# Formal Verification of Neural Networks using Linearity Grafting

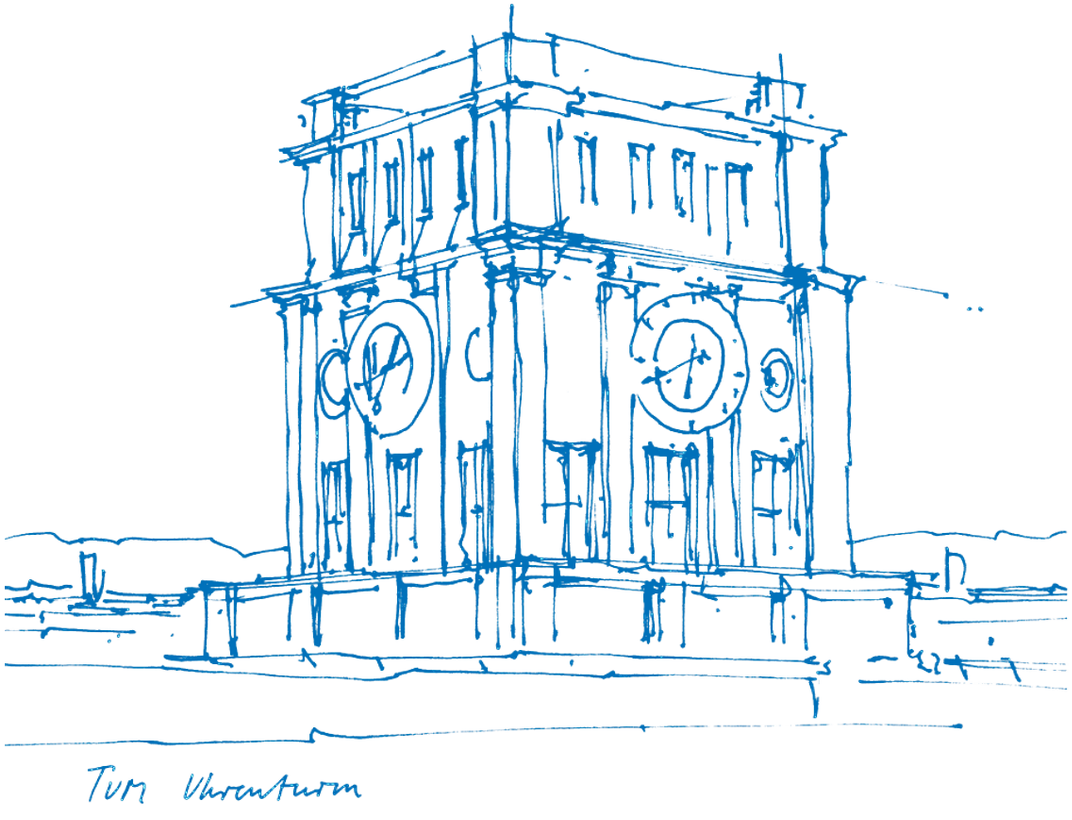
## Bachelor’s Thesis Presentation

Felix Brandis

Supervisor: Prof. Dr. Jan Křetínský
Advisor: Stefanie Mohr

Technical University of Munich
Chair for Foundations of Software Reliability and Theoretical Computer Science

06.10.2023



# Overview

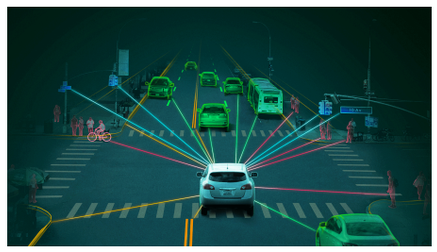
**Motivation & Recap:**
- Neural Network Robustness
- Adversarial Examples
- Verification

**Linearity Grafting:**
- Idea and original results
- My matters of interest
- Own experiments and results

**Wrap-up:**
- Conclusions
- Outlook

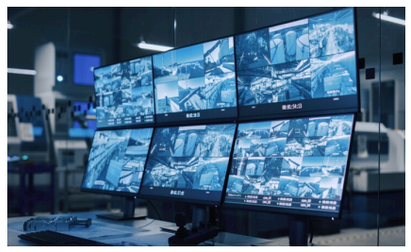
# Motivation: Robustness of Neural Networks

## 1) Deployment of NN-based Technology in safety-critical domains



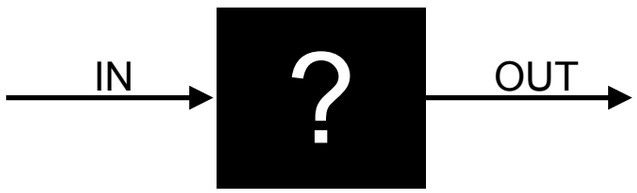
Autonomous Vehicles [1]

Medical AI [2]

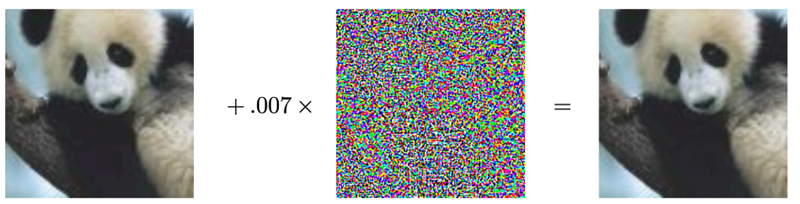


Security and Surveillance Systems [3]

## 2) Black Box Nature of NNs




## 3) Existence of Adversarial Attacks

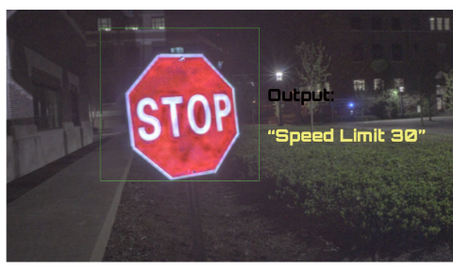



$\boldsymbol{x}$
"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

[4]




Stop sign classified as speed limit sign [5]

[1] https://researchleap.com/wp-content/uploads/2021/12/AI_Drive_Reasoning-002.png
[2] https://www.intel.de/content/dam/www/public/us/en/images/iot/rwd/a1042070-medical-imaging-iot-healtchare-brain-scans-rwd.jpg.rendition.intel.web.576.324.jpg
[3] https://erepublic.brightspotcdn.com/dims4/default/4904193/2147483647/strip/true/crop/1000x521+0+21/resize/840x438!/quality/90/?url=http%3A%2F%2Ferepublic-brightspot.s3.us-west-2.amazonaws.com%2Fd8%2Fcf%2F686632c344a09d77b7f7f40bcf96%2Fsurveillance-room.jpg
[4] From „Explaining and harnessing adversarial examples“ by Goodfellow et al. 2015
[5] "Optical adversarial attack" by Gnanasambandam et al., ICCV 2021

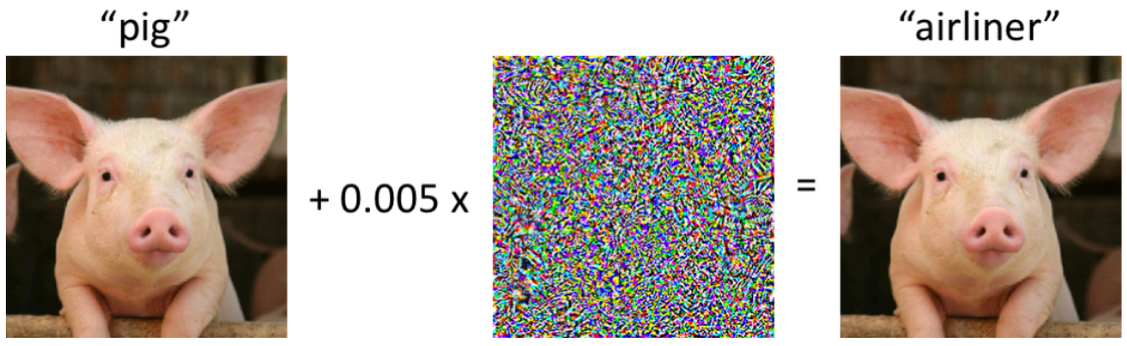
# Recap: Adversarial Examples

**Optimization problem** of finding a minimal perturbation to the input to cause a misclassification (optionally with specified target class)

