Reconstructing proprietary video streaming algorithms

Maximilian Grüner, Melissa Licciardello, Ankit Singla

ETH zürich

Photo: ETH Zürich / Gian Marco Castelberg

Roadmap

Roadmap

- Introduction

Roadmap

- Introduction
- Reconstruction

Roadmap

- Introduction
- Reconstruction
- Experiments

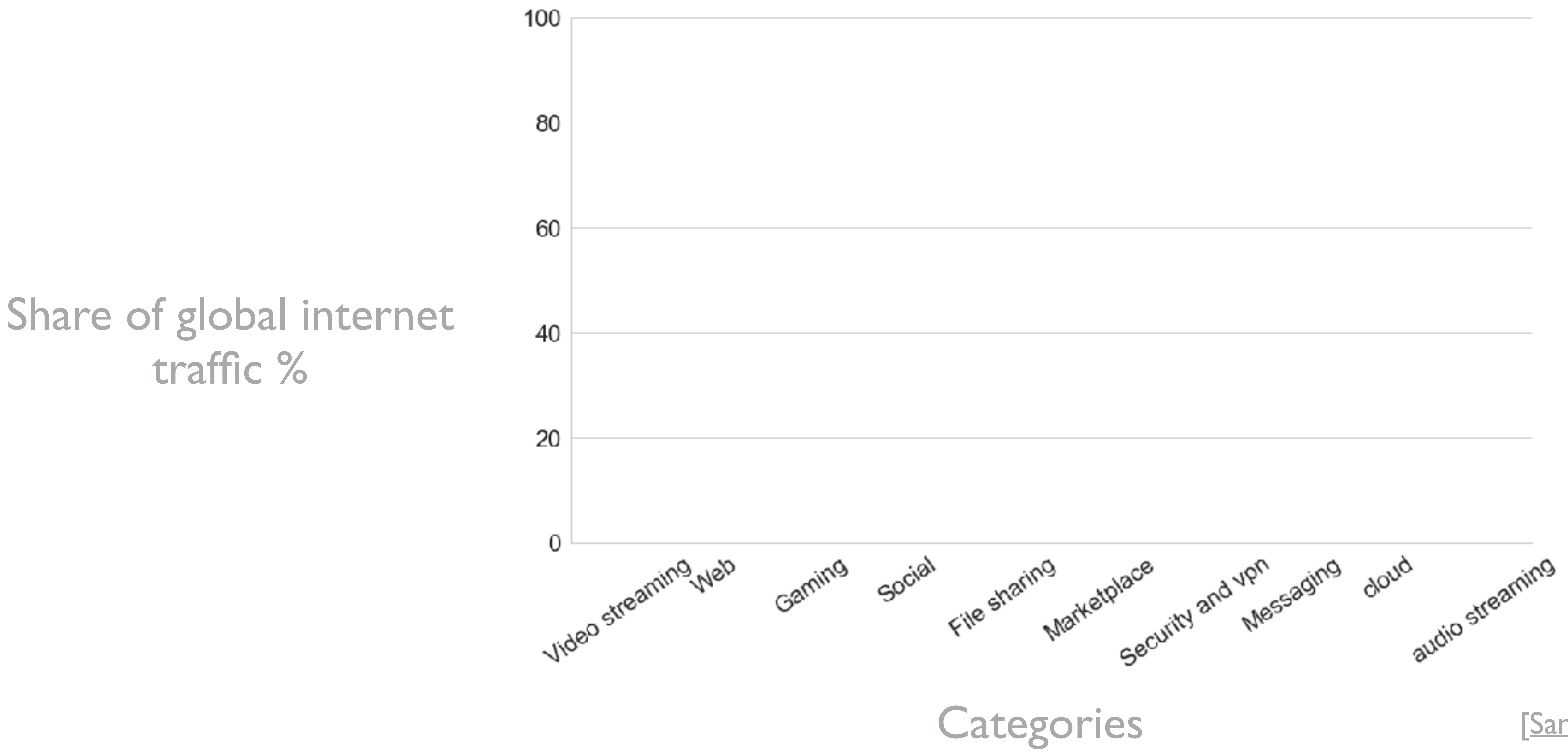
Roadmap

- Introduction
- Reconstruction
- Experiments
- Limitations and future work

Roadmap

- Introduction
- Reconstruction
- Experiments
- Limitations and future work
- Takeaways

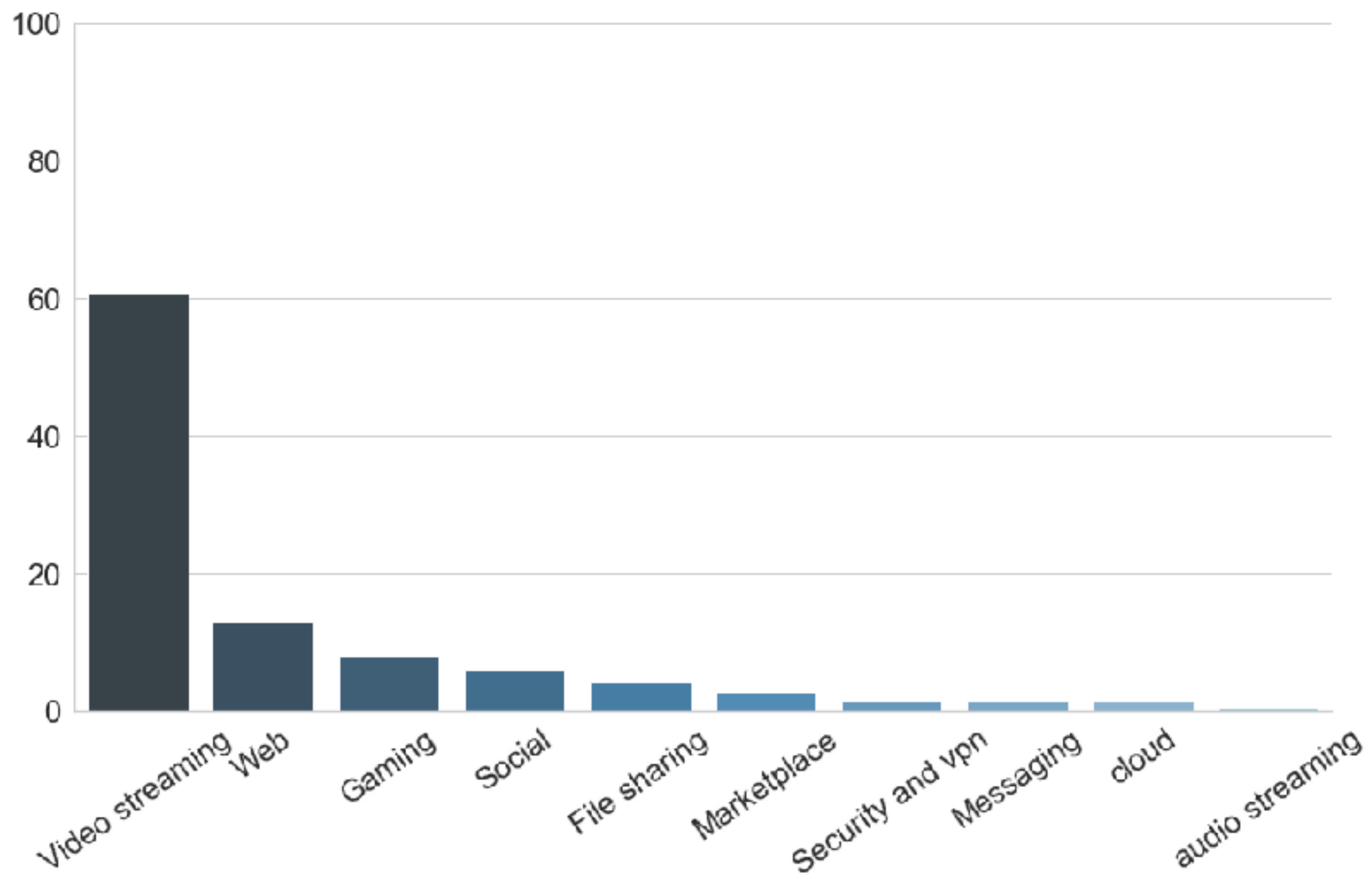
Streaming in Industry



[Sandvine 2019]

Streaming in Industry

Share of global internet traffic %

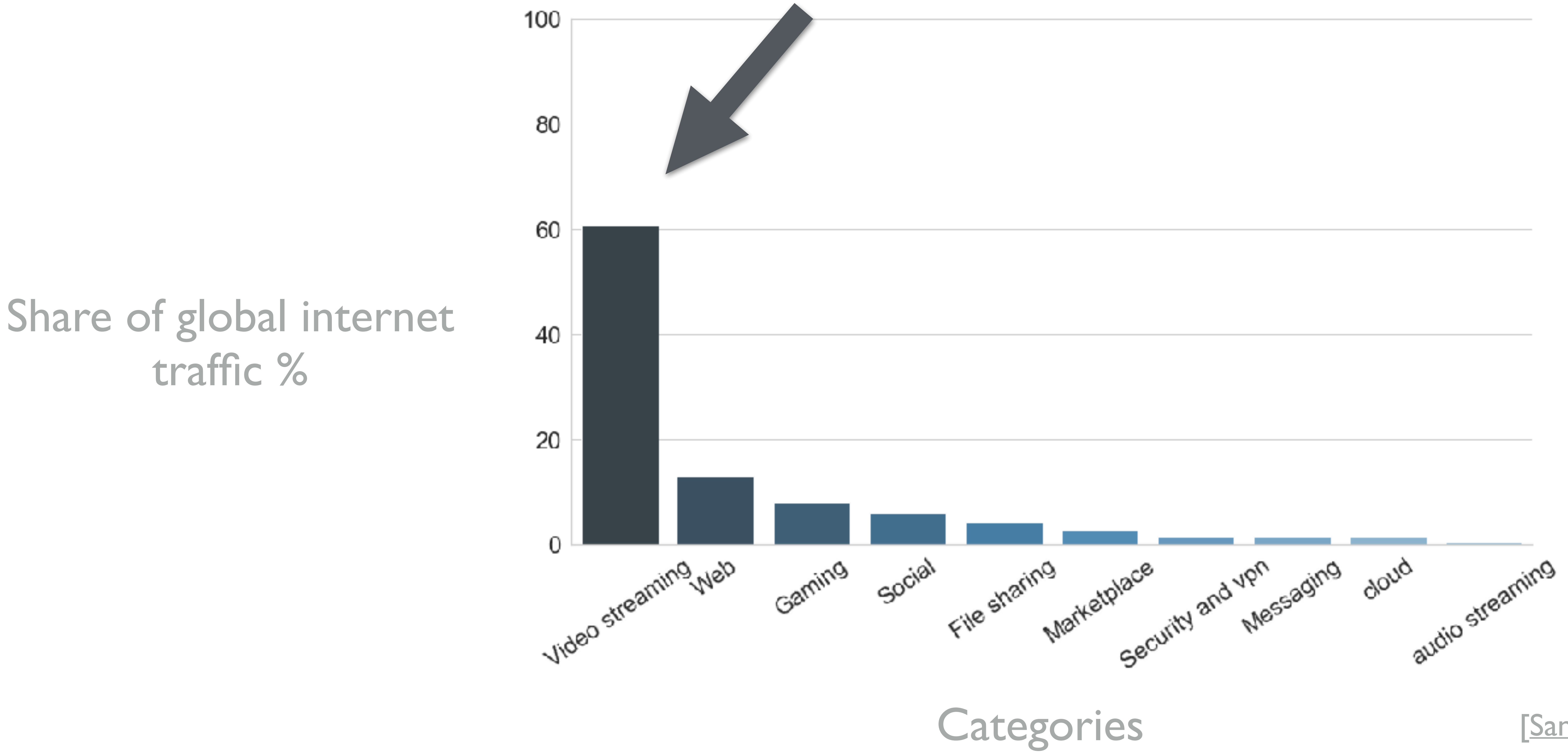


Categories

[Sandvine 2019]

