TECHNISCHE UNIVERSITÄT BERLIN

When Papers Choose their Reviewers: Adversarial Machine Learning in Peer Review

Konrad Rieck

VISP Distinguished Lecture

Machine Learning and Security

BIFOLD

Background Paper

No more Reviewer #2: Subverting Automatic Paper-Reviewer Assignment using Adversarial Learning

USENIX Security Symposium, August 2023


Thorsten Eisenhofer, **Erwin Quiring**,
Jonas Möller, Doreen Riepel,
Thorsten Holz, and Konrad Rieck

Automatic Paper-Reviewer Assignment

Papers and Reviews

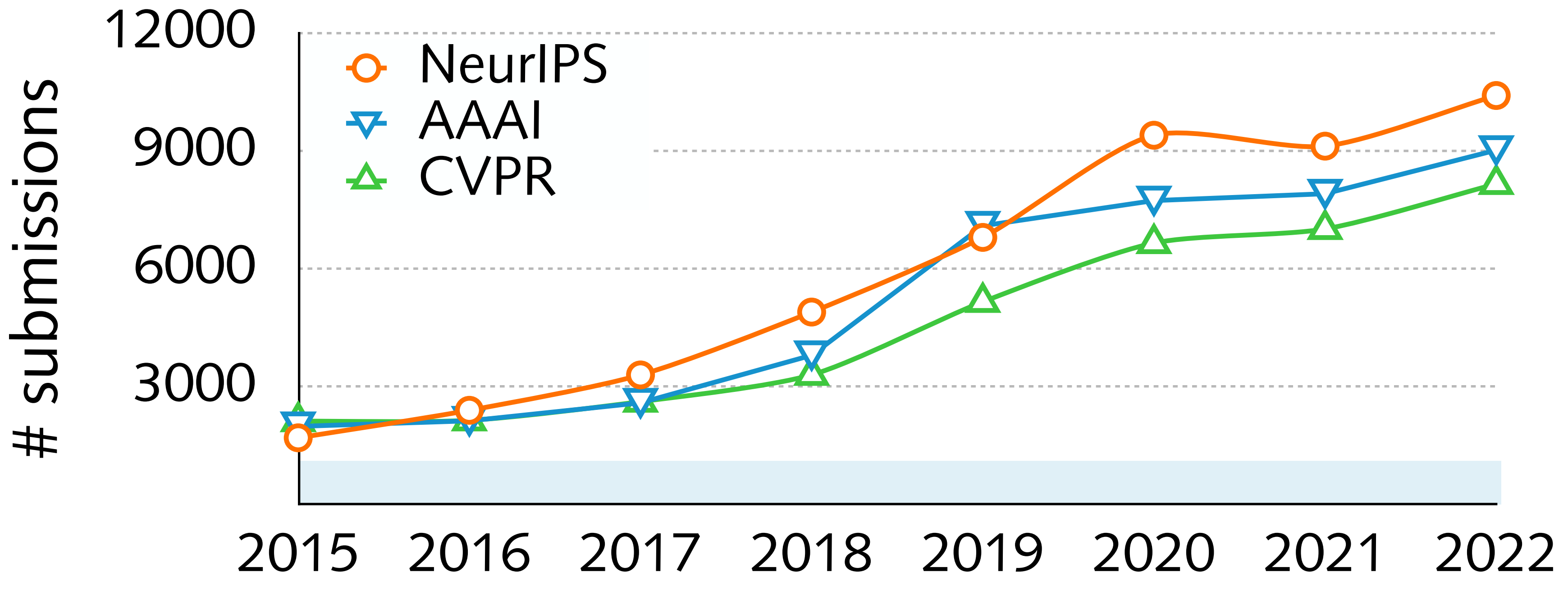
- **Peer review**
  - Independent evaluation of scientic papers by reviewers
  - Instrument for quality control and selection of publications
  - Process with many weaknesses — little alternatives yet
- Initial Step: **Paper-Reviewer Assignment**
  - Assignment of qualified reviewers to each paper
  - Good match of topic (paper) and expertise (reviewer)

Assignment Process

- **Traditional assignment process**
  - Classic assignment by journal editor or program committee chair
  - “Bidding” of reviewers on papers and semi-automatic assignment

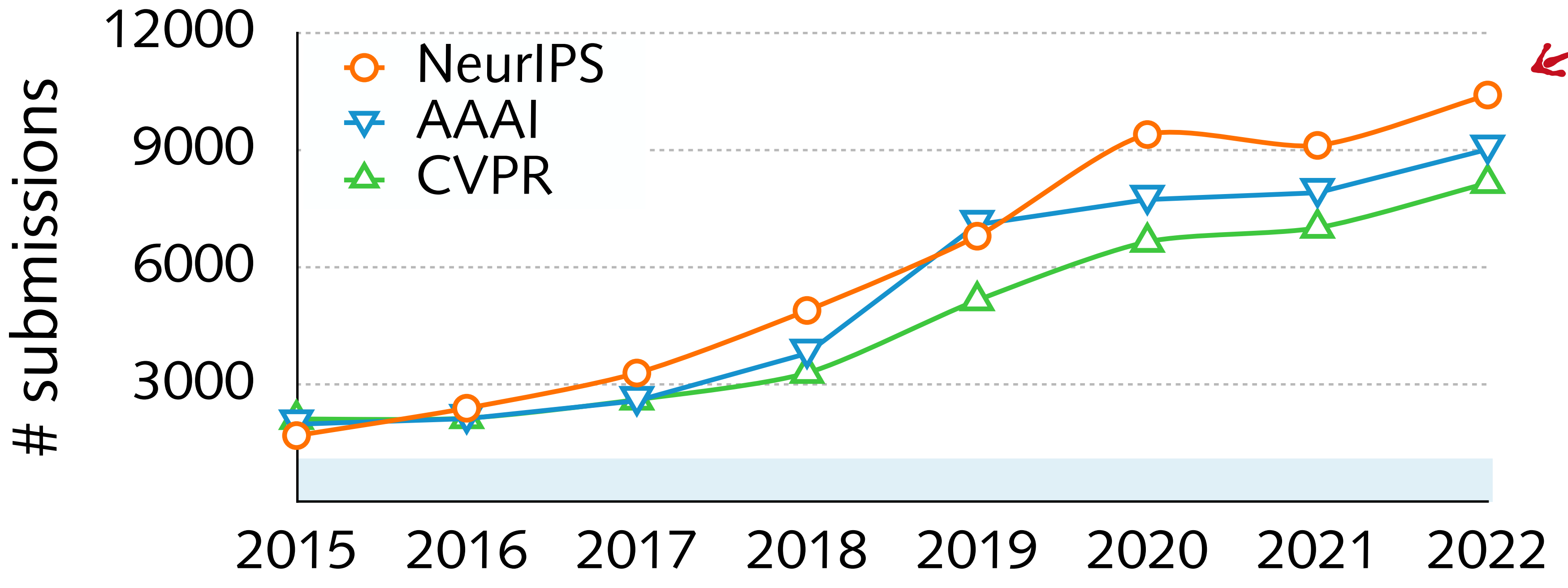
Assignment Process

- **Traditional assignment process**
  - Classic assignment by journal editor or program committee chair
  - “Bidding” of reviewers on papers and semi-automatic assignment
- **Manual bidding increasingly impossible for hot topics** 🔥



Assignment Process

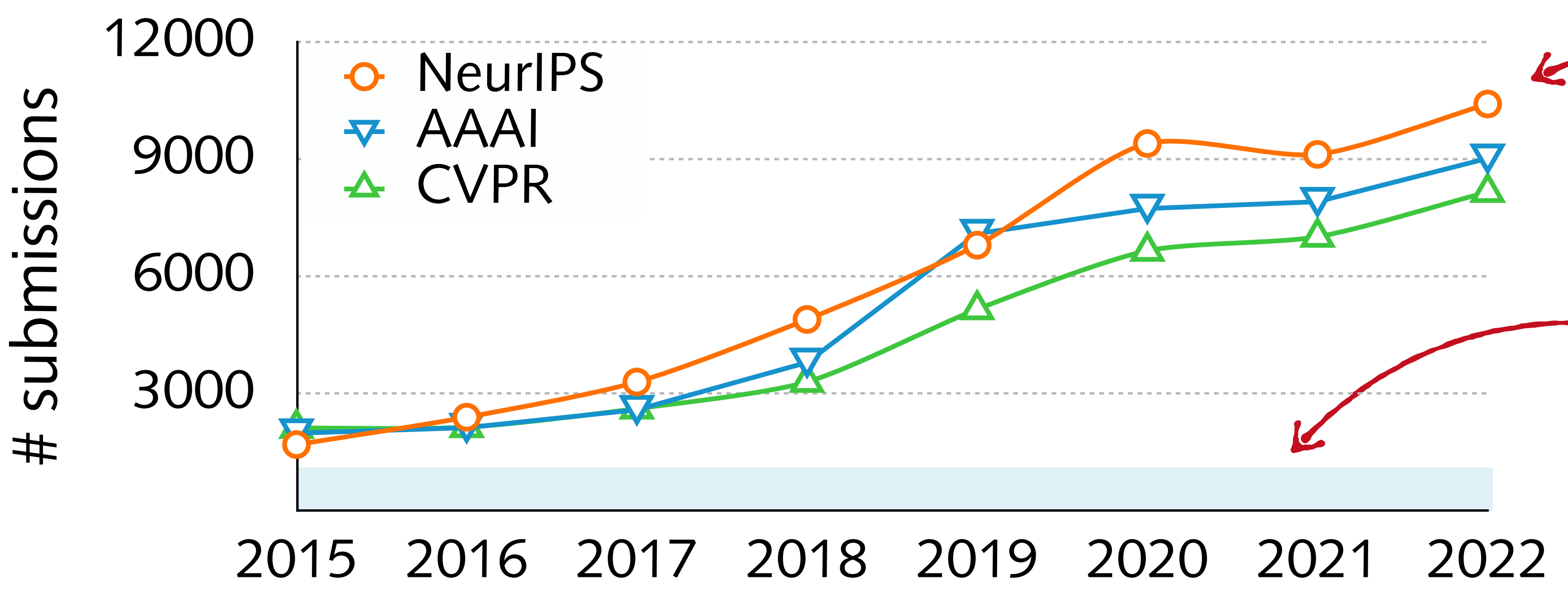
- **Traditional assignment process**
  - Classic assignment by journal editor or program committee chair
  - "Bidding" of reviewers on papers and semi-automatic assignment
- **Manual bidding increasingly impossible for hot topics** 🔥



10.000 submissions. Reading each paper's title (~3s) takes 8 hours!

Assignment Process

- **Traditional assignment process**
  - Classic assignment by journal editor or program committee chair
  - "Bidding" of reviewers on papers and semi-automatic assignment
- **Manual bidding increasingly impossible for hot topics** 🔥



10.000 submissions. Reading each paper's title (~3s) takes 8 hours!