Find $x^*$ with

$$\boldsymbol{x}^* = \boldsymbol{x} + \arg\min_{\delta \boldsymbol{x}} \{ \|\delta \boldsymbol{x}\| \mid \hat{k}(\boldsymbol{x} + \delta \boldsymbol{x}) \neq \hat{k}(\boldsymbol{x}) \}$$

- Multitude of attack strategies
- Examples often transferable between models

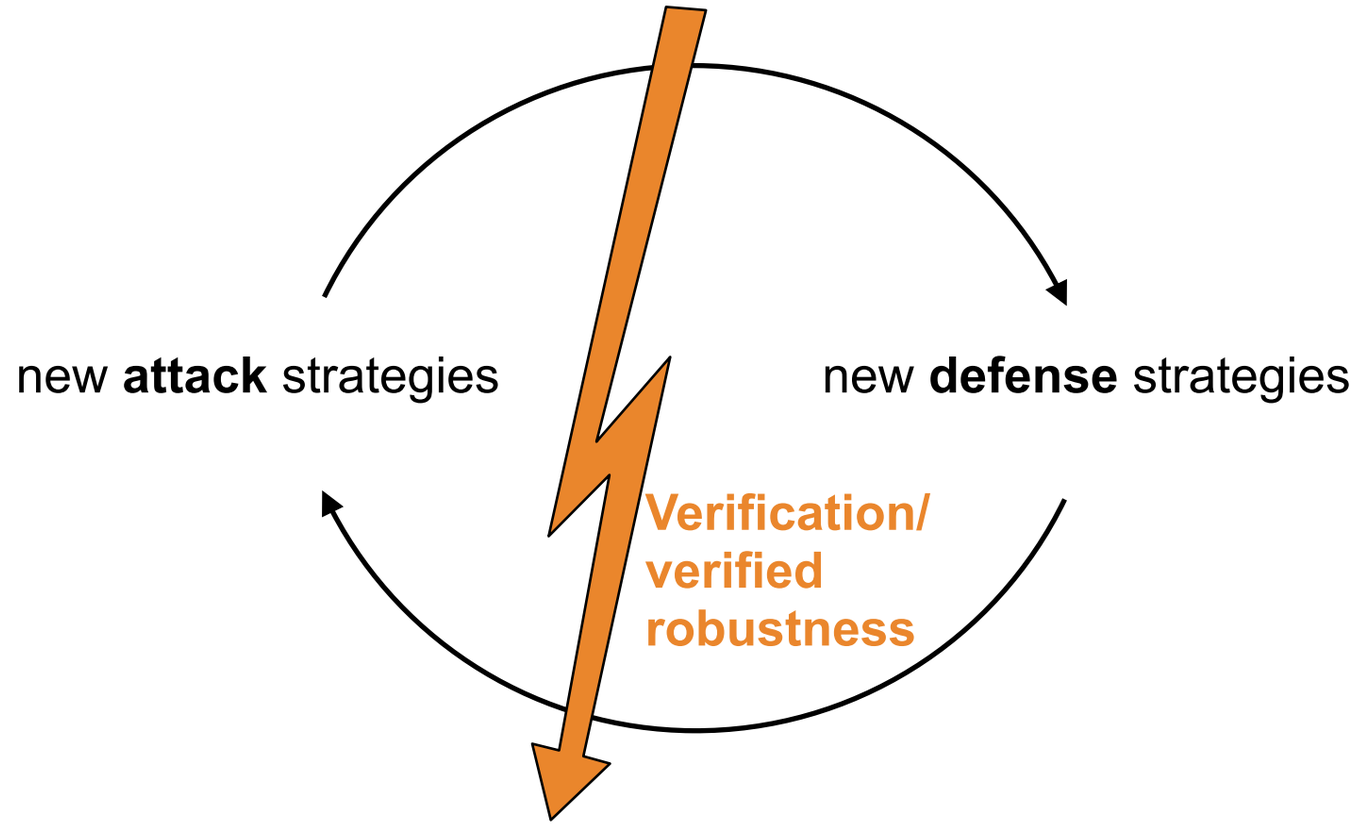



[1]

[1] https://gradientscience.org/images/piggie.png

# Motivation: Verification

**Empirical Robustness:**

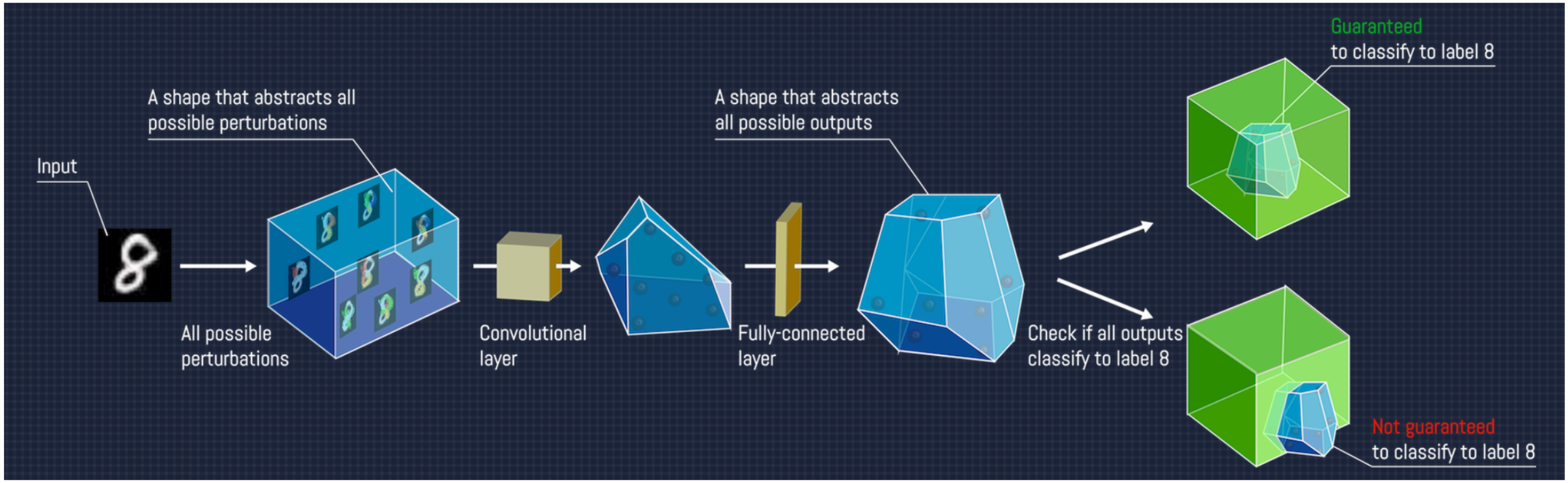



- Evaluation of new strategies not as standardized as in more traditional security related fields [1]
- Generally little resources for reproduction of results
- Numerous flaws shown in many defense techniques [2]

→ **Elegant solution:** Show **non- existence** of adversarial examples within a given perturbation radius = Verification

