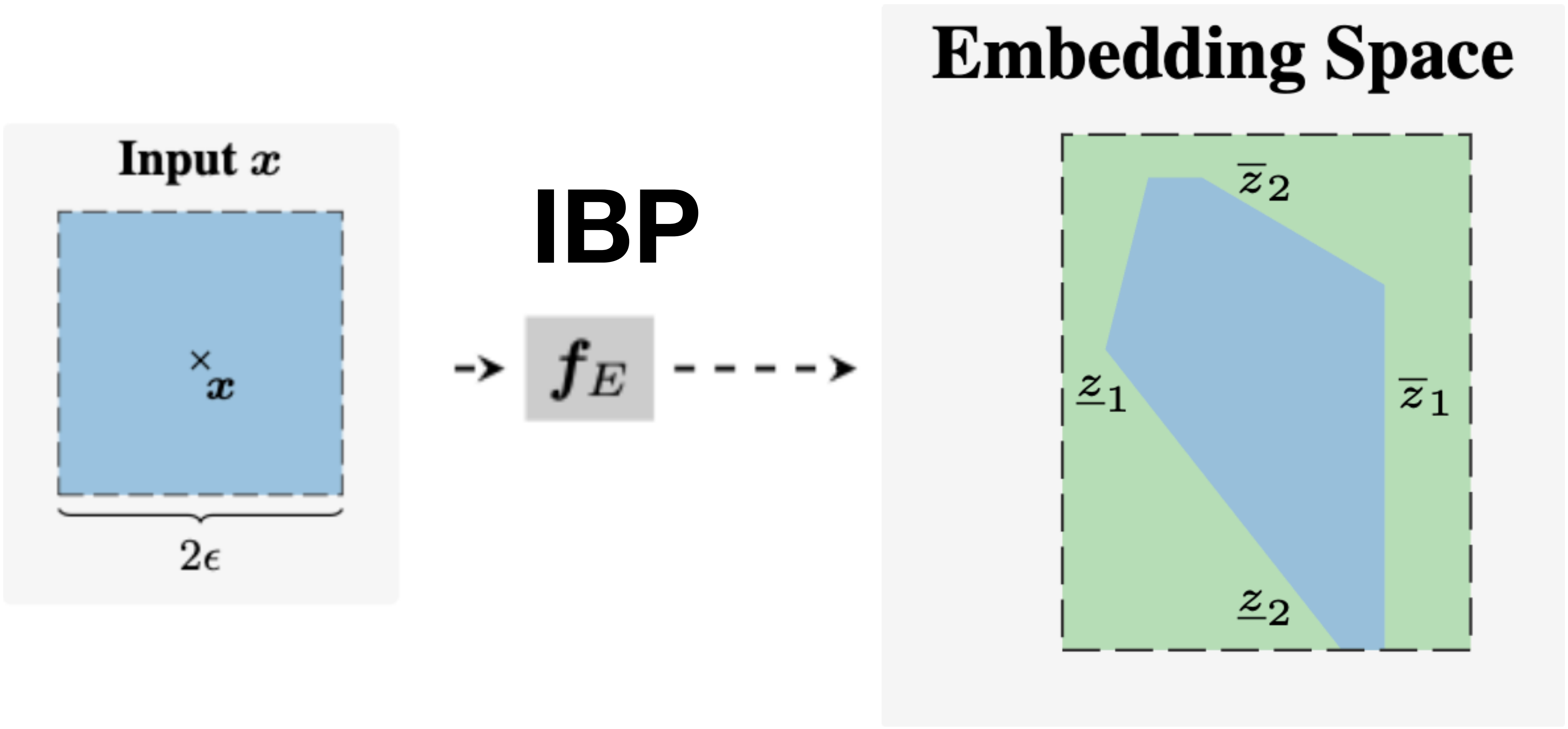
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

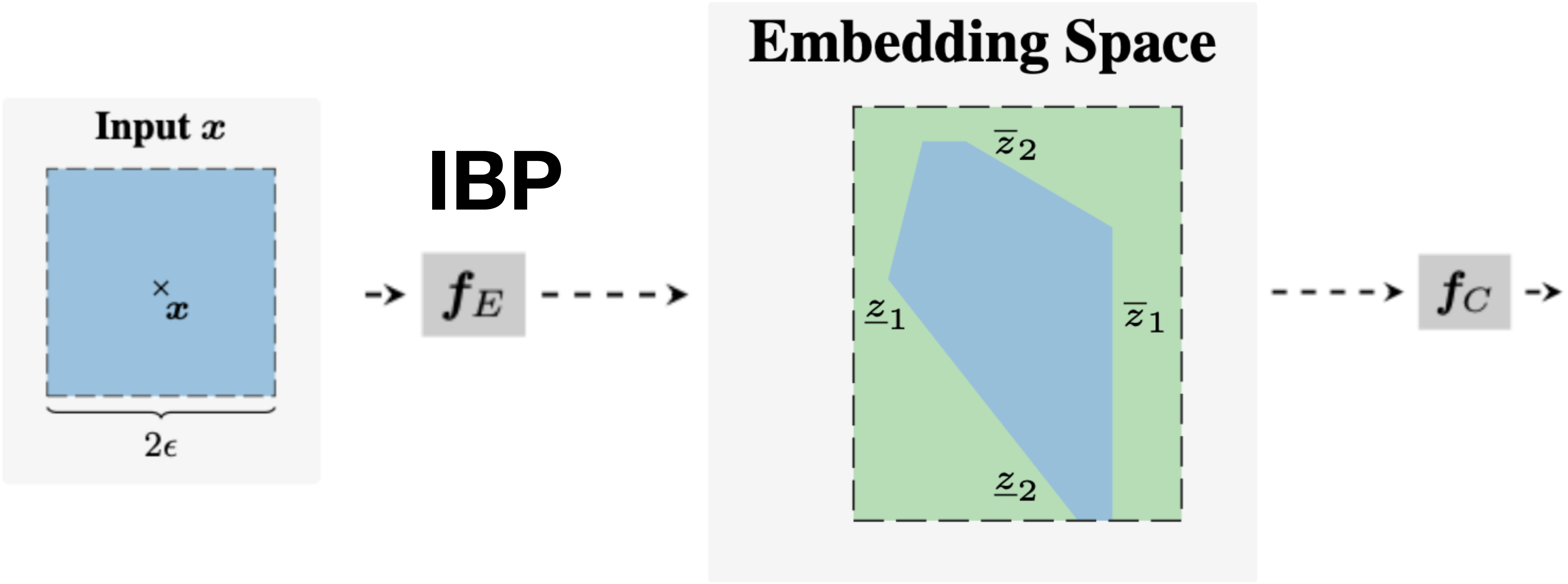


Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

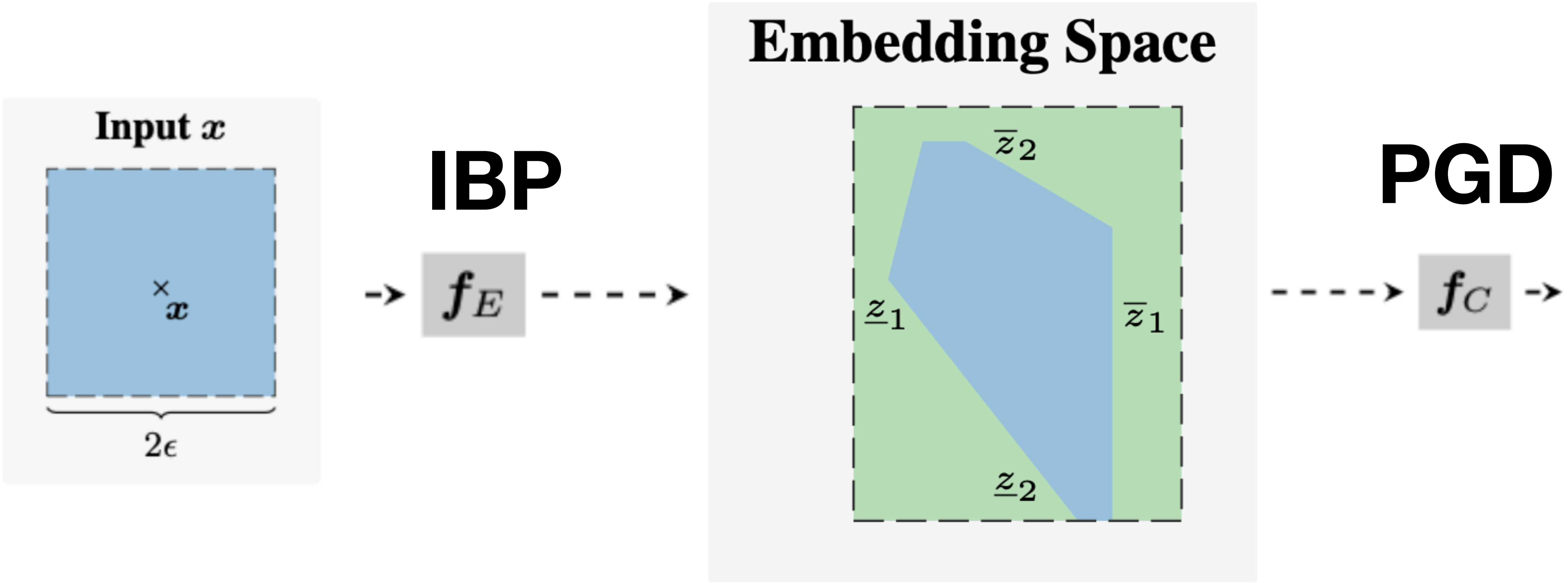
Training via Adversarially Propagating Subnetworks



Training via Adversarially Propagating Subnetworks

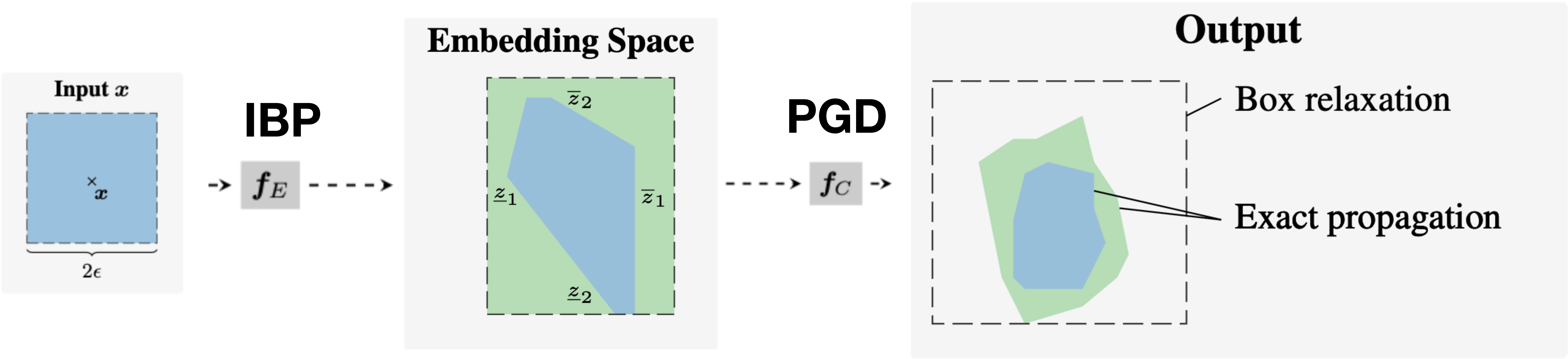


Training via Adversarially Propagating Subnetworks



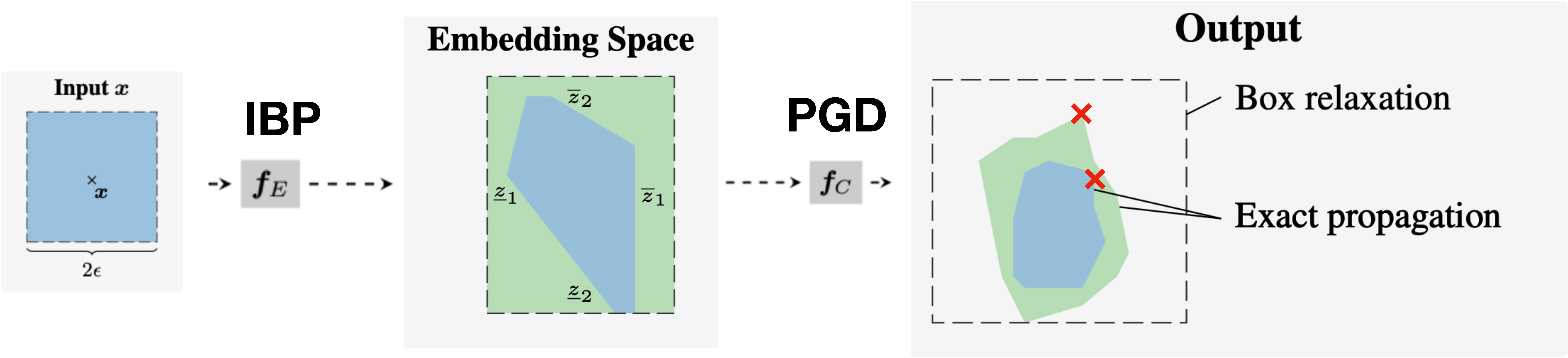
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

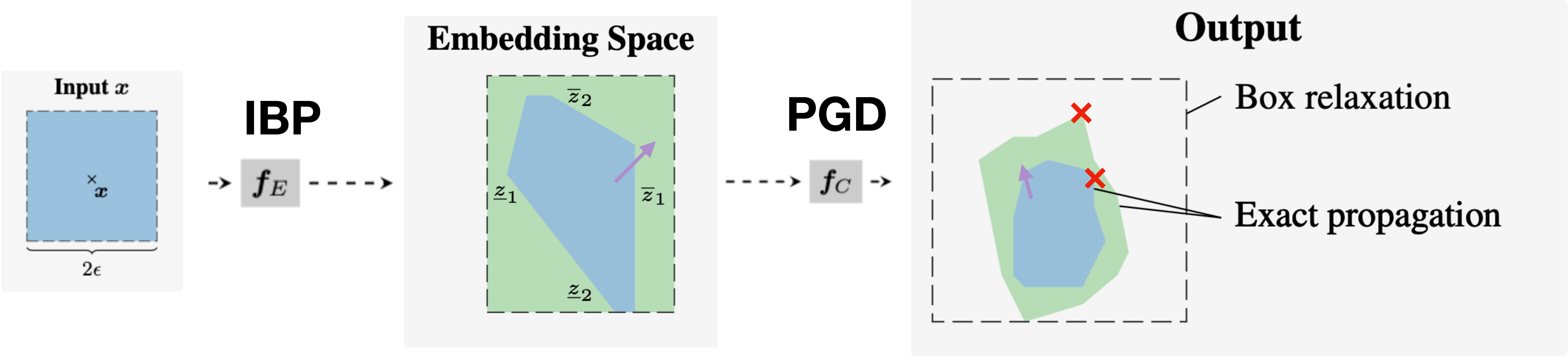


Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

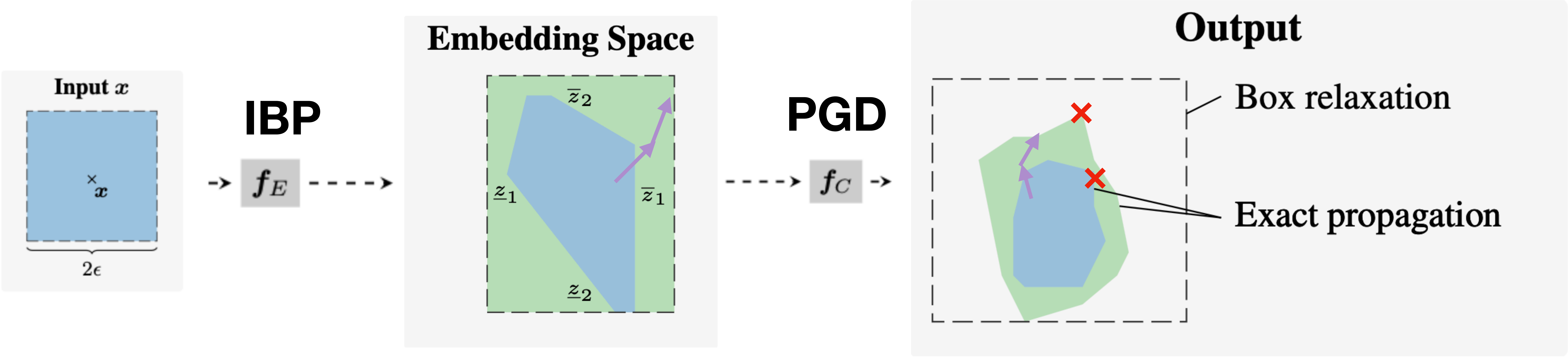


Training via Adversarially Propagating Subnetworks



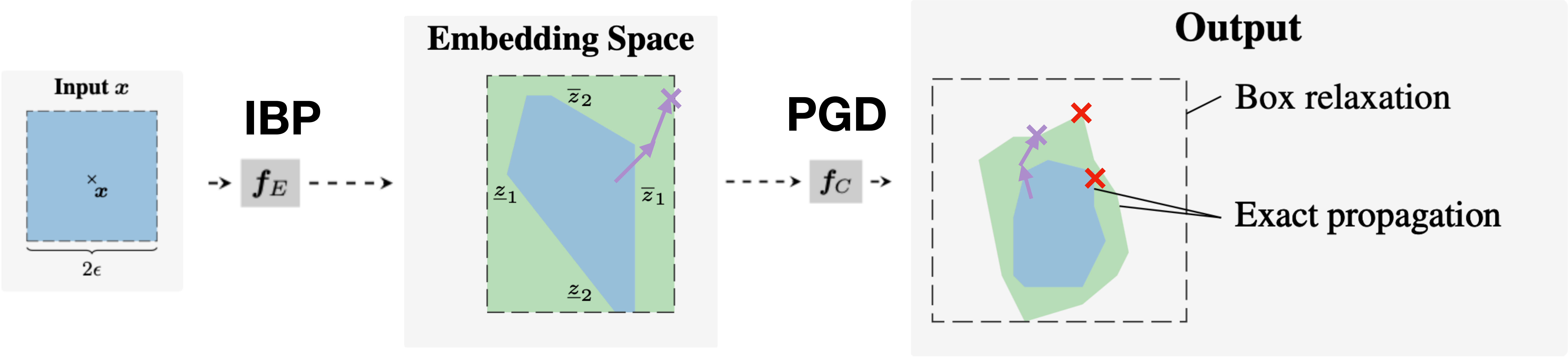
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