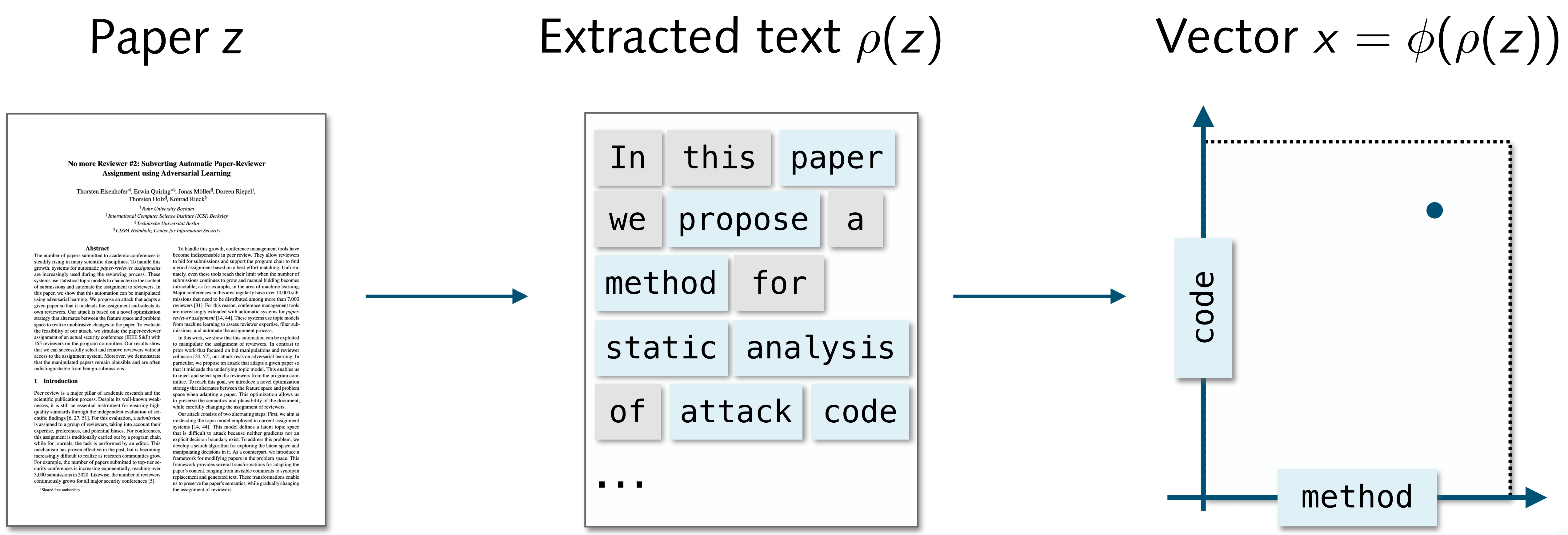
Not so hot research topics, e.g. computer security

Automatic Assignment

- Idea: **Assignment of reviewers to papers using machine learning**
  - First solutions developed already in 2010 for NeurIPS
  - Two systems available: TPMS and AutoBid (open-source variant of TPMS)
  - TPMS de-facto standard employed by several conferences
- Main principle: **Topic modeling**
  - Extraction of topics from corpus of representative publications
  - Matching of papers with reviewers in the topic space

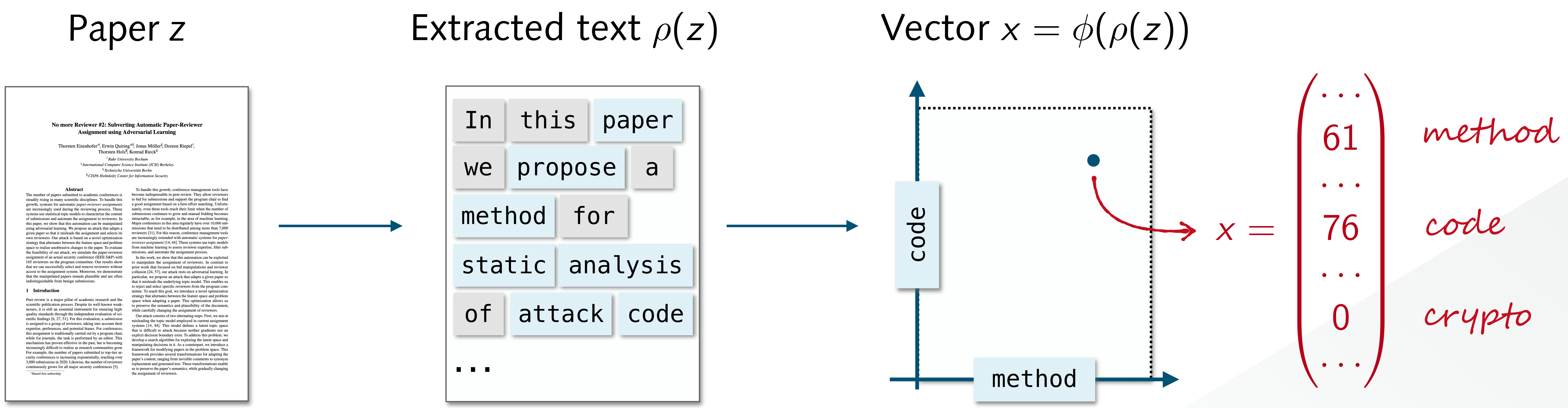
From Papers to Vectors

- **Step 1: Mapping of papers to a feature space**
  - Extraction and preprocessing of text from paper document (e.g. PDF)
  - Paper z represented as bag-of-words vector $x \in \mathbb{N}^{|V|}$ over vocabulary V



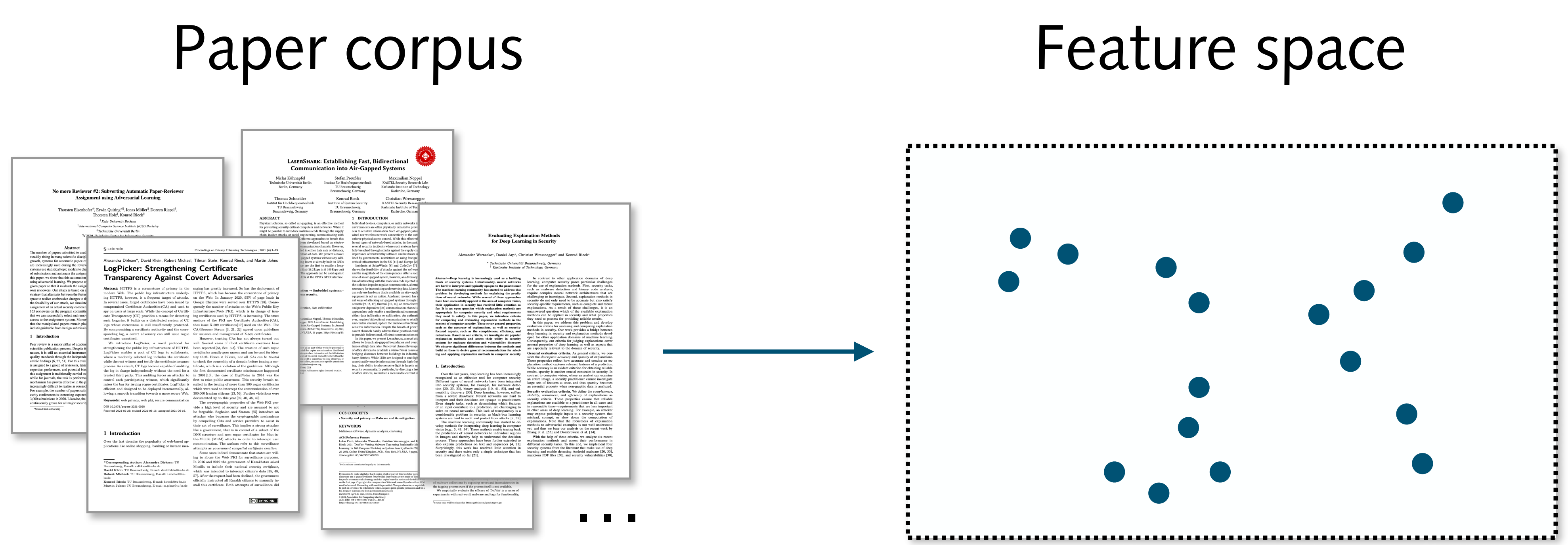
From Papers to Vectors

- **Step 1: Mapping of papers to a feature space**
  - Extraction and preprocessing of text from paper document (e.g. PDF)
  - Paper z represented as bag-of-words vector $x \in \mathbb{N}^{|V|}$ over vocabulary V



From Vectors to Topics

- **Step 2: Automatic discovery of topics from feature vectors**
  - Topic = set of co-occuring words (e.g., “crypto” and “key”)
  - Different algorithms for topic modelling available, e.g. LDA
  - Each feature vector represented as mixture of topics

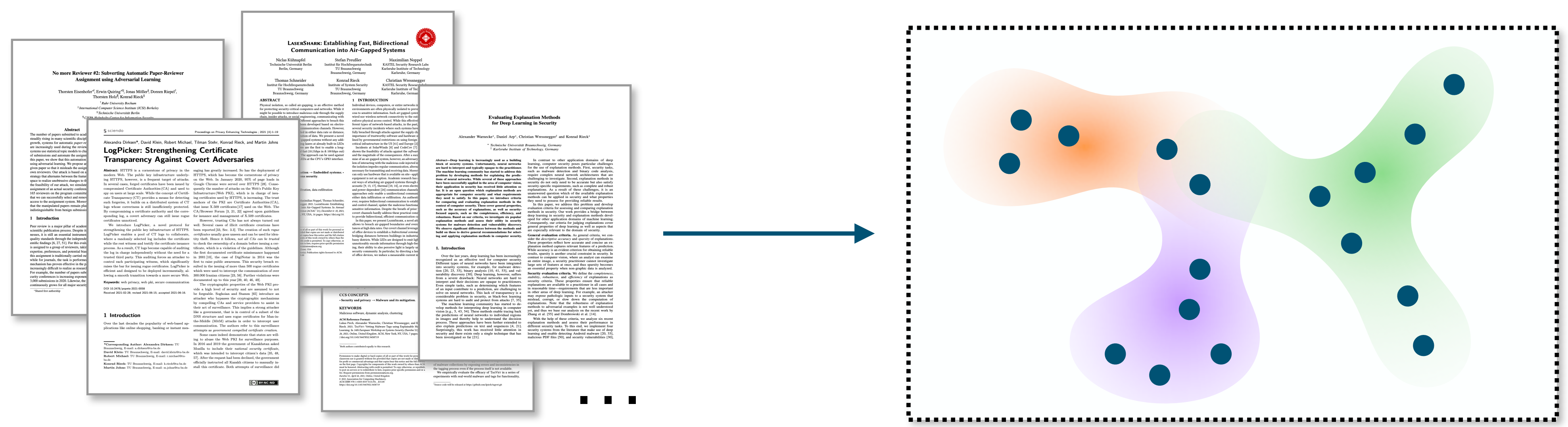


From Vectors to Topics

- **Step 2: Automatic discovery of topics from feature vectors**
  - Topic = set of co-occuring words (e.g., "crypto" and "key")
  - Different algorithms for topic modelling available, e.g. LDA
  - Each feature vector represented as mixture of topics