[1] “On evaluating adversarial robustness“ by Carlini et al., 2019
[2] „Adversarial examples are not easily detected: Bypassing ten detection methods“ by Carlini and Wagner, 2017

# Recap: Verification




Visualization of bound propagation [1]

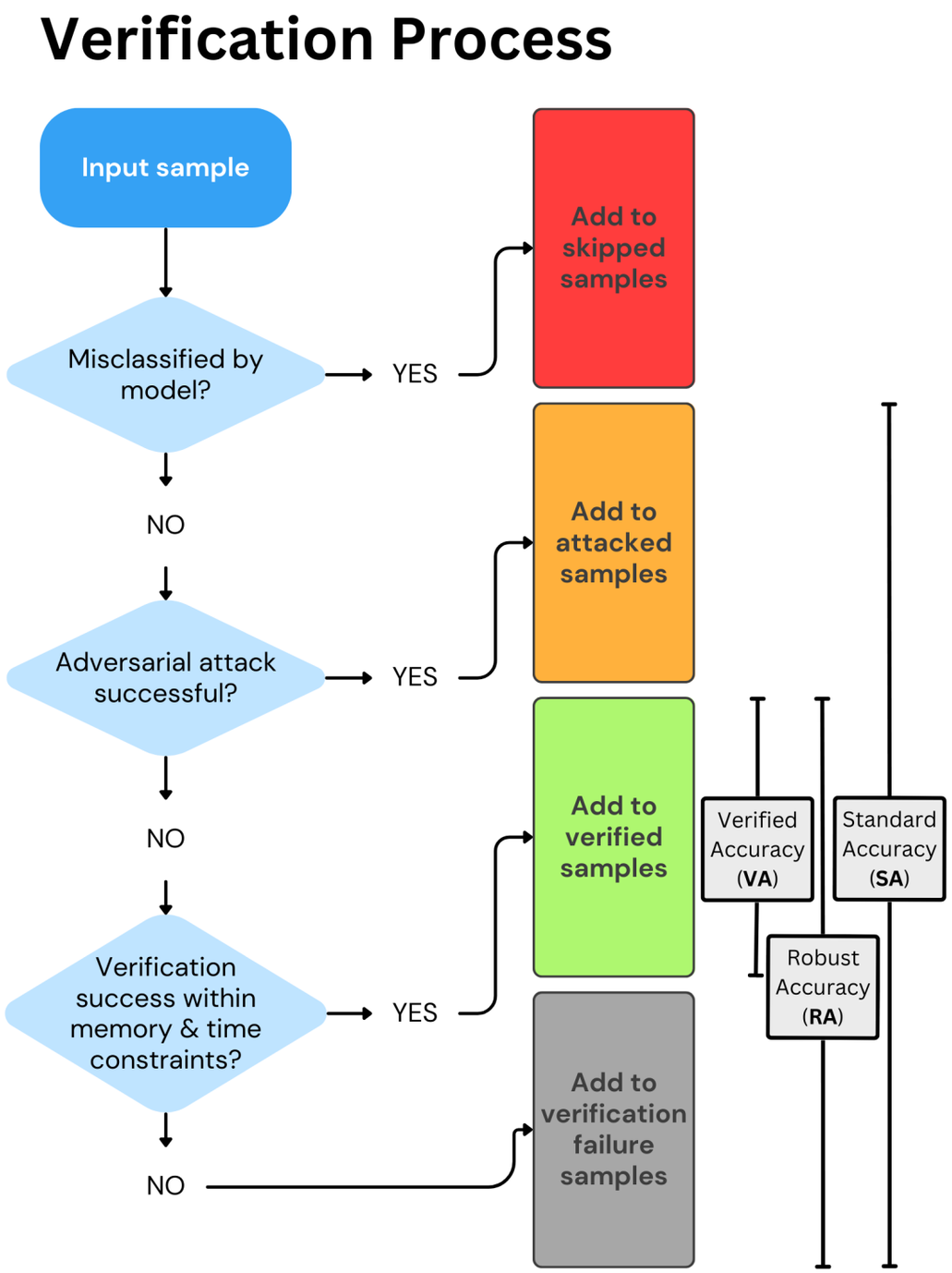
[1] https://safeai.ethz.ch/

# Recap: Verification

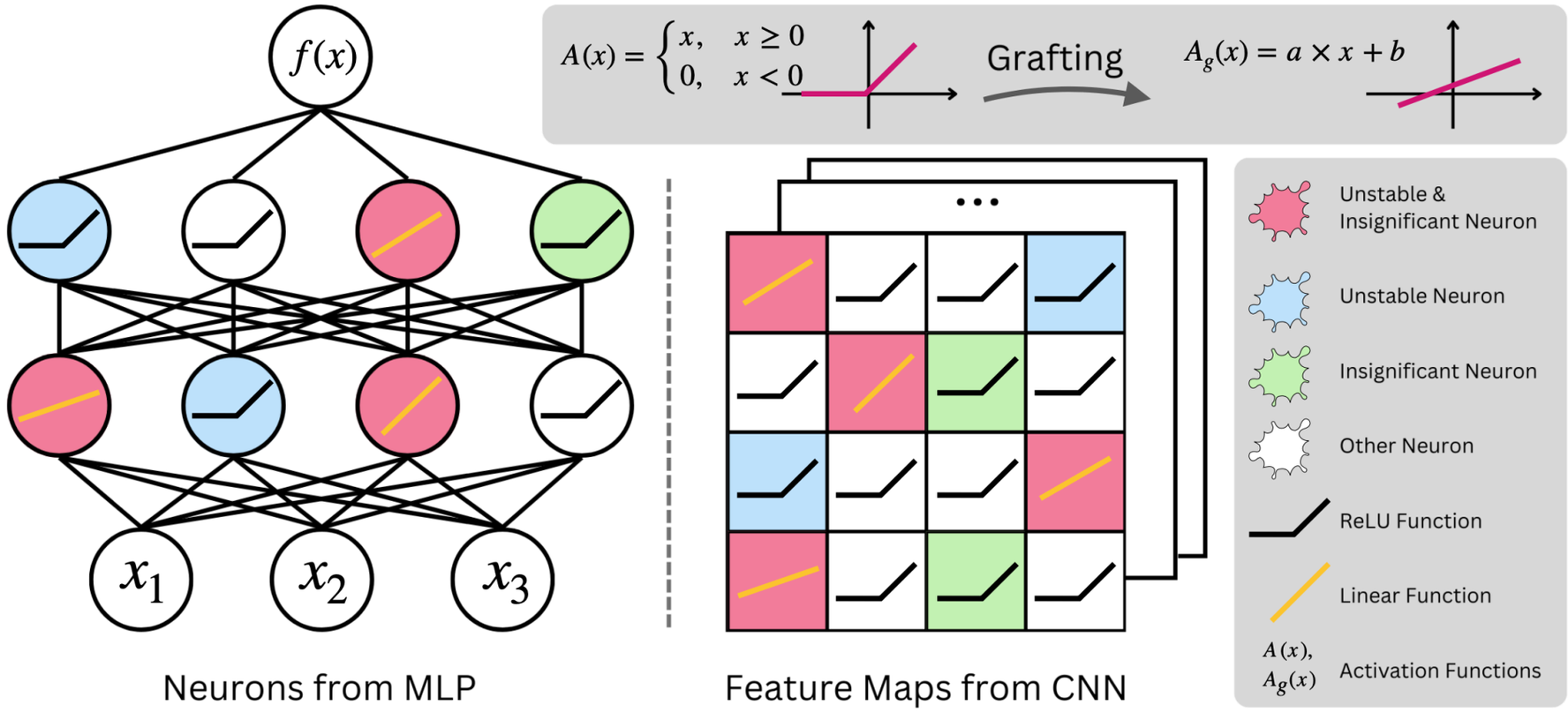
**Optimization problem** of finding minimal classification value gap to ground truth class (verified if positive)

$$g^* = \min_{x' \in \mathcal{P}} \left( f_{c_{true}}(x') - \max_{\substack{c \neq c_{true} \\ c \in C}} f_c(x') \right)$$