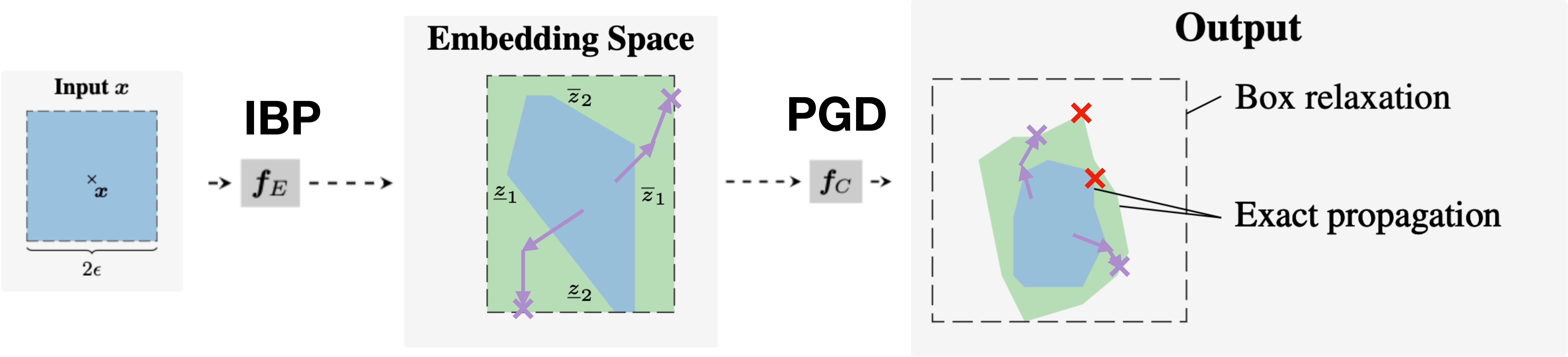
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



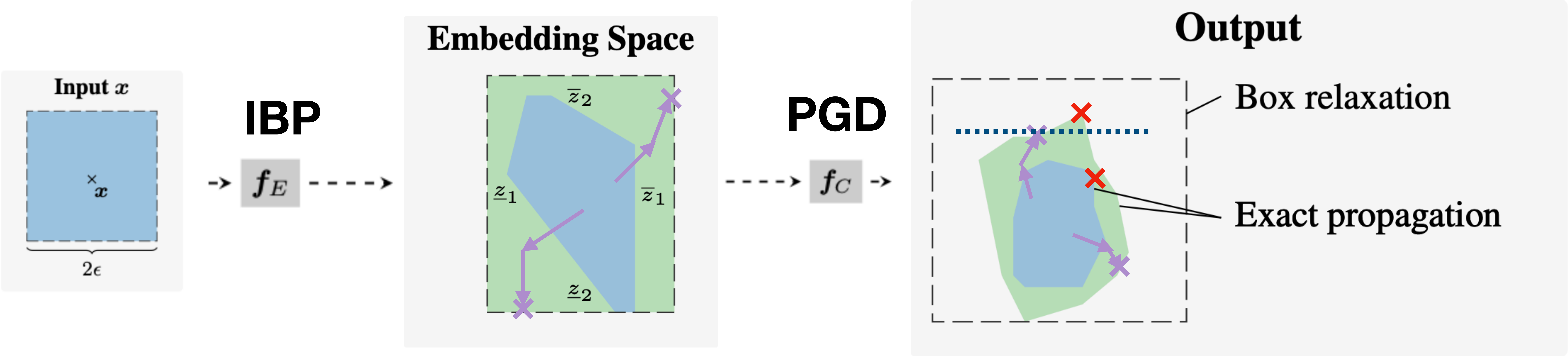
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

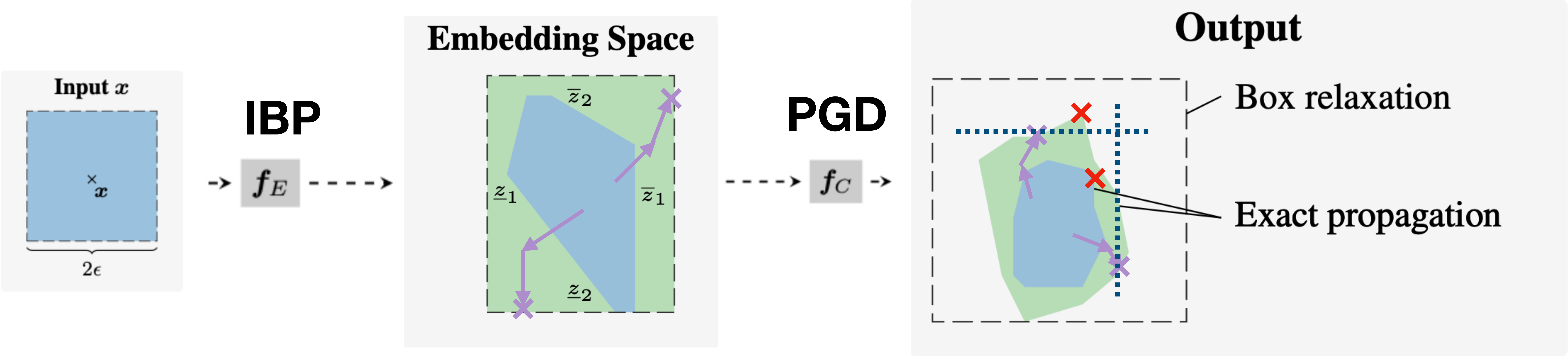


Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

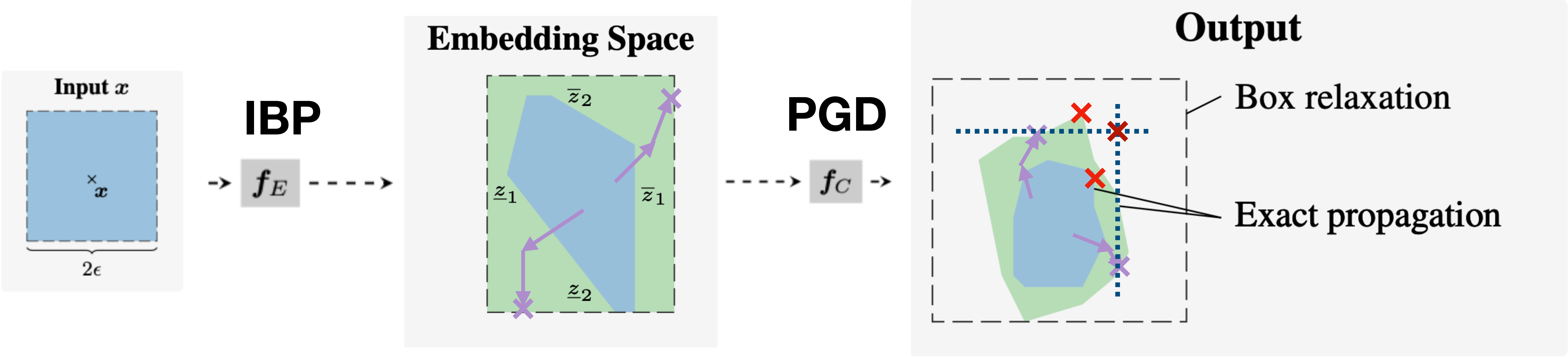
Training via Adversarially Propagating Subnetworks



Training via Adversarially Propagating Subnetworks

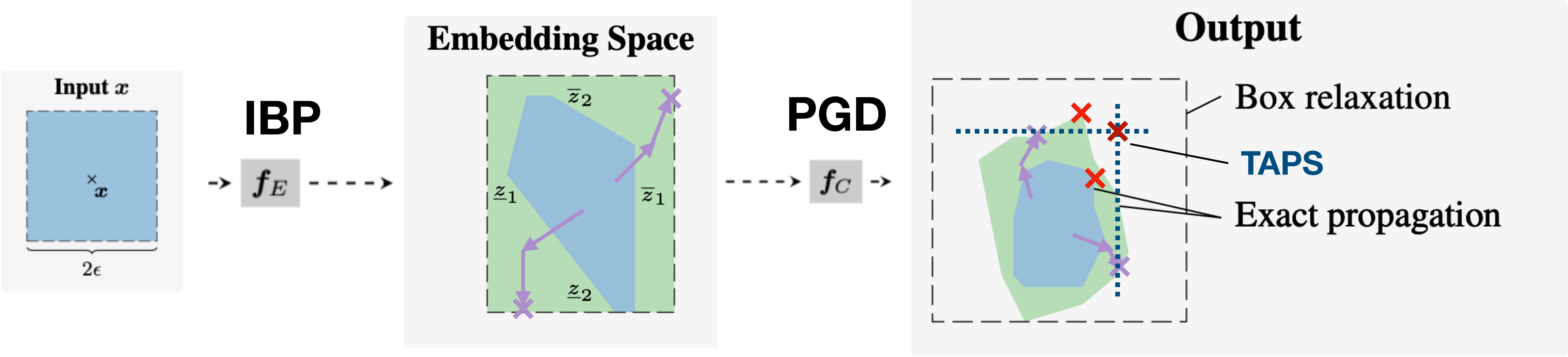


Training via Adversarially Propagating Subnetworks



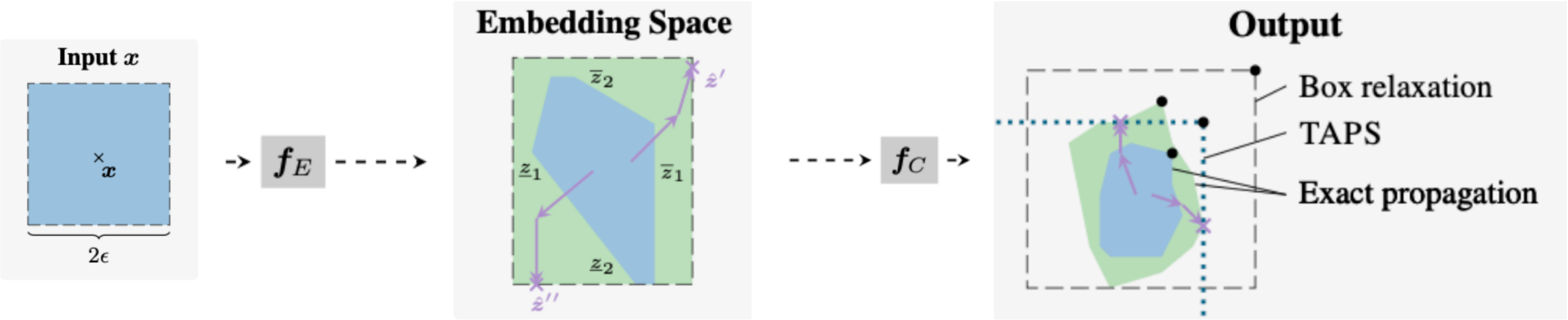
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



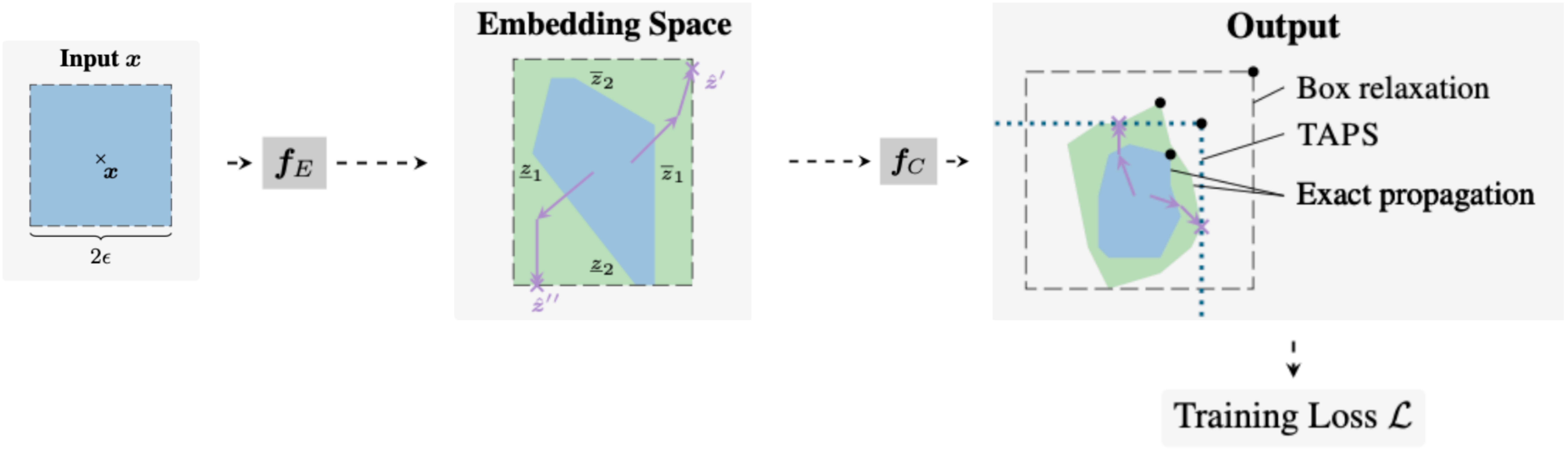
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



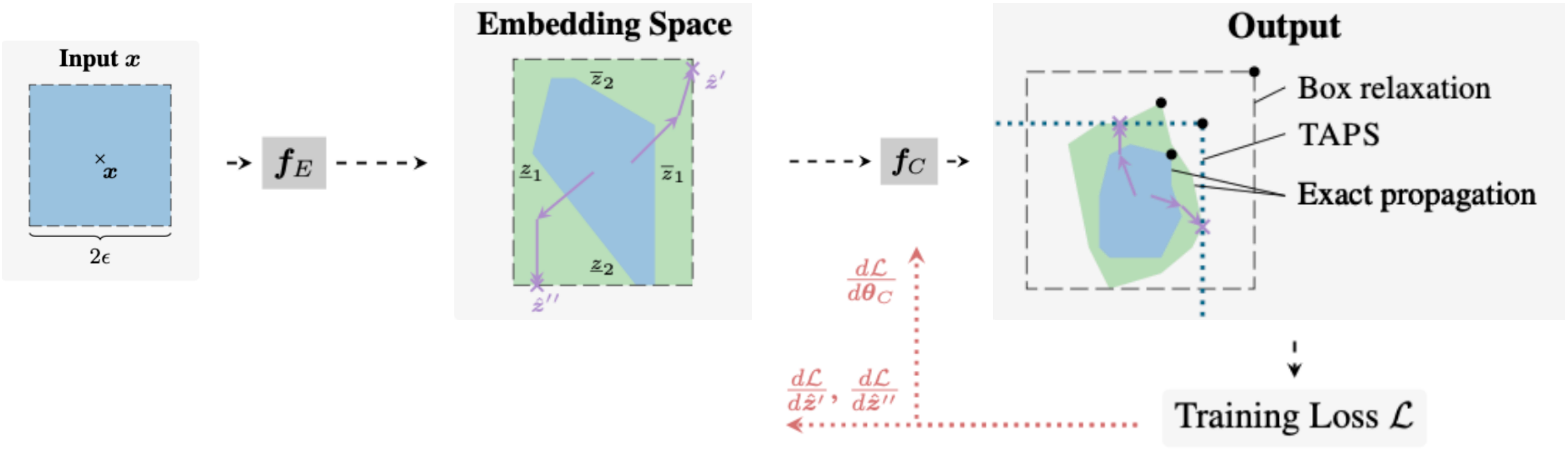
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



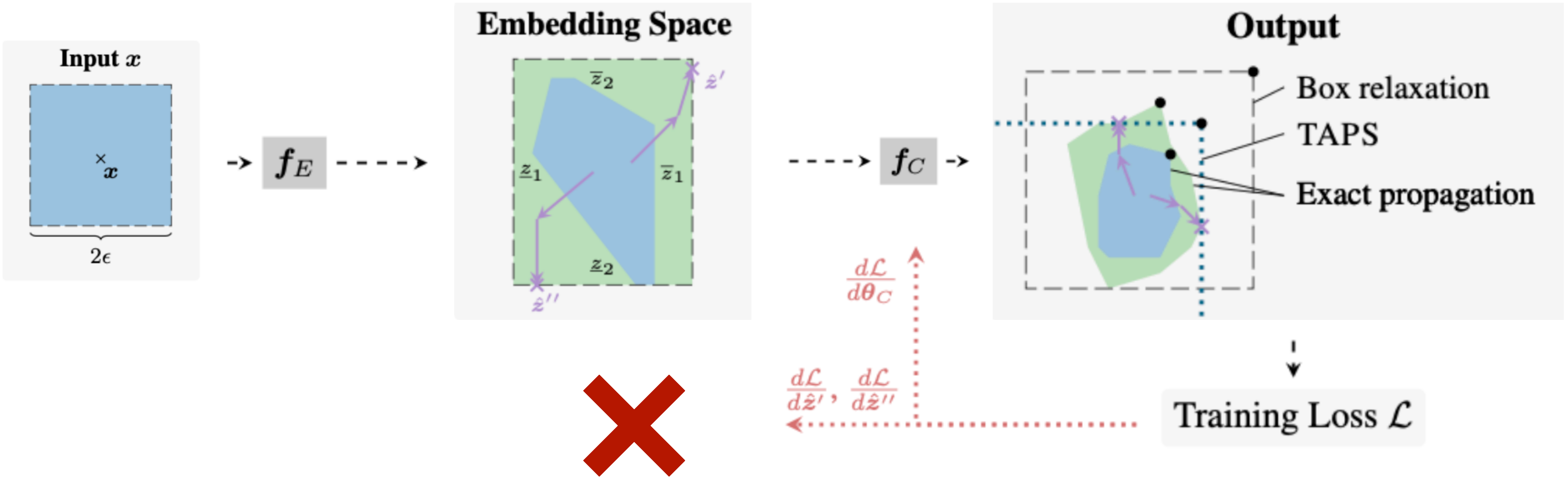
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