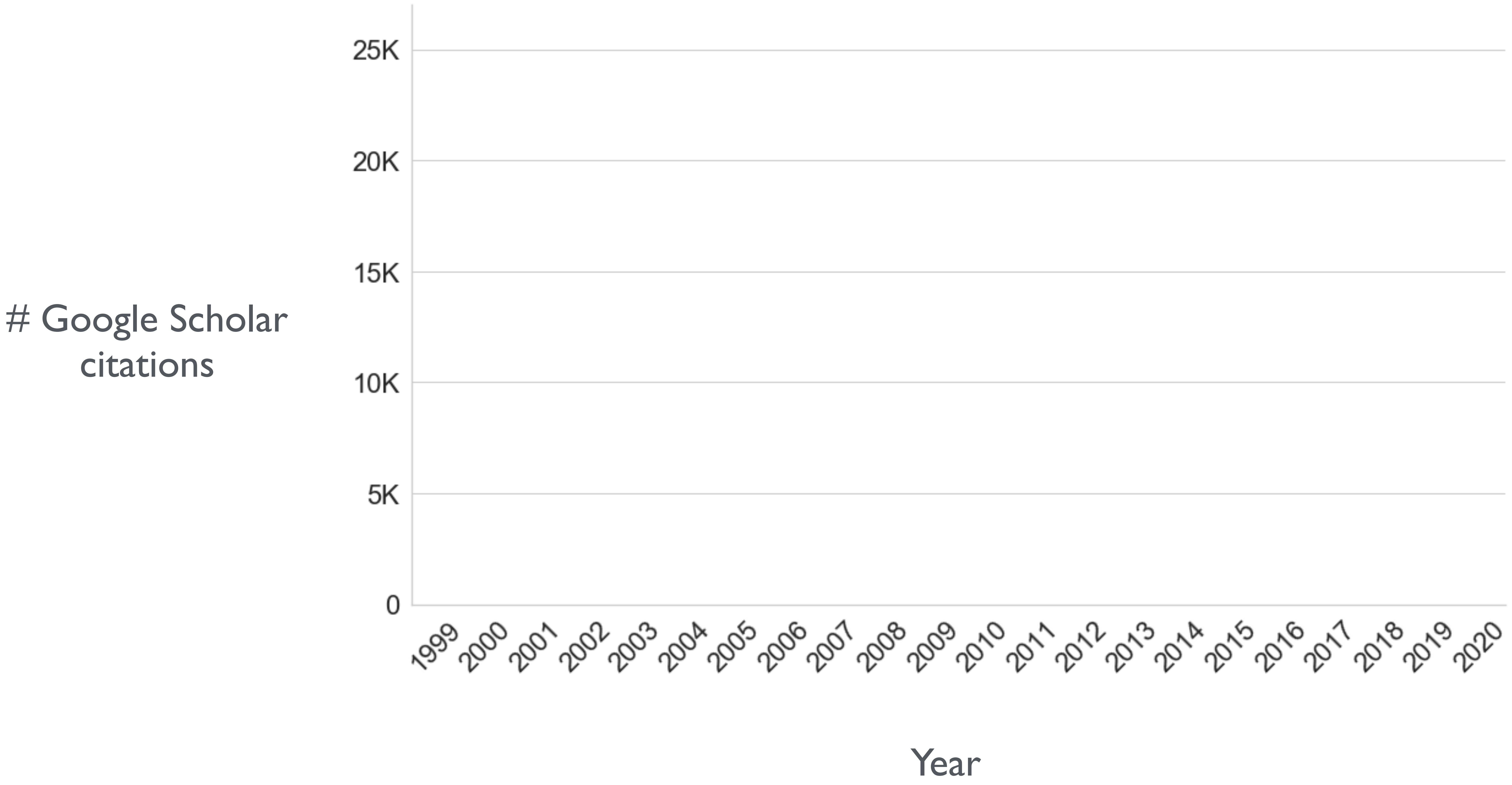
Streaming in Industry

23.8 % Various, 23.1 % Netflix, 12.7 % YouTube, 4.2 % Twitch…

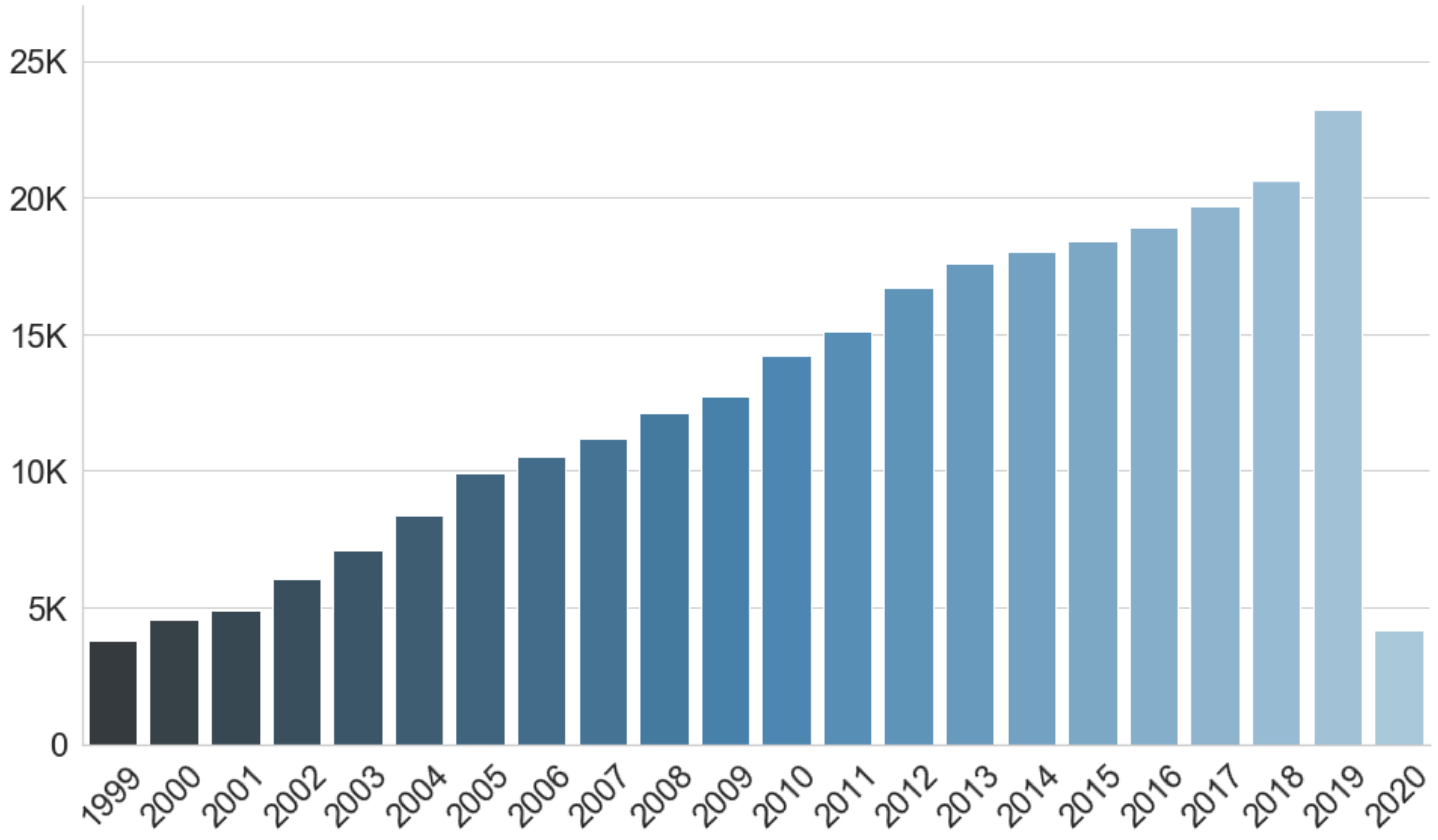


[Sandvine 2019]