- Highly nonlinear
- NP complete
- Obviously no brute force
- Bound propagation
- Branch and Bound → solve subproblems
- Limited by time and memory constraints
- Recent improvements in practice due to
  - domain-informed adjustments
  - parallelization



# Linearity Grafting: Idea




[1]

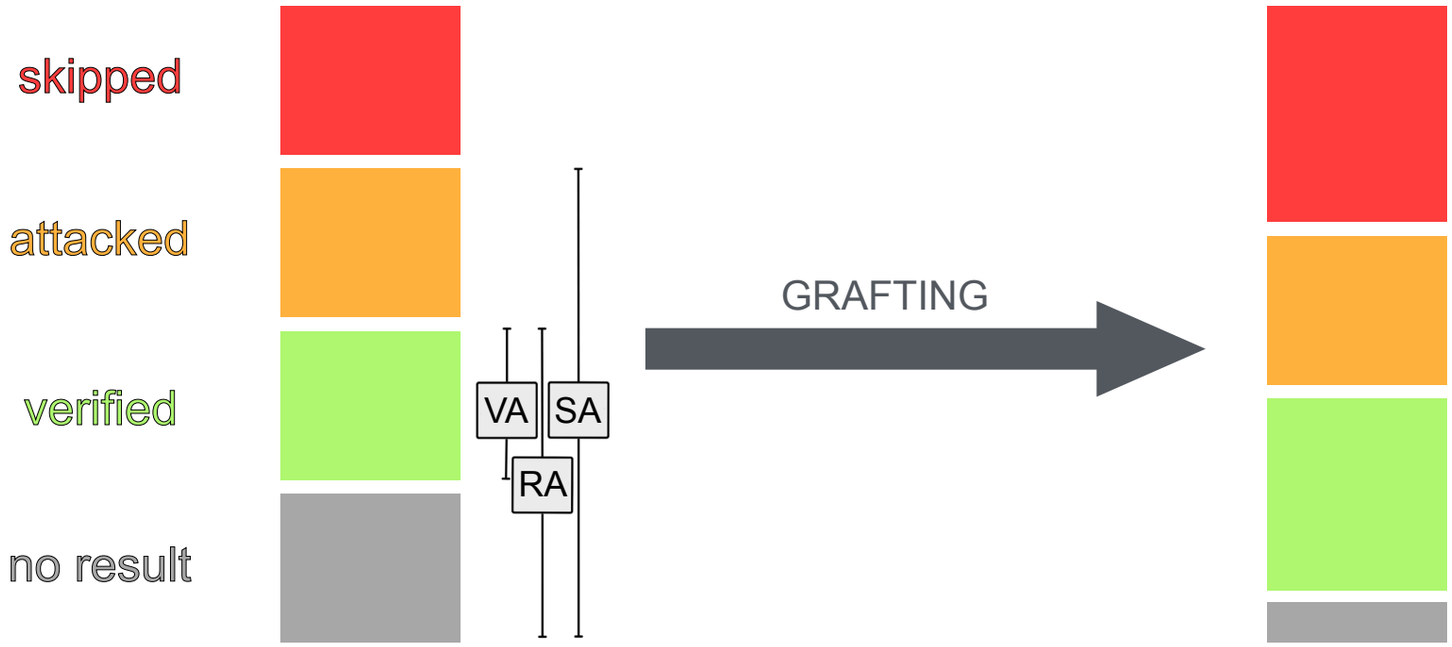
**Multistep Process:**

1. **Train network** using adversarial training for improved empirical robustness
2. Identify insignificant and unstable neurons and **linearize their activation functions**
3. **Tune network** by optimizing slope and intercept values and by adjusting weights
4. Run **robustness verification** with a complete verifier

[1] Adapted from “Linearity Grafting: Relaxed Neuron Pruning Helps Certifiable Robustness“ by Chen et al., 2015

# Linearity Grafting: Original Results

**Generally: increased VA, reduced UNR, but also reduced SA and RA**




**Largely optimistic conclusions:**

- Improved verification results (better SA - VA tradeoff)
- Improved certification scalability
- Competitive certified accuracy for large scale network
- Reduced time requirements
- Superiority to other pruning techniques

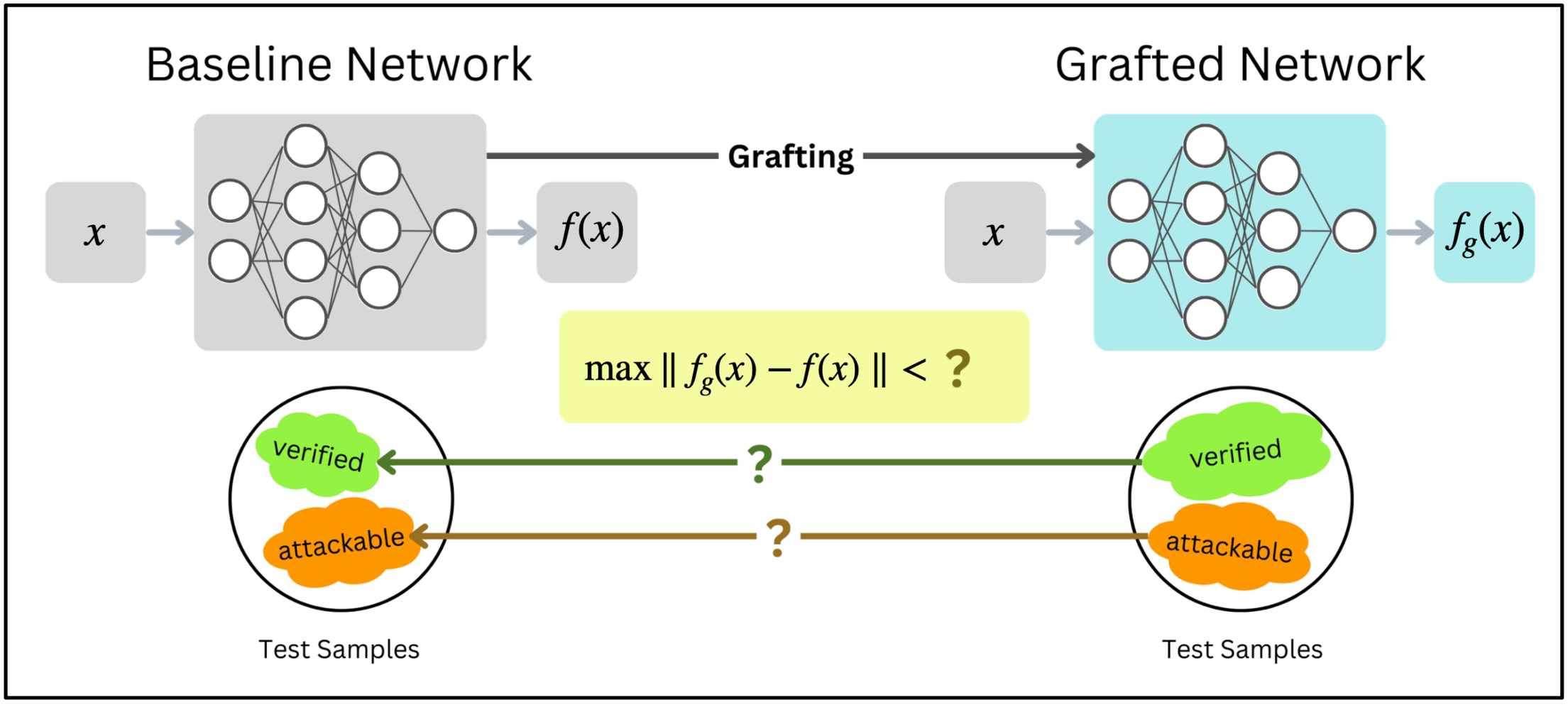
# My Part: Main matters of interest

1) **Reproducibility** of results from original publication

2) **Influence of neuron selection heuristics** and other grafting parameters on performance

3) Properties of the **relationship between baseline and grafted network**:
   - Effectiveness of grafting without weight retraining
   - Changes in verification status for individual test inputs
   - Transferability of adversarial examples

# My Part: Main matters of interest




**Relationship between baseline and grafted network:** Can their difference be bounded? Can verification or attack results be transferred back?