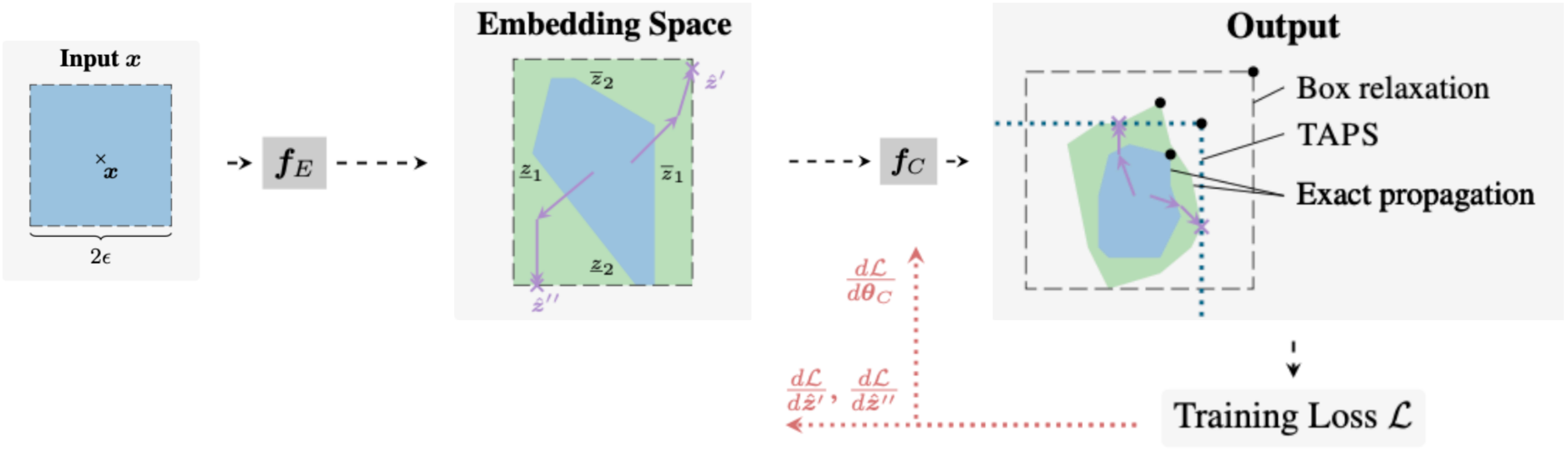


Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

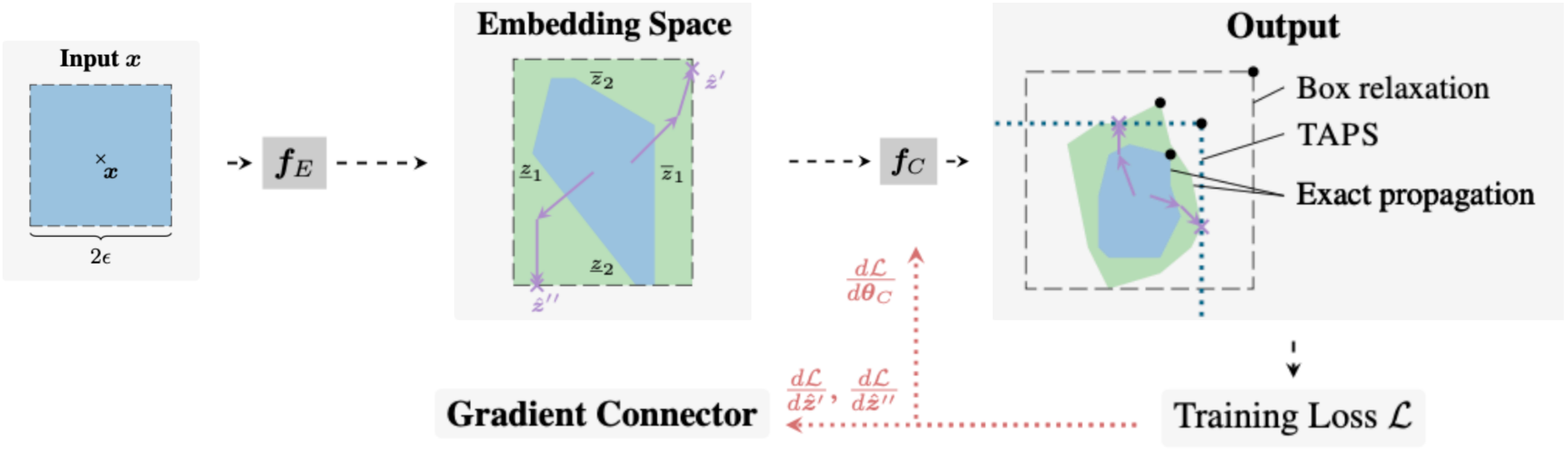


Training via Adversarially Propagating Subnetworks



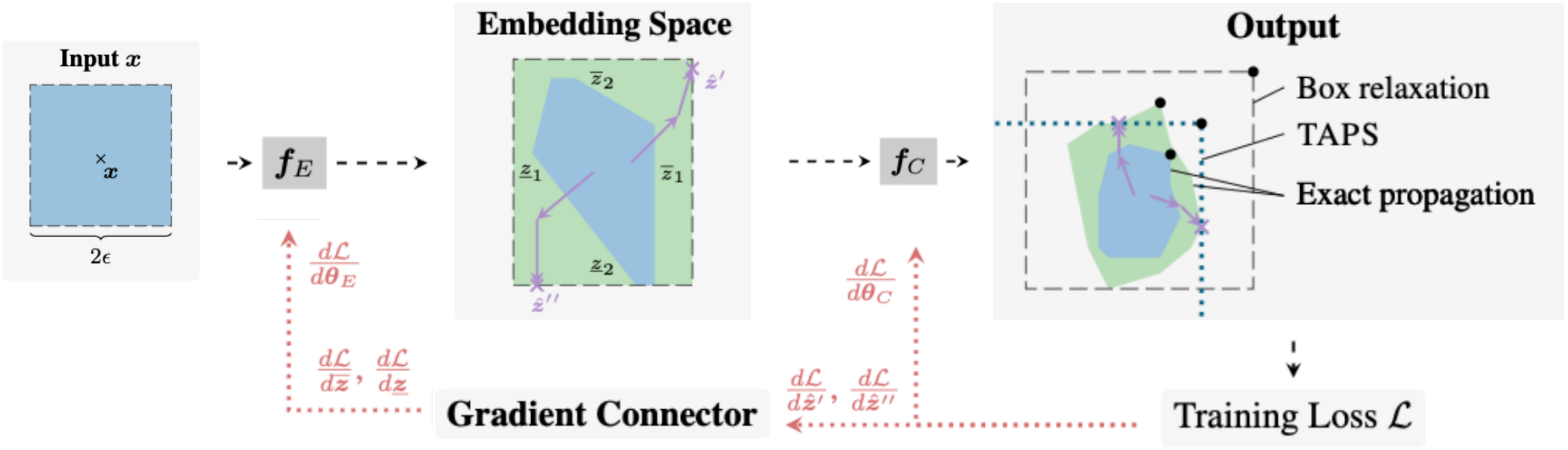
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



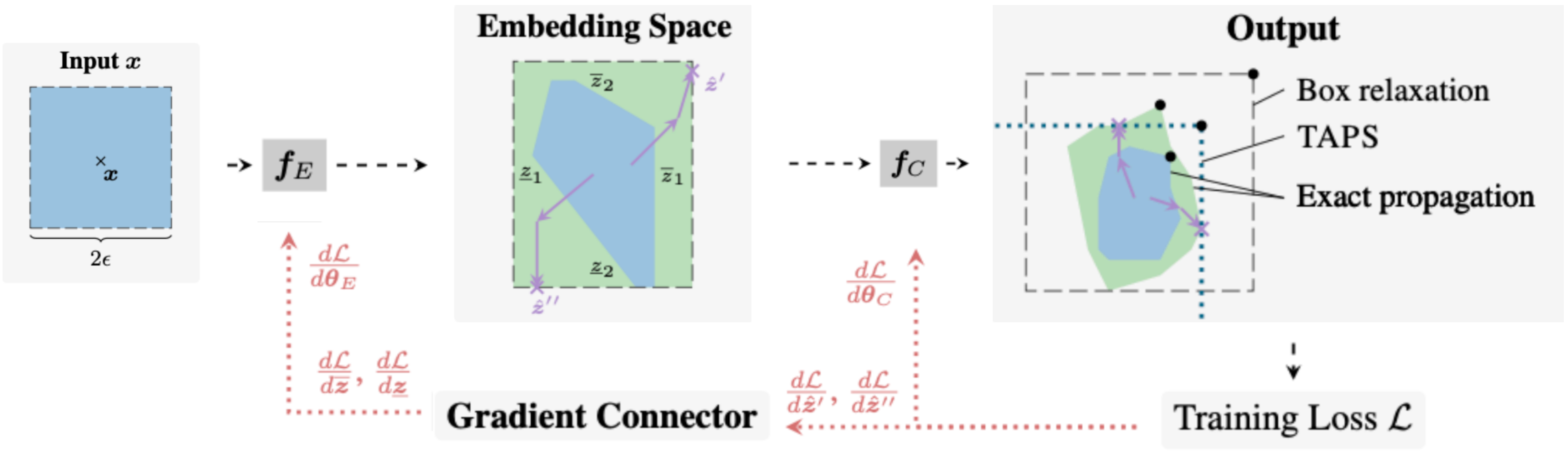
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

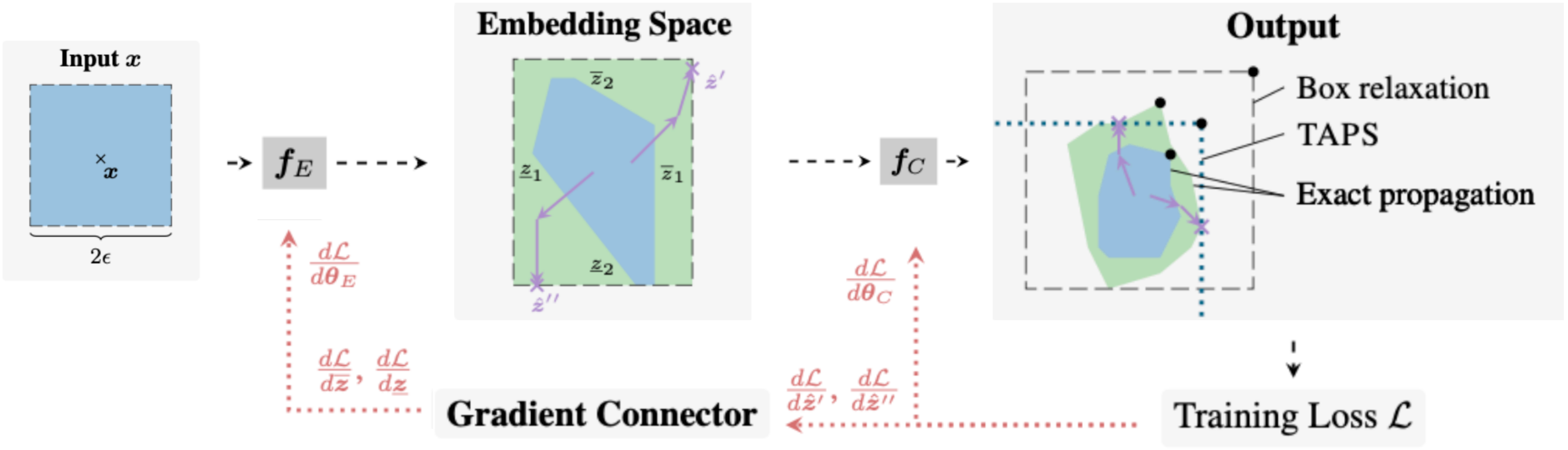
Training via Adversarially Propagating Subnetworks



Connecting Adversarial Examples with Bounds

Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks

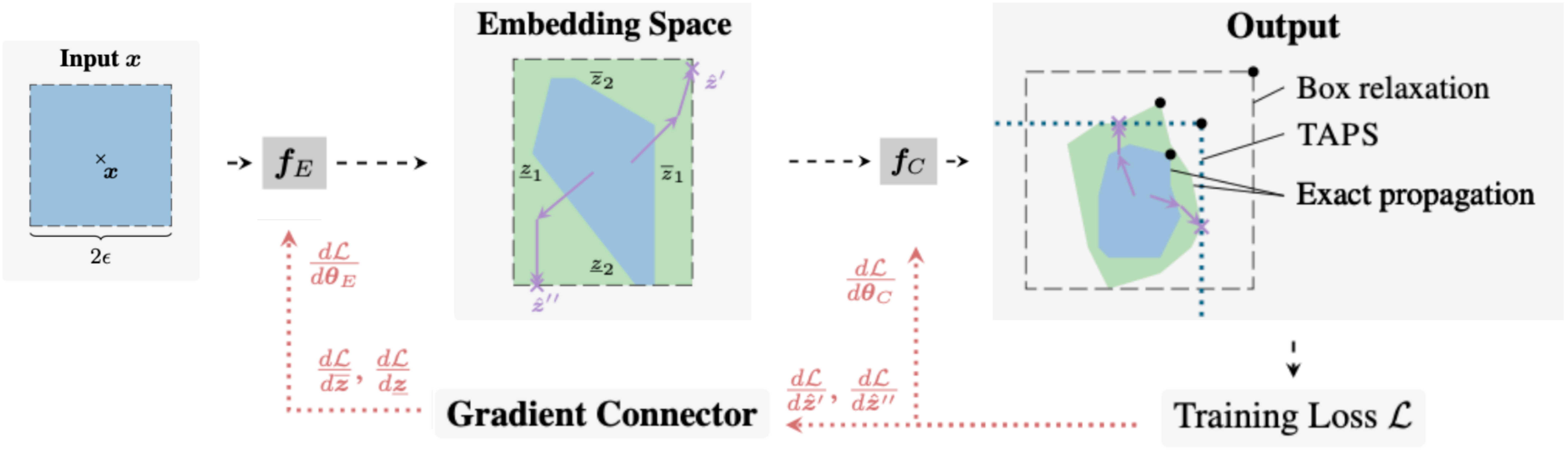


Connecting Adversarial Examples with Bounds

$$\frac{dL}{d\underline{z}_i} = \sum_j \frac{dL}{d\hat{z}_j} \frac{\partial \hat{z}_j}{\partial \underline{z}_i} = \frac{dL}{d\hat{z}_i} \frac{\partial \hat{\boldsymbol{z}}_i}{\partial \underline{\boldsymbol{z}}_i}$$

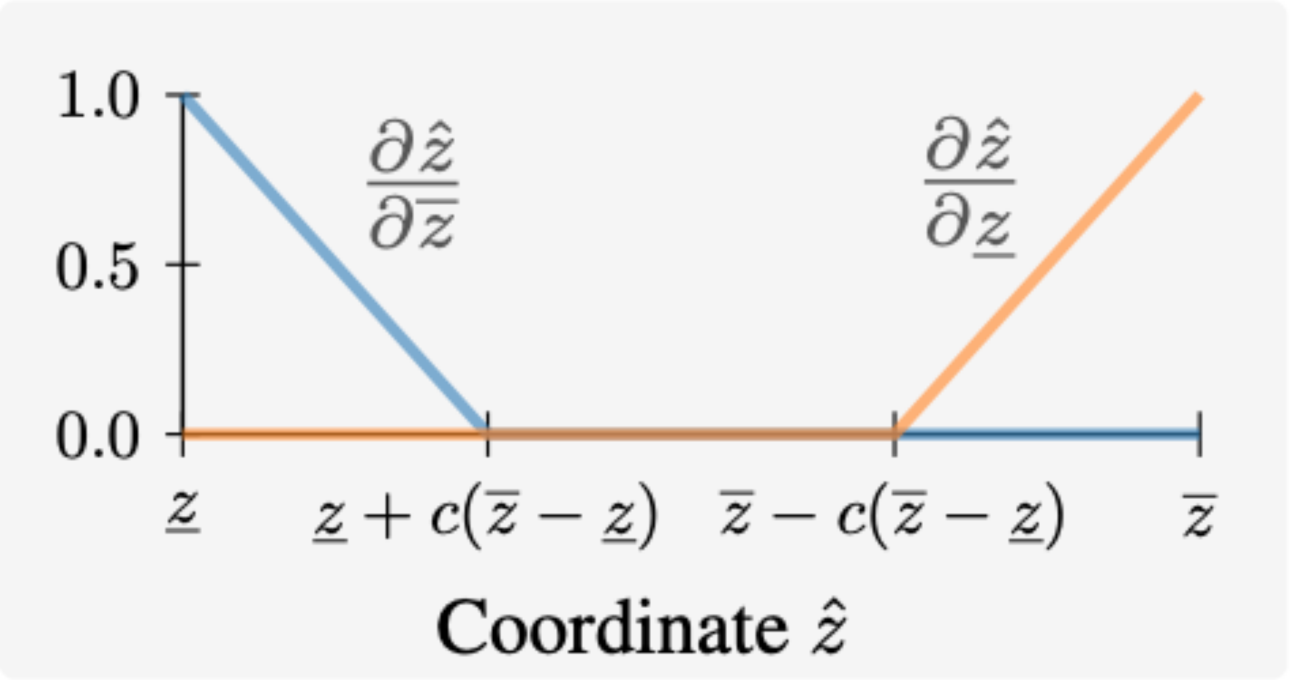
Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Training via Adversarially Propagating Subnetworks



Connecting Adversarial Examples with Bounds

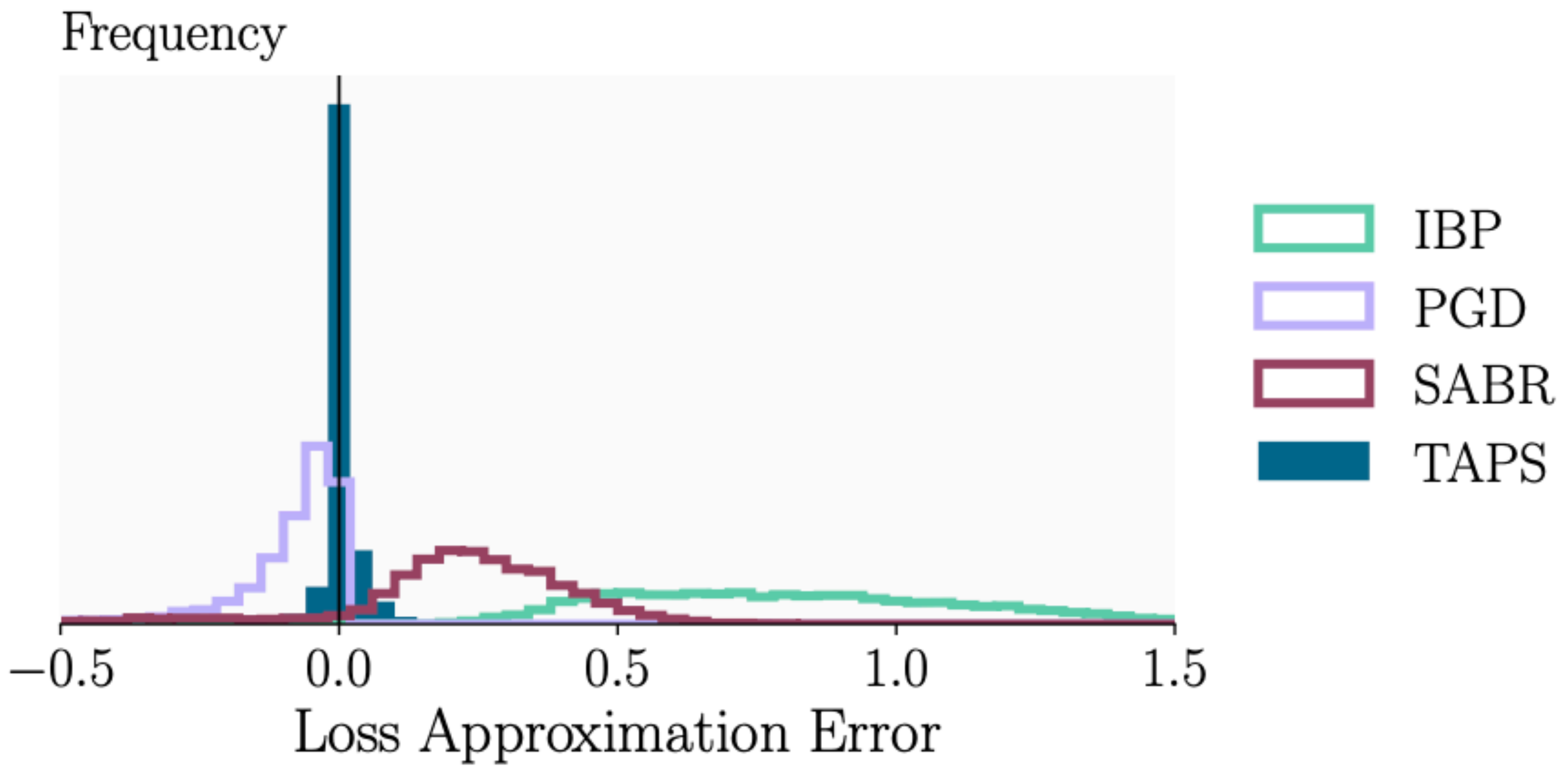
$$\frac{dL}{d\underline{z}_i} = \sum_j \frac{dL}{d\hat{z}_j}\frac{\partial \hat{z}_j}{\partial \underline{z}_i} = \frac{dL}{d\hat{z}_i}\frac{\partial \hat{\boldsymbol{z}}_i}{\partial \underline{\boldsymbol{z}}_i}$$



Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Precise Approximation

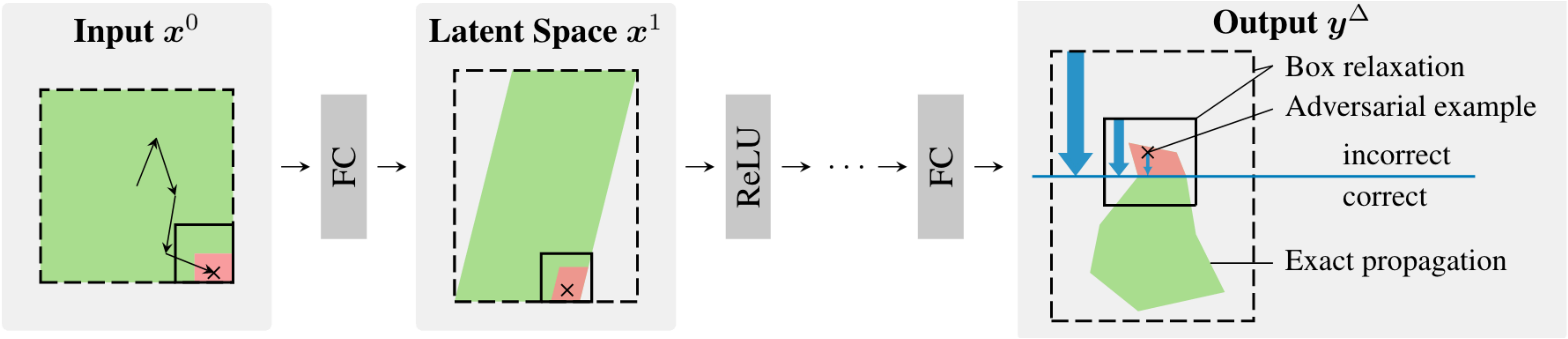
Precise Approximation



Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Complement Previous SOTA

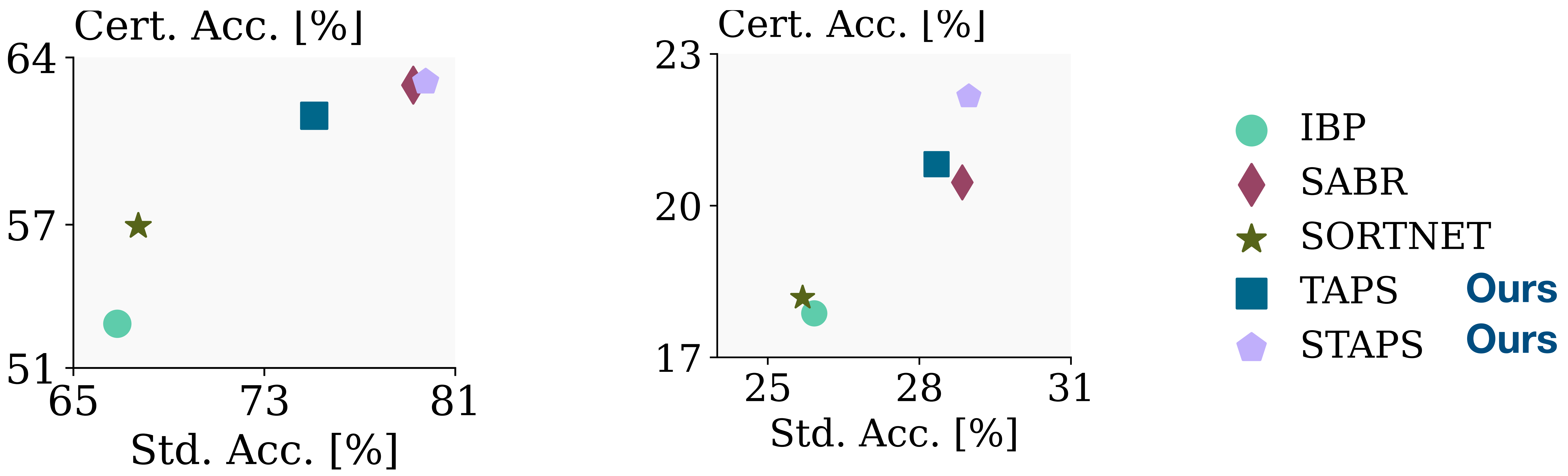
Small Adversarial Bound Regions
+Training via Adversarially Propagating Subnetworks
(SABR+TAPS=STAPS)



Plot taken from SABR paper.

Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

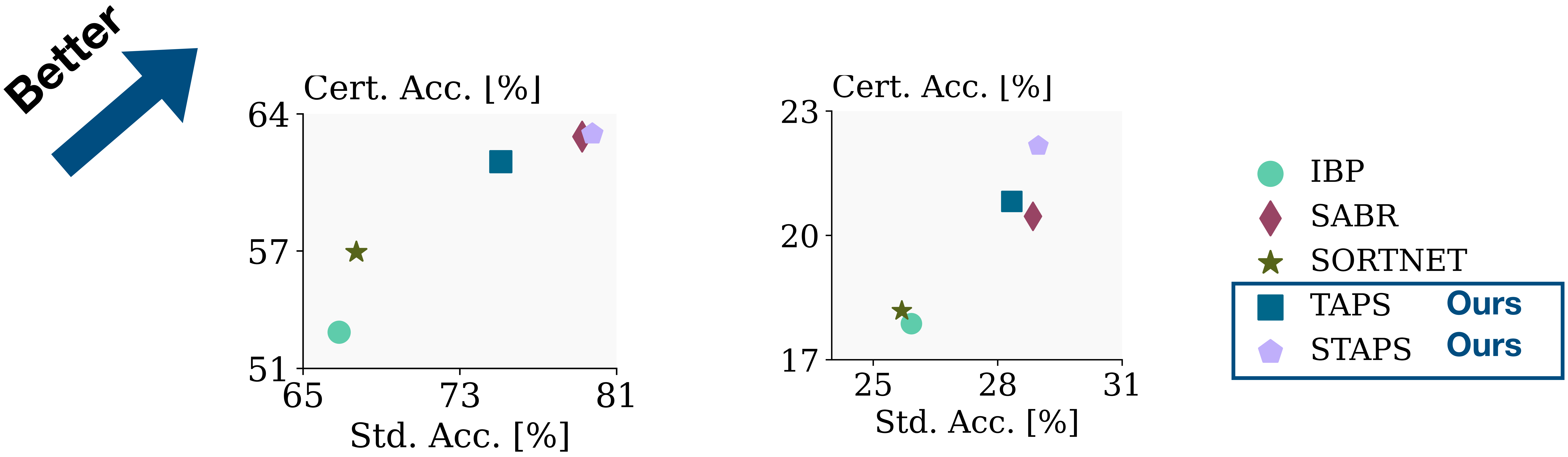
Empirical Results



CIFAR
$\epsilon = 2/255$

TinyImageNet
$\epsilon = 1/255$

Empirical Results



CIFAR
$\epsilon = 2/255$

TinyImageNet
$\epsilon = 1/255$

Mao et. al., Connecting Adversarial and Certified Training, NeurIPS'23

Take-away

Take-away

- We develop TAPS, a framework that sequentially connects certified and adversarial training to yield more precise approximation of the worst-case error.

Take-away

- We develop TAPS, a framework that sequentially connects certified and adversarial training to yield more precise approximation of the worst-case error.

- We present the idea of gradient connector, a novel tool for connecting their gradients and thus enable joint training.

Part 3

Understanding the Success of Interval Bound Propagation

Research Question

Baader et. al., Universal Approximation with Certified Networks, ICLR'20
Wang et. al., Interval Universal Approximation for Neural Networks, POPL'22

Research Question

- Interval Bound Propagation (IBP) and IBP-based methods get SOTA certified accuracy than more precise domains.

Baader et. al., Universal Approximation with Certified Networks, ICLR'20
Wang et. al., Interval Universal Approximation for Neural Networks, POPL'22

Research Question

- Interval Bound Propagation (IBP) and IBP-based methods get SOTA certified accuracy than more precise domains.

- There exists a neural network that approximates every continuous function and IBP bounds are nearly optimal, up to ϵ error. However, finding this network is strictly harder than NP-complete problems.

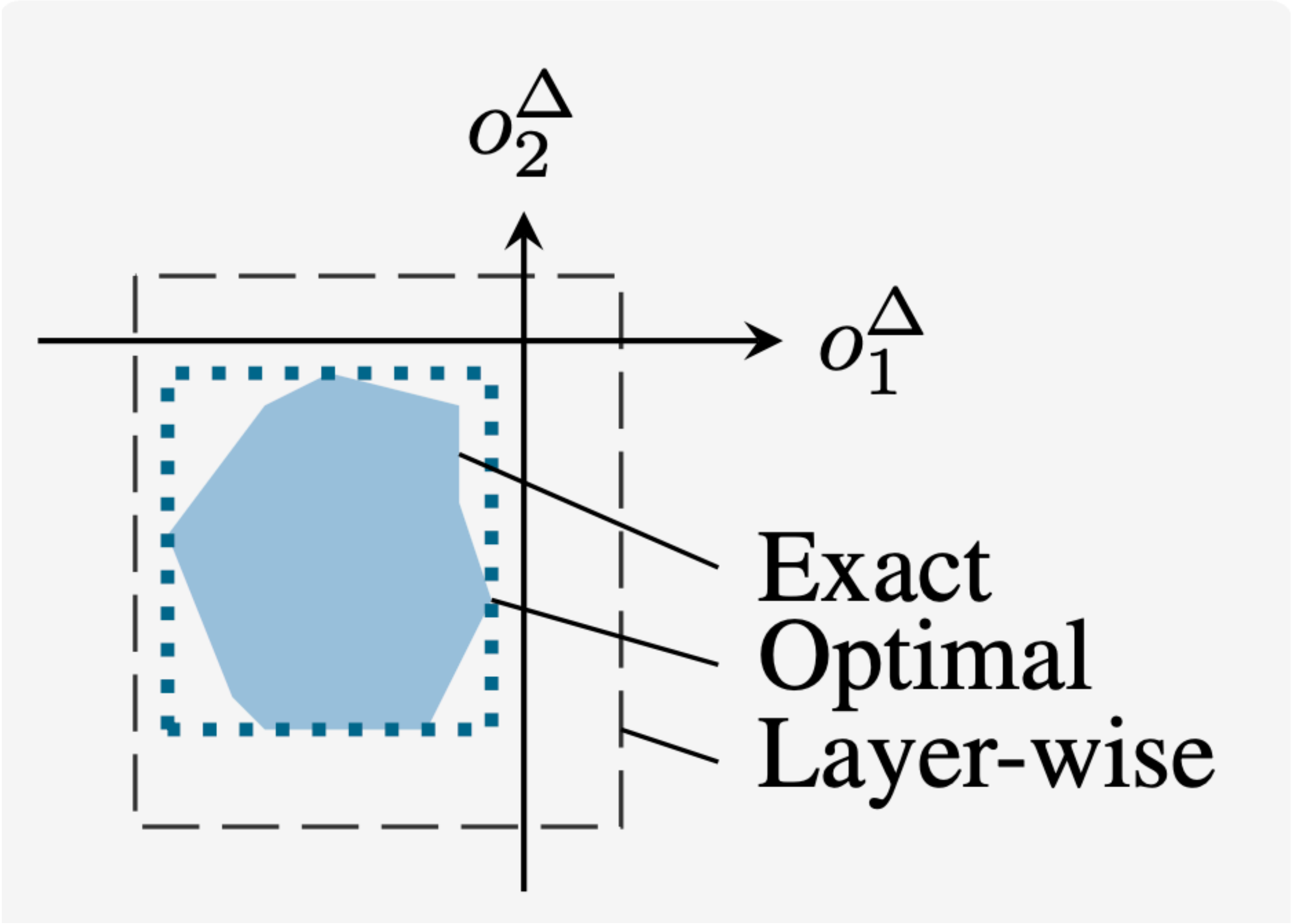
Baader et. al., Universal Approximation with Certified Networks, ICLR'20
Wang et. al., Interval Universal Approximation for Neural Networks, POPL'22

Research Question

- Interval Bound Propagation (IBP) and IBP-based methods get SOTA certified accuracy than more precise domains.
- There exists a neural network that approximates every continuous function and IBP bounds are nearly optimal, up to ϵ error. However, finding this network is strictly harder than NP-complete problems.
- Understanding how IBP works with the least tight relaxation is critical to future development.

Baader et. al., Universal Approximation with Certified Networks, ICLR'20
Wang et. al., Interval Universal Approximation for Neural Networks, POPL'22