Streaming in academia



Streaming in academia

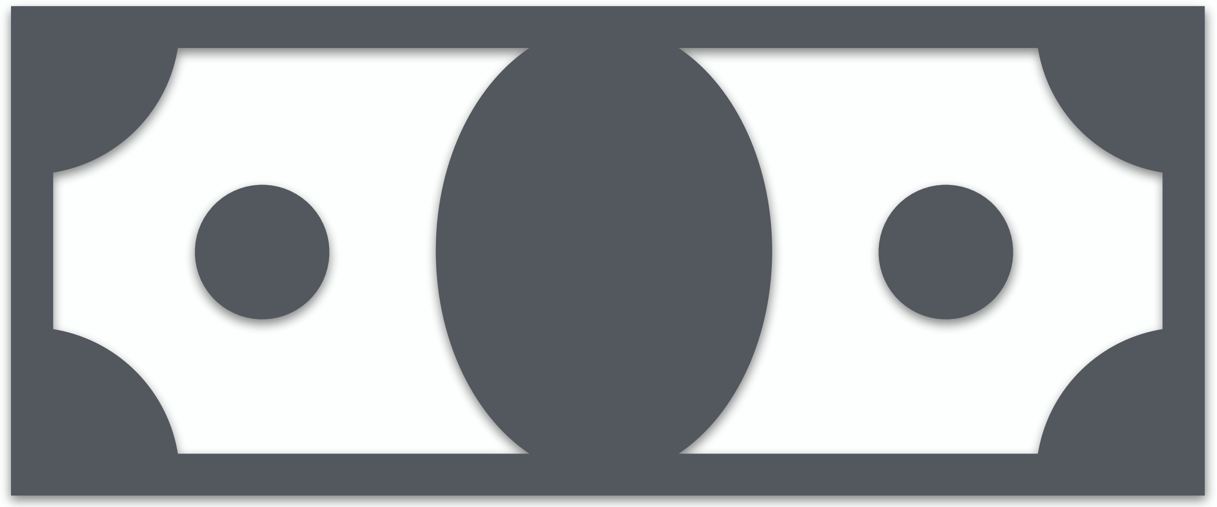


# Google Scholar citations

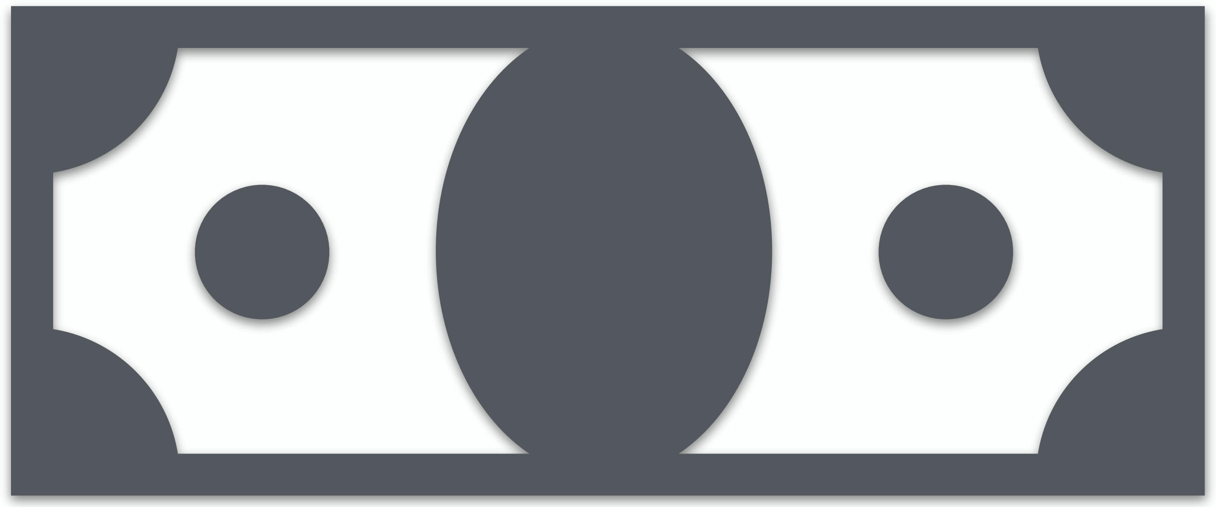
Year

Applications of reconstruction

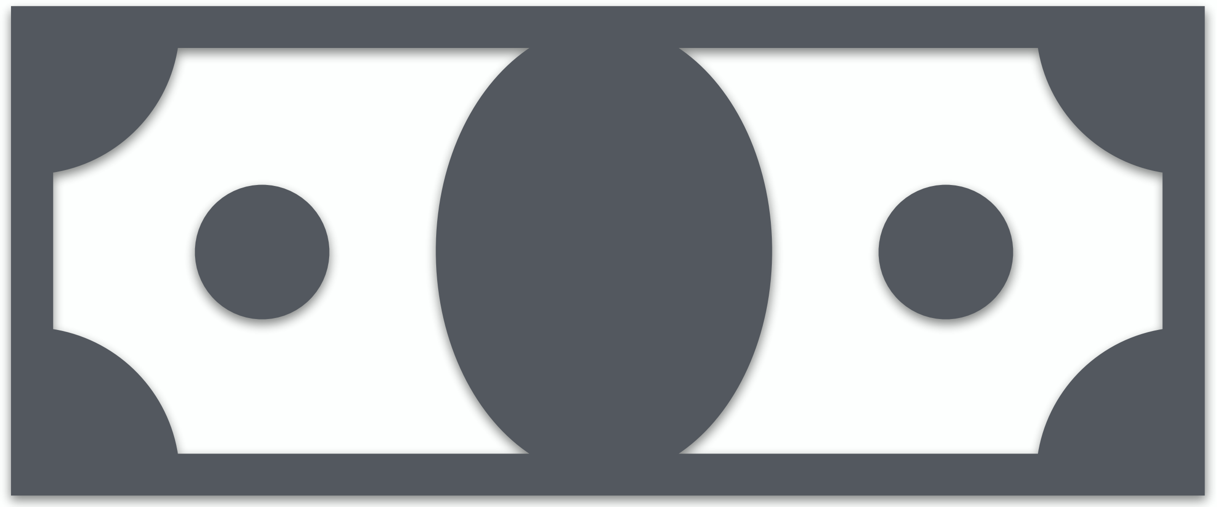
Applications of reconstruction

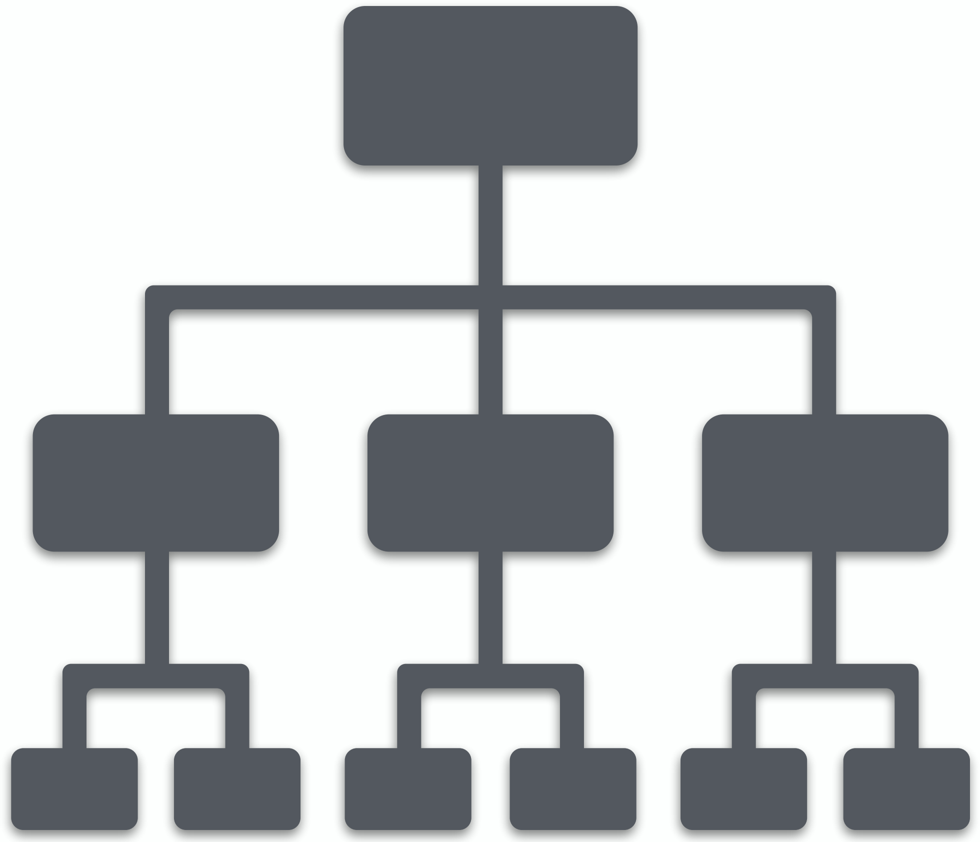


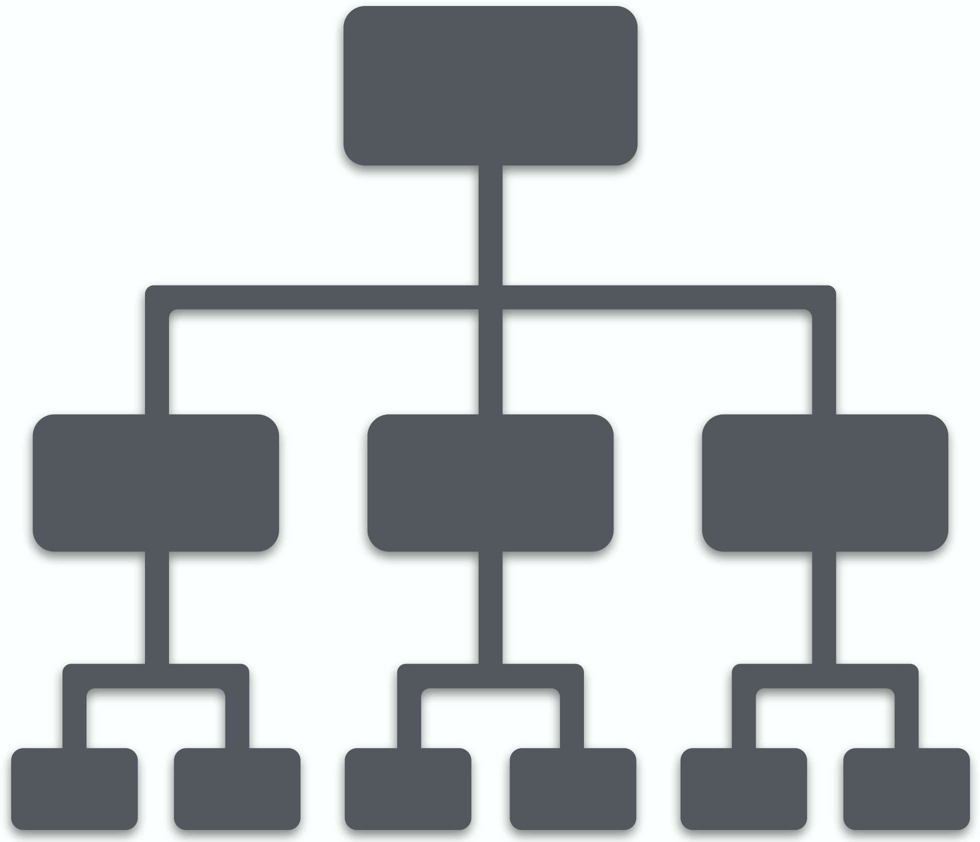
Applications of reconstruction

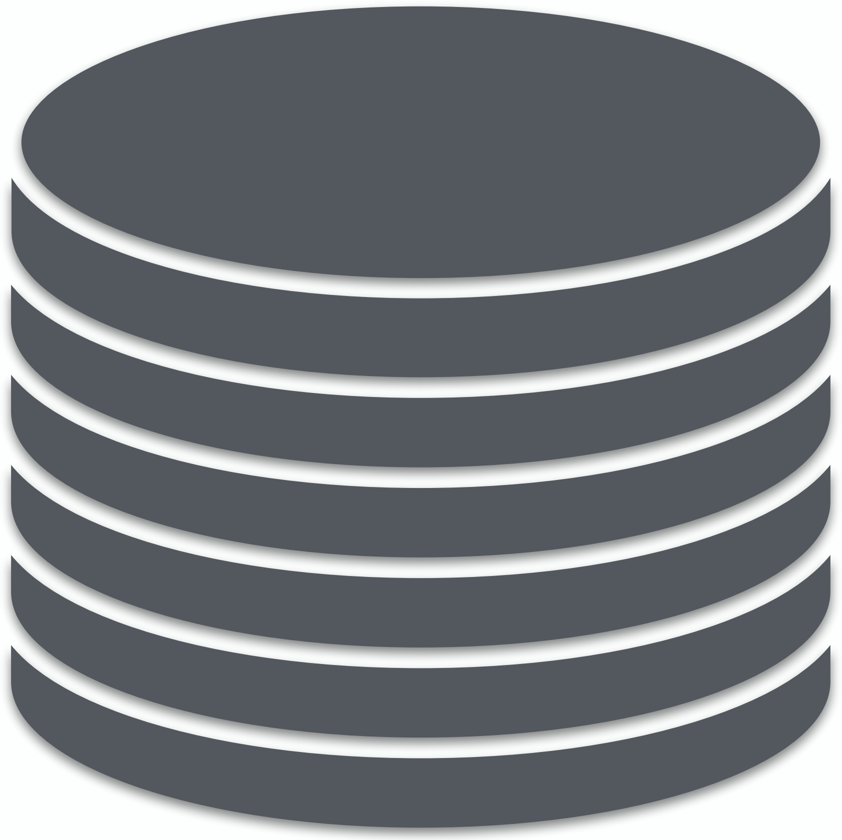


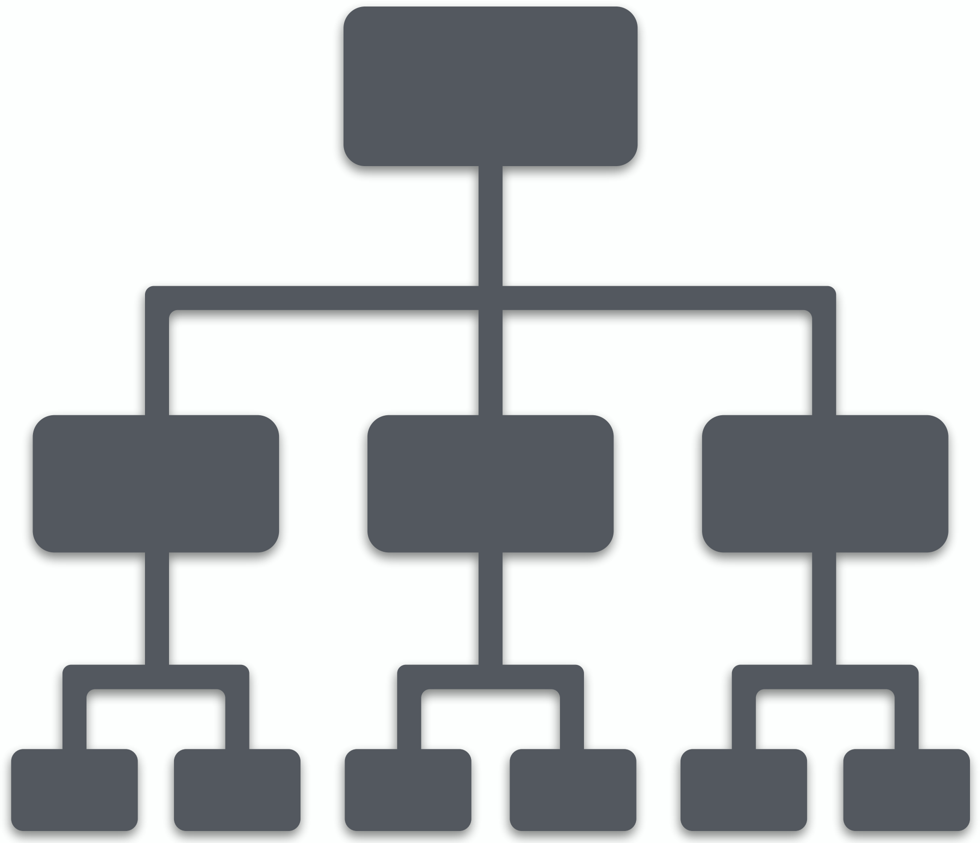
Applications of reconstruction

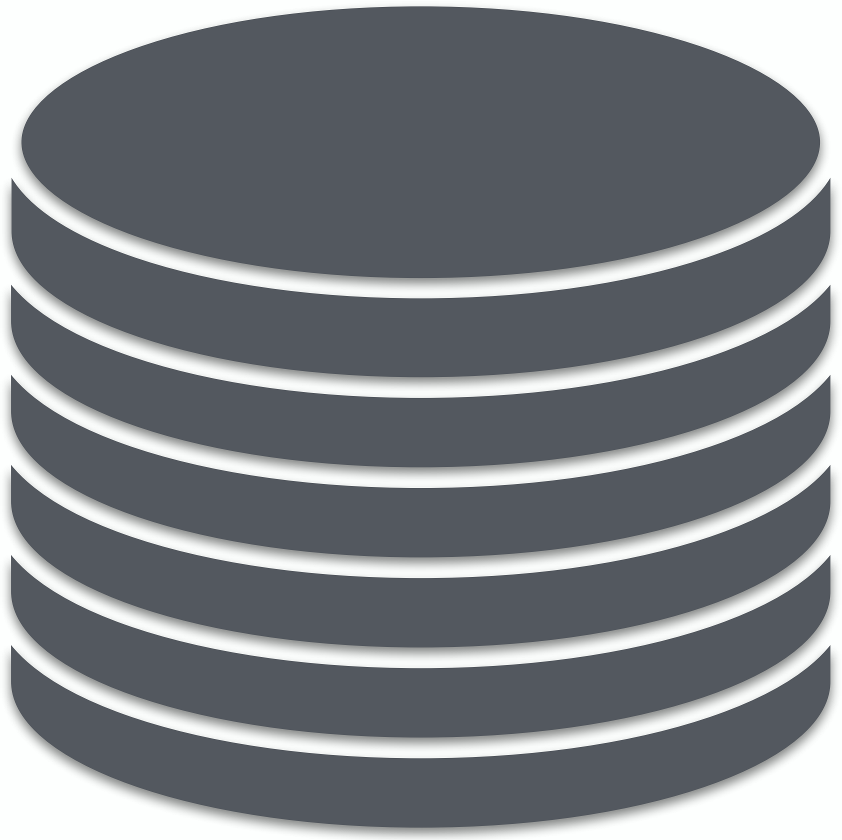


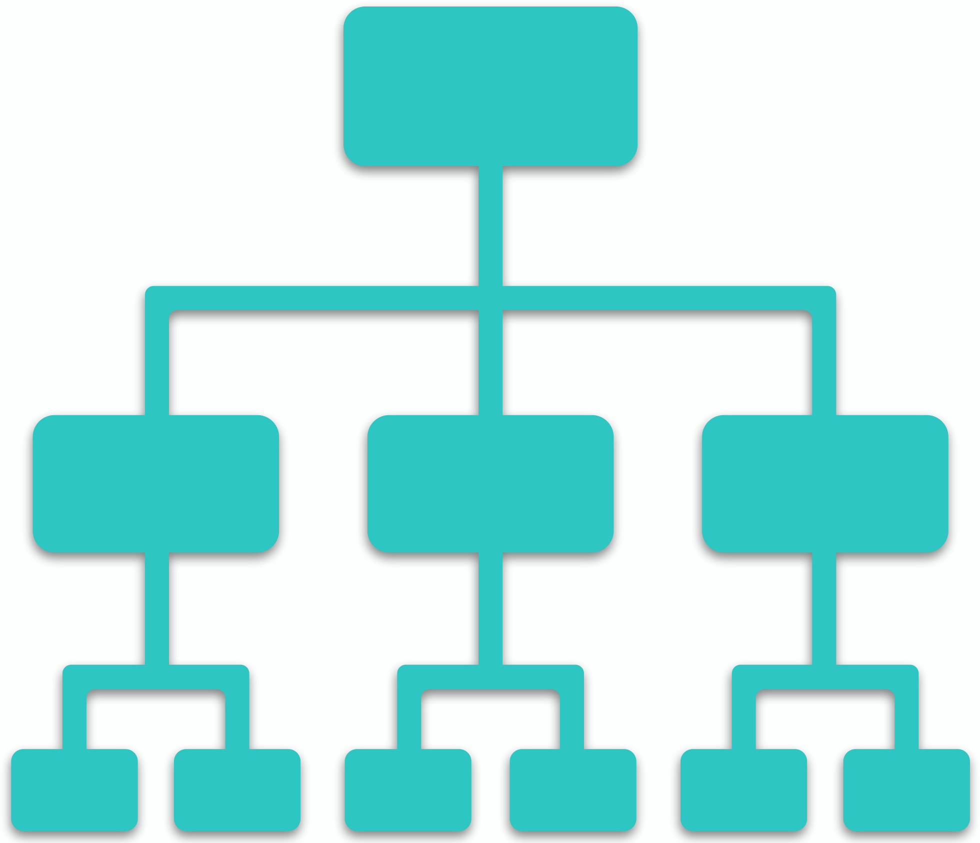


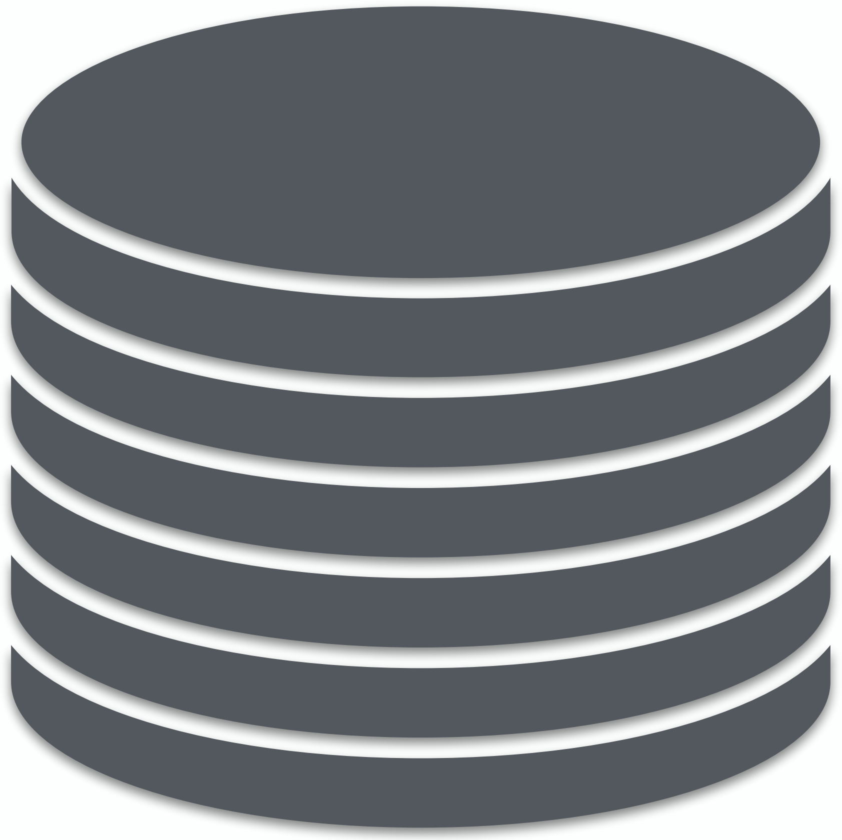