# Own Experiments

| **Architecture** | **CNN-B** | **ConvBig** |
|---|---|---|
| | 4-Layer CNN | 7-Layer CNN |
| **Datasets** | CIFAR10 | MNIST, CIFAR10 |
| **# Neurons** | 16.6 K | 62.5 K |
| **# Params** | 2.1 M | 2.5 M |

**Verifier:** $\alpha$-$\beta$-CROWN [1]

**GPU:** NVIDIA A40, 48 GB

**Implementation:** Code from [2] with own additions, mainly for data extraction and visualization

[1] https://github.com/Verified-Intelligence/alpha-beta-CROWN
[2] “Linearity Grafting: Relaxed Neuron Pruning Helps Certifiable Robustness“ by Chen et al., 2015

# Challenges

- **Ambiguities** in textual descriptions (ranking, grafting ratio (global or local), decaying $\gamma$ values)
- **Missing implementations** (instability indicator calculation, UNR calculation, mask creation, result extraction, ranking and score calculation)
- **Few models** and masks provided
- **Sparse documentation**

➡ Failed to pass only possible sanity check for mask creation (max 76% overlap)

➡ Limited comparability of method and results

# Results: Neuron Heuristics and Grafting Parameters

## Neuron Selection

calculate **instability** indicators

↓

rank neurons

↓

normalize rank to obtain $r_u$

calculate **significance** indicators

↓

rank neurons

↓

normalize rank to obtain $r_s$

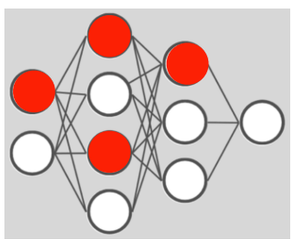
↘ ↙

combine and select neurons by finding

$$\arg\max_i \gamma \times r_u^{(i)} - r_s^{(i)}$$

↓

create mask



## Variables

significance heuristic, algorithm for calculating instability indicator

ranking algorithm

$\gamma$ value(s): static/dynamic, grafting mode: global/local

# Results: Neuron Heuristics and Grafting Parameters

**Main Findings**

- Newly implemented significance heuristic did not have big influence on neuron selection
- Varying ranking algorithm (not specified) led to greater difference in selected neuron sets than varying $\gamma$ values or significance heuristic

**Variables**

significance heuristic, algorithm for calculating instability indicator

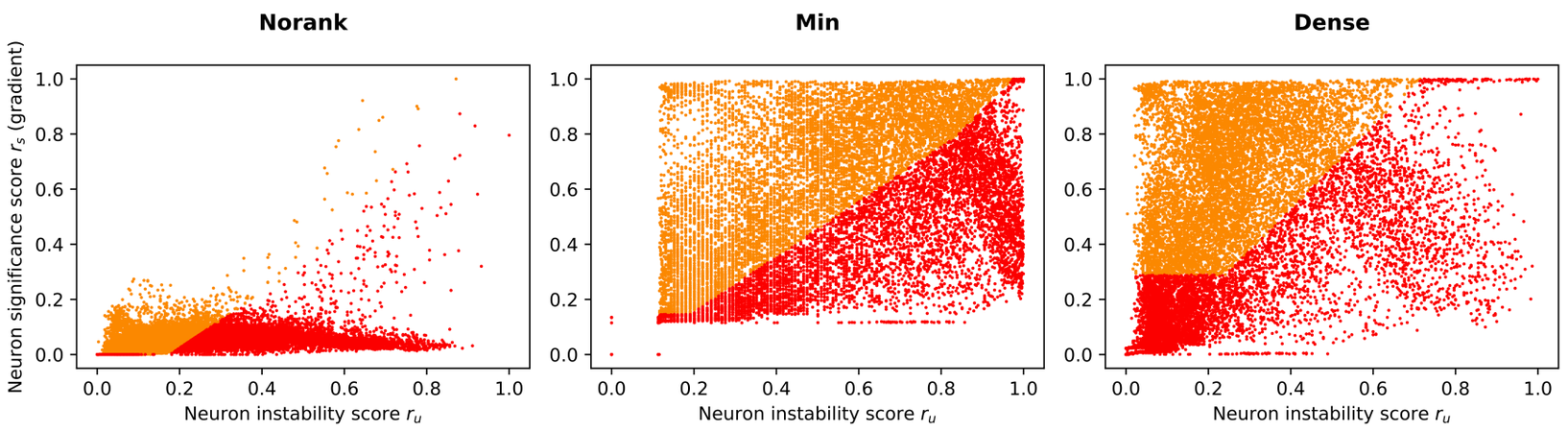
ranking algorithm

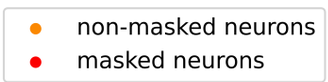
$\gamma$ value(s): static/dynamic, grafting mode: global/local

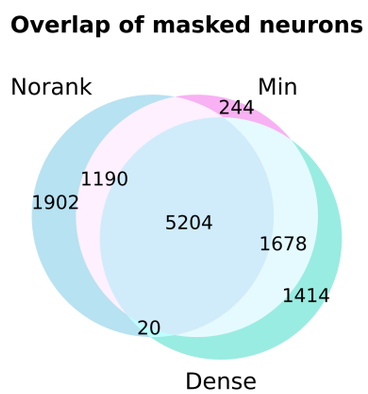
# Results: Neuron Heuristics and Grafting Parameters

## Mine

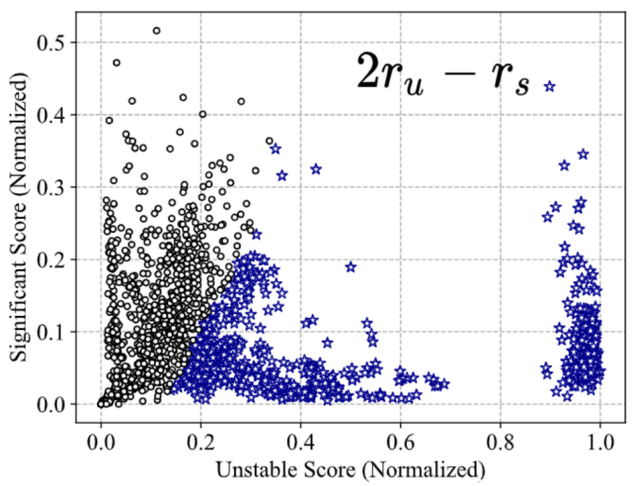
CIFAR10 CNN-B (pretrained) neuron selection depending on ranking (standard $\gamma$ decay)