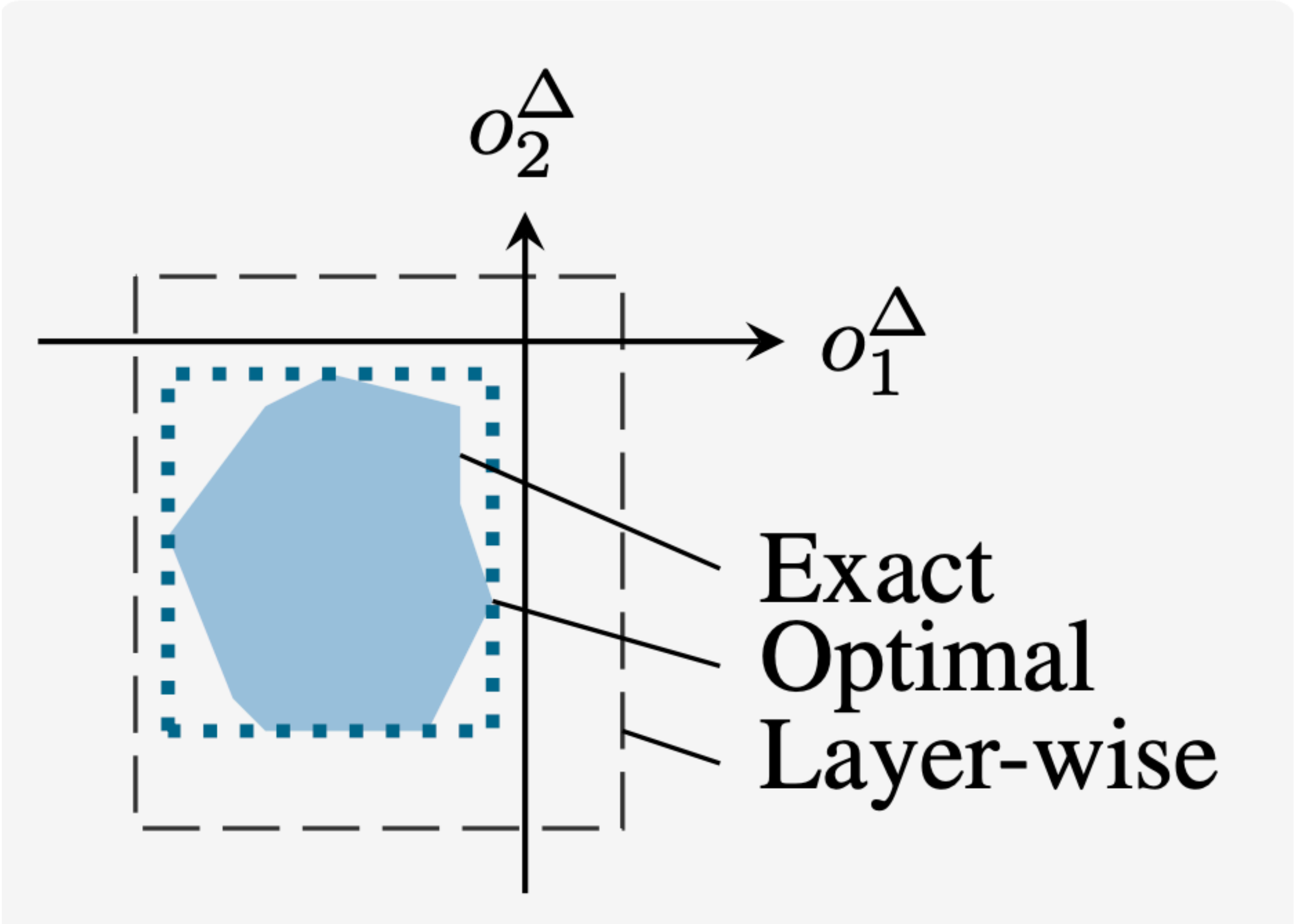
Notions



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Notions

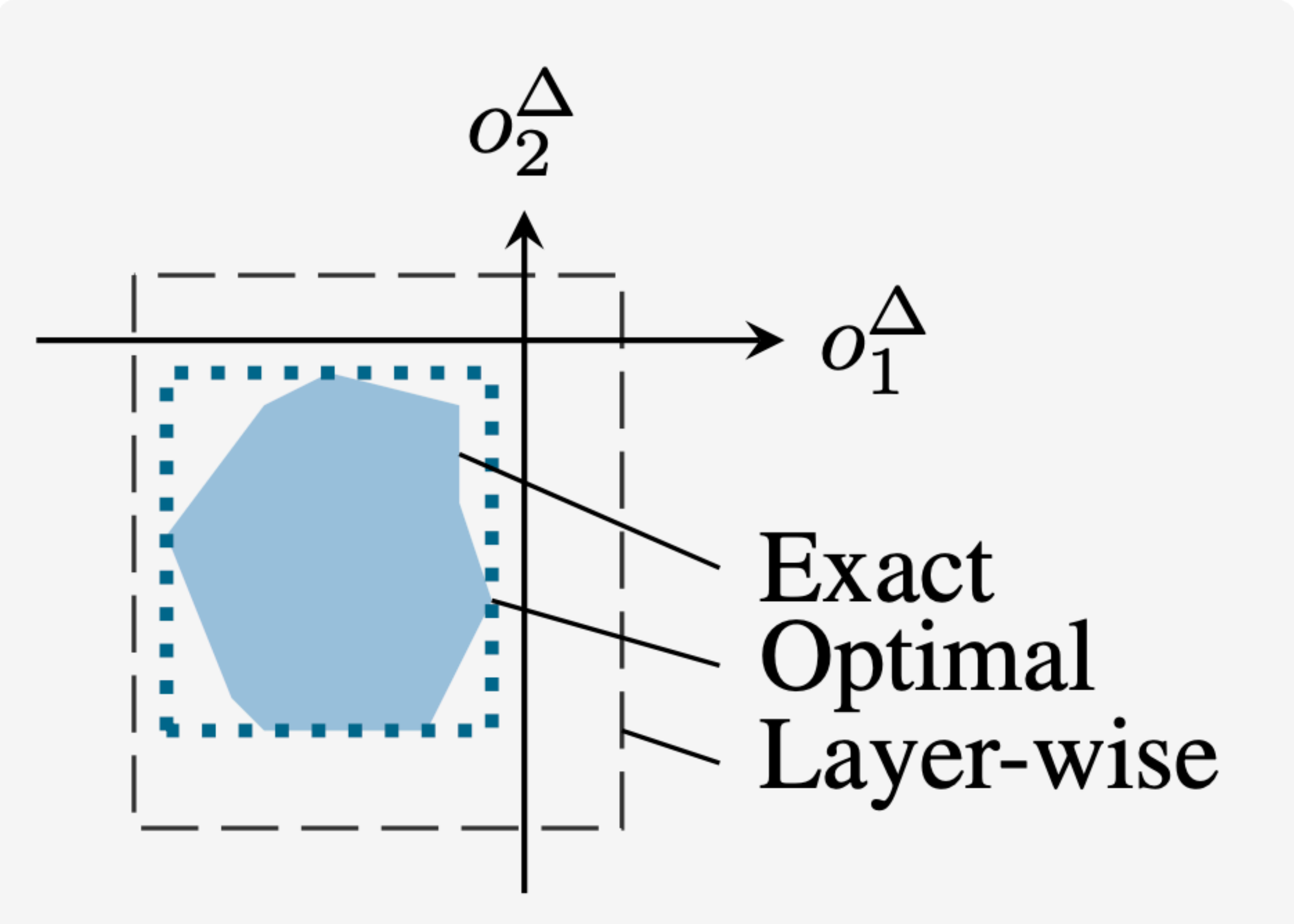
- **Layer-wise Approximation** $\text{Box}^\dagger(\boldsymbol{f}, B^\epsilon(\boldsymbol{x})) = [\underline{z}^\dagger, \overline{z}^\dagger]$: apply optimal approximation layer-wisely, i.e., IBP approximation.



Notions

- **Layer-wise Approximation** $\mathrm{Box}^{\dagger}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x})) = [\underline{z}^{\dagger}, \overline{z}^{\dagger}]$: apply optimal approximation layer-wisely, i.e., IBP approximation.

- **Optimal Approximation** $\mathrm{Box}^{*}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$: smallest hyper-box $[\underline{z}^{*}, \overline{z}^{*}]$ such that $\boldsymbol{f}(\boldsymbol{x}') \in [\underline{z}^{*}, \overline{z}^{*}], \forall \boldsymbol{x}' \in B^{\epsilon}(\boldsymbol{x})$.



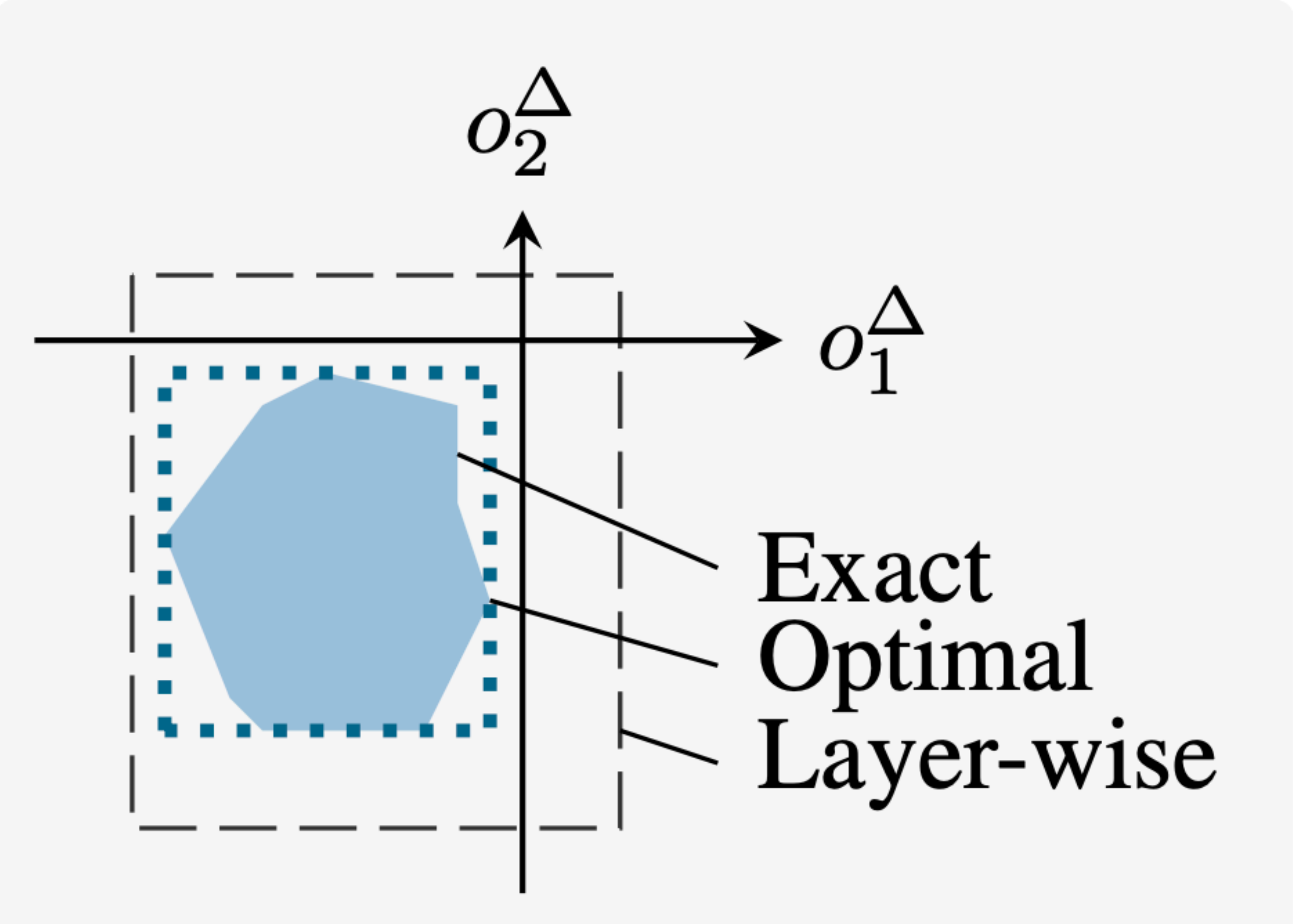
Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Notions

- **Layer-wise Approximation** $\mathrm{Box}^\dagger(\boldsymbol{f}, B^\epsilon(\boldsymbol{x})) = [\underline{z}^\dagger, \overline{z}^\dagger]$: apply optimal approximation layer-wisely, i.e., IBP approximation.

- **Optimal Approximation** $\mathrm{Box}^*(\boldsymbol{f}, B^\epsilon(\boldsymbol{x}))$: smallest hyper-box $[\underline{z}^*, \overline{z}^*]$ such that $\boldsymbol{f}(\boldsymbol{x}') \in [\underline{z}^*, \overline{z}^*], \forall \boldsymbol{x}' \in B^\epsilon(\boldsymbol{x})$.

- **Propagation Invariance**: a network is propagation invariant if $\mathrm{Box}^\dagger(\boldsymbol{f}, B^\epsilon(\boldsymbol{x})) = \mathrm{Box}^*(\boldsymbol{f}, B^\epsilon(\boldsymbol{x}))$, i.e., IBP is exact.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR’24

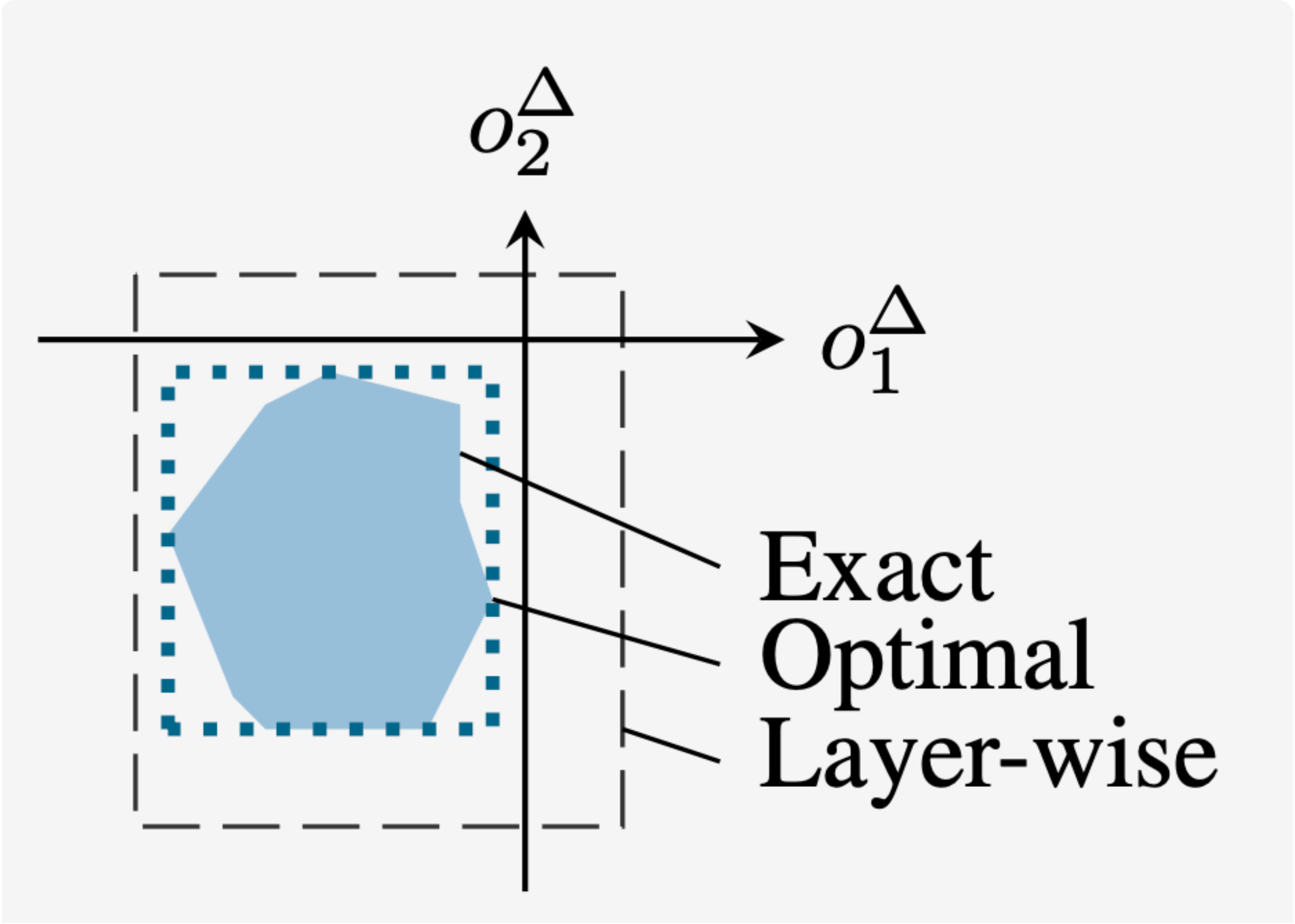
Notions

- **Layer-wise Approximation** $\text{Box}^{\dagger}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x})) = [\underline{z}^{\dagger}, \overline{z}^{\dagger}]$: apply optimal approximation layer-wisely, i.e., IBP approximation.

- **Optimal Approximation** $\text{Box}^{*}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$: smallest hyper-box $[\underline{z}^{*}, \overline{z}^{*}]$ such that $\boldsymbol{f}(\boldsymbol{x}') \in [\underline{z}^{*}, \overline{z}^{*}], \forall \boldsymbol{x}' \in B^{\epsilon}(\boldsymbol{x})$.

- **Propagation Invariance**: a network is propagation invariant if $\text{Box}^{\dagger}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x})) = \text{Box}^{*}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$, i.e., IBP is exact.

- **Propagation Tightness**: $\boldsymbol{\tau} = (\underline{z}^{*} - \overline{z}^{*})/(\overline{z}^{\dagger} - \underline{z}^{\dagger})$, i.e., the ratio of optimal and layer-wise box sizes.



Explicit IBP for Deep Linear Network (DLN)

Explicit IBP for Deep Linear Network (DLN)

- For DLN $\boldsymbol{f} = \prod_{k=1}^{L} \boldsymbol{W}^{(k)}$ and input box with radius $\boldsymbol{\epsilon}$, the size of approximations are:

Explicit IBP for Deep Linear Network (DLN)

- For DLN $\boldsymbol{f} = \prod_{k=1}^{L} \boldsymbol{W}^{(k)}$ and input box with radius $\boldsymbol{\epsilon}$, the size of approximations are:

$$\overline{z}^* - \underline{z}^* = 2 \left| \Pi_{k=1}^{L} \boldsymbol{W}^{(k)} \right| \boldsymbol{\epsilon}$$

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Explicit IBP for Deep Linear Network (DLN)

- For DLN $\boldsymbol{f} = \prod_{k=1}^{L} \boldsymbol{W}^{(k)}$ and input box with radius $\boldsymbol{\epsilon}$, the size of approximations are:

$$\bar{\boldsymbol{z}}^{*} - \underline{\boldsymbol{z}}^{*} = 2 \left| \Pi_{k=1}^{L} \boldsymbol{W}^{(k)} \right| \boldsymbol{\epsilon}$$

$$\bar{\boldsymbol{z}}^{\dagger} - \underline{\boldsymbol{z}}^{\dagger} = 2 \left(\Pi_{k=1}^{L} \left| \boldsymbol{W}^{(k)} \right| \right) \boldsymbol{\epsilon}$$

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Explicit IBP for Deep Linear Network (DLN)

- For DLN $\boldsymbol{f} = \prod_{k=1}^{L} \boldsymbol{W}^{(k)}$ and input box with radius $\boldsymbol{\epsilon}$, the size of approximations are:

$$\bar{\boldsymbol{z}}^{*} - \underline{\boldsymbol{z}}^{*} = 2 \left| \Pi_{k=1}^{L} \boldsymbol{W}^{(k)} \right| \boldsymbol{\epsilon}$$

$$\bar{\boldsymbol{z}}^{\dagger} - \underline{\boldsymbol{z}}^{\dagger} = 2 \left(\Pi_{k=1}^{L} \left| \boldsymbol{W}^{(k)} \right| \right) \boldsymbol{\epsilon}$$

- DLN with all non-negative weights is propagation invariant.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Propagation Invariance

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Propagation Invariance

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is propagation invariant if and only if $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \geq 0$ for all k or $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \leq 0$ for all k.

Propagation Invariance

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is propagation invariant if and only if $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \geq 0$ for all k or $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \leq 0$ for all k.

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is not propagation invariant if $\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i,j}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i,j'}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i',j}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i',j'} < 0$ for some i, j.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Propagation Invariance

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is propagation invariant if and only if $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \geq 0$ for all k or $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \leq 0$ for all k.

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is not propagation invariant if $\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i,j}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i,j'}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i',j}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i',j'} < 0$ for some i, j.

- $\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)} = \begin{pmatrix} 1 & 2 \\ -3 & 4 \end{pmatrix}$ ➔ not propagation invariant.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Propagation Invariance

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is propagation invariant if and only if $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \geq 0$ for all k or $\boldsymbol{W}^{(2)}_{i,k} \cdot \boldsymbol{W}^{(1)}_{k,j} \leq 0$ for all k.

- A two-layer DLN $\boldsymbol{f} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}$ is not propagation invariant if $\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i,j}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i,j'}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i',j}\left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)_{i',j'} < 0$ for some i, j.

- $\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)} = \begin{pmatrix} 1 & 2 \\ -3 & 4 \end{pmatrix}$ ➔ not propagation invariant.