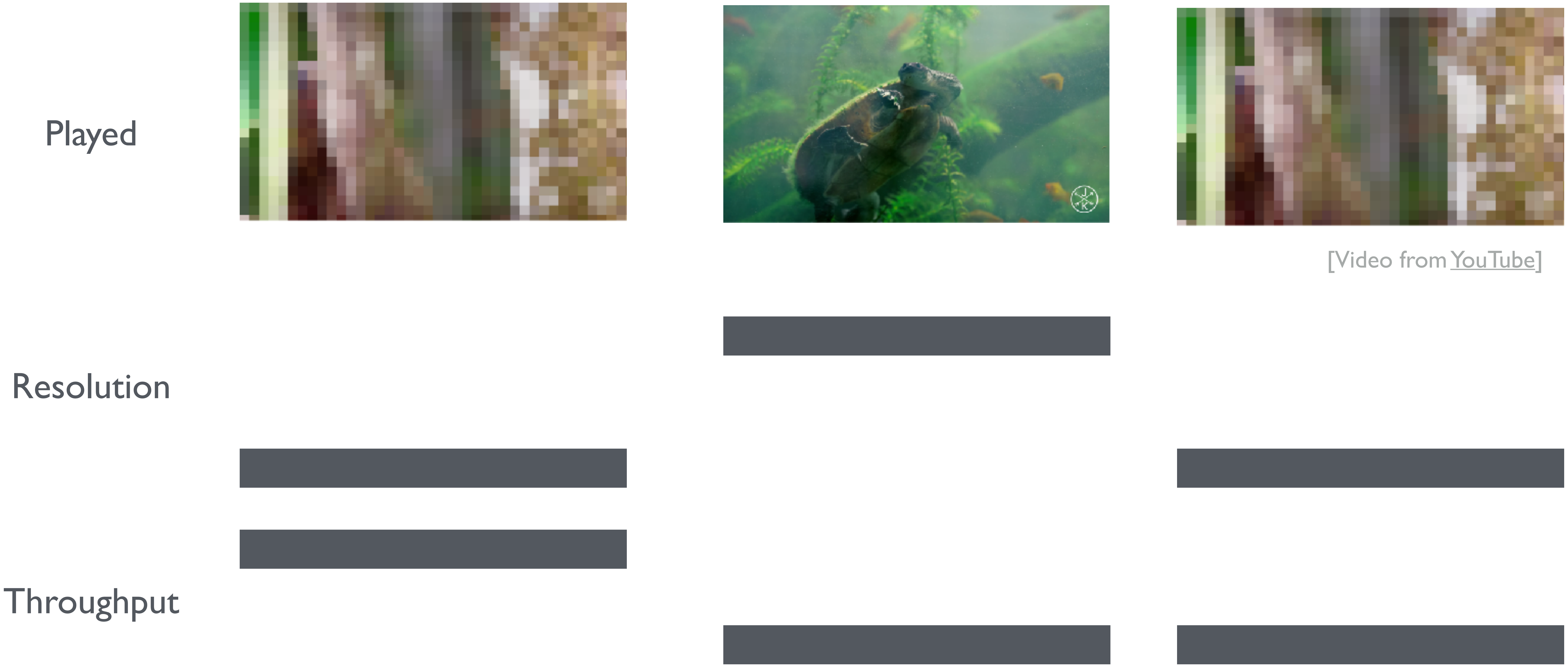








Introduction into ABRs



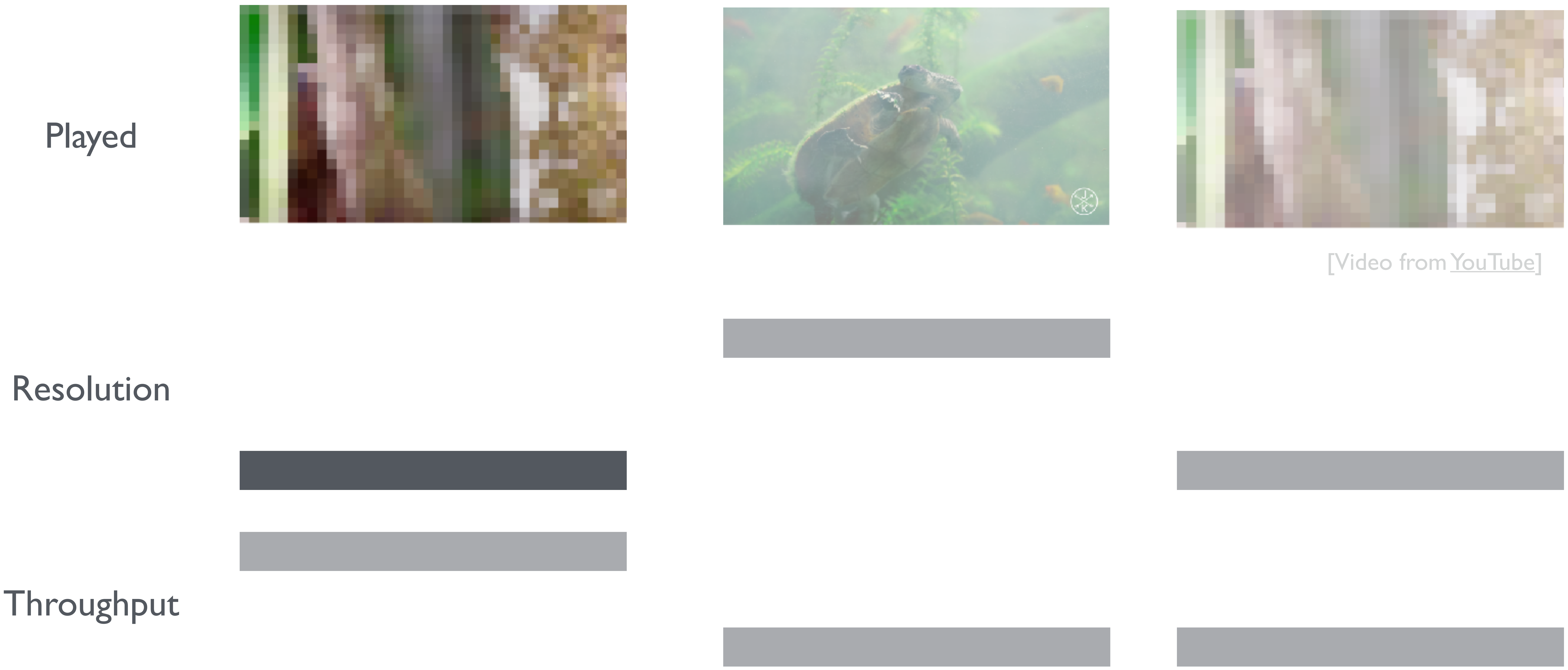
Played

Resolution

Throughput

[Video from YouTube]

Introduction into ABRs



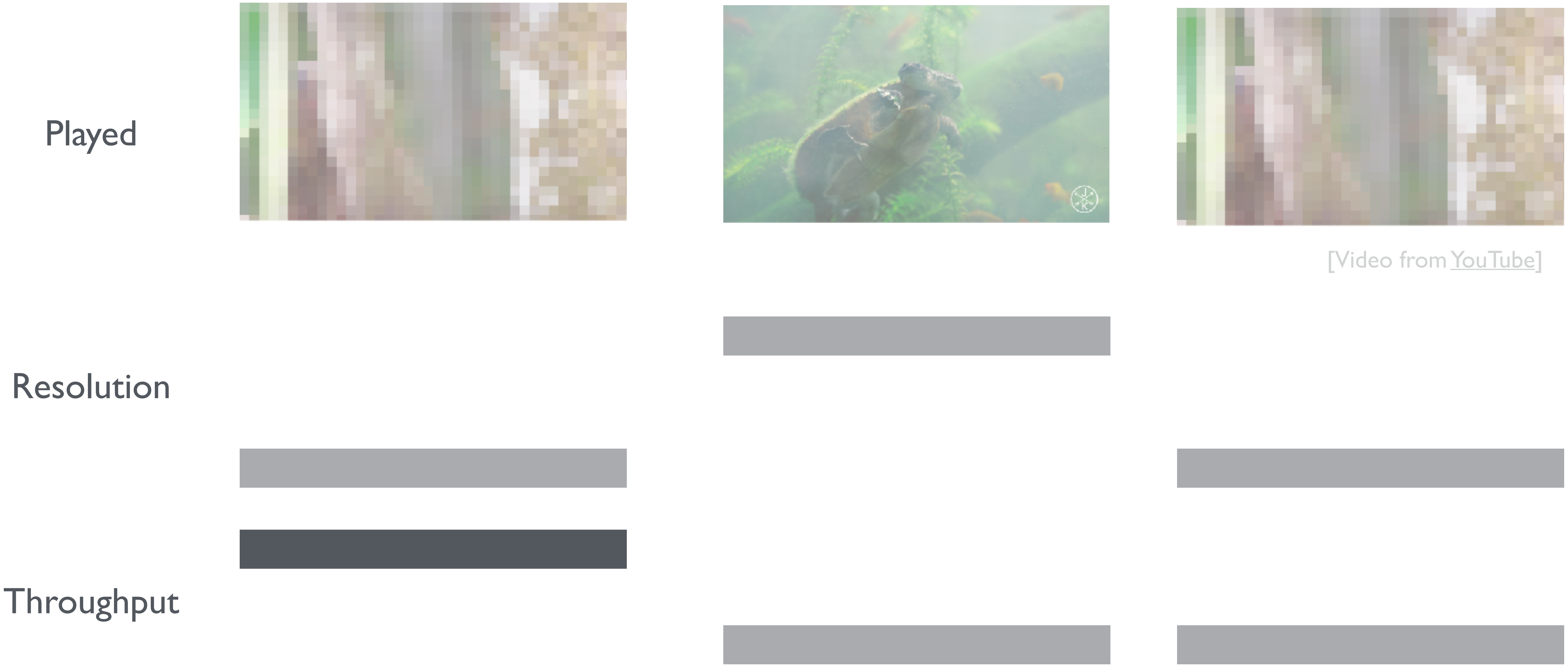
Played

Resolution

Throughput

[Video from YouTube]

Introduction into ABRs



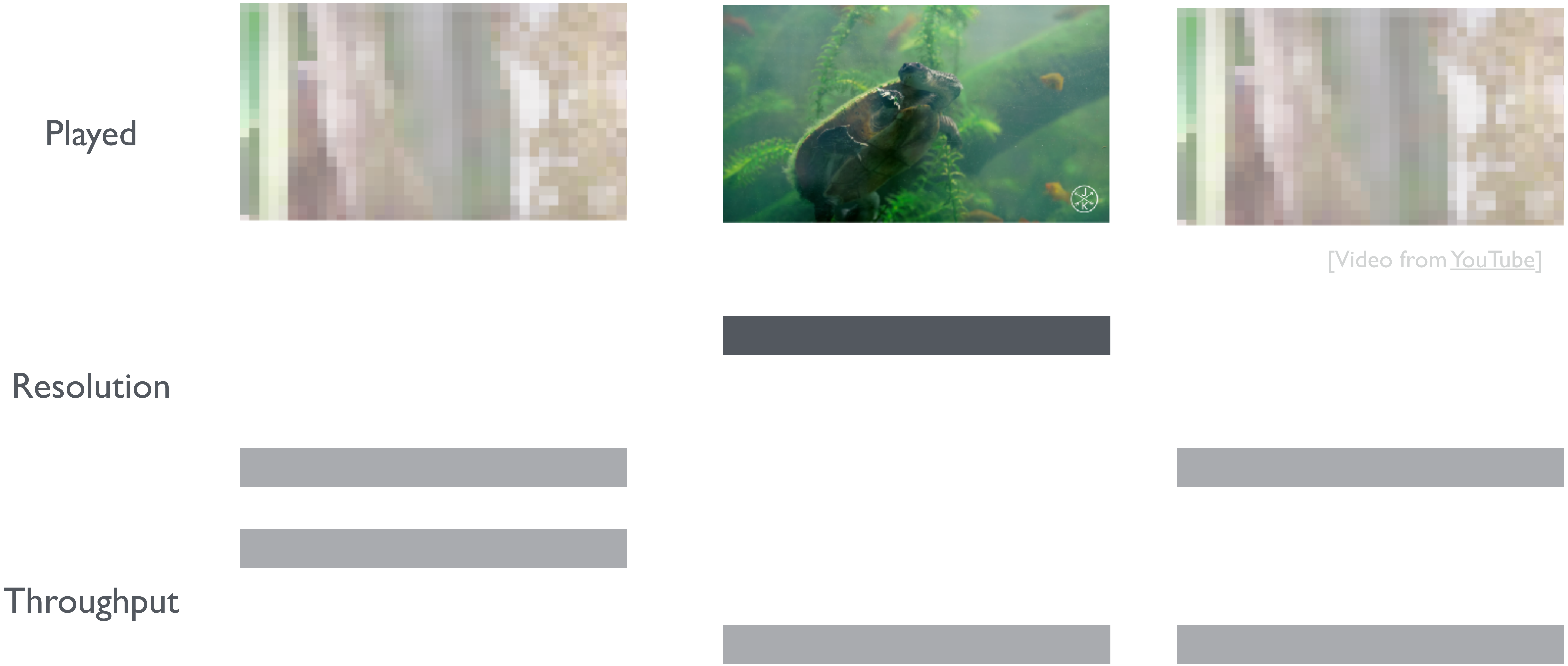
Played

Resolution

Throughput

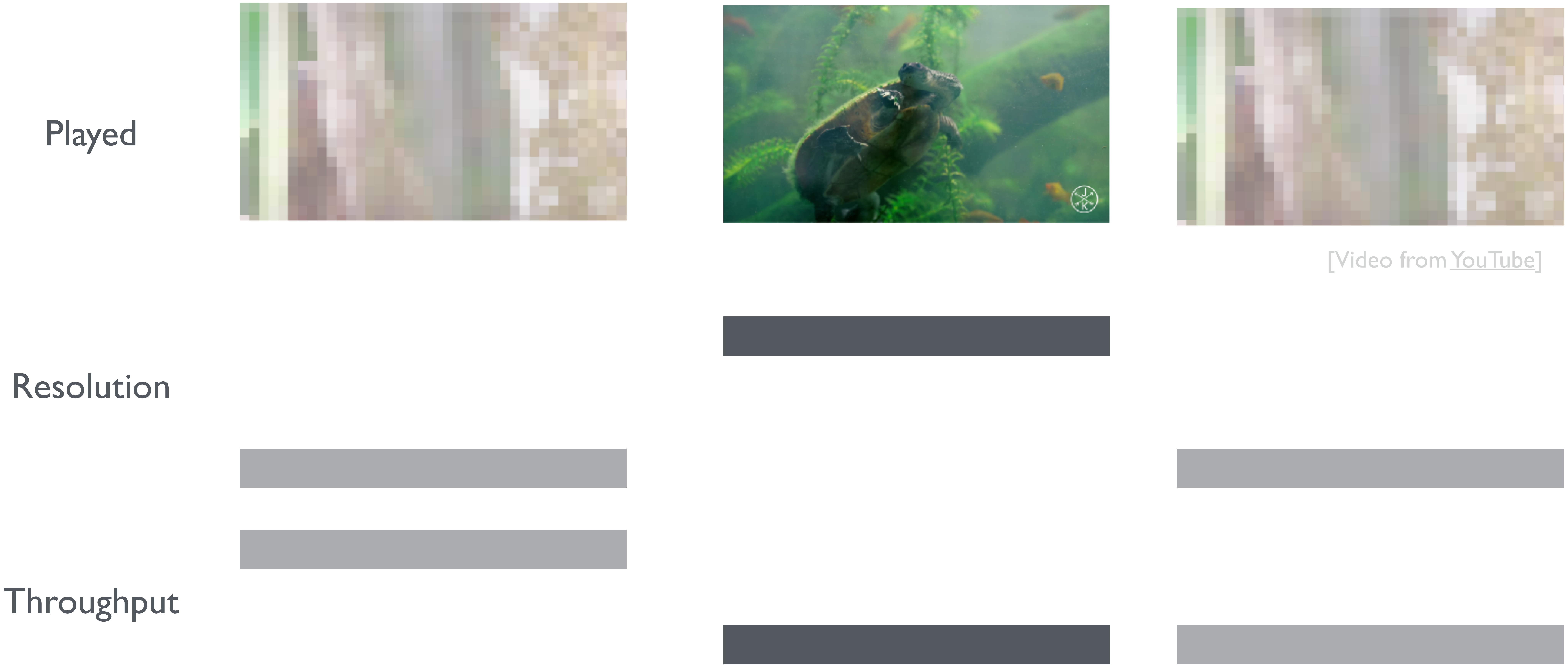
[Video from YouTube]

Introduction into ABRs



[Video from YouTube]

Introduction into ABRs



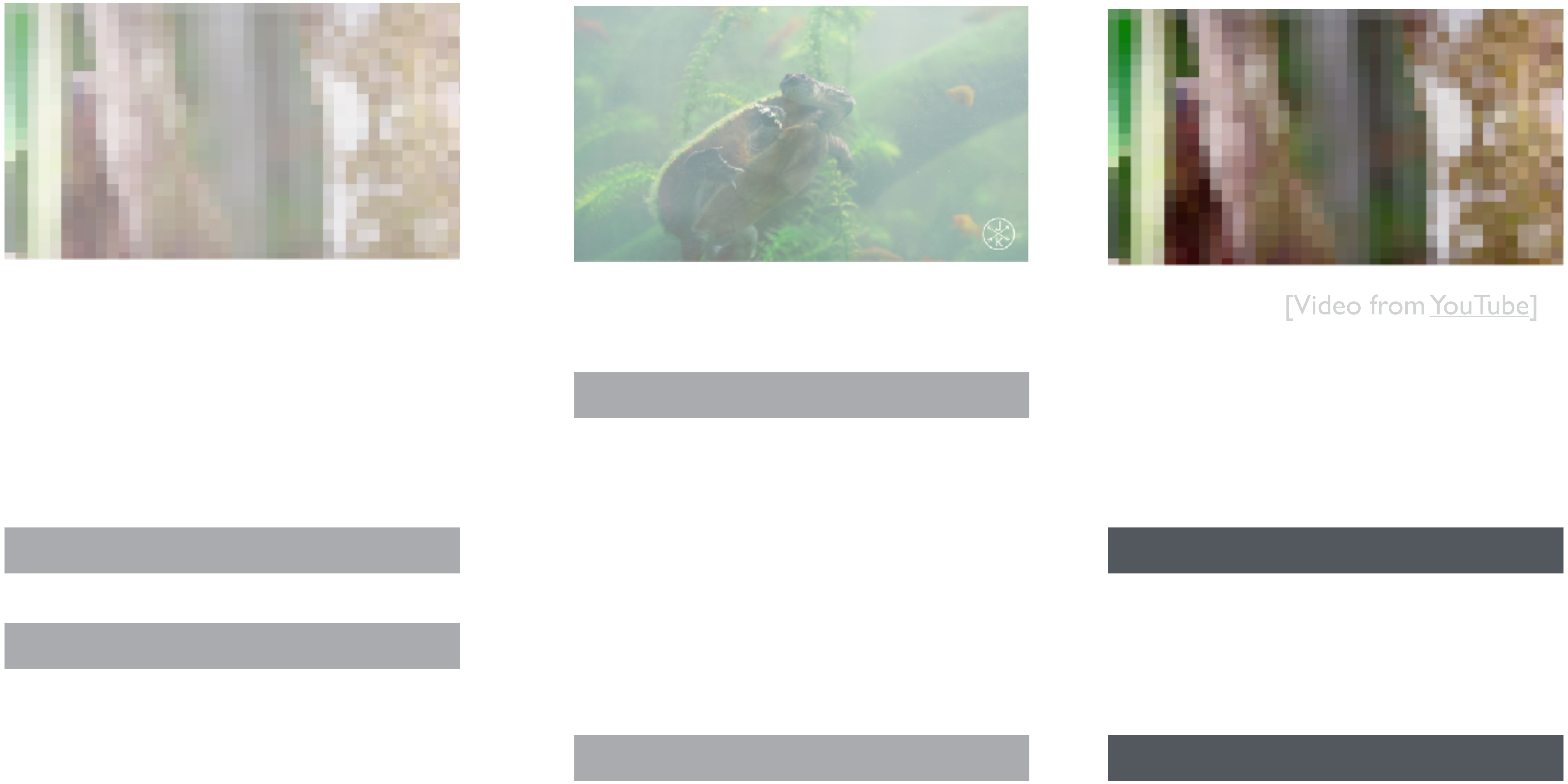
[Video from YouTube]