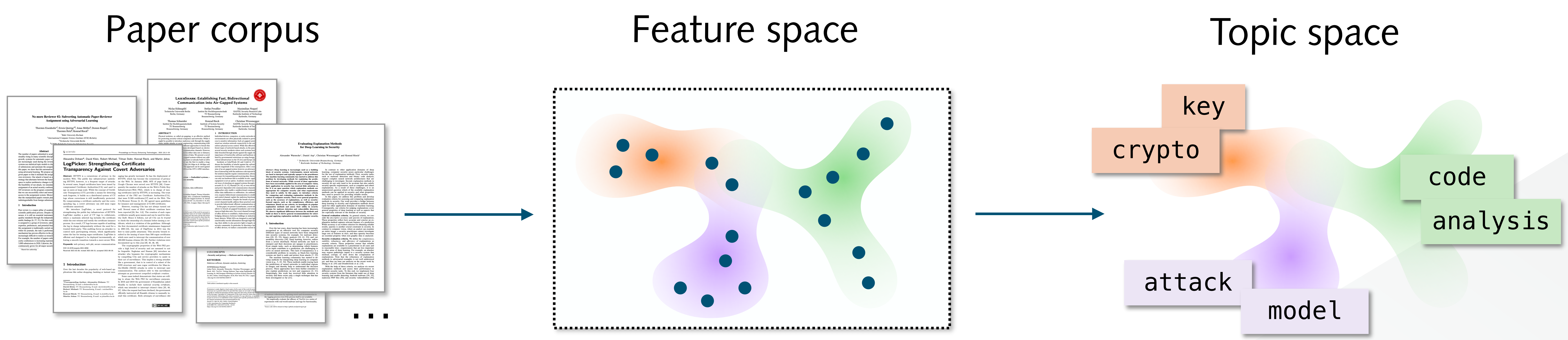
Paper corpus

Feature space



From Vectors to Topics

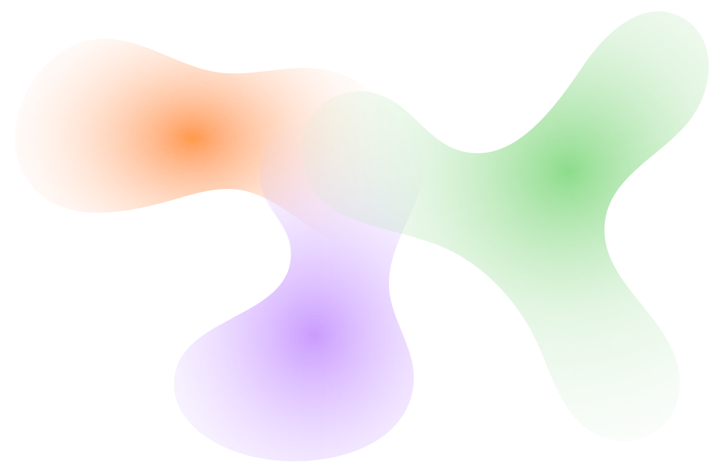
- **Step 2: Automatic discovery of topics from feature vectors**
  - Topic = set of co-occuring words (e.g., "crypto" and "key")
  - Different algorithms for topic modelling available, e.g. LDA
  - Each feature vector represented as mixture of topics



From Topics to Expertise

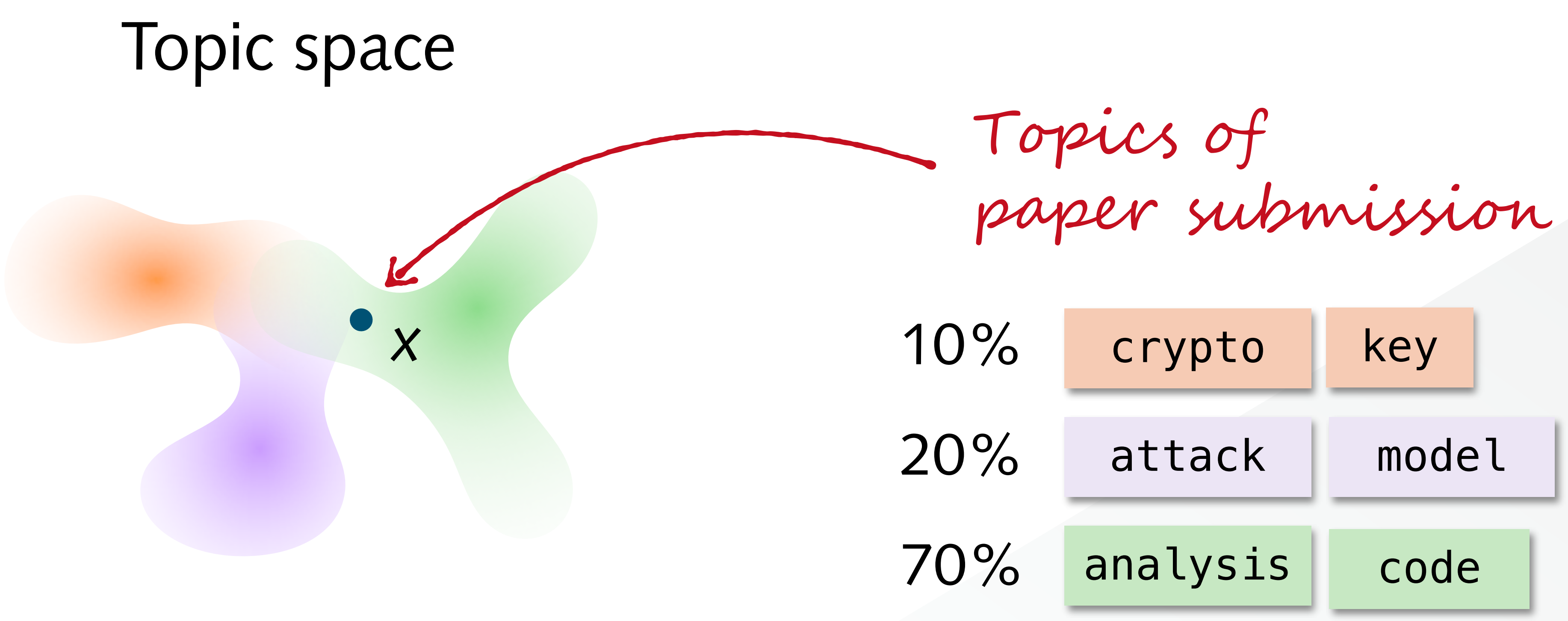
- **Step 3: Matching of reviewers and papers along topics**
  - Paper submission mapped to feature vector *x*
  - Combined publications of each reviewer also mapped to vectors
  - Ranking of reviewers based on similarity in topic space

Topic space



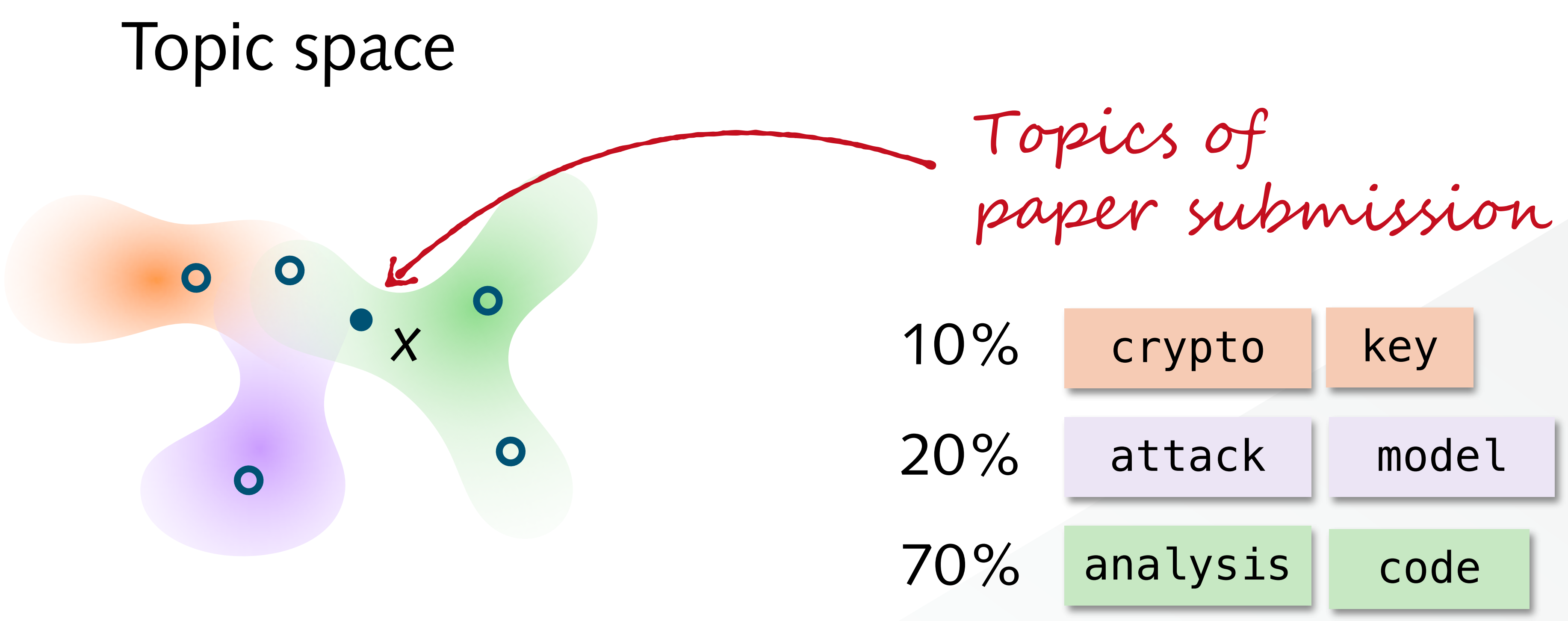
From Topics to Expertise

- **Step 3: Matching of reviewers and papers along topics**
  - Paper submission mapped to feature vector *x*
  - Combined publications of each reviewer also mapped to vectors
  - Ranking of reviewers based on similarity in topic space



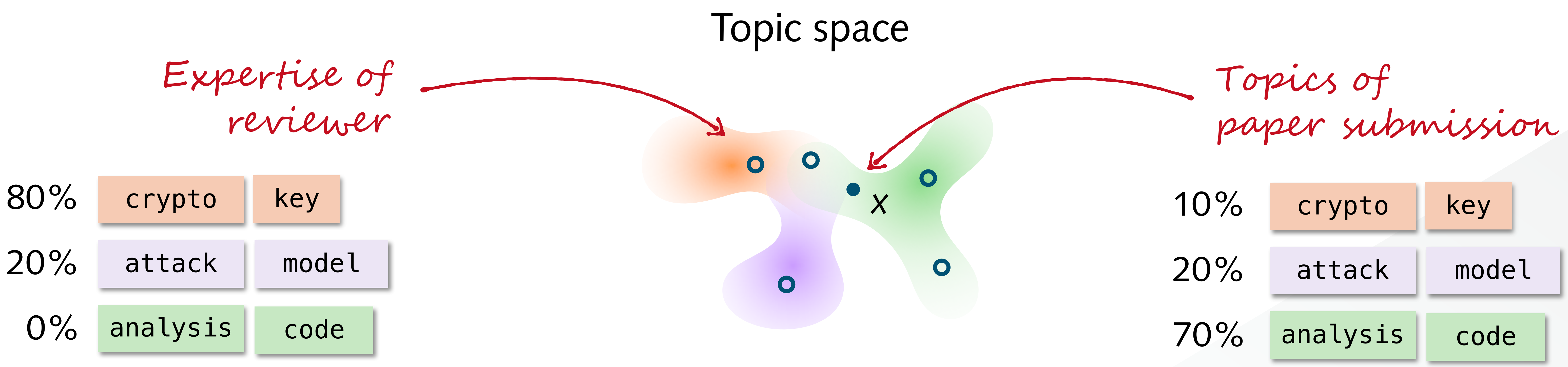
From Topics to Expertise

- **Step 3: Matching of reviewers and papers along topics**
  - Paper submission mapped to feature vector *x*
  - Combined publications of each reviewer also mapped to vectors
  - Ranking of reviewers based on similarity in topic space