- A two-layer propagation invariant DLN has $O(N)$ degree of freedom for parameter signs, while a general two-layer DLN has $O(N^2)$.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Box Reconstruction Error

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

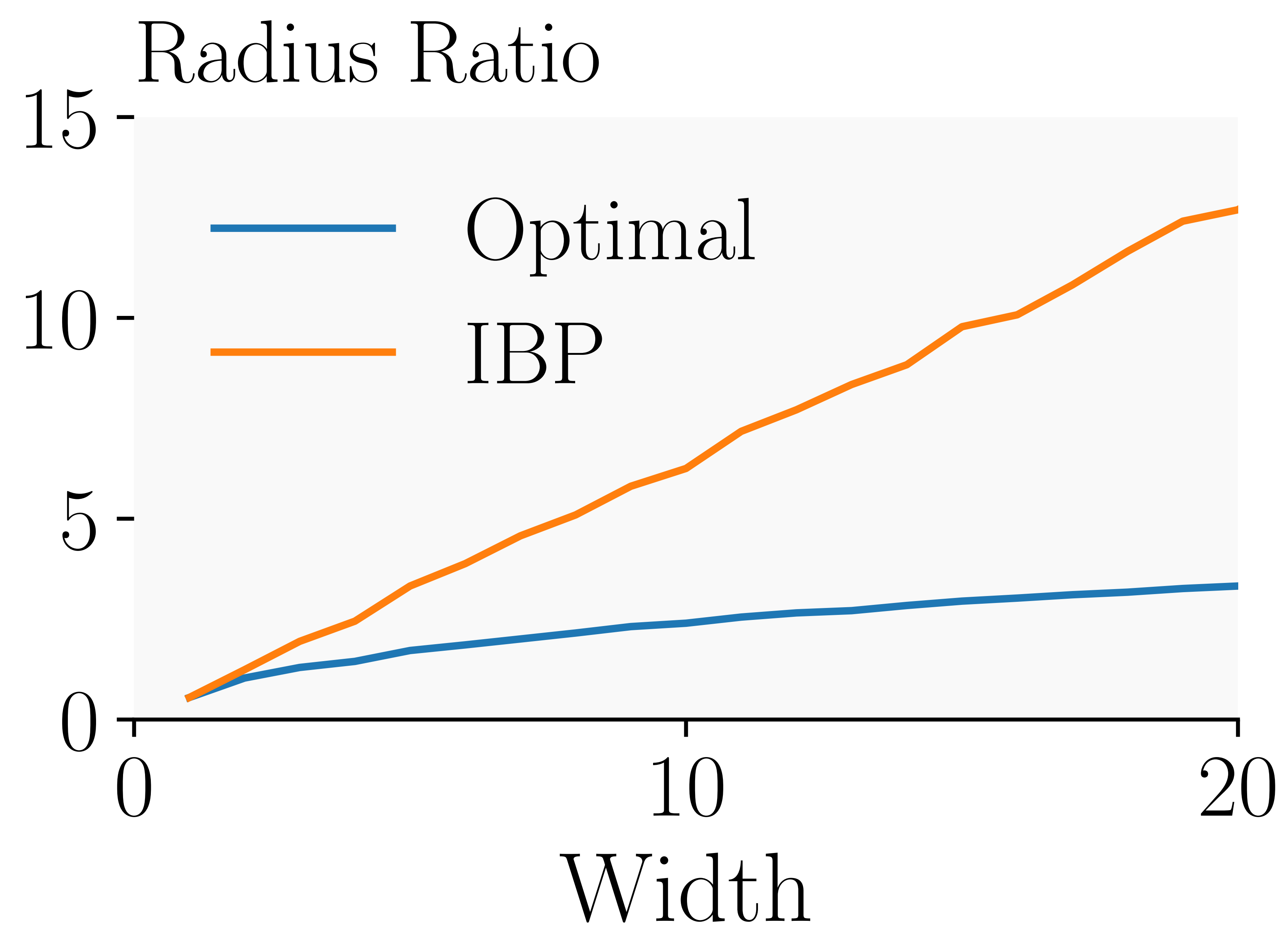
Box Reconstruction Error

- For linearly separable data, PCA (optimal) weights lead to linear growth of layer-wise box size and sqrt growth of optimal box size with regard to instrinsic dimension.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Box Reconstruction Error

- For linearly separable data, PCA (optimal) weights lead to linear growth of layer-wise box size and sqrt growth of optimal box size with regard to instrinsic dimension.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Tightness at Initialization

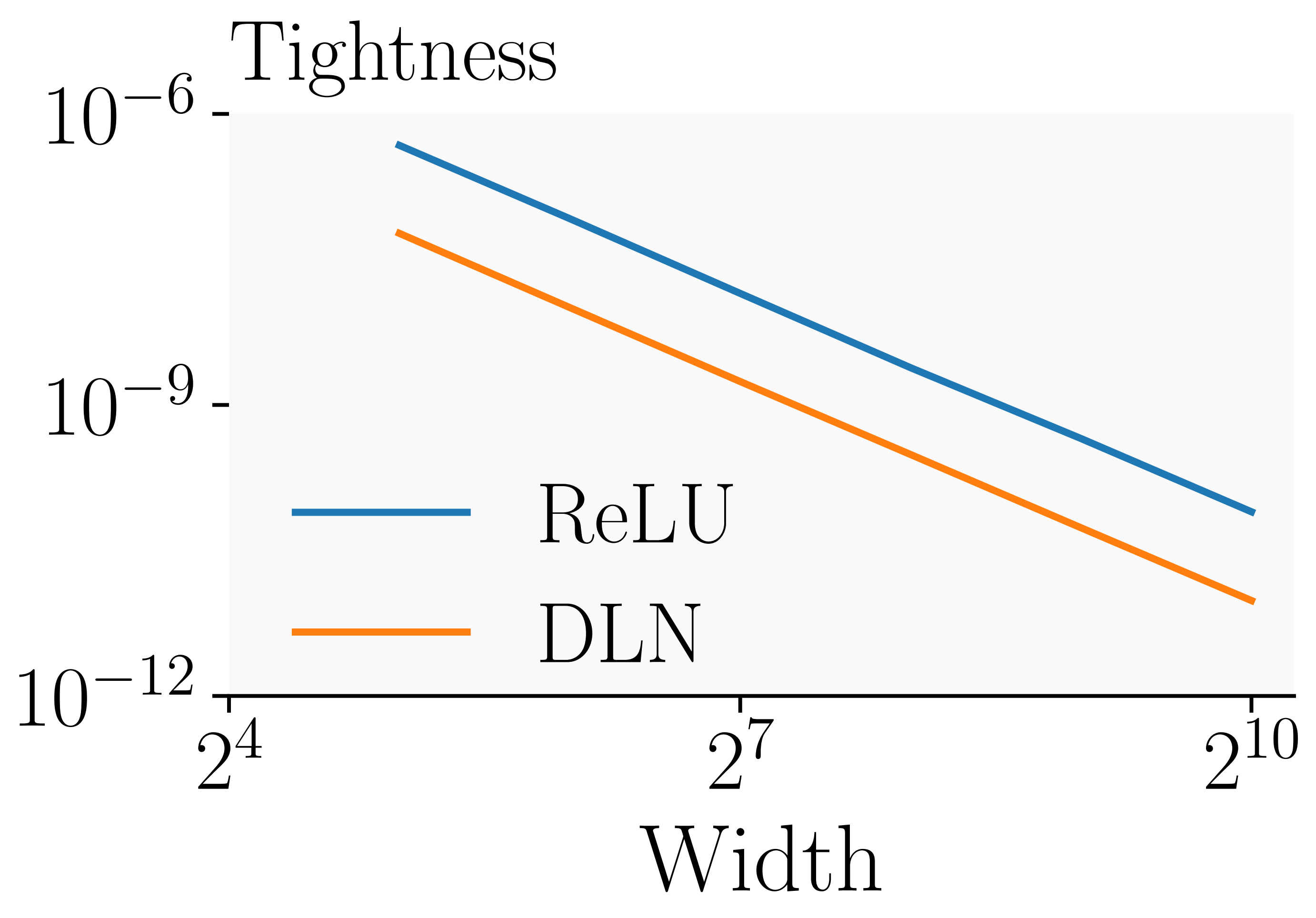
Tightness at Initialization

- For two-layer DLN with weights sampled from i.i.d. Gaussian distribution and hidden dimension d, tightness decreases in squared root order of d: $\boldsymbol{\tau} = \Theta(d^{-1/2})$.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Tightness at Initialization

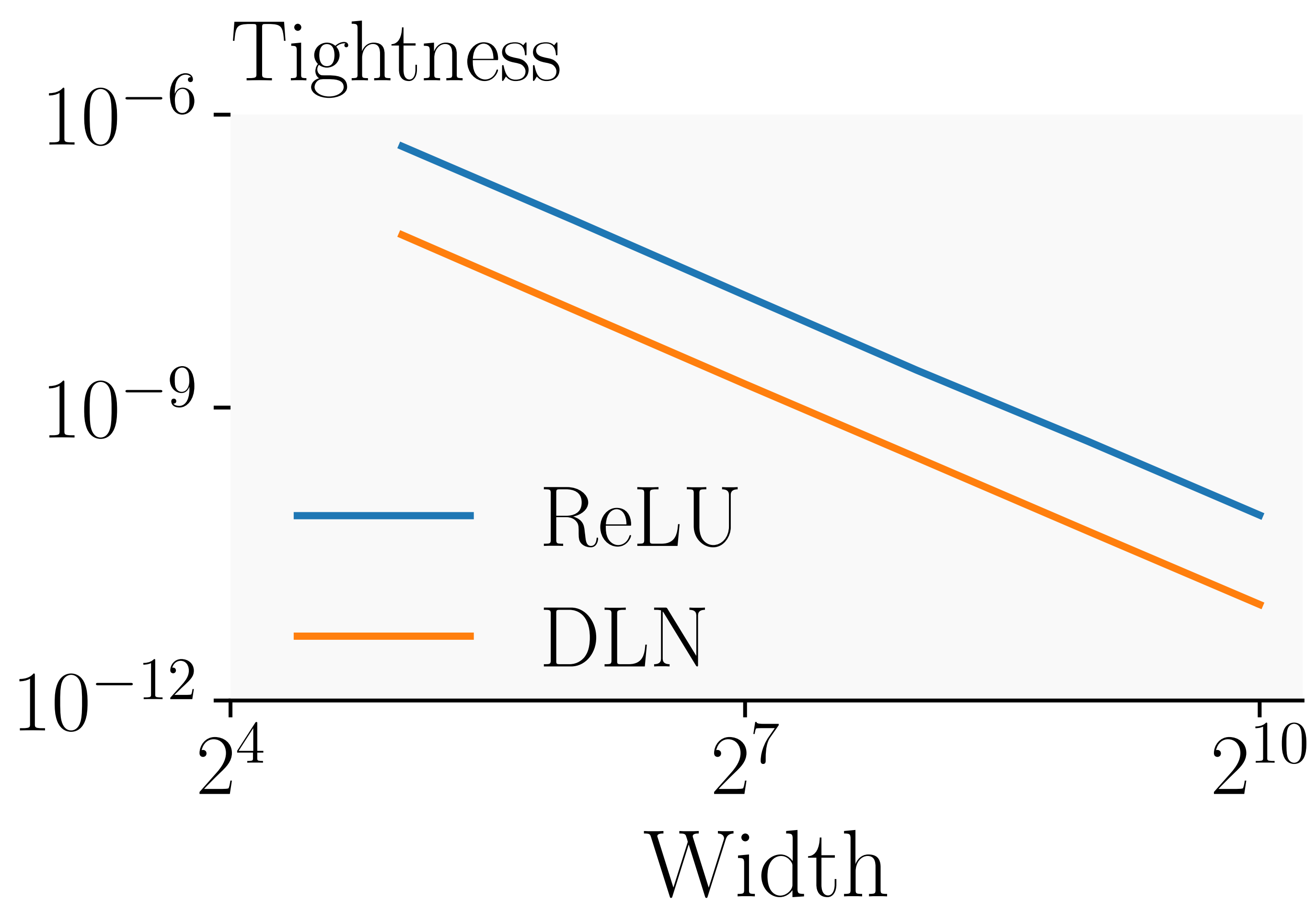
- For two-layer DLN with weights sampled from i.i.d. Gaussian distribution and hidden dimension d, tightness decreases in squared root order of d: $\boldsymbol{\tau} = \Theta(d^{-1/2})$.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Tightness at Initialization

- For two-layer DLN with weights sampled from i.i.d. Gaussian distribution and hidden dimension d, tightness decreases in squared root order of d: $\boldsymbol{\tau} = \Theta(d^{-1/2})$.

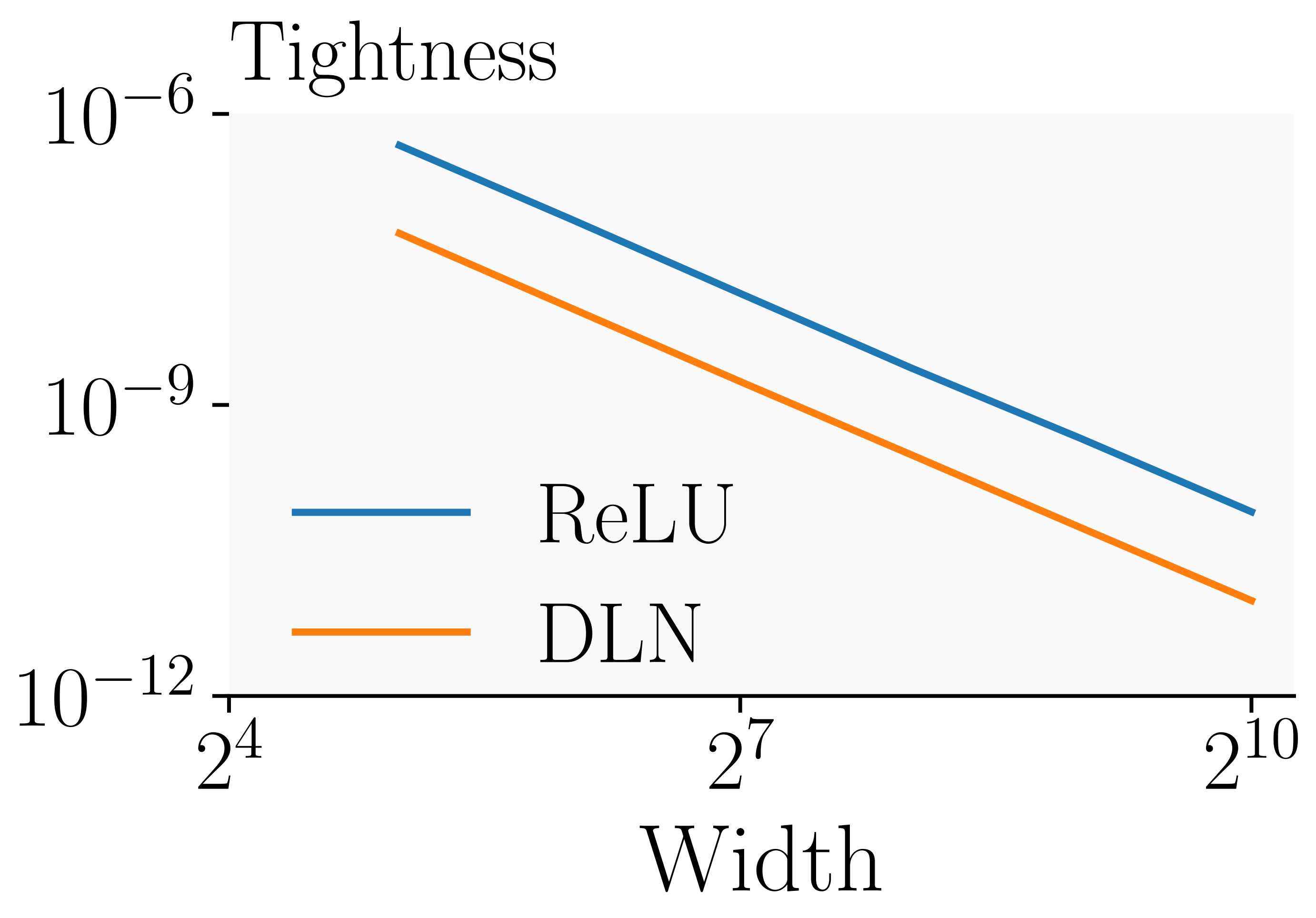


- For L-layer DLN randomly initialized with i.i.d. Gaussian and minimum hidden dimension d, tightness decreases in exponential order of L: $\boldsymbol{\tau} = O(d^{-\lfloor L/4 \rfloor})$.

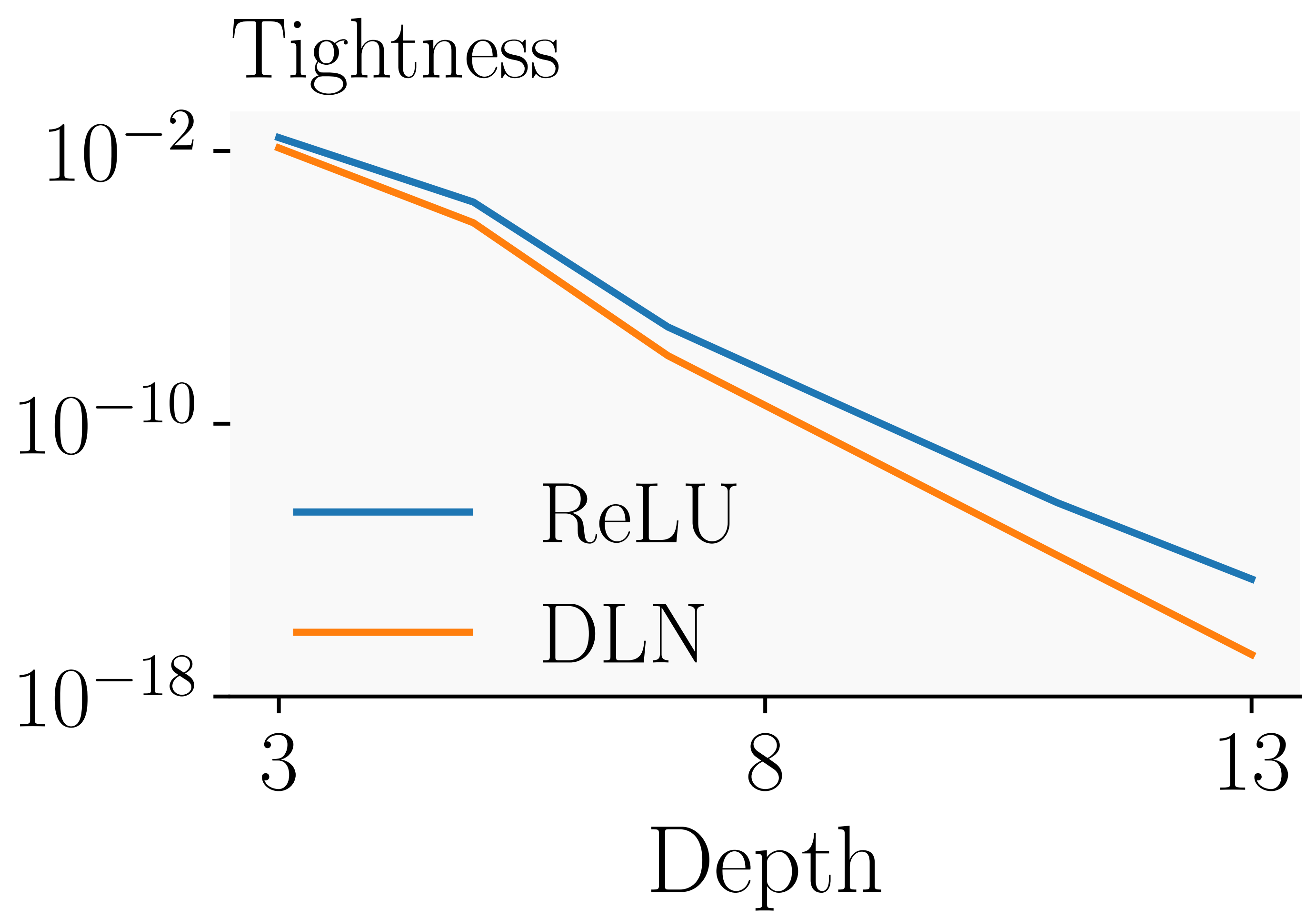
Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Tightness at Initialization

- For two-layer DLN with weights sampled from i.i.d. Gaussian distribution and hidden dimension d, tightness decreases in squared root order of d: $\boldsymbol{\tau} = \Theta(d^{-1/2})$.