Introduction into ABRs

Played

Resolution

Throughput



[Video from YouTube]

Simple ABR in code

```
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ⬇⬇
    else:  ## throughput_estimate == high
        next_quality = ⬆⬆
    return next_quality
```

Simple ABR in code

```
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ⬇⬇

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

Simple ABR in code

```
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ⬇⬇

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

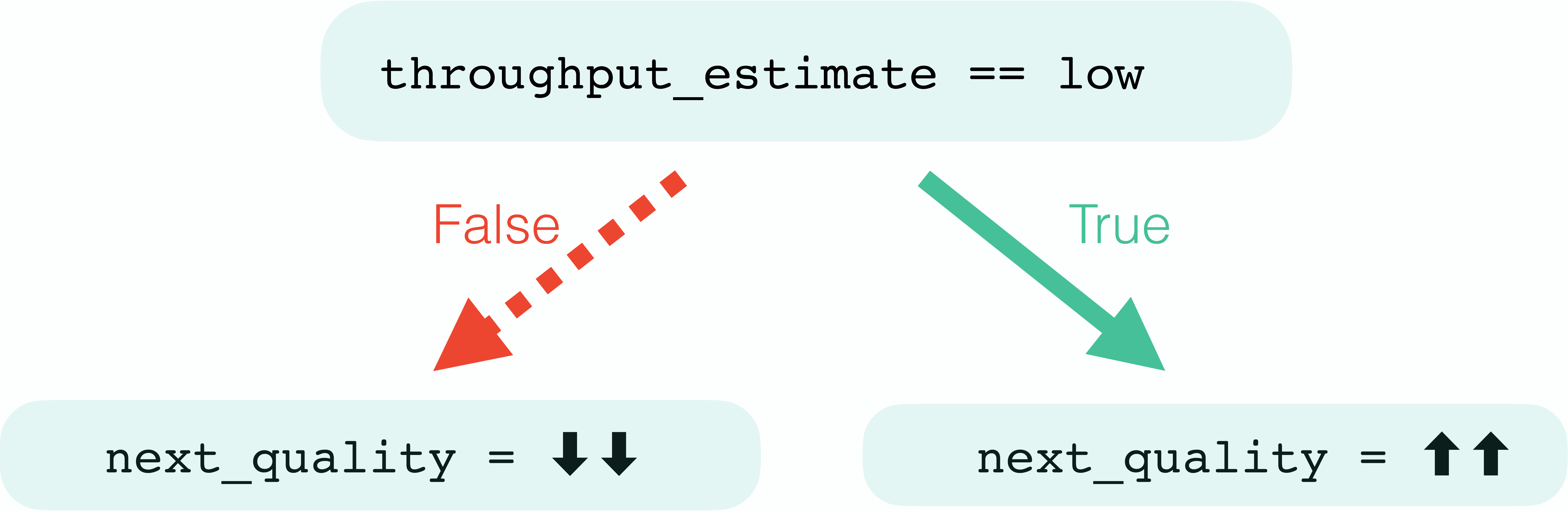
Simple ABR in code

```
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ⬇⬇
    else:  ## throughput_estimate == high
        next_quality = ⬆⬆
    return next_quality
```

Simple ABR in code

```
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ⬇⬇

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

Simple ABR as a tree



`throughput_estimate == low`

False

True

`next_quality = ⬇⬇`

`next_quality = ⬆⬆`

(Slightly more) complex ABR in code

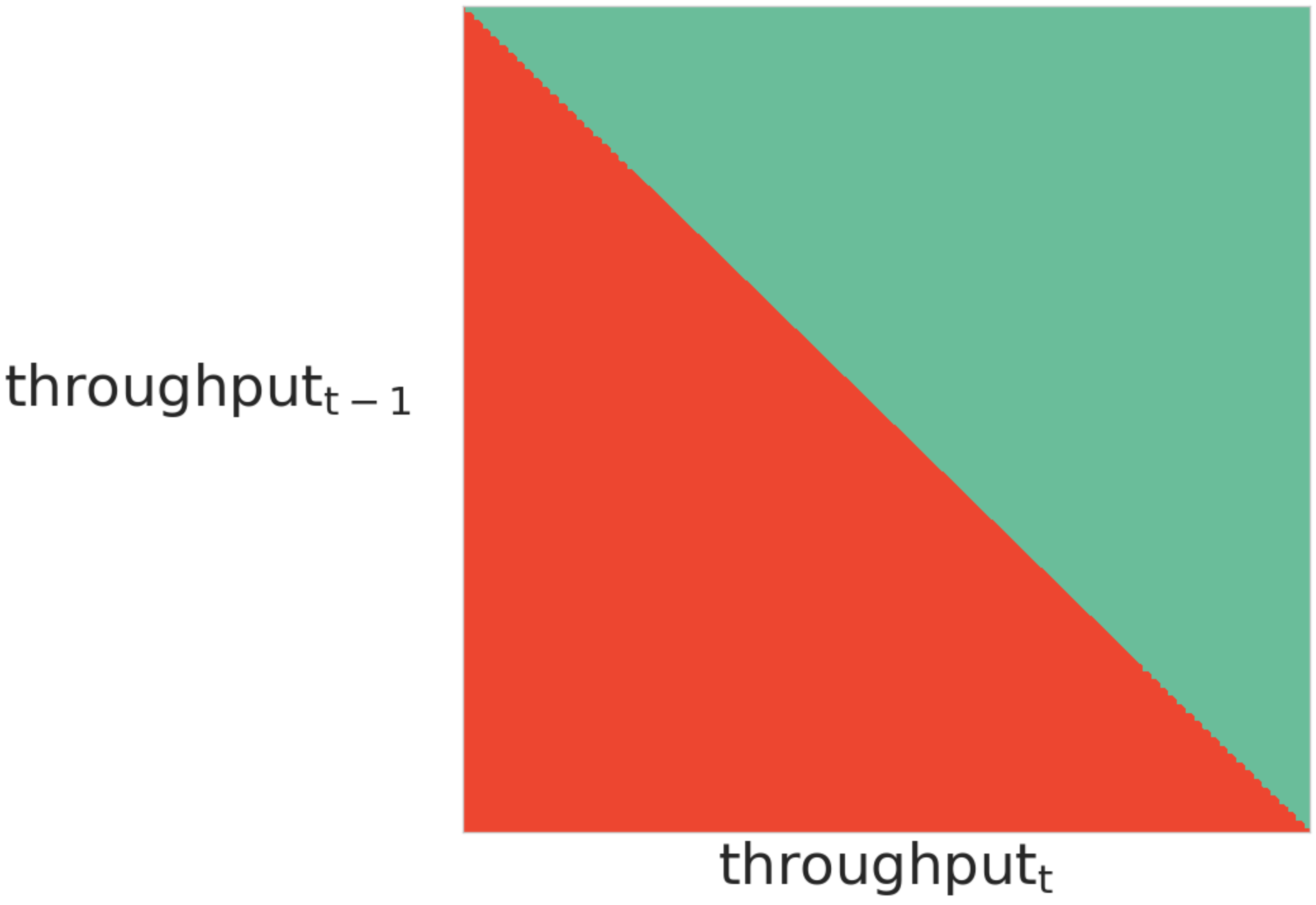
```
def choose_next_quality(throughput_history):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        next_quality = ⬇⬇

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

(Slightly more) complex ABR in code

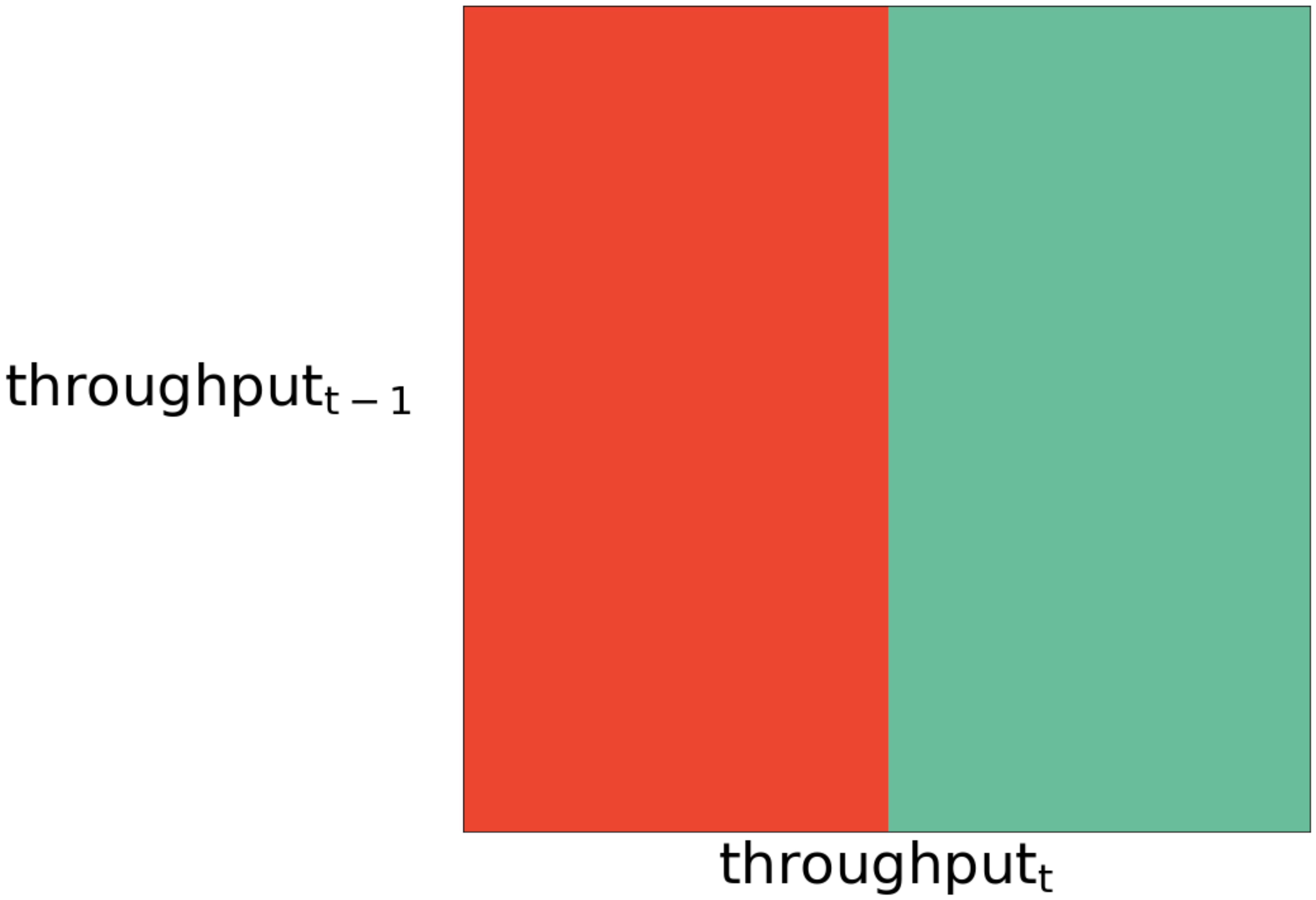
```
def choose_next_quality(throughput_history):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        next_quality = ⬇⬇
    else:   ## throughput_estimate == high
        next_quality = ⬆⬆
    return next_quality
```