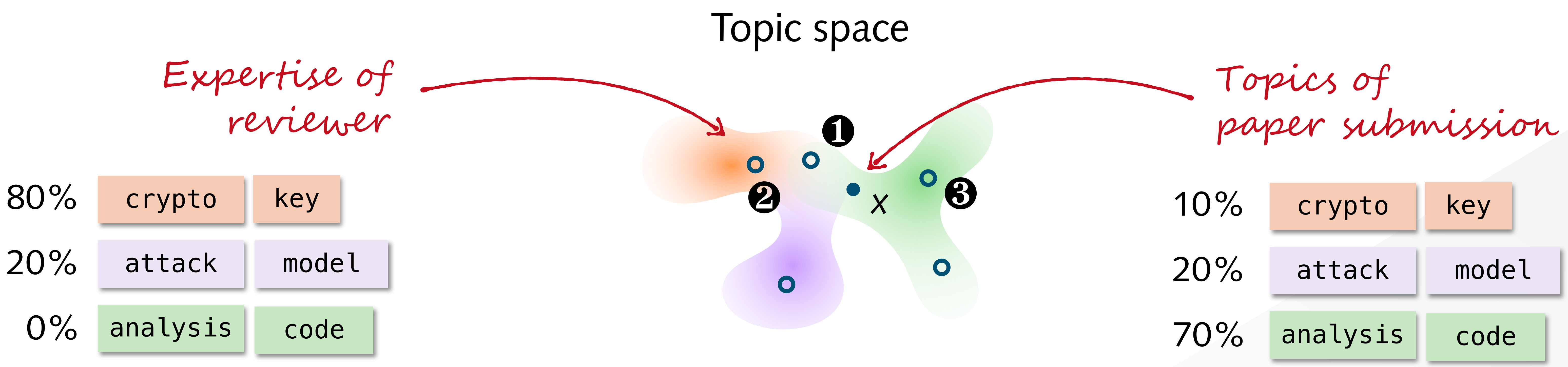
From Topics to Expertise

- **Step 3: Matching of reviewers and papers along topics**
  - Paper submission mapped to feature vector *x*
  - Combined publications of each reviewer also mapped to vectors
  - Ranking of reviewers based on similarity in topic space



From Topics to Expertise

- **Step 3: Matching of reviewers and papers along topics**
  - Paper submission mapped to feature vector *x*
  - Combined publications of each reviewer also mapped to vectors
  - Ranking of reviewers based on similarity in topic space



Real Examples

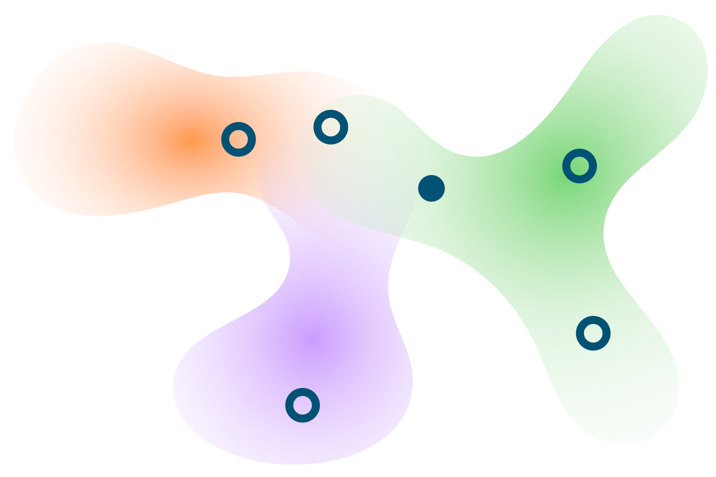
- Reviewer: **Martina Lindorfer**
  - Topic 33% `app` `android` `applic` `permiss` `user` ...
  - Topic 26% `malwar` `detect` `malici` `sampl` `featur` ...
  - Topic 08% `analysi` `input` `fuzz` `execut` `test` ...
- Reviewer: **Matteo Maffei**
  - Topic 26% `random` `signatur` `secur` `key` `scheme` ...
  - Topic 21% `transact` `bitcoin` `contract` `payment` `blockchain` ...
  - Topic 14% `protocol` `model` `secur` `messag` `session` ...

Construction of Adversarial Papers

Attack Overview

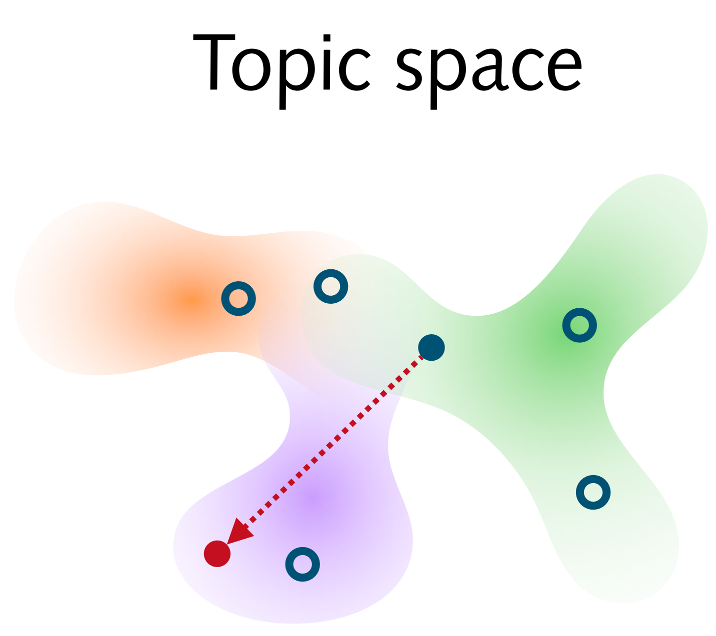
- Idea: **Adversarial Paper**
  - Smart changes to paper misleading reviewer assignment
  - Manipulation of ranking: Removal and addition of reviewers
  - Minimal and unobtrusive changes to paper only

Topic space



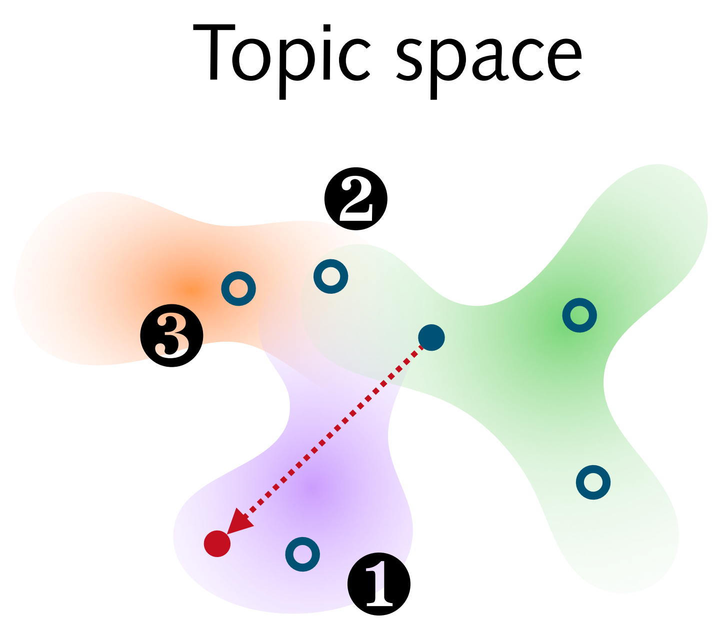
Attack Overview

- Idea: **Adversarial Paper**
  - Smart changes to paper misleading reviewer assignment
  - Manipulation of ranking: Removal and addition of reviewers
  - Minimal and unobtrusive changes to paper only



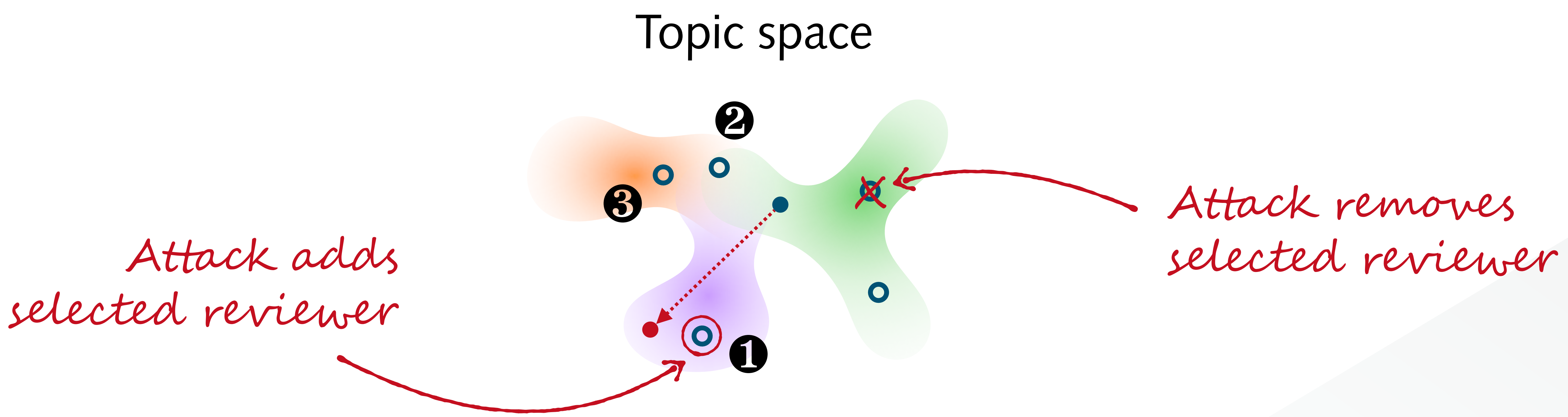
Attack Overview

- Idea: **Adversarial Paper**
  - Smart changes to paper misleading reviewer assignment
  - Manipulation of ranking: Removal and addition of reviewers
  - Minimal and unobtrusive changes to paper only



Attack Overview

- Idea: **Adversarial Paper**
  - Smart changes to paper misleading reviewer assignment
  - Manipulation of ranking: Removal and addition of reviewers
  - Minimal and unobtrusive changes to paper only