- For L-layer DLN randomly initialized with i.i.d. Gaussian and minimum hidden dimension d, tightness decreases in exponential order of L: $\boldsymbol{\tau} = O(d^{-\lfloor L/4 \rfloor})$.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

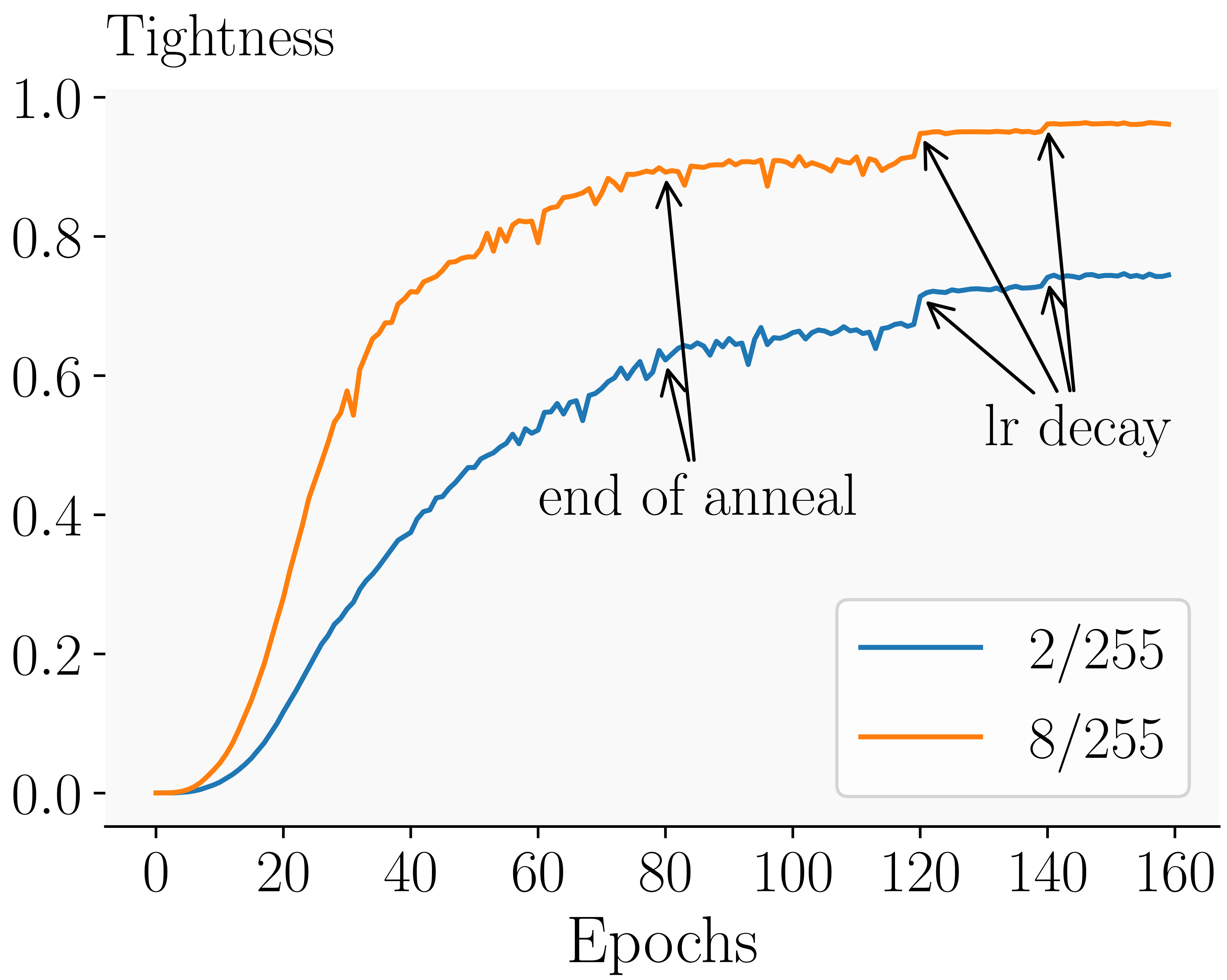
IBP Increases Tightness

IBP Increases Tightness

- If $\mathrm{Box}^{\dagger}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$ deviates too much from $\mathrm{Box}^{*}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$, then the gradient difference between IBP and standard loss is aligned with an increase in tightness, i.e., IBP-trained models have larger tightness.

IBP Increases Tightness

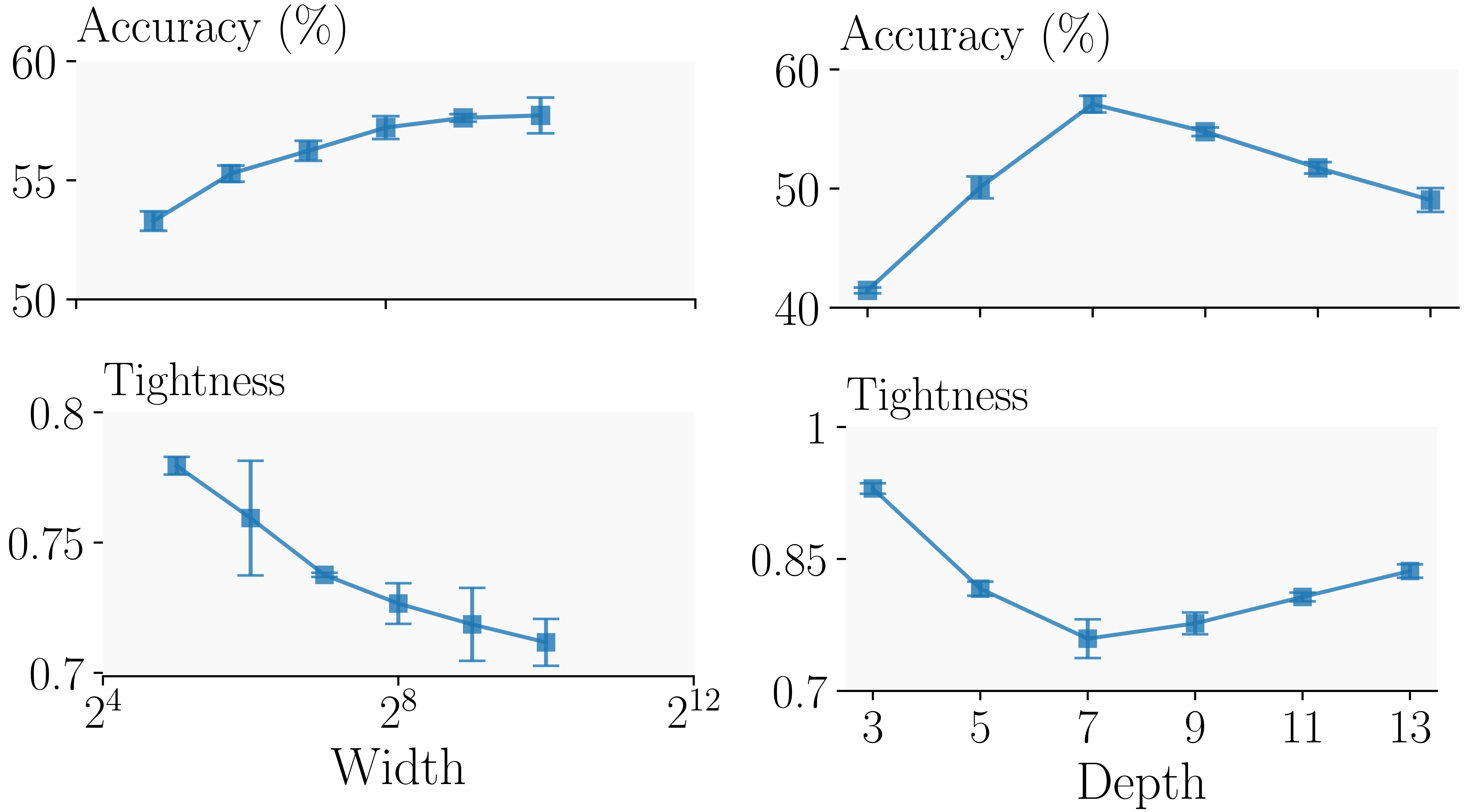
- If $\mathrm{Box}^{\dagger}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$ deviates too much from $\mathrm{Box}^{*}(\boldsymbol{f}, B^{\epsilon}(\boldsymbol{x}))$, then the gradient difference between IBP and standard loss is aligned with an increase in tightness, i.e., IBP-trained models have larger tightness.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Generalization to Trained ReLU Nets

Width brings less regularization than depth.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

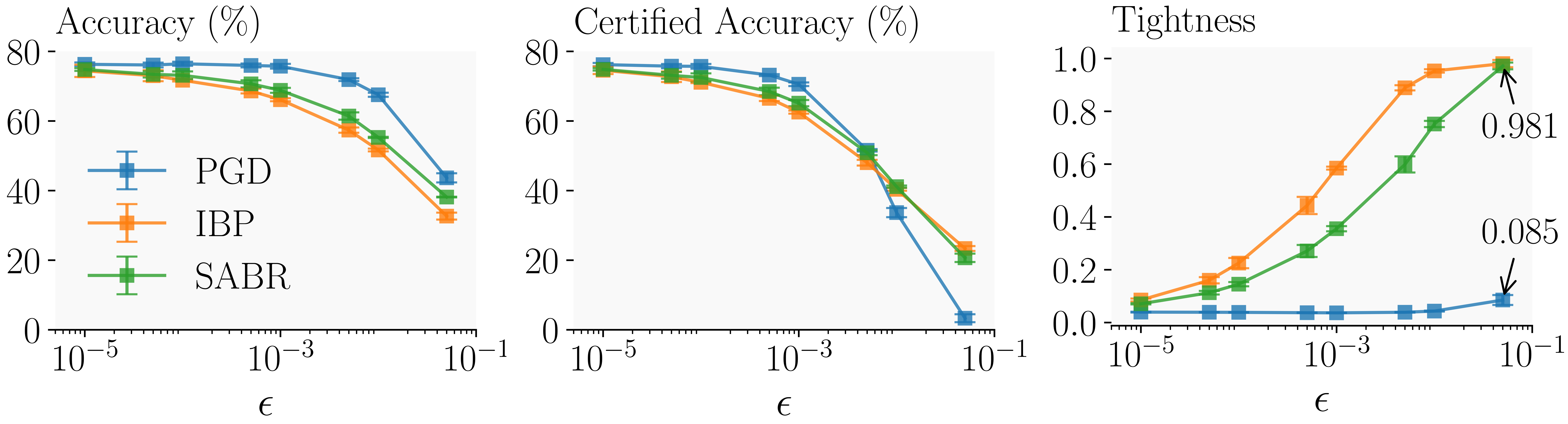
Generalization to Trained ReLU Nets

Width-scale Rule
Predicts Better Models.

Dataset	ϵ	Method	Width	Accuracy	Certified
MNIST	0.1	IBP	1×	98.83	98.10
			4×	98.86	98.23
		SABR	1×	98.99	98.20
			4×	**98.99**	**98.32**
	0.3	IBP	1×	97.44	93.26
			4×	97.66	93.35
		SABR	1×	**98.82**	93.38
			4×	98.48	**93.85**
CIFAR-10	$\frac{2}{255}$	IBP	1×	67.93	55.85
			2×	68.06	56.18
		IBP-R	1×	78.43	60.87
			2×	**80.46**	62.03
		SABR	1×	79.24	62.84
			2×	79.89	**63.28**
	$\frac{8}{255}$	IBP	1×	47.35	34.17
			2×	47.83	33.98
		SABR	1×	50.78	34.12
			2×	**51.56**	**34.95**
TinyImageNet	$\frac{1}{255}$	IBP	0.5×	24.47	18.76
			1×	25.33	19.46
			2×	25.40	19.92
		SABR	0.5×	27.56	20.54
			1×	28.63	21.21
			2×	**28.97**	**21.36**

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

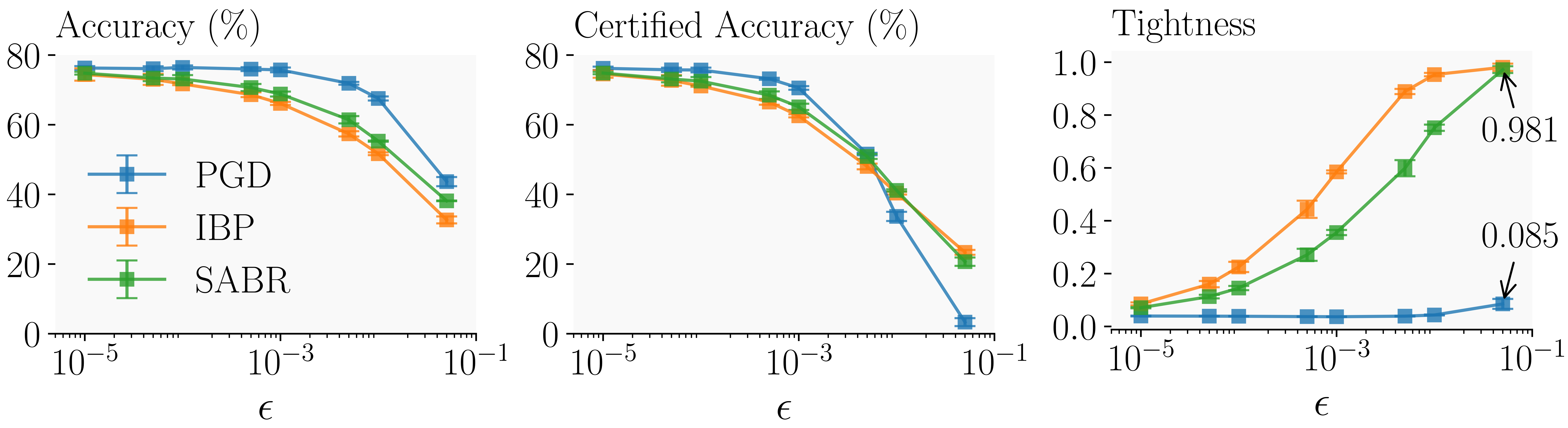
Generalization to Trained ReLU Nets



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Generalization to Trained ReLU Nets

Larger input box leads to larger tightness.

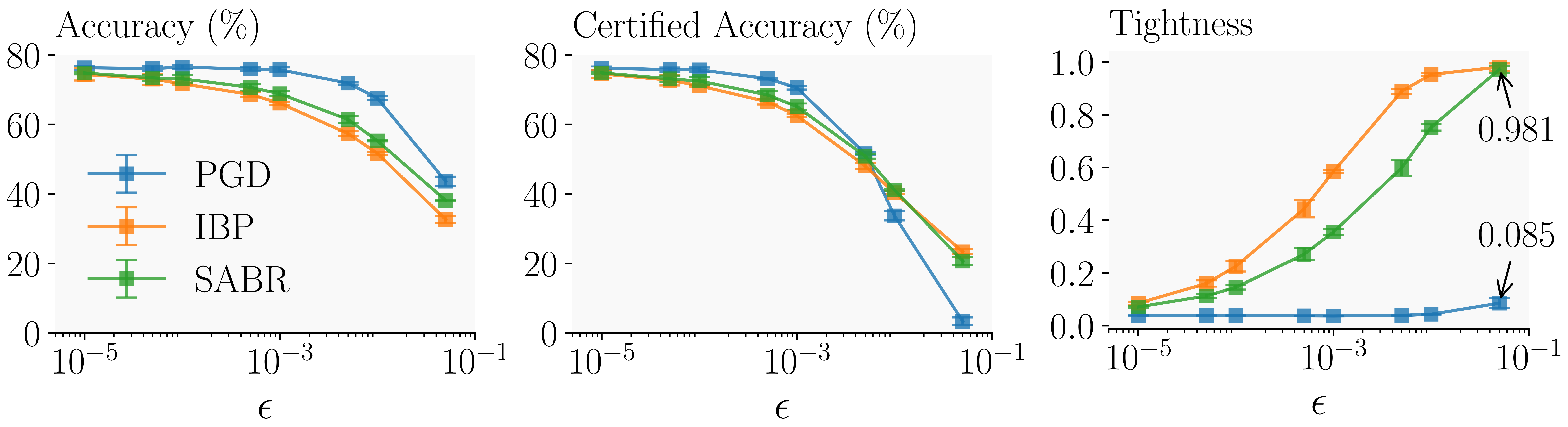


Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Generalization to Trained ReLU Nets

Larger input box leads to larger tightness.

Propagation Invariance is associated with strong regularization.



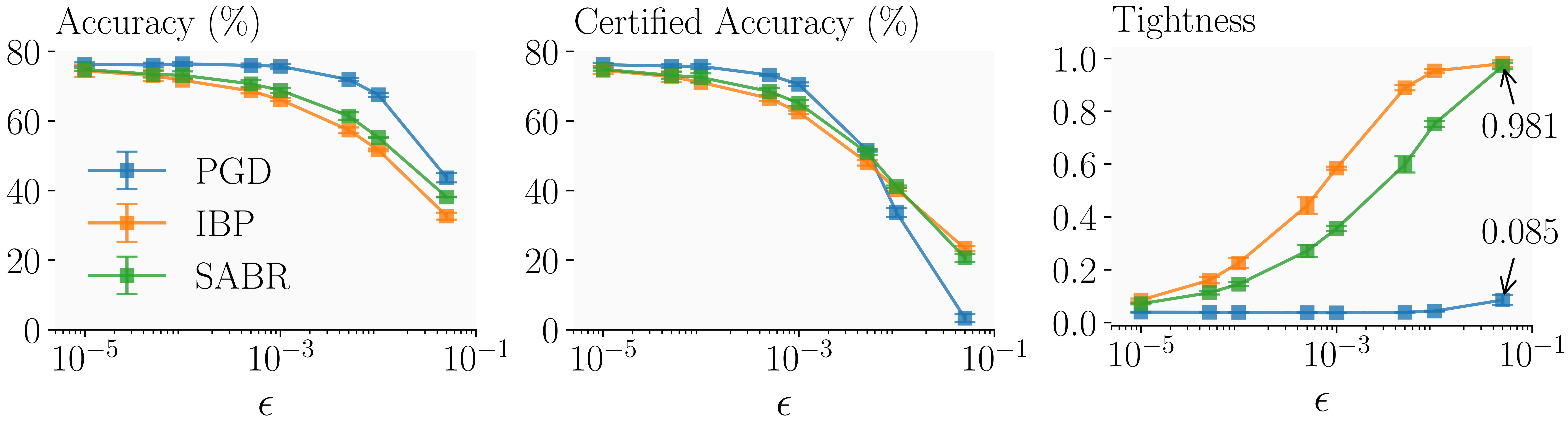
Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

Generalization to Trained ReLU Nets

Larger input box leads to larger tightness.