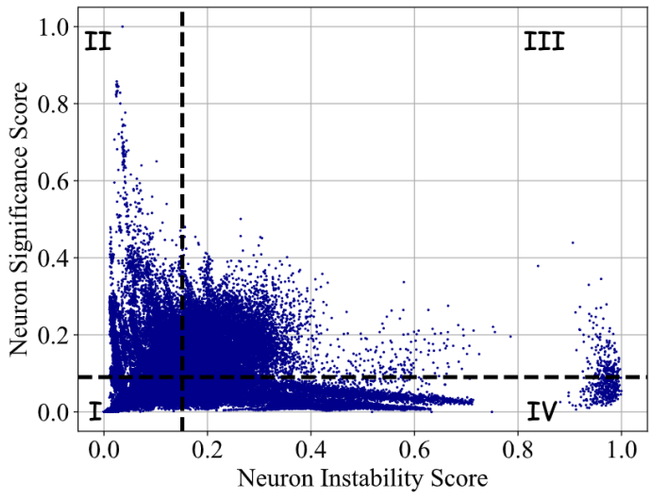



## Original



# Results: Neuron Heuristics and Grafting Parameters

**Main Findings**

- Newly implemented significance heuristic did not have big influence on neuron selection
- Varying ranking algorithm (not specified) led to greater difference in selected neuron sets than varying $\gamma$ values or significance heuristic
- Due to the varying distributions of indicator values, justification of specified optimal $\gamma$ values is questionable

**Variables**

significance heuristic, algorithm for calculating instability indicator

ranking algorithm

$\gamma$ value(s): static/dynamic, grafting mode: global/local

# Results: Reproducibility

**Main Findings**

- Performance of grafted networks and reproducibility heavily dependent on ranking algorithm used in grafting mask creation
- No configuration of grafting parameters consistently comes closest to reported results
- Choice of ranking seems to influence prioritization of instability vs insignificance and therefore the VA - SA tradeoff
- Networks grafted without retraining weights perform consistently and significantly worse in terms of SA, mixed results in terms of VA
- UNR differs considerably to given results
- For **MNIST ConvBig** model, RA value and extreme VA improvements not reproducible (my maximum: +32.38%p, reported: +82.20%p)

# Results: Relationship Original ↔ Grafted Network

**Changes in verification status for individual inputs**

Hypothesized subset properties:

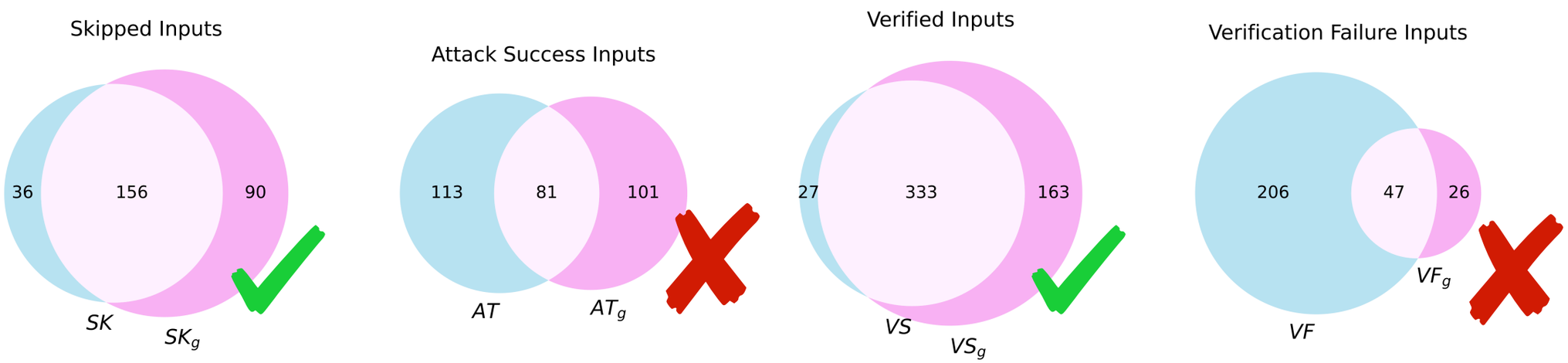
$$VS \overset{?}{\subseteq} VS_g \qquad \text{(verification success)}$$

$$SK \overset{?}{\subseteq} SK_g \qquad \text{(skipped)}$$

$$AT \overset{?}{\subseteq} AT_g \qquad \text{(attack success)}$$

$$VF_g \overset{?}{\subseteq} VF \qquad \text{(verification failure)}$$

Reality: CIFAR 10 CNN-B, Grafting with Weight Retraining



# Results: Relationship Original ↔ Grafted Network

**Changes in verification status for individual inputs**

Hypothesized subset properties:

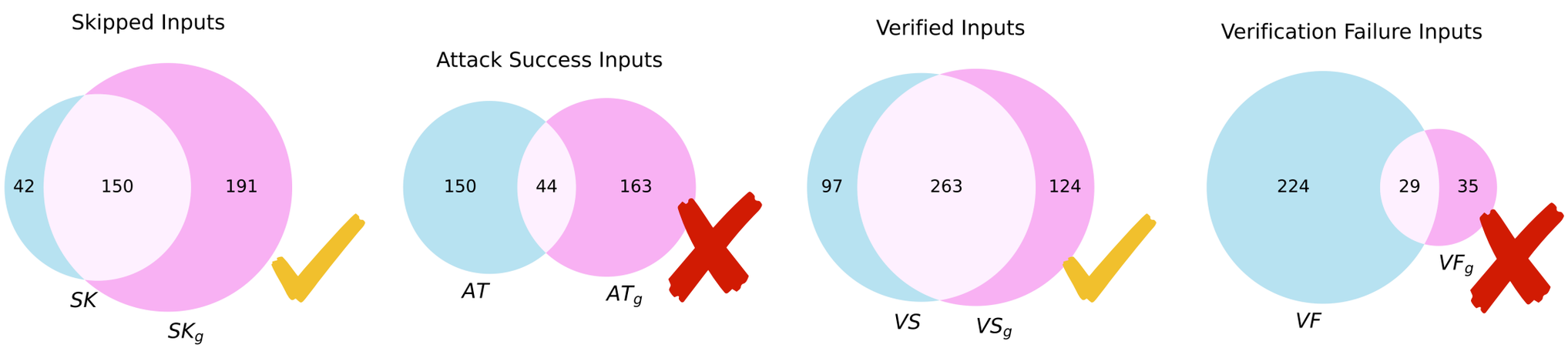
$$VS \overset{?}{\subseteq} VS_g \qquad \text{(verification success)}$$