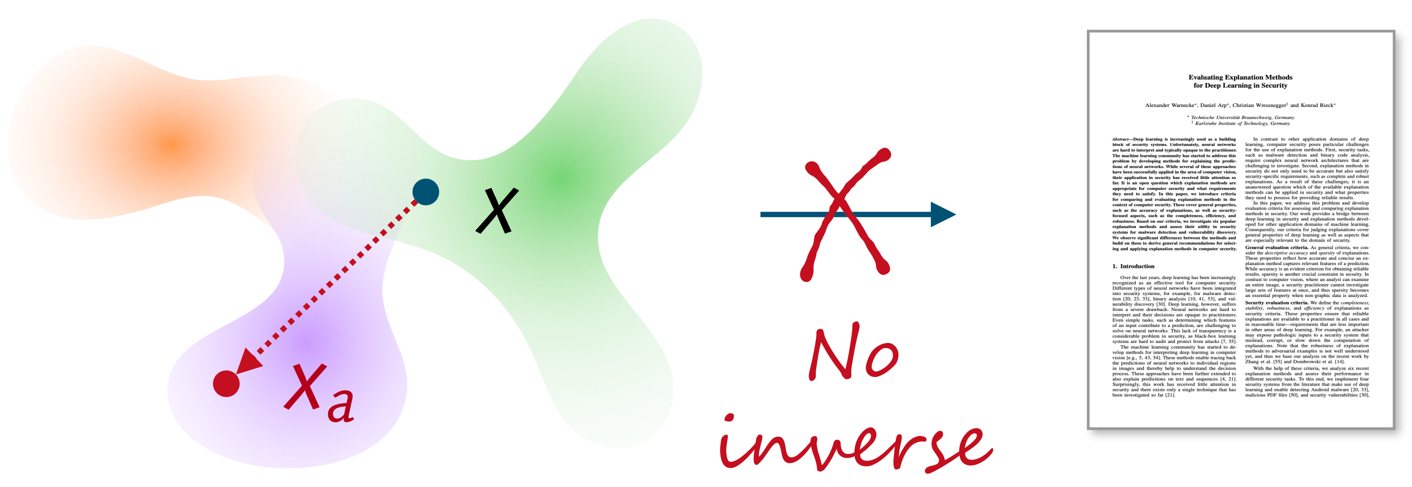
How hard could it be?

- **Despite hype on adversarial learning: No suitable work for us 😢**
- **Two tricky challenges**
  - No inverse map from topic space back to problem space
  - Unobtrusive changes lead to side effects in the feature space

How hard could it be?

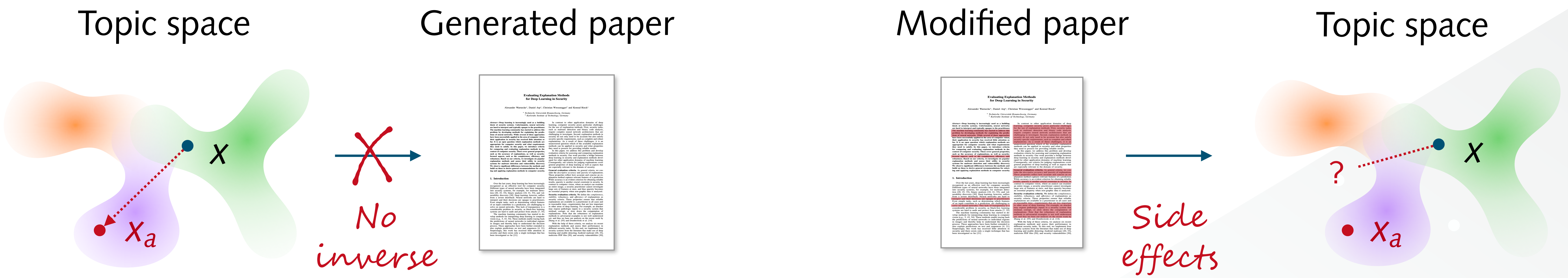
- **Despite hype on adversarial learning: No suitable work for us 😢**
- **Two tricky challenges**
  - No inverse map from topic space back to problem space
  - Unobtrusive changes lead to side effects in the feature space

Topic space Generated paper



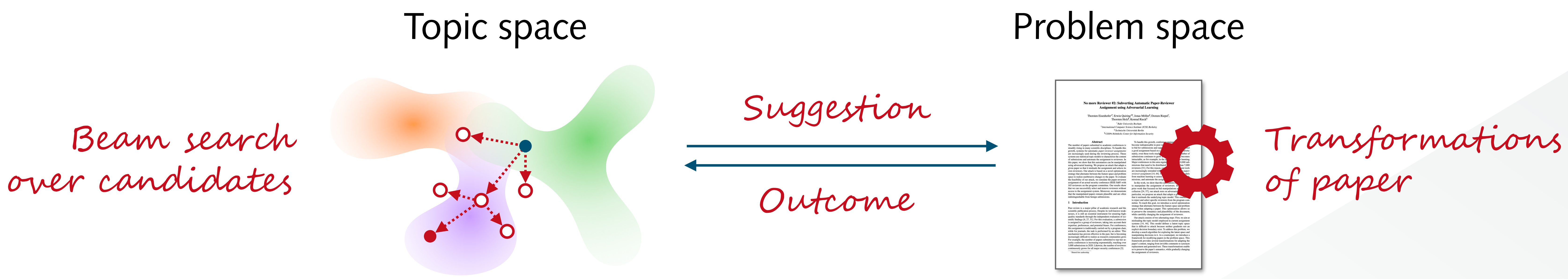
How hard could it be?

- **Despite hype on adversarial learning: No suitable work for us 😢**
- **Two tricky challenges**
  - No inverse map from topic space back to problem space
  - Unobtrusive changes lead to side effects in the feature space



Our Attack Strategy

- **Alternating beweeting topic space and problem space**
  - Beam search in topic space suggests small steps
  - Realization of steps using transformations in problem space
  - Iterative process moving towards selected positions

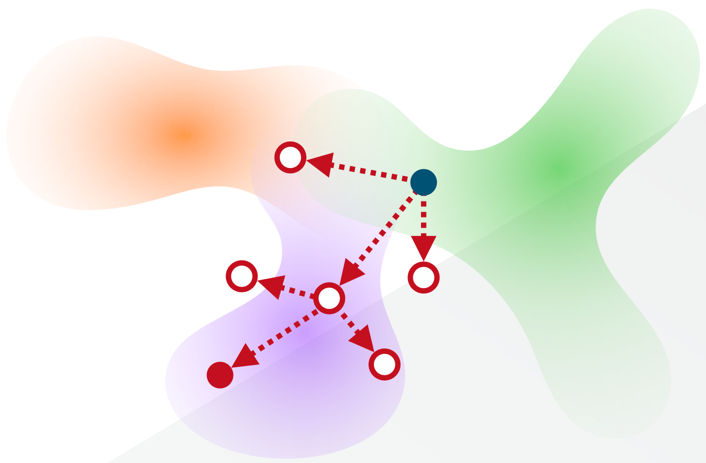


Navigation: Beam Search

- **Each reviewer represented by word probabilities of topics**
  - Restriction to words with minimal side effect (unique use)

80%	`crypto`	`key`
20%	`attack`	`model`

- **Search using k directions in parallel drawn from word probabilites**
  - Direction: Increments and decrements of words
  - L_1 Constraint on total modified words in paper
  - L_∞ Constraint on total modification per words



Driving: Transformations

- **Selection from set of available transformations**
  - Support for incrementing and decrementing words
  - Different level of stealthiness and side effects

- **Two groups of transformations**
  - Format and encoding: Dirty tricks on text representation in paper
  - Text transformation: Semantics-preserving changes

Driving: Format and Encoding

- **Large attack surface due to complex PDF format**
  - Support of accessibility features, scripting and several encodings

Driving: Format and Encoding

- **Large attack surface due to complex PDF format**
  - Support of accessibility features, scripting and several encodings

- Example: **Subsitution with accessibility feature**

Driving: Format and Encoding

- **Large attack surface due to complex PDF format**
  - Support of accessibility features, scripting and several encodings

- Example: **Subsitution with accessibility feature**

Paper z ... static program ... → Text $\rho(z)$ crypto analysis

crypto analysis *Alternate text*

- Example: **Deletion of words with encoding**

Paper z ... static program ... → Text $\rho(z)$ st?tic progr?m

Homoglyphs

Driving: Format and Encoding

- **Large attack surface due to complex PDF format**
  - Support of accessibility features, scripting and several encodings

- Example: **Subsitution with accessibility feature**

Paper z `... static program ...` ⟶ Text $\rho(z)$ `crypto analysis`

`crypto` `analysis` *Alternate text*

- Example: **Deletion of words with encoding**

Paper z `... static program ...` ⟶ Text $\rho(z)$ `st?tic progr?m`

Homoglyphs

u+0061 u+0430

a ≠ a

Machine Learning and Security

Driving: Text Transformations

Driving: Text Transformations

- **Neural word embedding trained on 11,000 security papers**
  - Removal of words using synonyms from embedding

Original text	`intrusion` `detection`	→	`attack` `identification`	Changed text
	`x86` `binary`	→	`i386` `software`	

Synonyms and similar words

Driving: Text Transformations

- **Neural word embedding trained on 11,000 security papers**
  - Removal of words using synonyms from embedding

Original text		→			Changed text
	`intrusion` `detection`	→	`attack` `identification`		
	`x86` `binary`	→	`i386` `software`		

Synonyms and similar words

- **Bibliography database of 11,000 security papers**
  - Insertion of words using additional bibliographic references

Requested words: `crypto` `model` →

New reference:

```
[22] M. Aizatulin, A. Gordon, J. Jürjens: Extracting
     and verifying cryptographic models from C protocol
     code by symbolic execution. CCS, 2011
```

Often helpful side effects!

Driving: Text Transformation

- **Large language model for fabricating text with given words**
  - Transformer model OPT-350m finetuned to text from security papers
  - With our resources reasonable text, but no comparison to larger models

`exempl` `broad` `think`

↓

The recent rise in popularity for social networking services (SNS) **exempl**ifies how users are using them today. Users can share content with others by posting it on their own **broad**ened **think**ing.