Propagation Invariance is associated with strong regularization.

IBP > SABR > PGD consistently in terms of tightness.



Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR’24

IBP-based vs non-IBP-based

Method	ϵ	Accuracy	Tightness	Certified
PGD	2/255	81.2	0.001	-
	8/255	69.3	0.007	-
COLT	2/255	78.4*	0.009	60.7*
	8/255	51.7*	0.057	26.7*
IBP-R	2/255	78.2*	0.033	62.0*
	8/255	51.4*	0.124	27.9*
SABR	2/255	75.6	0.182	57.7
	8/255	48.2	0.950	31.2
IBP	2/255	63.0	0.803	51.3
	8/255	42.2	0.977	31.0

* Literature result.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

IBP-based vs non-IBP-based

- IBP-based methods get significantly larger tightness (17x to 80x).

Method	ϵ	Accuracy	Tightness	Certified
PGD	2/255	81.2	0.001	-
	8/255	69.3	0.007	-
COLT	2/255	78.4*	0.009	60.7*
	8/255	51.7*	0.057	26.7*
IBP-R	2/255	78.2*	0.033	62.0*
	8/255	51.4*	0.124	27.9*
SABR	2/255	75.6	0.182	57.7
	8/255	48.2	0.950	31.2
IBP	2/255	63.0	0.803	51.3
	8/255	42.2	0.977	31.0

* Literature result.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

IBP-based vs non-IBP-based

- IBP-based methods get significantly larger tightness (17x to 80x).
- Certified method with no IBP component (COLT) still has significantly larger tightness than PGD (8x).

Method	ϵ	Accuracy	Tightness	Certified
PGD	2/255	81.2	0.001	-
	8/255	69.3	0.007	-
COLT	2/255	78.4*	0.009	60.7*
	8/255	51.7*	0.057	26.7*
IBP-R	2/255	78.2*	0.033	62.0*
	8/255	51.4*	0.124	27.9*
SABR	2/255	75.6	0.182	57.7
	8/255	48.2	0.950	31.2
IBP	2/255	63.0	0.803	51.3
	8/255	42.2	0.977	31.0

* Literature result.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR'24

IBP-based vs non-IBP-based

- IBP-based methods get significantly larger tightness (17x to 80x).
- Certified method with no IBP component (COLT) still has significantly larger tightness than PGD (8x).
- Large tightness seems necessary for large ϵ (see SABR).

Method	ϵ	Accuracy	Tightness	Certified
PGD	2/255	81.2	0.001	-
	8/255	69.3	0.007	-
COLT	2/255	78.4*	0.009	60.7*
	8/255	51.7*	0.057	26.7*
IBP-R	2/255	78.2*	0.033	62.0*
	8/255	51.4*	0.124	27.9*
SABR	2/255	75.6	0.182	57.7
	8/255	48.2	0.950	31.2
IBP	2/255	63.0	0.803	51.3
	8/255	42.2	0.977	31.0

* Literature result.

Mao et. al., Understanding Certified Training with Interval Bound Propagation, ICLR’24

Take-away

Take-away

- We quantify Interval Bound Propagation, the key component of all SOTA methods in recent years, in terms of approximation error.

Take-away

- We quantify Interval Bound Propagation, the key component of all SOTA methods in recent years, in terms of approximation error.
- We theoretically prove that (1) it leads to strong regularization on the parameter signs, (2) it requires more model capacity, and (3) it benefits more from width than depth.

Take-away

- We quantify Interval Bound Propagation, the key component of all SOTA methods in recent years, in terms of approximation error.
- We theoretically prove that (1) it leads to strong regularization on the parameter signs, (2) it requires more model capacity, and (3) it benefits more from width than depth.
- Based on our insights, we explain the improvement of recent SOTA over IBP and successfully push SOTA further by simply increasing the model width.

Part 4

The Future of (Deterministic) Neural Network Verification

Infeasibility of Single-Neuron Relaxation

Baader et. al., Expressivity of ReLU-networks Under Convex Relaxation, ICLR'24.
Ferrari et. al., Complete Verification via Multi-Neuron Relaxation Guided Branch-and-Bound, ICLR'22.

Infeasibility of Single-Neuron Relaxation

- The most precise single-neuron convex relaxation (triangle) is unable to precisely encode $max(x_1, x_2)$ with arbitrary ReLU network.

Baader et. al., Expressivity of ReLU-networks Under Convex Relaxation, ICLR'24.
Ferrari et. al., Complete Verification via Multi-Neuron Relaxation Guided Branch-and-Bound, ICLR'22.

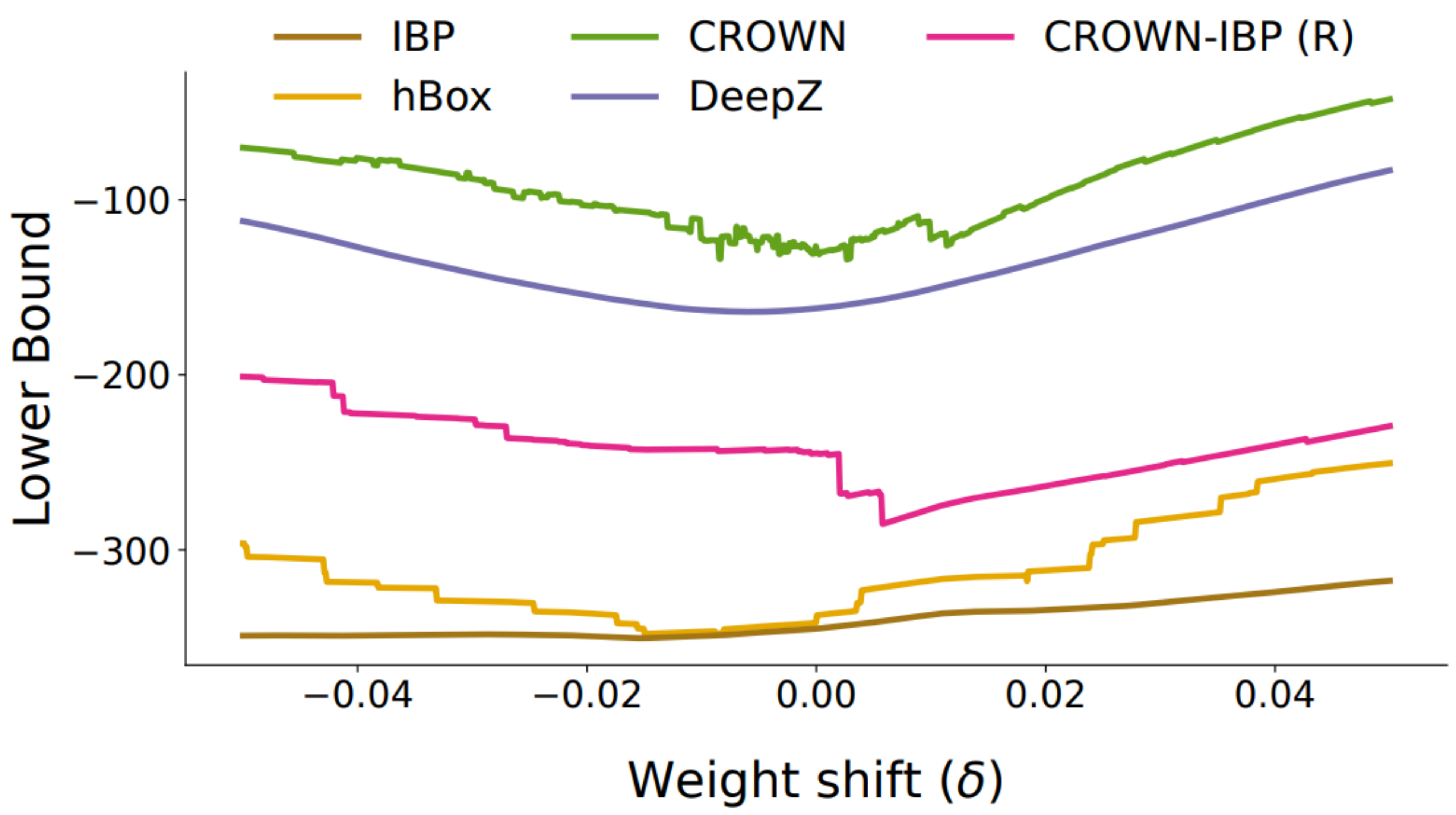
Infeasibility of Single-Neuron Relaxation

- The most precise single-neuron convex relaxation (triangle) is unable to precisely encode $max(x_1, x_2)$ with arbitrary ReLU network.
- Multi-neuron relaxation is key to designing future verifiers.

Baader et. al., Expressivity of ReLU-networks Under Convex Relaxation, ICLR'24.
Ferrari et. al., Complete Verification via Multi-Neuron Relaxation Guided Branch-and-Bound, ICLR'22.

Bad Gradients from Precise Relaxation

Relaxation	Tightness	Certified (%)
IBP / Box	0.73	86.8
hBox / Symbolic Intervals	1.76	83.7
CROWN / DeepPoly	3.36	70.2
DeepZ / CAP / FastLin / Neurify	3.00	69.8
CROWN-IBP (R)	2.15	75.4

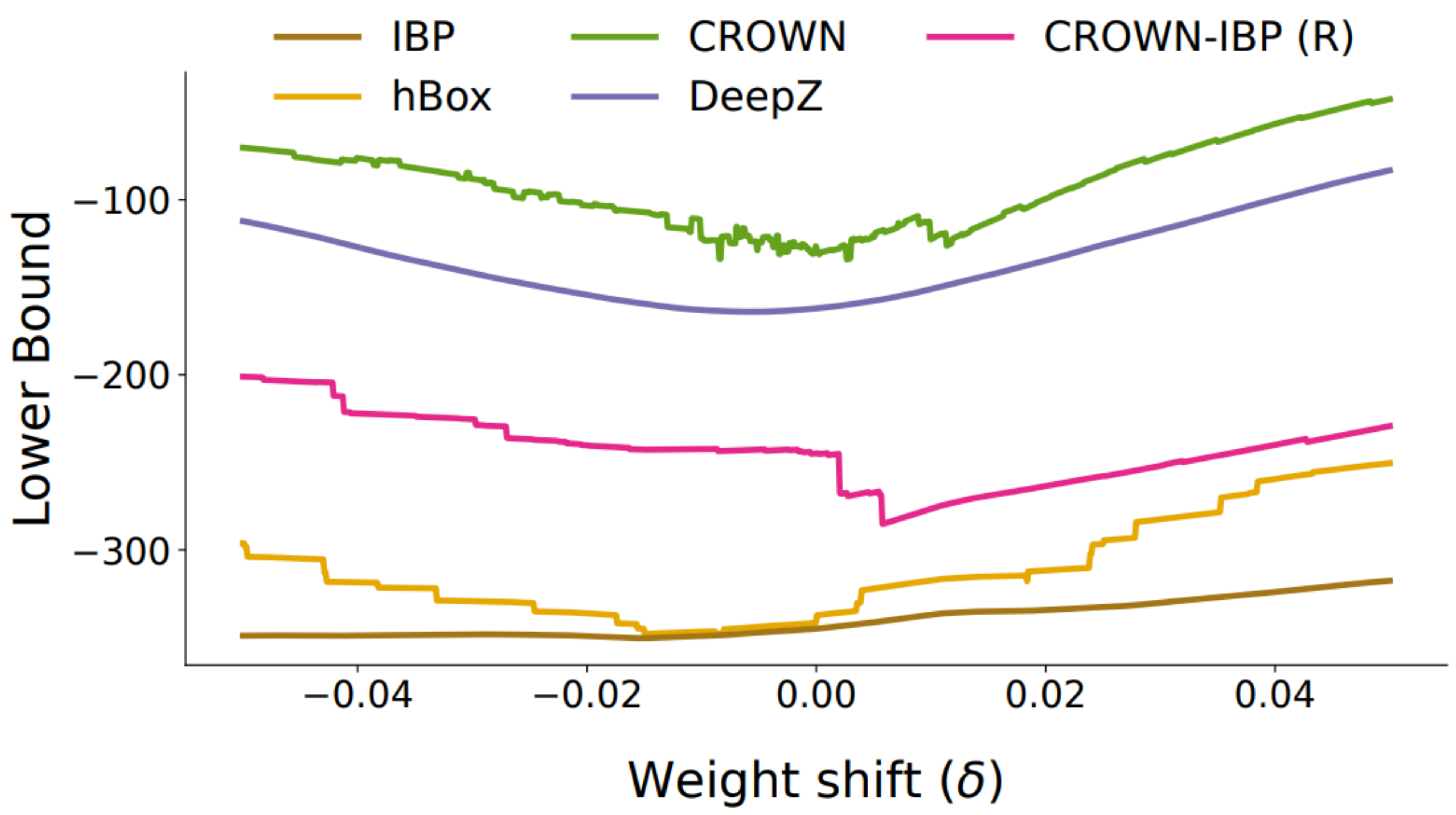


Jovanovic et. al., On the paradox of certified training, TMLR’22.

Bad Gradients from Precise Relaxation

- While being the least precise, IBP training gets better results than all the other precise domains.

Relaxation	Tightness	Certified (%)
IBP / Box	0.73	86.8
hBox / Symbolic Intervals	1.76	83.7
CROWN / DeepPoly	3.36	70.2
DeepZ / CAP / FastLin / Neurify	3.00	69.8
CROWN-IBP (R)	2.15	75.4

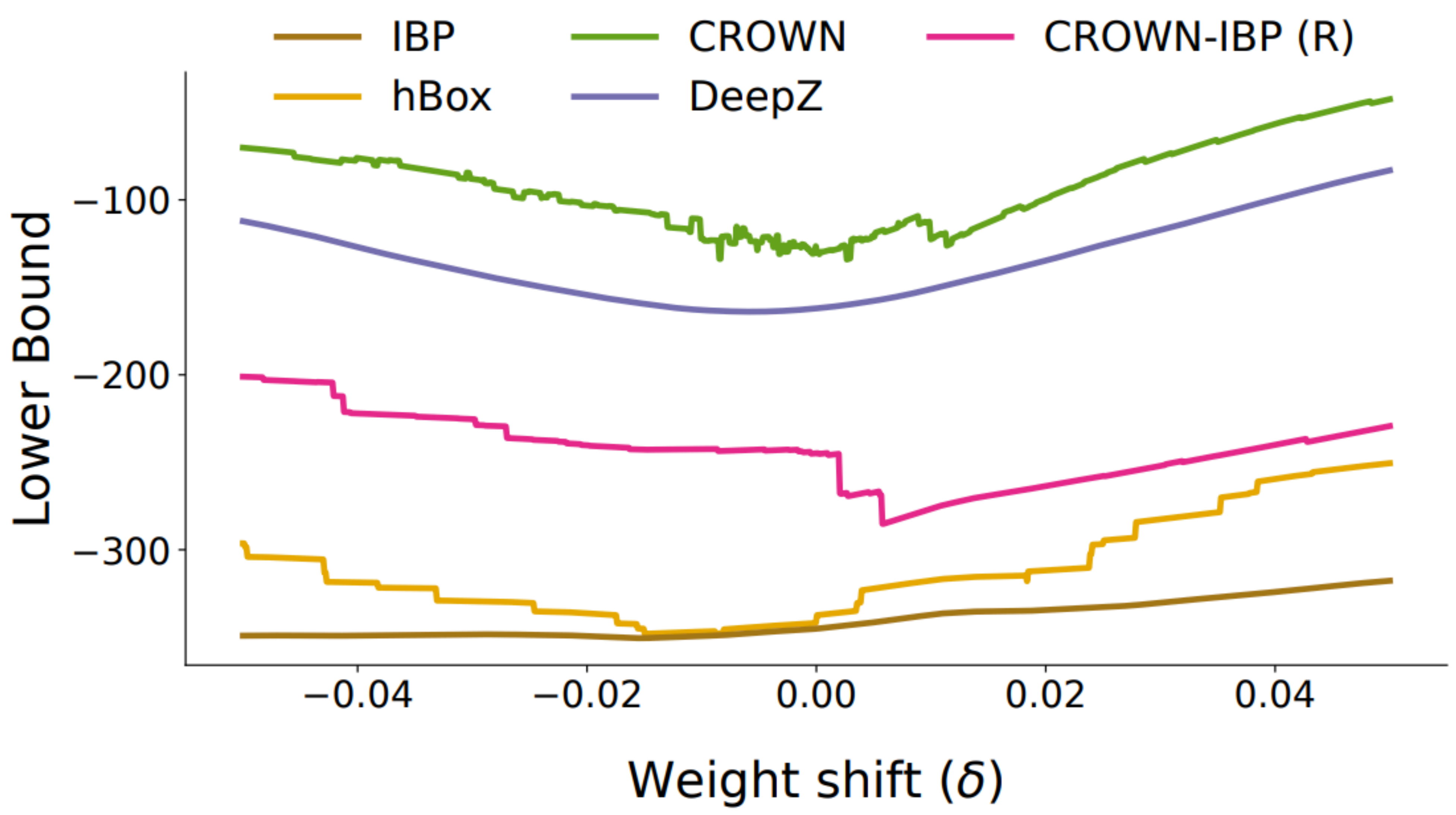


Bad Gradients from Precise Relaxation

- While being the least precise, IBP training gets better results than all the other precise domains.

- More precise methods with decent gradient quality is key to future certified training methods, e.g., SABR and TAPS.

Relaxation	Tightness	Certified (%)
IBP / Box	0.73	86.8
hBox / Symbolic Intervals	1.76	83.7
CROWN / DeepPoly	3.36	70.2
DeepZ / CAP / FastLin / Neurify	3.00	69.8
CROWN-IBP (R)	2.15	75.4



Jovanovic et. al., On the paradox of certified training, TMLR'22.