$$SK \overset{?}{\subseteq} SK_g \qquad \text{(skipped)}$$

$$AT \overset{?}{\subseteq} AT_g \qquad \text{(attack success)}$$

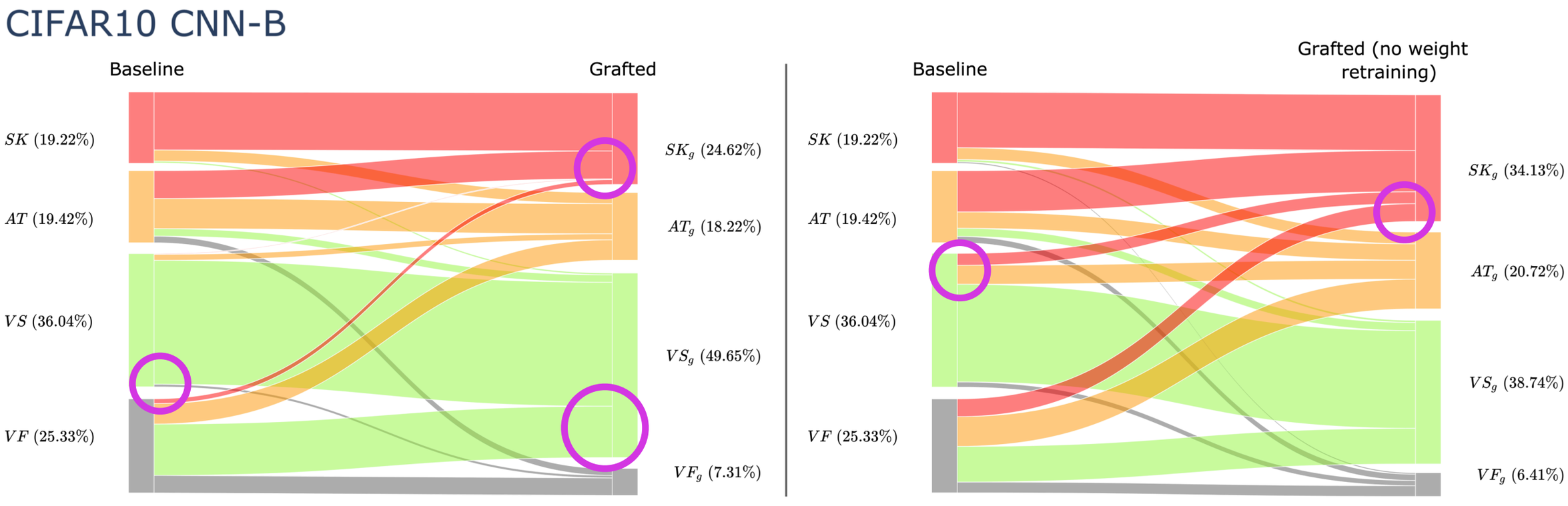
$$VF_g \overset{?}{\subseteq} VF \qquad \text{(verification failure)}$$

Reality: CIFAR 10 CNN-B, Grafting without Weight Retraining



# Results: Relationship Original ↔ Grafted Network

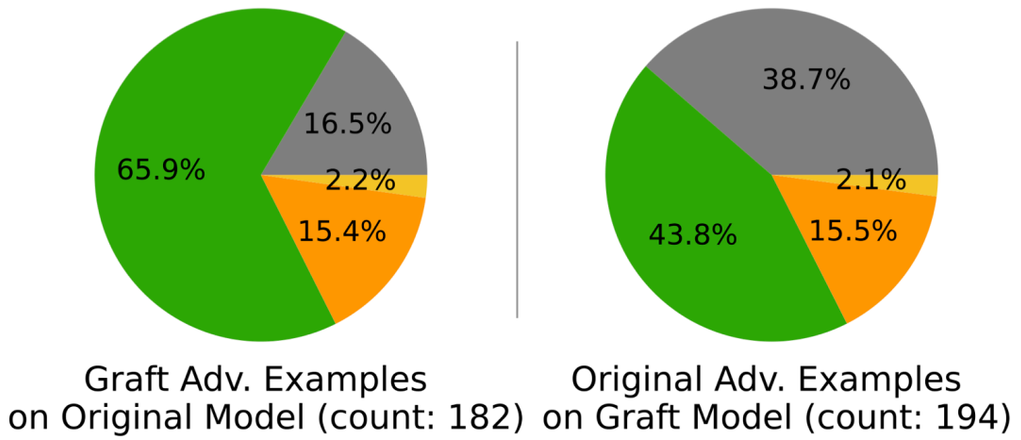
## Changes in verification status for individual inputs



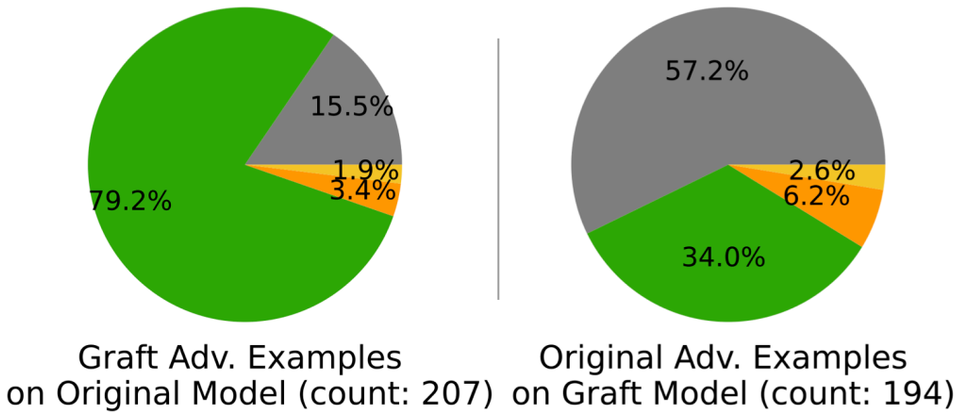
# Results: Relationship Original ↔ Grafted Network

## Transferability of Adversarial Examples

CIFAR10 CNN-B, Grafting with Weight Retraining

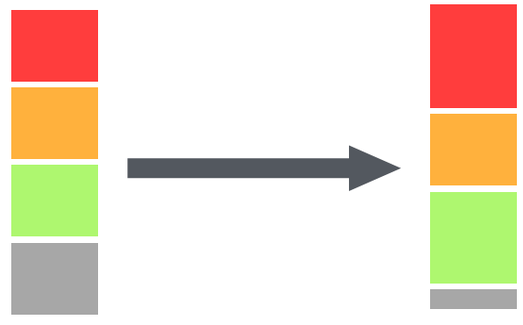



CIFAR10 CNN-B, Grafting without Weight Retraining




# Conclusions

- Results only **partially reproducible**
- Linearity Grafting **reduces uncertainty** at the cost of worsening performance → which one is the better model?
- Linearity Grafting needs **verification of baseline model** (weakens time advantage)
- Other pruning techniques can offer **computational advantages** due to structured reduction
- **Scalability advantage questionable** (28.30% VA causes 90.7% → 62.6% SA drop, there exists a better ungrafted CNN-B model w.r.t. all metrics)



# Conclusions

**Different conclusions from similar results:** Linearity Grafting can close the gap between problem solving capacity of a model and inherent complexity of a problem because it reduces the expressiveness of an overparameterized model (e.g. **ConvBig** on MNIST)

Verifiability mainly dictated by inherent problem complexity?

**Grafting without weight retraining:**

- worse verification results
- no promising results regarding transfer of attacks
- does not exhibit more desirable properties regarding the connection (e.g. subsets)

➡ No motivation for further efforts into establishing a more formal connection between the model versions

**Limits:** many more experiments and hyperparameter choices possible, compare performance and not only index set differences, more architectures / grafting ratios / heuristics

# Outlook

- **Lab setting:** Arithmetic distances ≠ Human perception of distinguishability
- **Foundation** for more complex and realistic scenarios, but **scalability** question remains present
- **Relevance:** at the moment not the most obvious vulnerability of NN technology, but will probably rise
- Other **advantages of adversarial training** apart from robustness: models exhibit closer alignment to human decision procedures [1]

[1] "Robustness may be at odds with accuracy", Tsipras et al., 2018

# Thank you!

## Questions?

---

Materials:

github.com/brandisf/linearity-grafting-materials

# Backup Slides

# Results: Reproducibility

| | Baseline | | Grafted | | | | |
|---|---|---|---|---|---|---|---|
| | own | reference | wrt | | nwrt | | reference |
| | | | norank | min | norank | min | |
| **SA %** | 84.85 | 84.90 | 74.09 | 83.34 | 59.19 | 73.08 | 62.23 |
| **RA %** | 68.93 | 68.10 | 57.07 | 65.54 | 44.84 | 54.02 | 47.73 |
| **VA %** | 1.40 | 1.30 | 37.8 | 5.6* | 35.2 | 5.0* | 39.12 |
| **UNR %** | 17.27 | 17.75 | 20.97 | 19.10* | 16.9 | 17.21* | 4.32 |