`lip` `lobe` `inaud` `speaker` `demot`

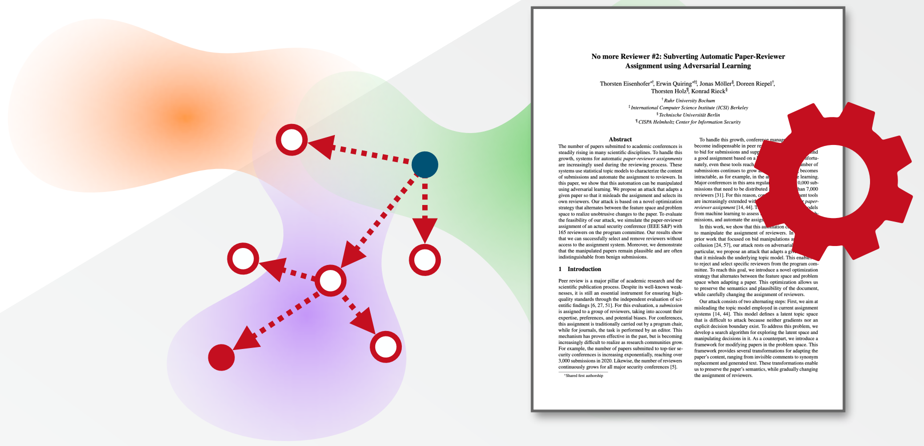
↓

The **lip speaker**s **inaud**ible voice assistants are **demot**ed away from human listeners by adding an additional layer between them (**lobe**s). This approach can potentially mitigate some attacks …

Addition of multiple words in one paragraph

Navigation & Driving: Putting it together

- **Each transformation assigned a stealth level and a budget**
  - Stealth transformations preferred until their budgets exceeded
  - Encoding and format tricks only when no text budget left
  - Example: 10 synonyms, 10 references, 10 generations, …
- **Iterative process alternating between search and transformations**
  - Control using total attack budget and number of switches



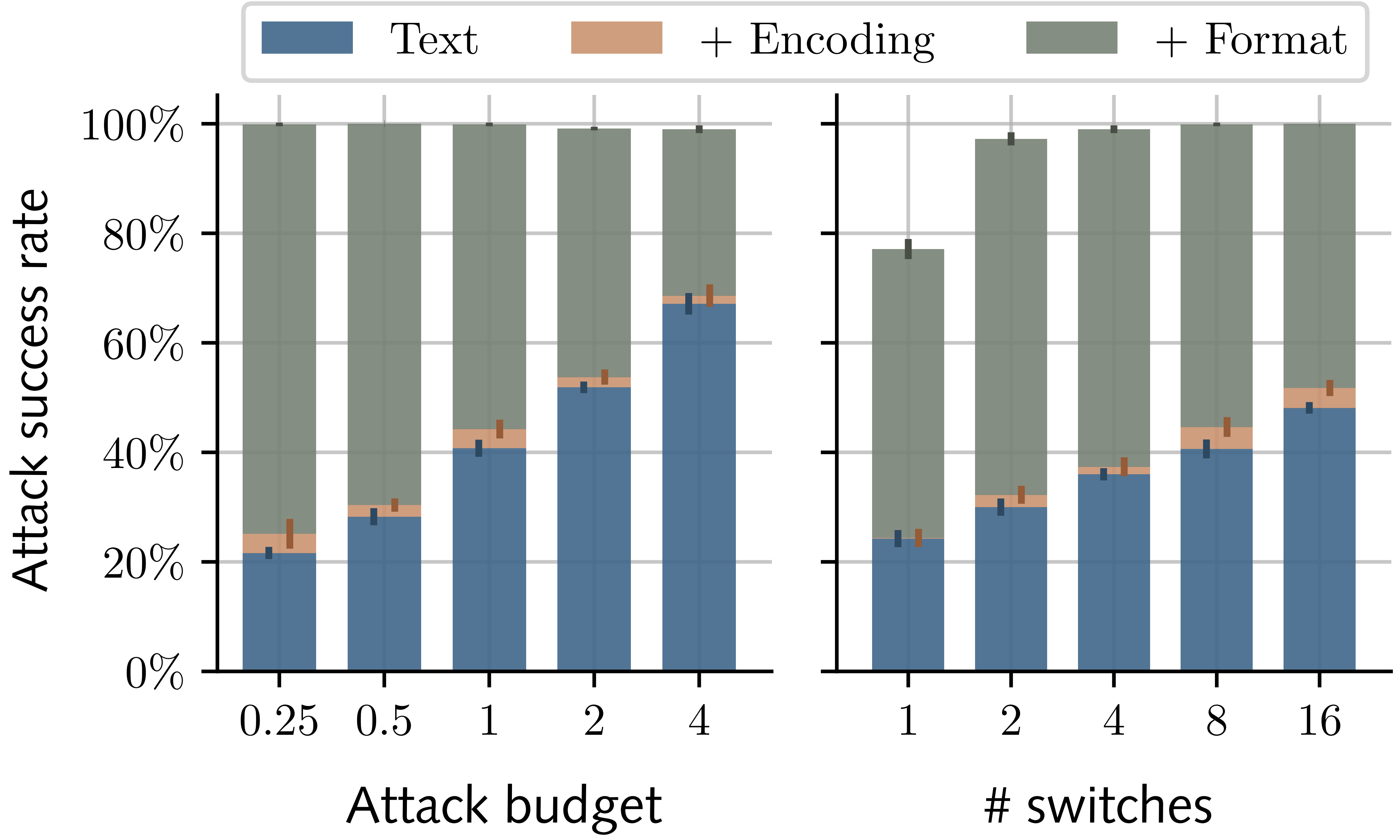
Empirical Evaluation

Simulated Conference

- **Simulation of IEEE Symposium on Security and Privacy 2020**
  - PC of 165 reviewers, each represented by 20 of their papers
  - 32 real paper submissions with source code from arXiv
  - Top-5 ranked reviewers assigned to each submission (no load balancing)
- **Two attack scenarios**
  - White-box attack: Adversary has direct access to topic model
  - Black-box attack: Adversary trains own surrogate models

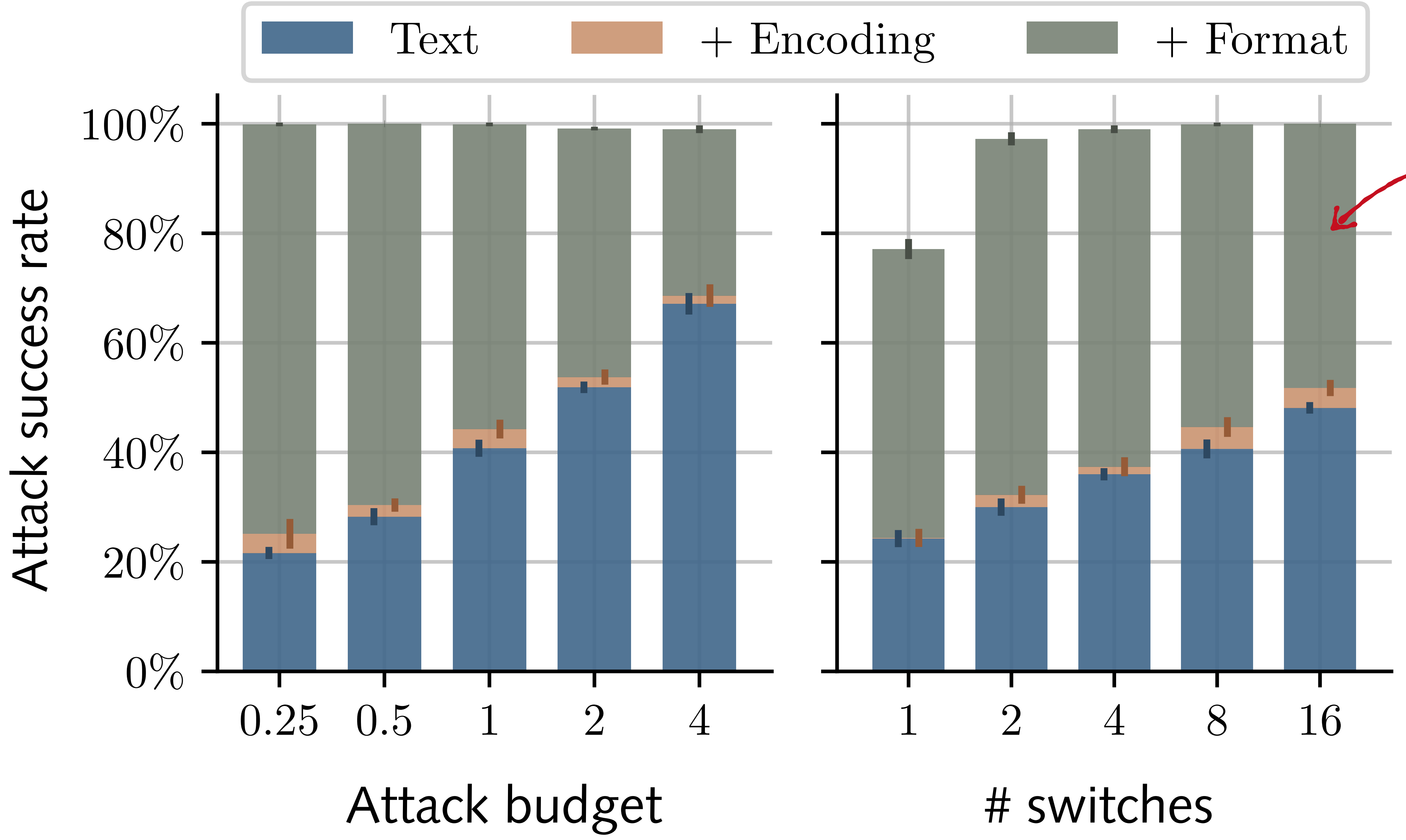
White-Box Scenario

- Experiment: **Selection and rejection of reviewers within Top-10**
  - Evaluation of attack budget and number of switches



White-Box Scenario

- Experiment: **Selection and rejection of reviewers within Top-10**
  - Evaluation of attack budget and number of switches

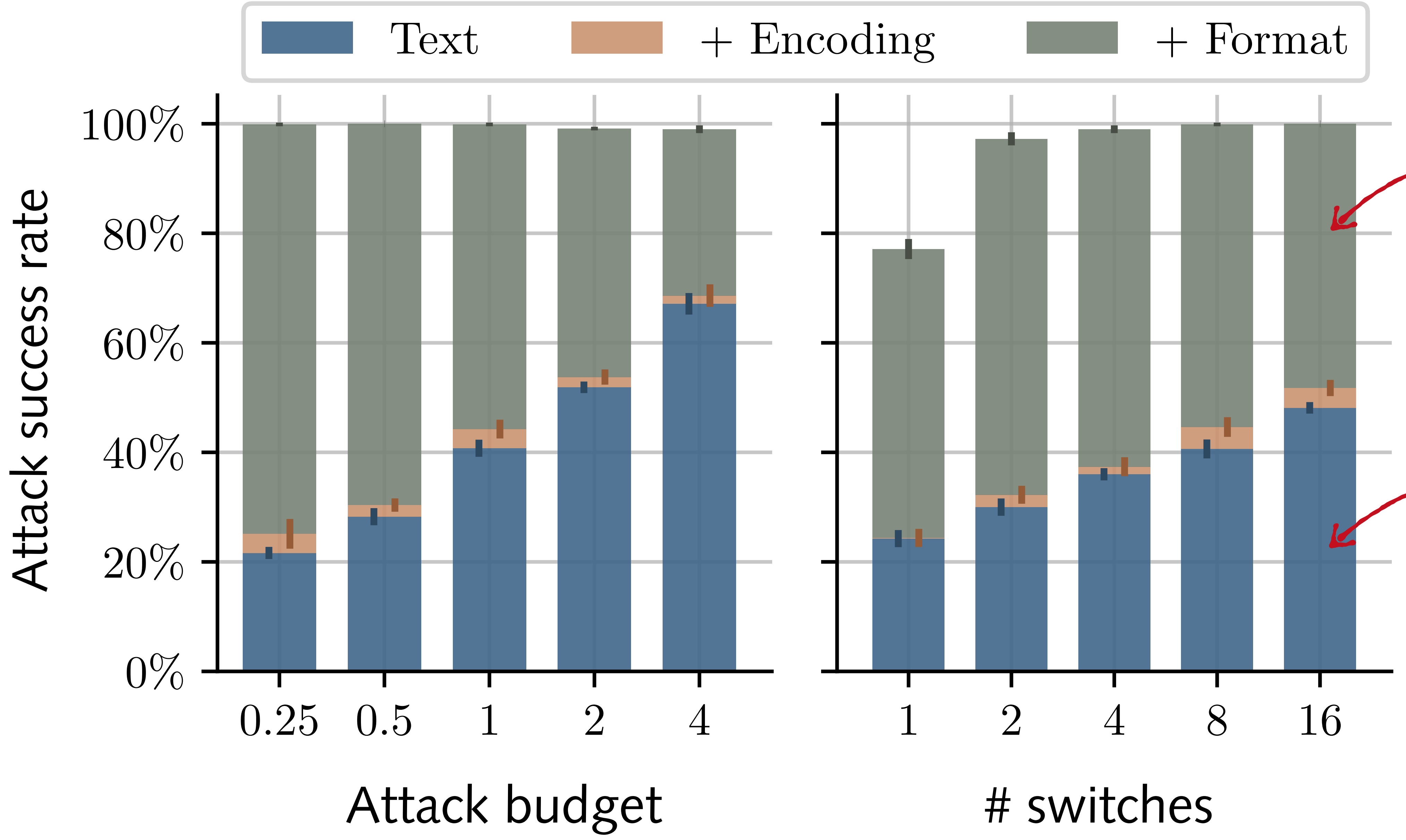


99% success rate if all transformations are used

Machine Learning and Security

White-Box Scenario

- Experiment: **Selection and rejection of reviewers within Top-10**
  - Evaluation of attack budget and number of switches