Why do we need primitives

Why do we need primitives

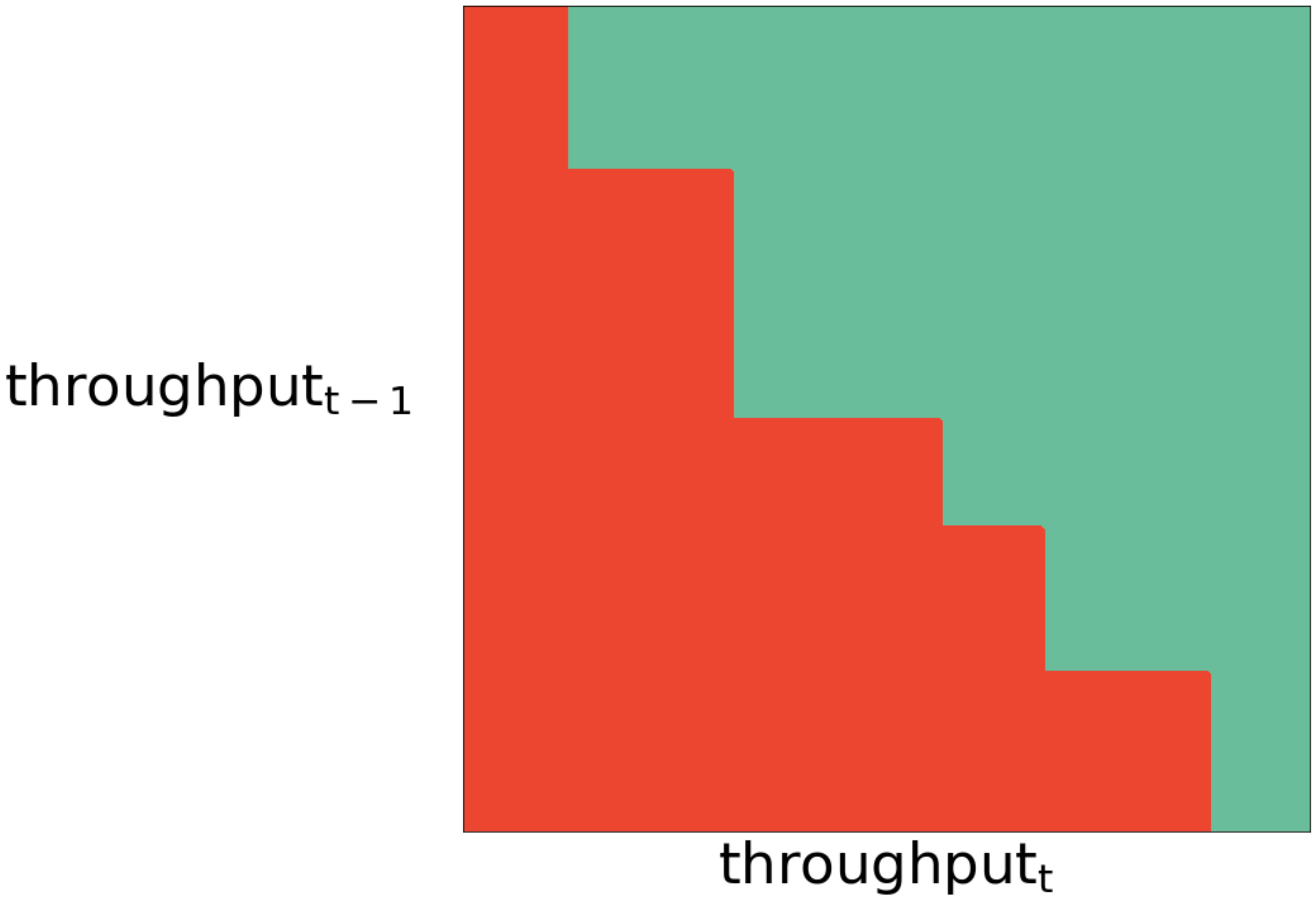


Why do we need primitives



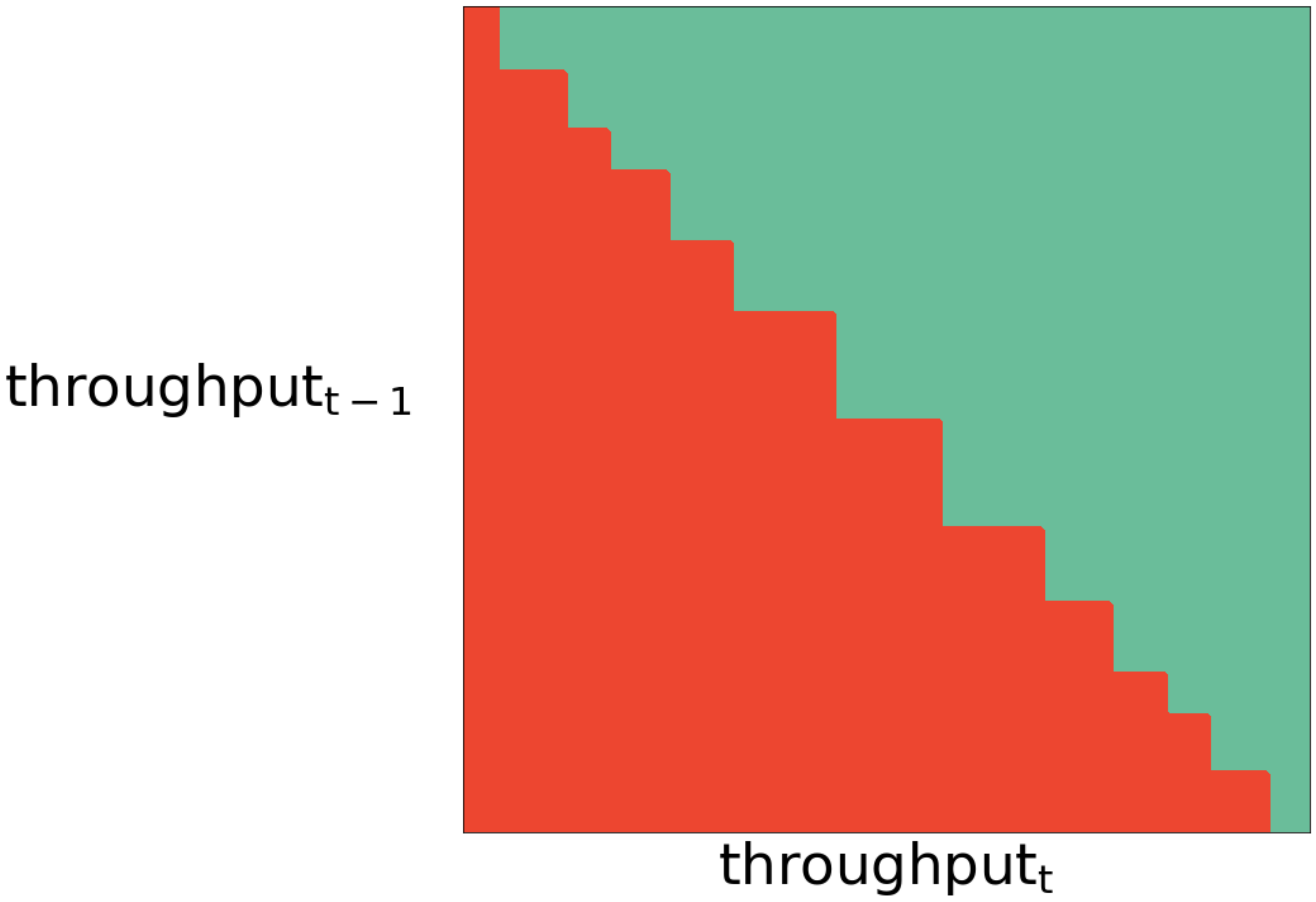
2 Rules

Why do we need primitives



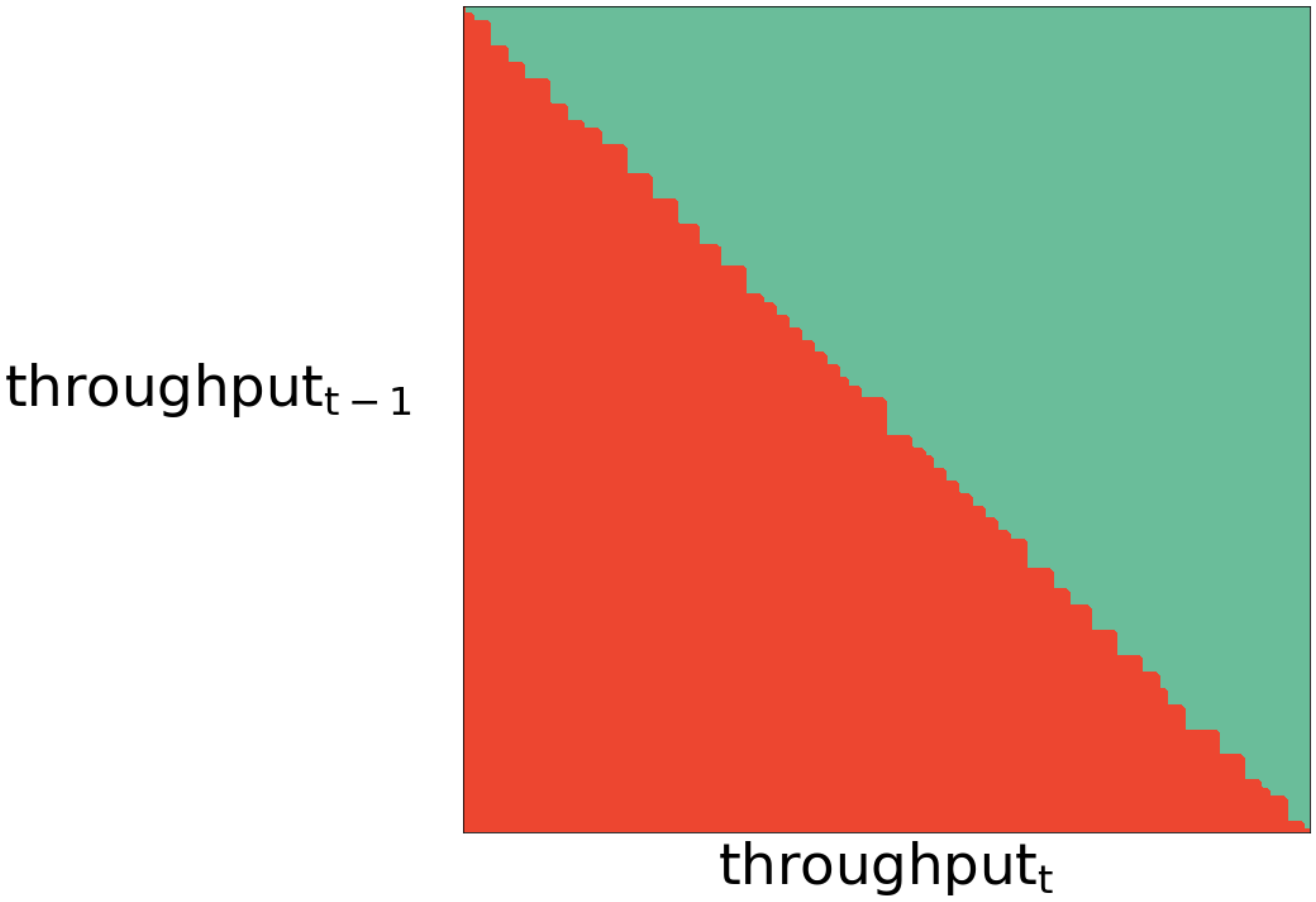
5 Rules

Why do we need primitives



25 Rules

Why do we need primitives



100 Rules

Complex ABR in code

```
def choose_next_quality(throughput_history, buffer_size):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        download_time = chunk_size_high / throughput_estimate
        if download_time < buffer_size:
            next_quality = ⬇⬇

        else:
            next_quality = ⬆⬆

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

Complex ABR in code

```
def choose_next_quality(throughput_history, buffer_size):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        download_time = chunk_size_high / throughput_estimate
        if download_time < buffer_size:
            next_quality = ⬇⬇

        else:
            next_quality = ⬆⬆

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

Complex ABR in code

```
def choose_next_quality(throughput_history, buffer_size):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        download_time = chunk_size_high / throughput_estimate
        if download_time < buffer_size:
            next_quality = ⬇⬇

        else:
            next_quality = ⬆⬆

    else:  ## throughput_estimate == high
        next_quality = ⬆⬆

    return next_quality
```

Number of primitives

Number of primitives

Features

Number of primitives

Features X Lookback

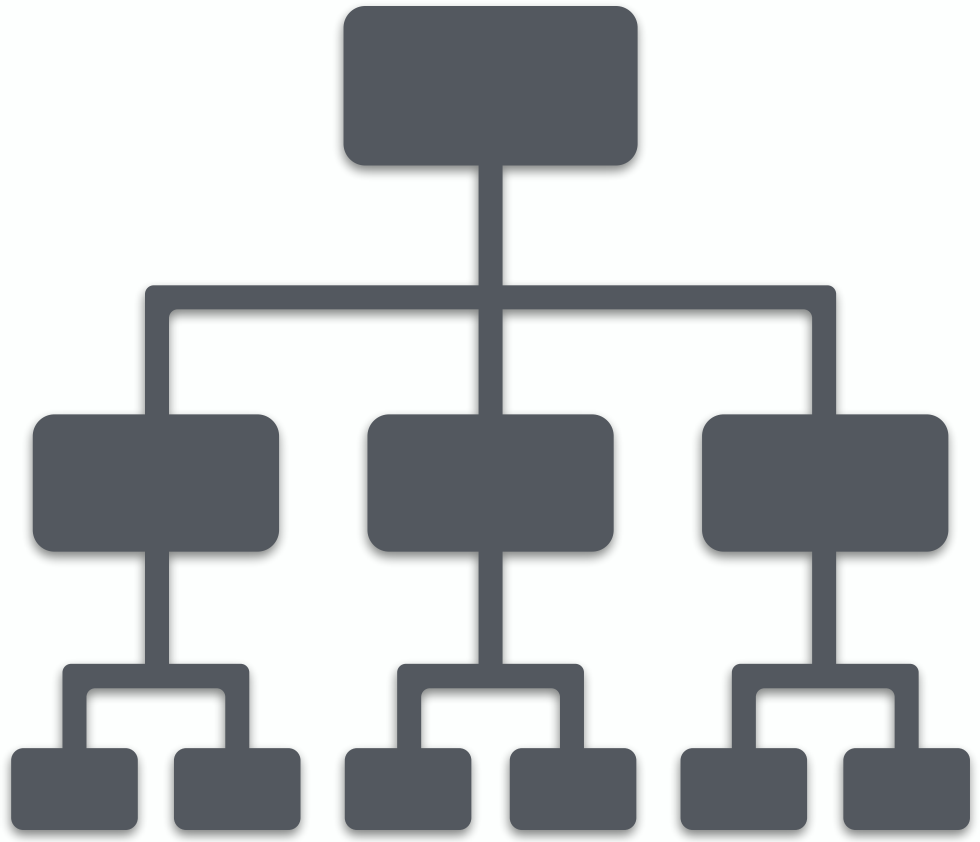
Number of primitives

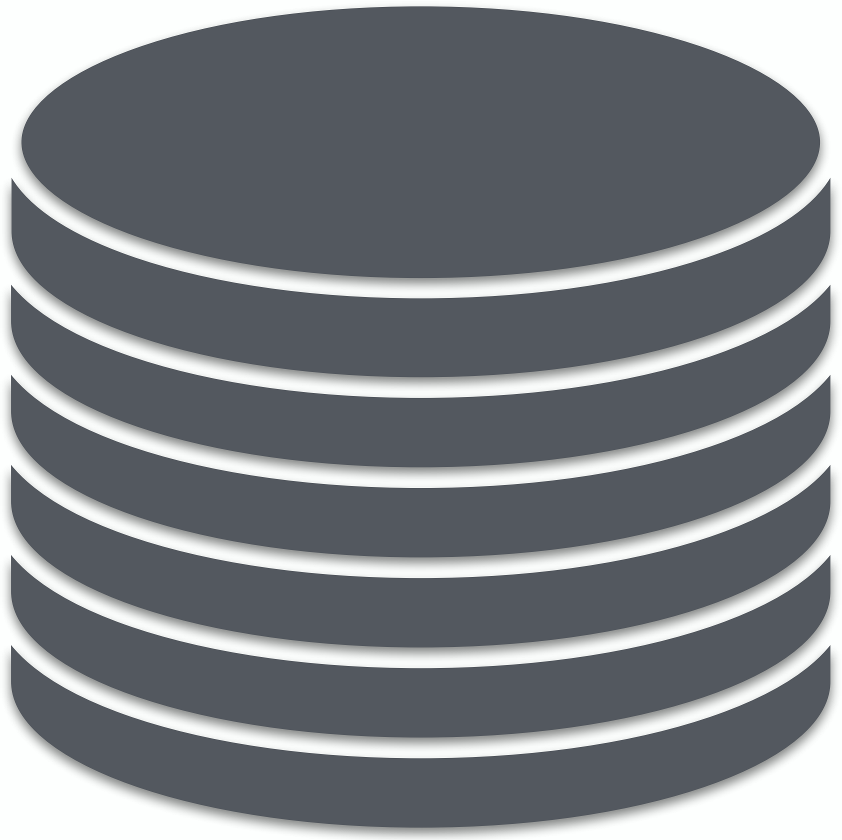
Features X Lookback X Actions

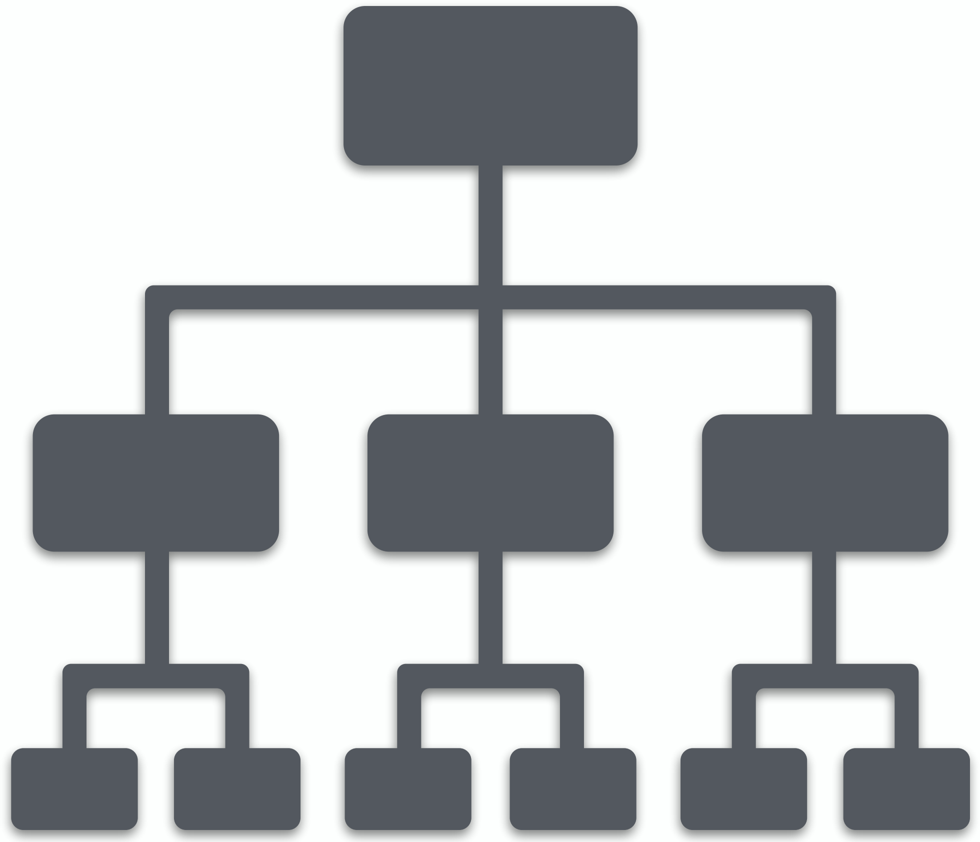
Number of primitives

Features X Lookback X Actions

~ 1300 Primitives







Services of interest

Provider	Description	Alexa Rank

Services of interest

Provider	Description	Alexa Rank
YouTube	Broad coverage	2, Global

Services of interest

Provider	Description	Alexa Rank
YouTube	Broad coverage	2, Global
ZDF	German Public Service	47, Germany