Table 4.1.: Results for self-trained CIFAR10 ConvBig model, (n)wrt stands for (no) weights retrained, UNR and VA results marked with an asterisk come from a test set size of 500, reference values taken from [2]

# Results: Reproducibility

| | Baseline | | Grafted | | | | |
|---|---|---|---|---|---|---|---|
| | own | reference | wrt | | nwrt | | reference |
| | | | norank | min | norank | min | |
| **SA %** | 97.95 | 99.29 | 98.19 | 98.32 | 95.14 | 98.01 | 98.68 |
| **RA %** | 44.6** | 97.14 | 64.41 | 97.71 | 53.79 | 97.5 | 92.73 |
| **VA %** | 0.0 | 0.10 | 11.01 | 0.0* | 32.38 | 0.2* | 82.30 |
| **UNR %** | 33.41 | 31.27 | 42.95 | 34.86* | 39.75 | 35.6* | 5.85 |

Table 4.2.: Results for self-trained MNIST ConvBig model, (n)wrt stands for (no) weights retrained, UNR and VA results marked with an asterisk come from a test set size of 500, RA results marked with a double asterisk come from a test set size of 1000, reference values taken from [2]

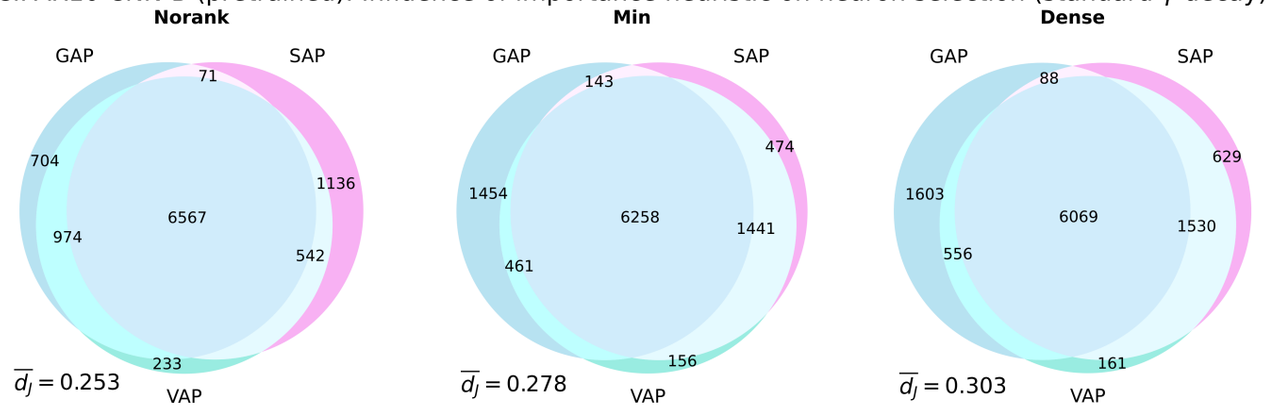
# Results: Reproducibility

| | **Baseline** | | **Grafted** | | | |
|---|---|---|---|---|---|---|
| | self-trained | provided | self-trained | | provided mask | |
| | | | norank | min | wrt | nwrt |
| **SA %** | 78.1 | *79.95* | 66.88 | 73.22 | *74.08* | 64.01 |
| **RA %** | 61.19 | *61.4* (62.23)* | 51.46 | 56.97 | *57.2* (58.76)* | 45.2* |
| **VA %** | 41.5 | *36.04 (37.40)* | 47.0 | 49.2 | *49.65 (50.40)* | 38.74 |
| **UNR %** | 13.28 | *15.85 (15.85)* | 14.23 | 15.45 | *13.44 (5.36)* | 14.17 |

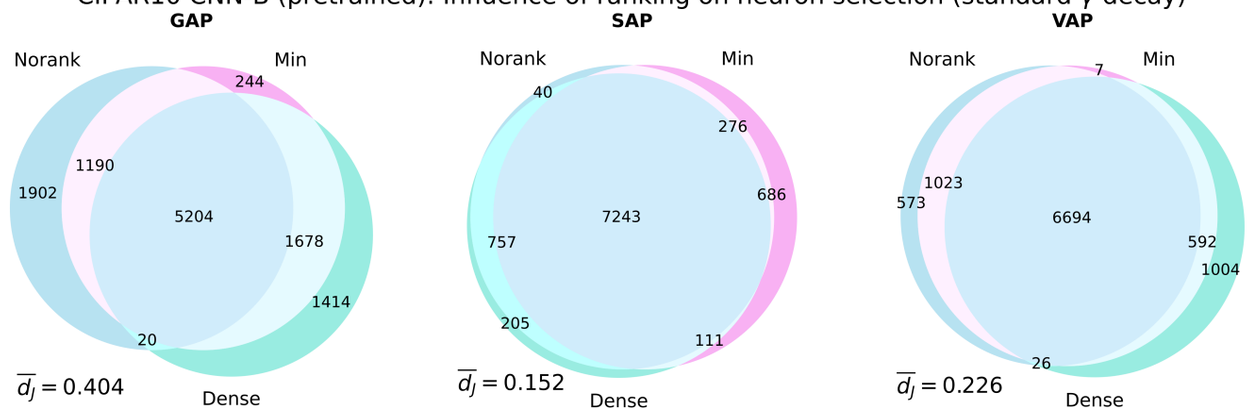
Table 4.3.: Results for CIFAR10 CNN-B model, (n)wrt stands for (no) weights retrained, results of pre-trained models in italic with reference values in brackets, RA results marked with an asterisk come from a test set size of 1000

# Results: Neuron Heuristics and Grafting Parameters

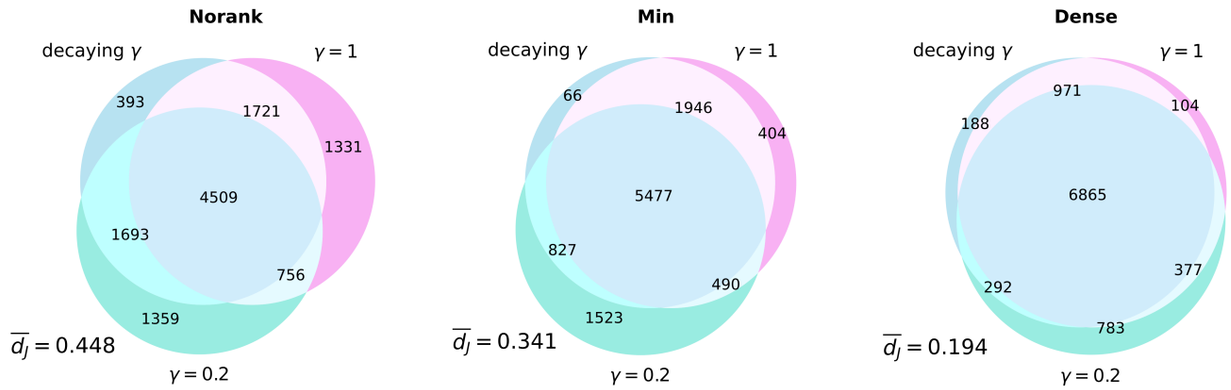
CIFAR10 CNN-B (pretrained): Influence of importance heuristic on neuron selection (standard $\gamma$ decay)




CIFAR10 CNN-B (pretrained): Influence of ranking on neuron selection (standard $\gamma$ decay)




CIFAR10 CNN-B (pretrained): Influence of $\gamma$ values on neuron selection (GAP heuristic)



# Results: Neuron Heuristics and Grafting Parameters