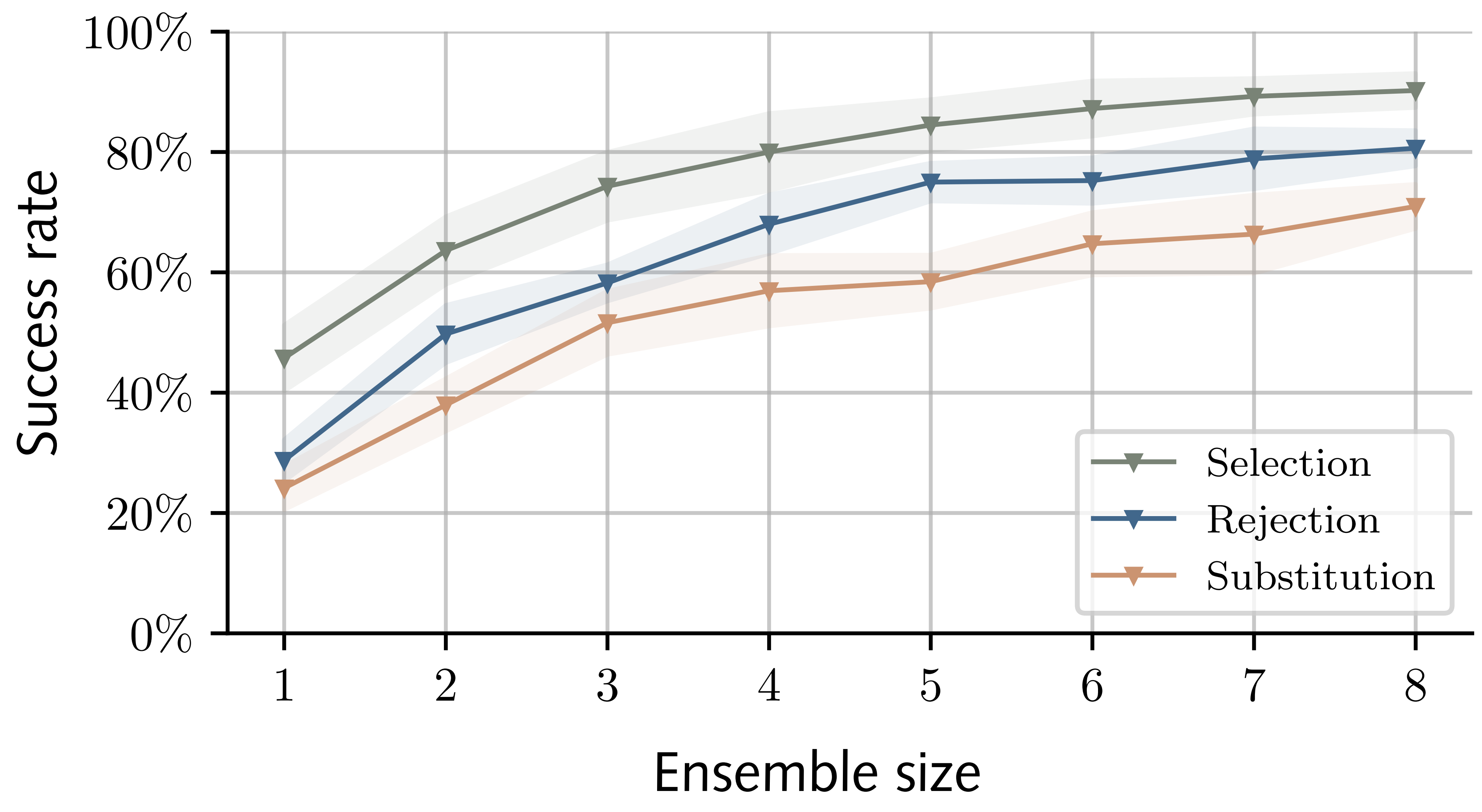
99% success rate if all transformations are used

20-60% attacks succesful with text changes only

Machine Learning and Security

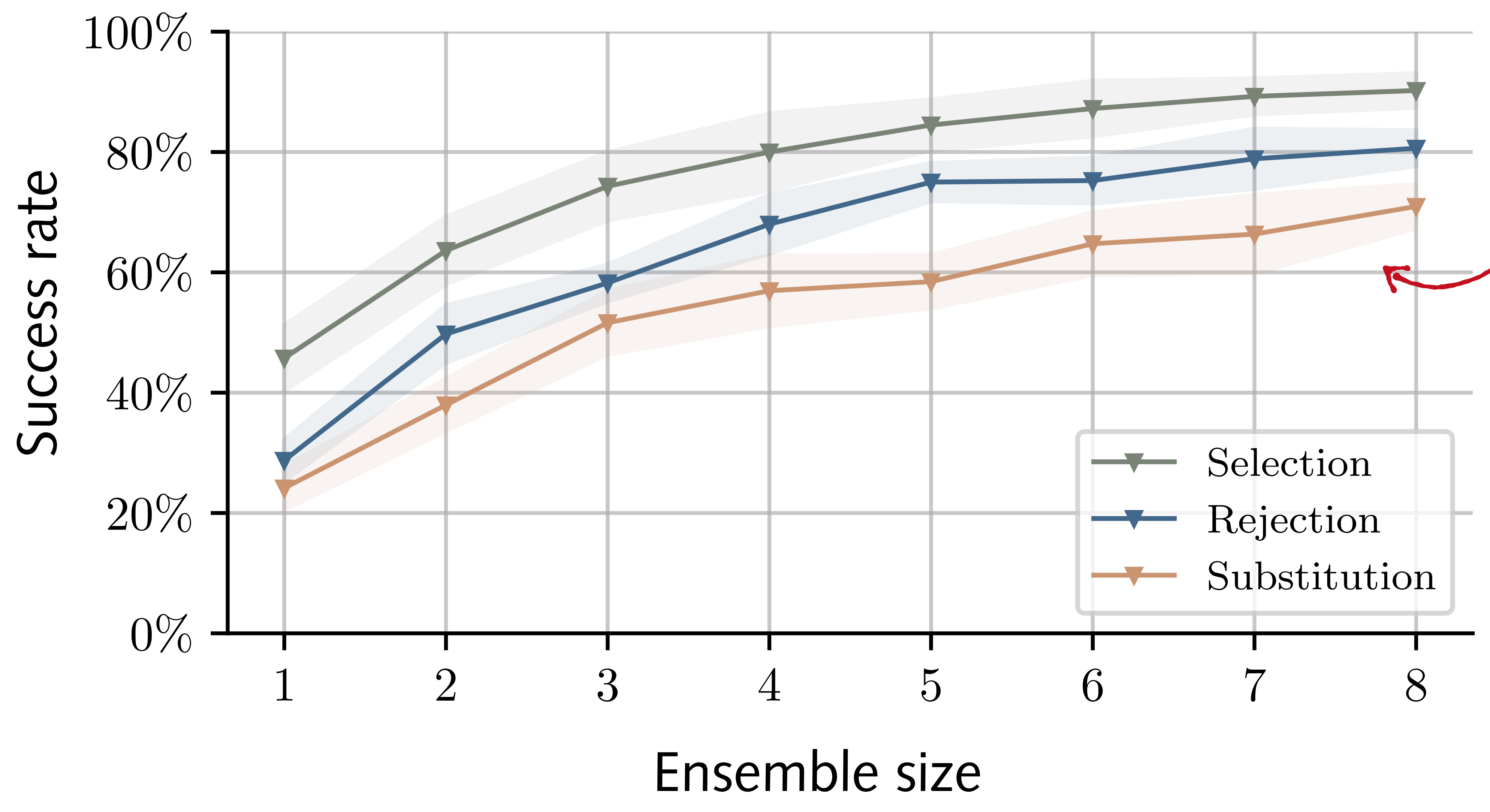
Black-Box Scenario

- Experiment: **Attacks with surrogate models**
  - Training of ensemble of surrogate models on 70% of original data
  - Transfer of best attack to topic model of conference system



Black-Box Scenario

- Experiment: **Attacks with surrogate models**
  - Training of ensemble of surrogate models on 70% of original data
  - Transfer of best attack to topic model of conference system