Services of interest

Provider	Description	Alexa Rank
YouTube	Broad coverage	2, Global
ZDF	German Public Service	47, Germany
Pornhub	Pornographic video sharing website	46, Global

Services of interest

Provider	Description	Alexa Rank
YouTube	Broad coverage	2, Global
ZDF	German Public Service	47, Germany
Pornhub	Pornographic video sharing website	46, Global
Arte	French-German,cultural	270,France
Fandom	Gaming, pop-culture	91, Global
SRF	Swiss Public Service	45, Switzerland
TubiTV	Movies and series of all genres	1330, USA
Twitch	Live and VoD streaming service, gaming	39, Global
Vimeo	Artistic content	188, Global
XVideos	Pornographic video sharing website	67, Global

Inference features

Inference features

Providers

Inference features

Providers X Net-traces

Inference features

Providers X Net-traces X Videos

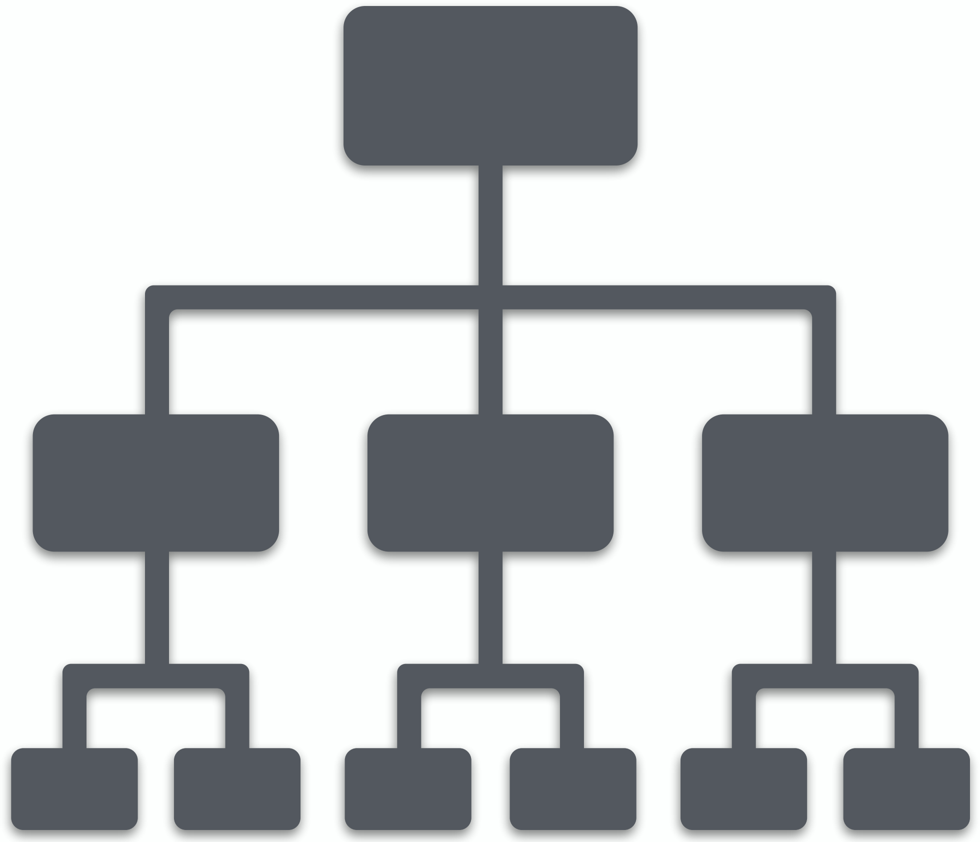
Inference features

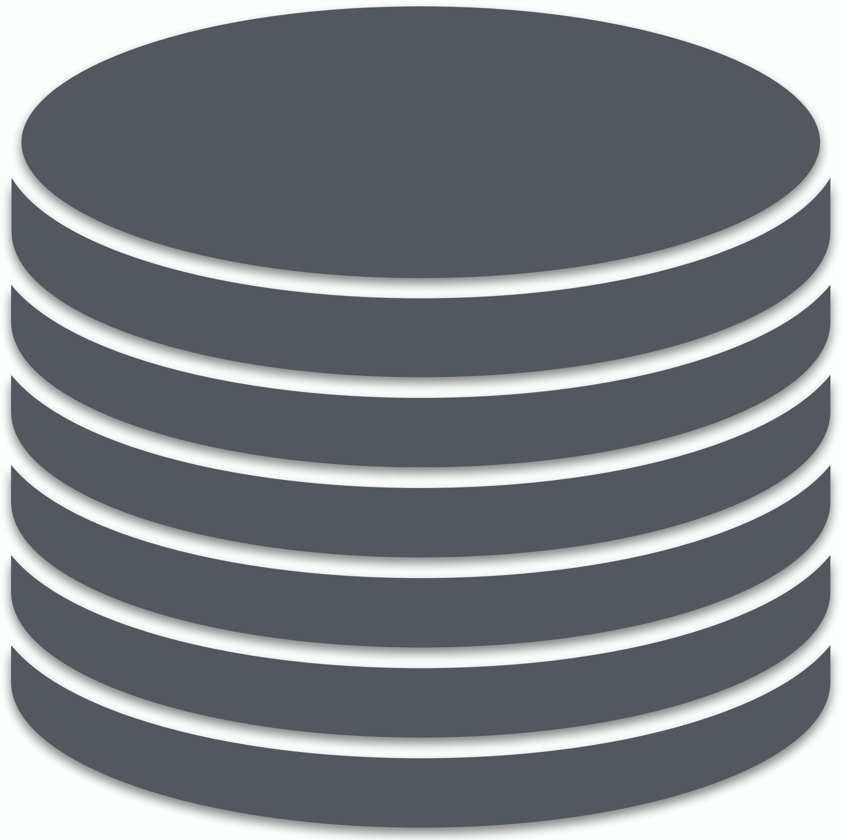
Providers X Net-traces X Videos X Length

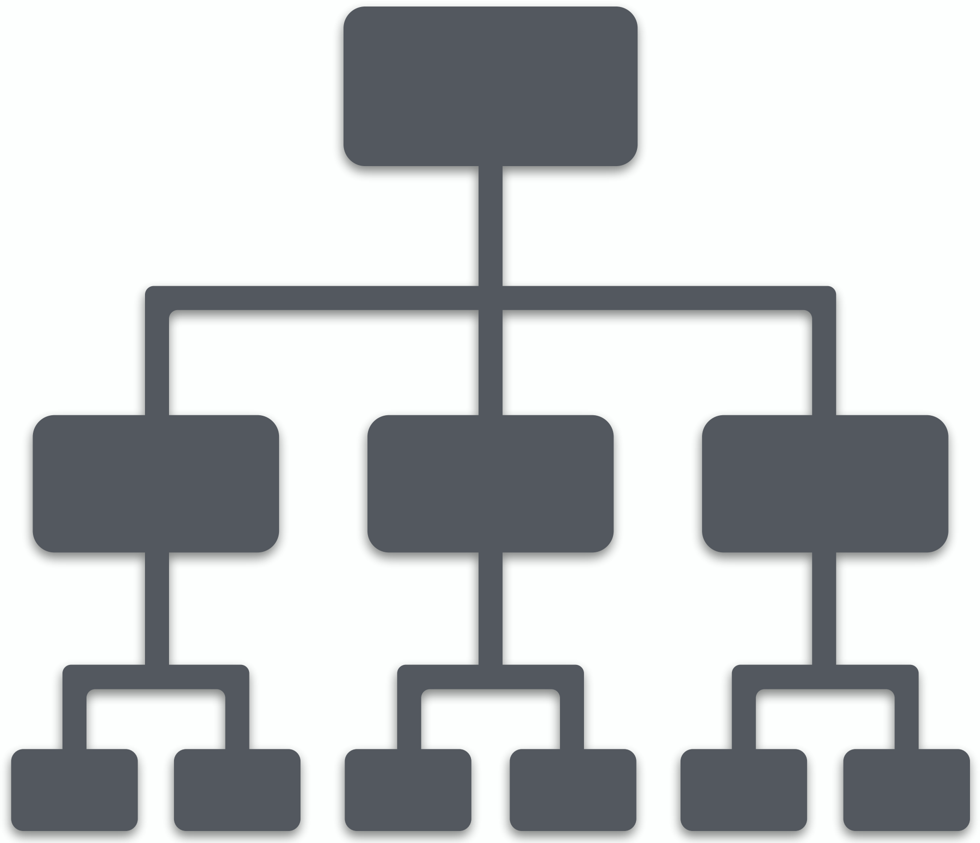
Inference features

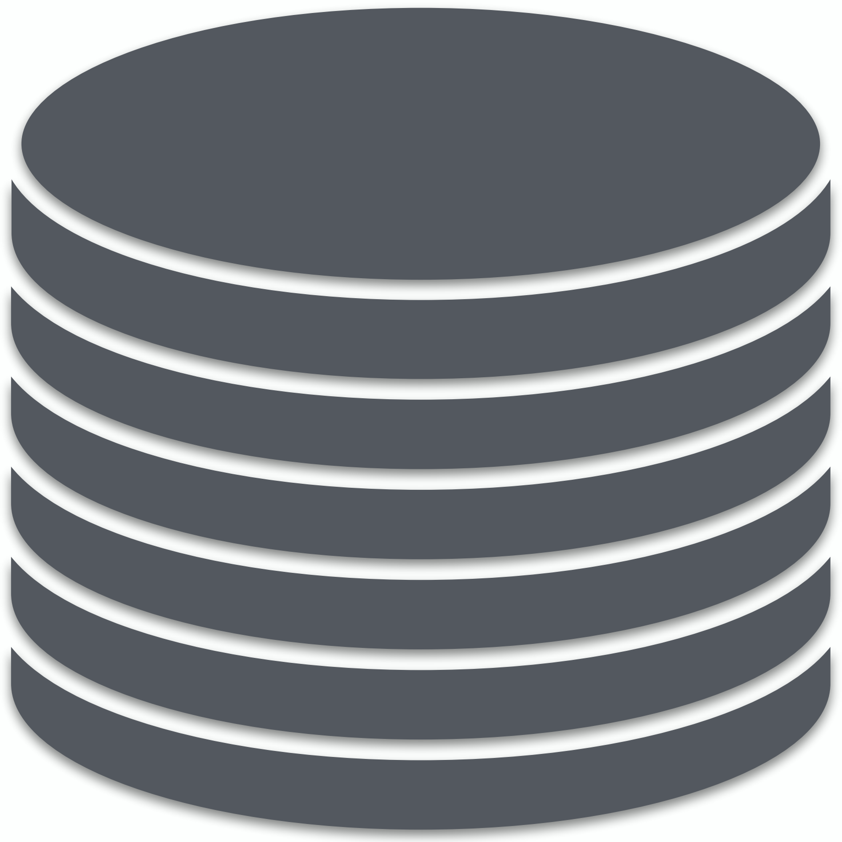
Providers X Net-traces X Videos X Length

~ 9 hours of training/testing streaming time





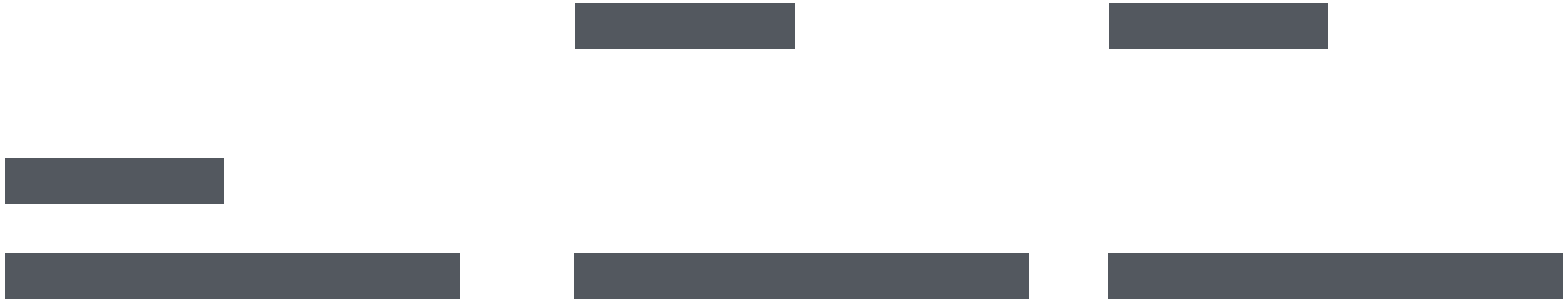




Why is agreement not sufficient

Why is agreement not sufficient

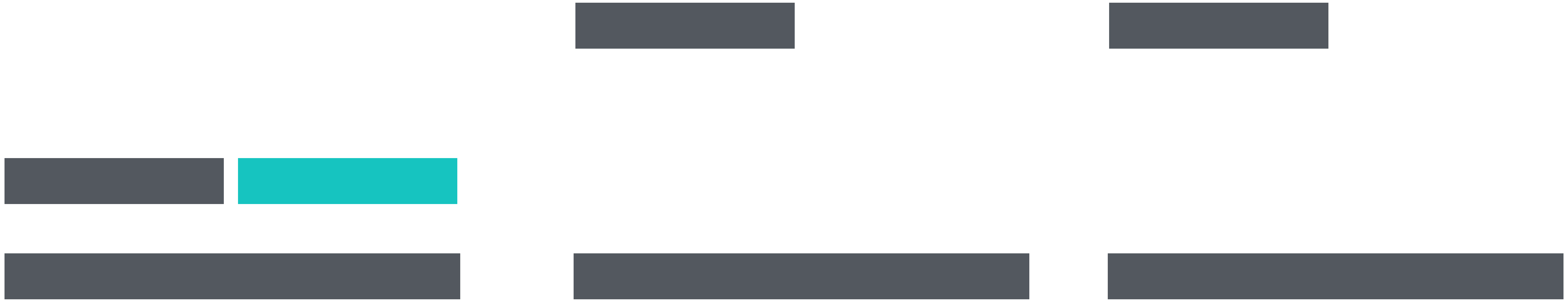
[Video from YouTube]