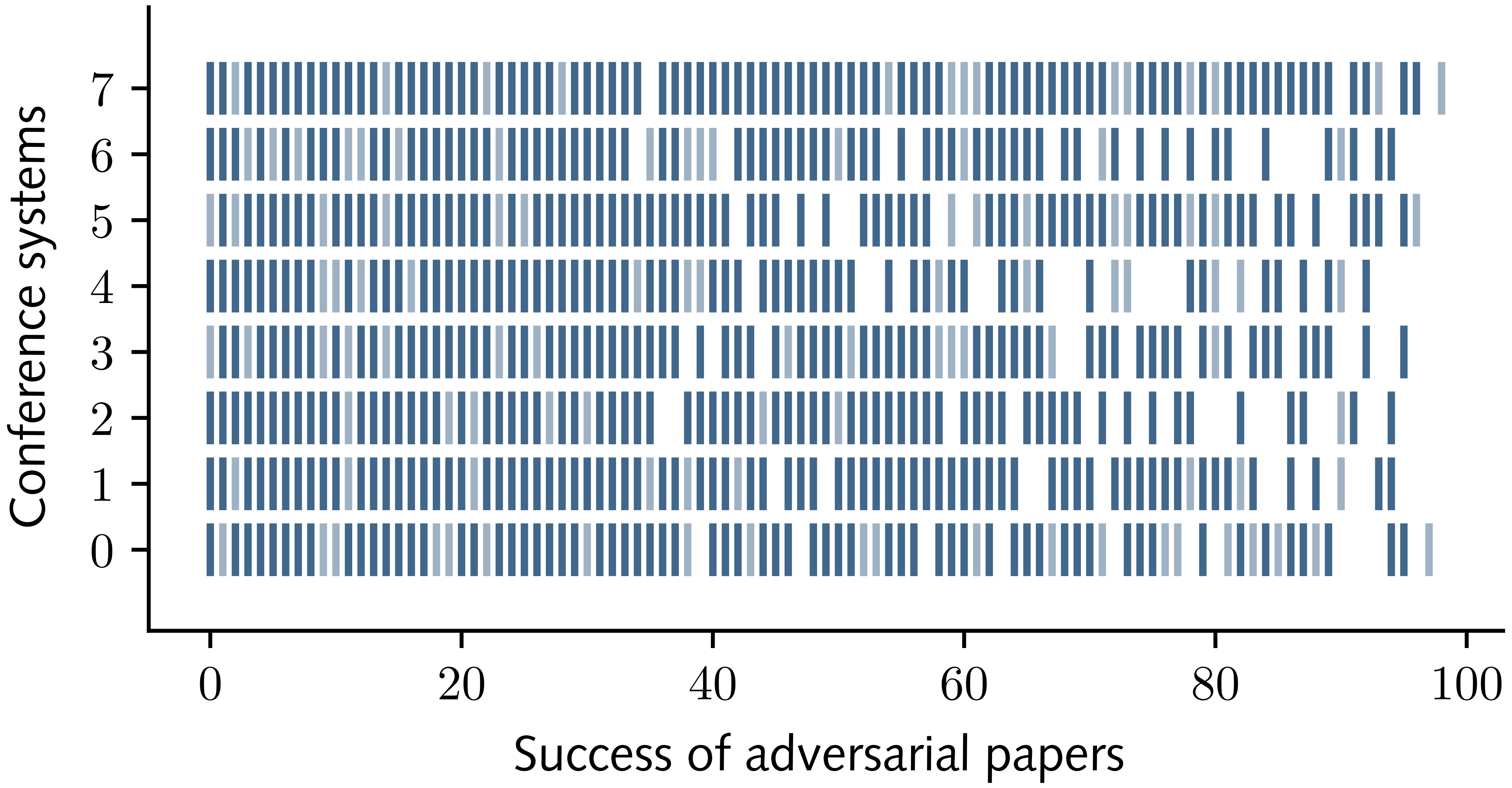


Good performance when ensemble size increased

70%–90% success rate of attacks

Black-Box Scenario

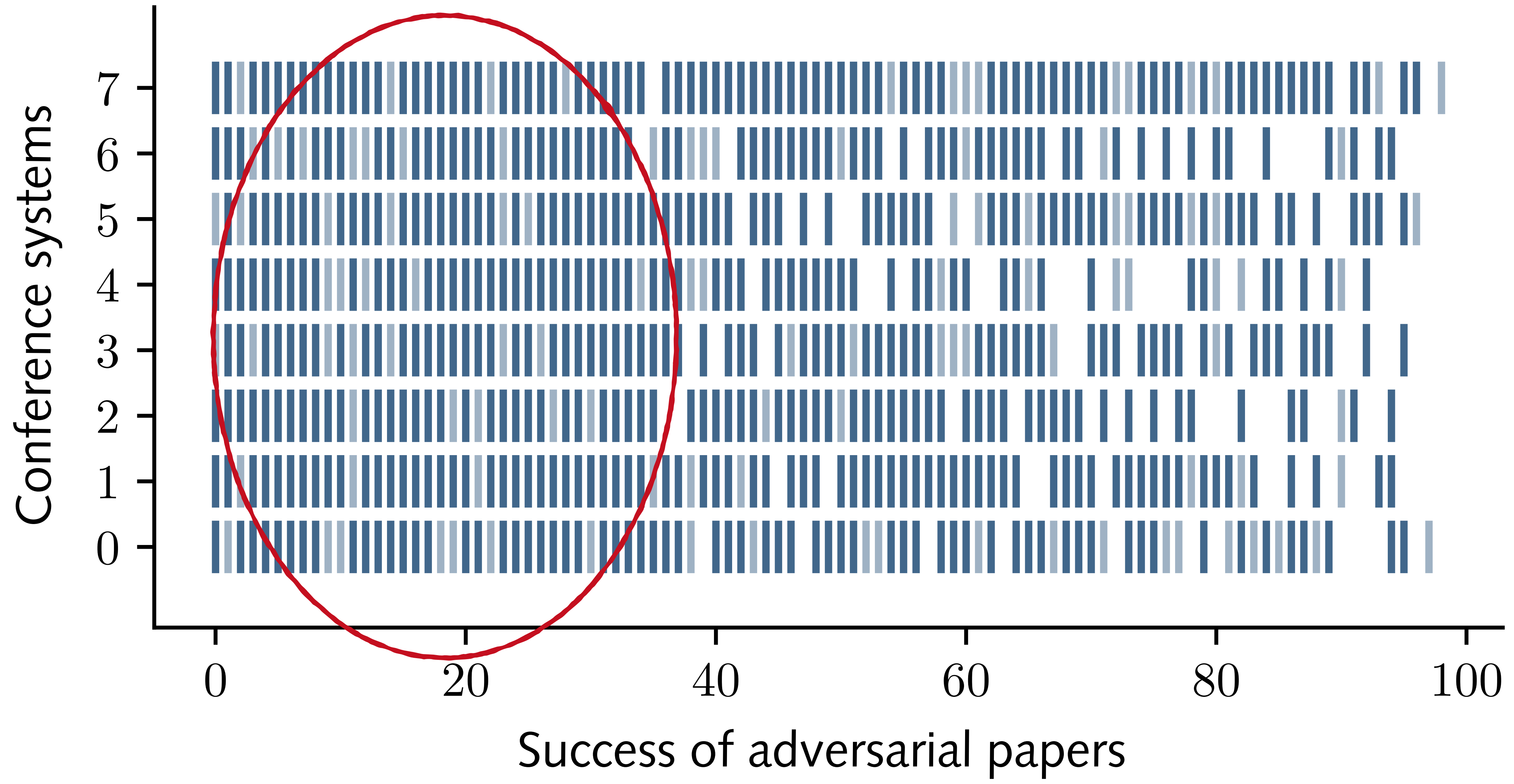
- Experiment: **Transferability for different conference systems**
  - Attacks from 8 surrogate models transfered to conference systems



Machine Learning and Security

Black-Box Scenario

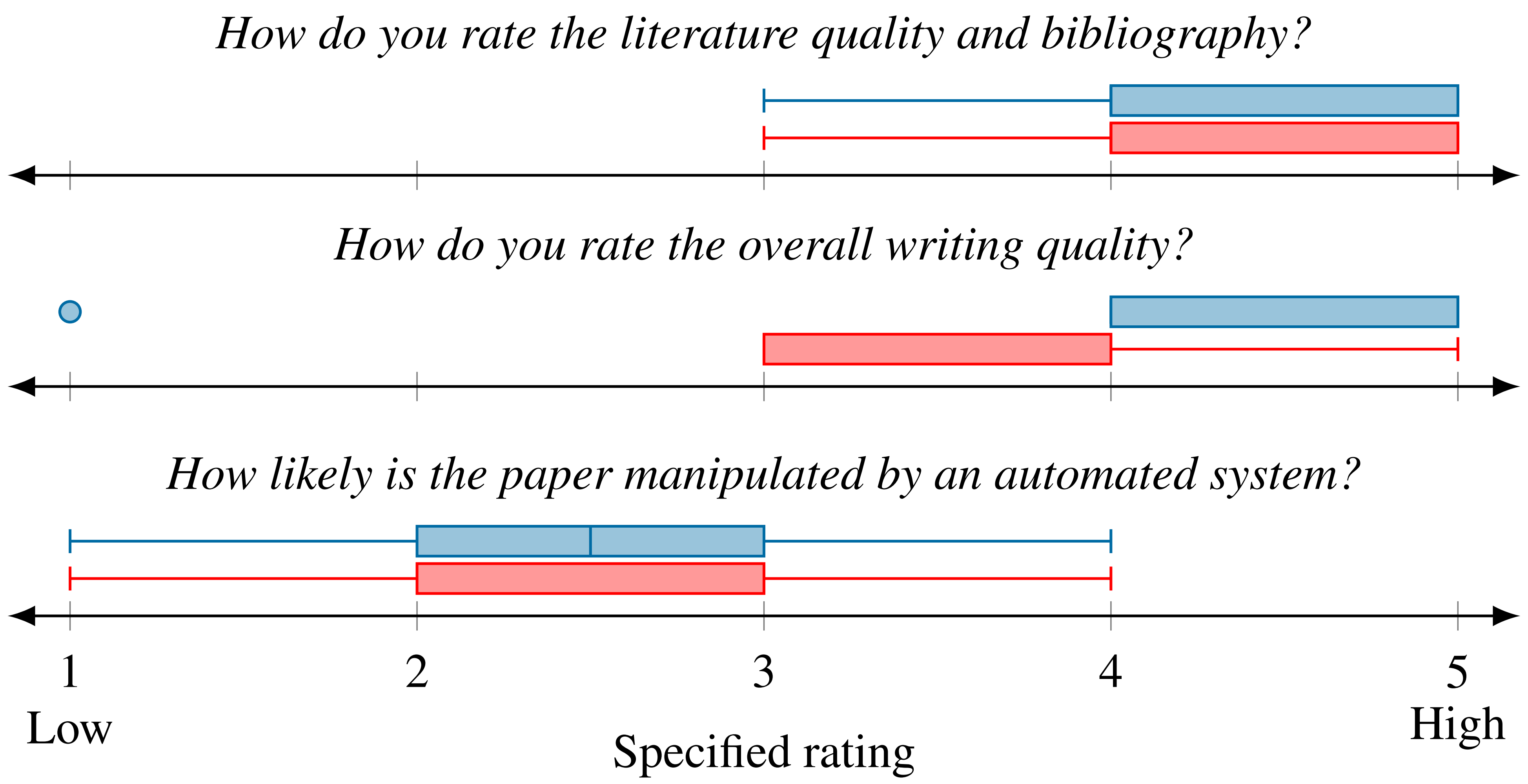
- Experiment: **Transferability for different conference systems**
  - Attacks from 8 surrogate models transfered to conference systems



34% papers effective against all eight systems

Plausibility

- **Evaluation of plausibility with small user study**
  - 21 security researchers perform mini-reviews on papers
  - Participants asked about quality of paper and suspiciousness



No significant difference observed

Conclusions

Aftermath

- **Possible defenses**
  - Sanitization and anomaly detection in PDF files
  - Prevention of format and encoding tricks with OCR recognition
  - Defenses against text transformations currently unknown
- **Notification of TPMS and AutoBid developers**
  - Positive email exchange — No time for defenses currently 🙈
- **Is this a threat? Personal take: Yes!**

Conclusions

- **New attack against automatic reviewer-paper assignment**
  - Hybrid attack strategy in feature space and problem space
  - Minimal and unobtrusive transformations of papers
- **Broader perspective**
  - Decisions based on learning models inherently insecure
  - More to explore off the beaten path of adversarial learning
- More at https://github.com/rub-syssec/adversarial-papers

Thanks! Questions?