Why is agreement not sufficient

Played

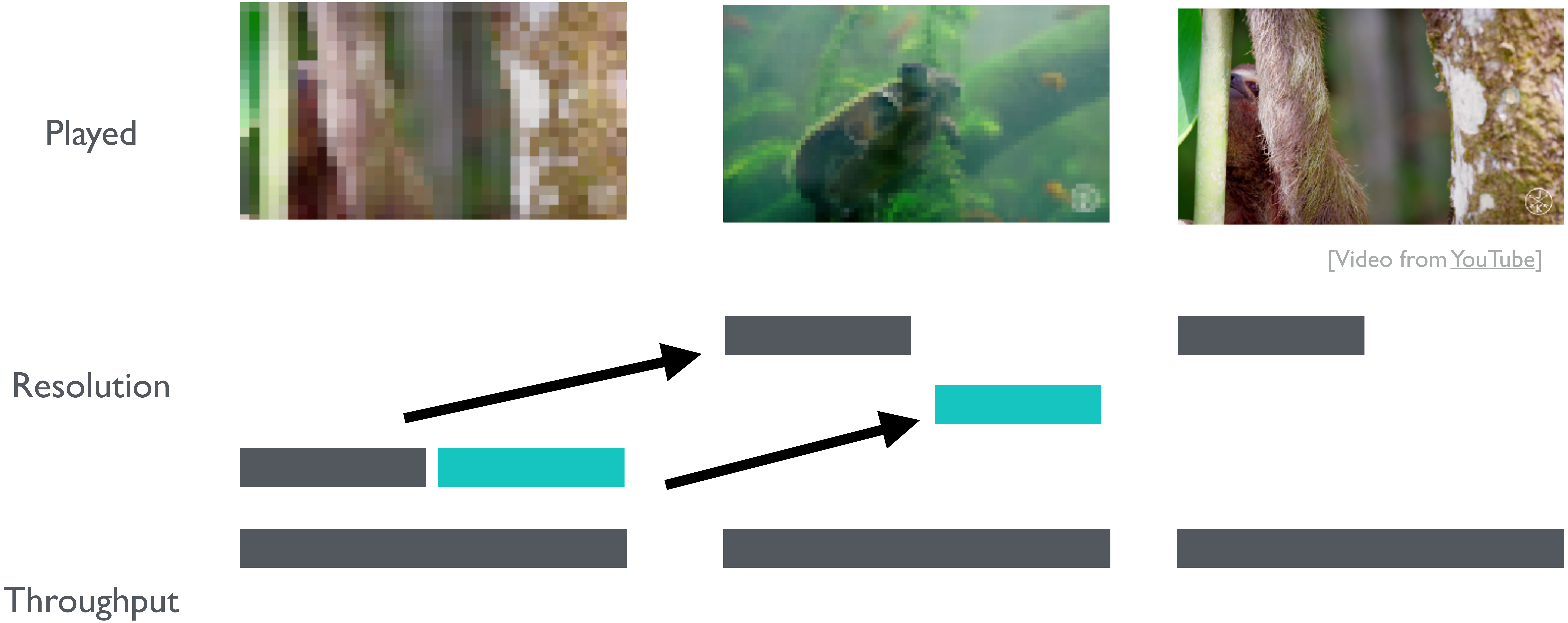
[Video from YouTube]



Resolution

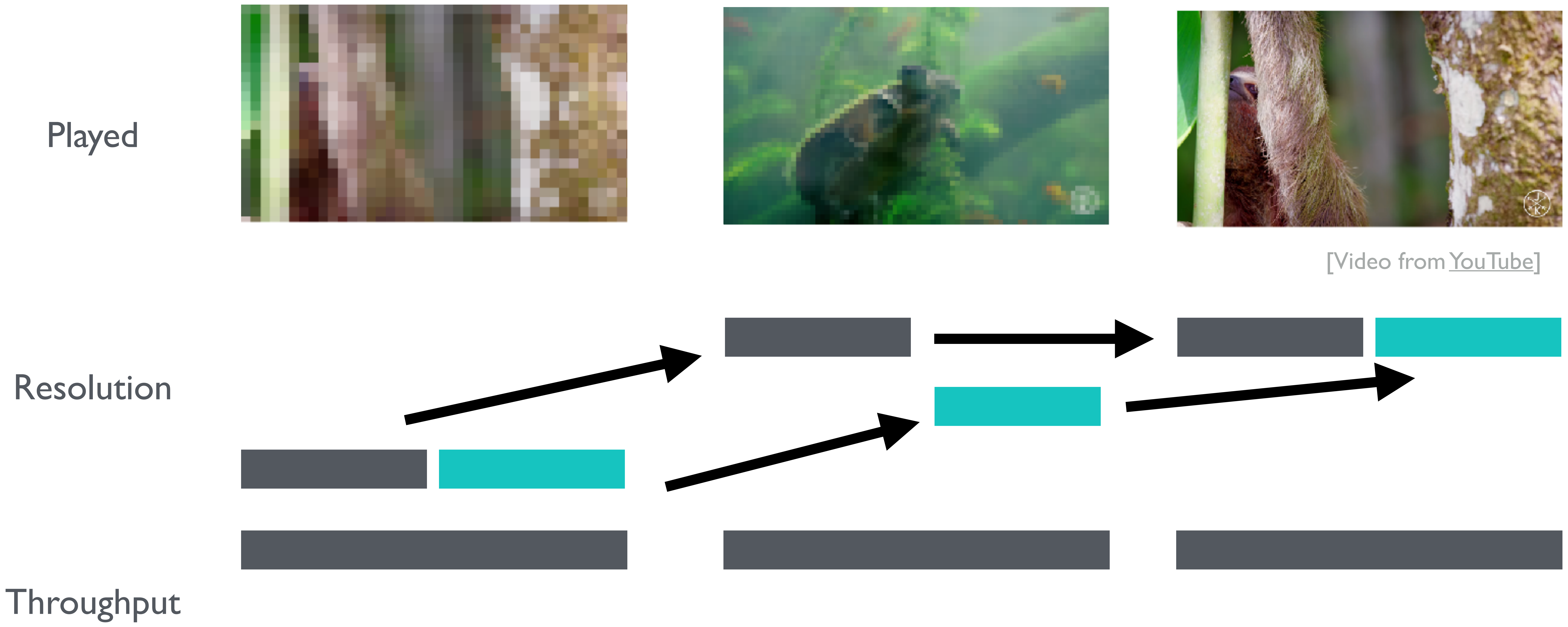
Throughput

Why is agreement not sufficient



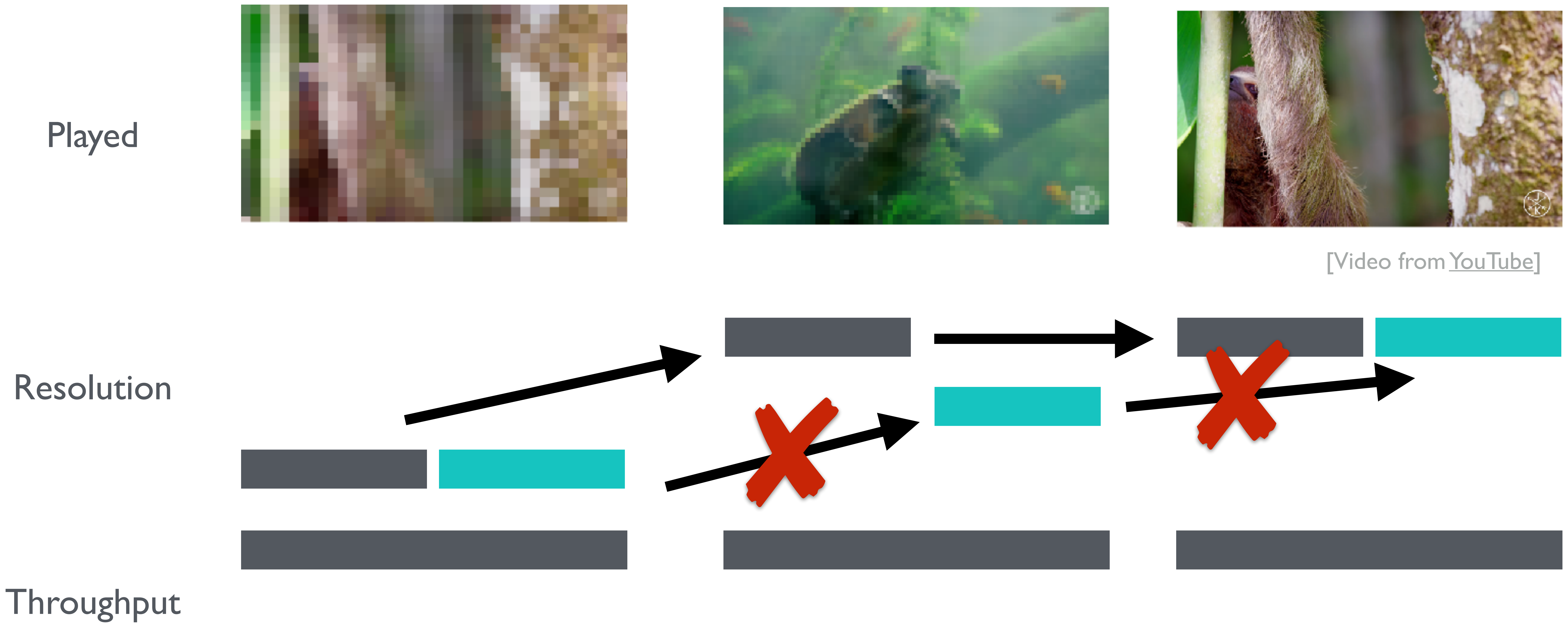
[Video from YouTube]

Why is agreement not sufficient



[Video from YouTube]

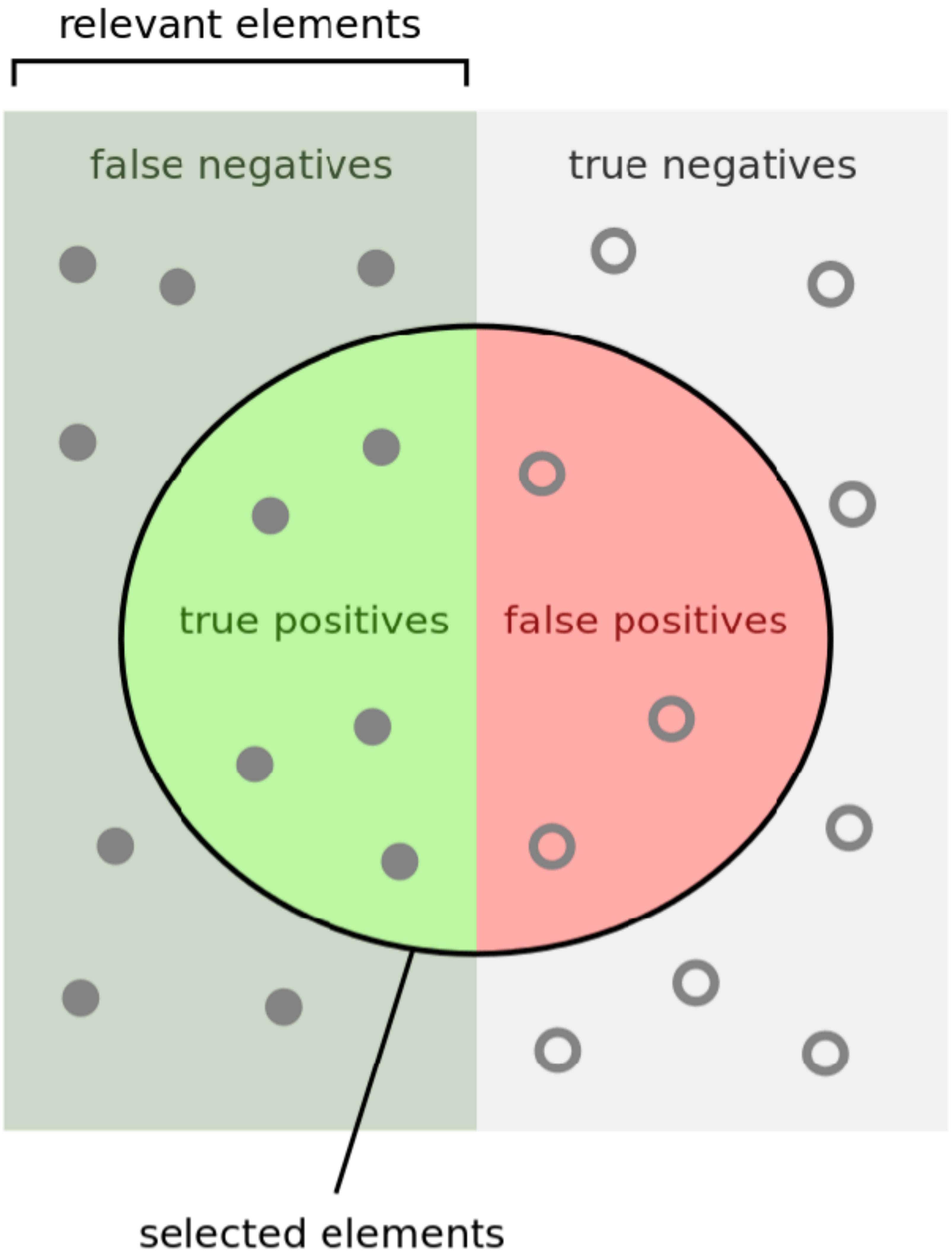
Why is agreement not sufficient



[Video from YouTube]

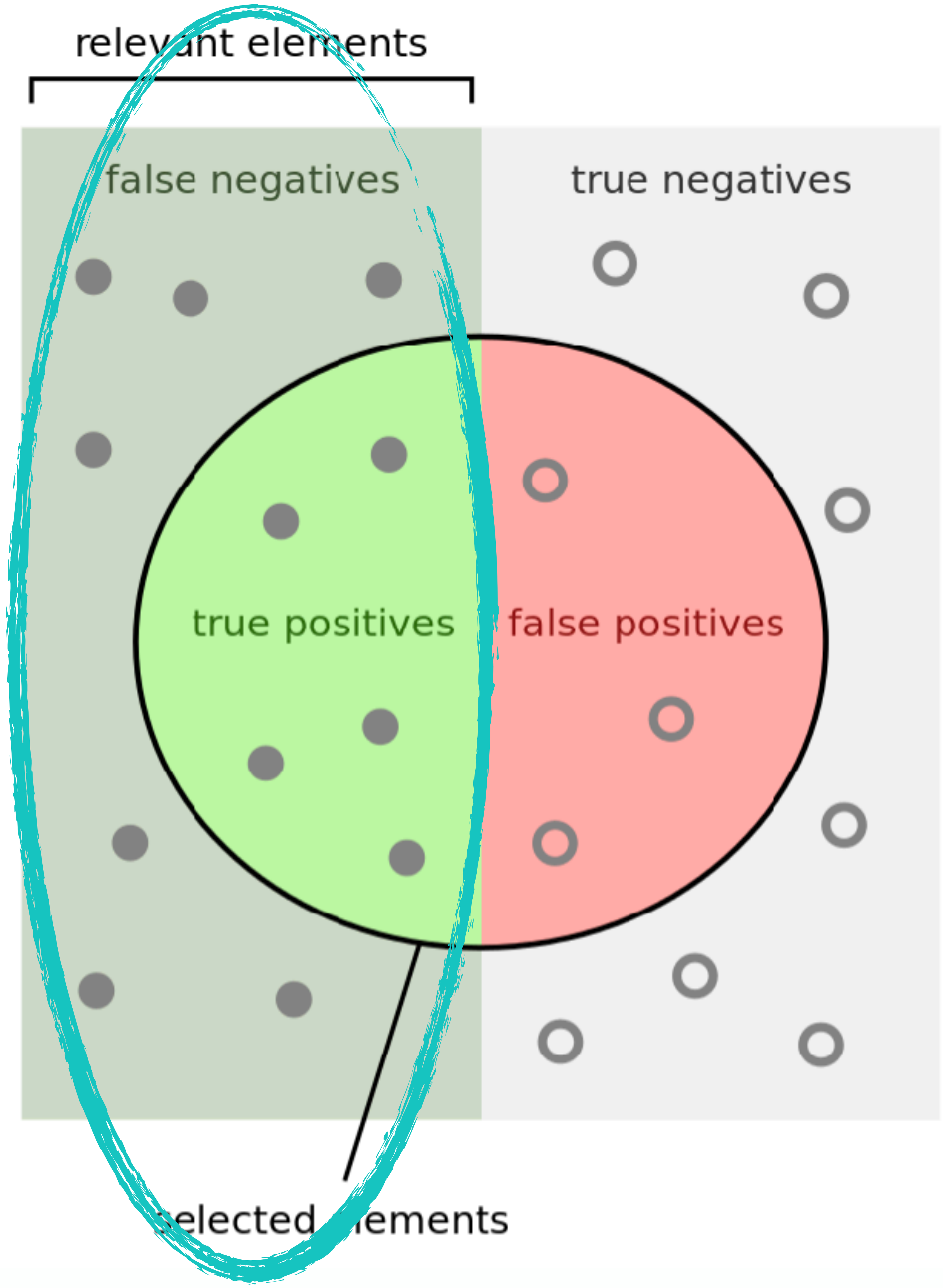
Agreement

Agreement



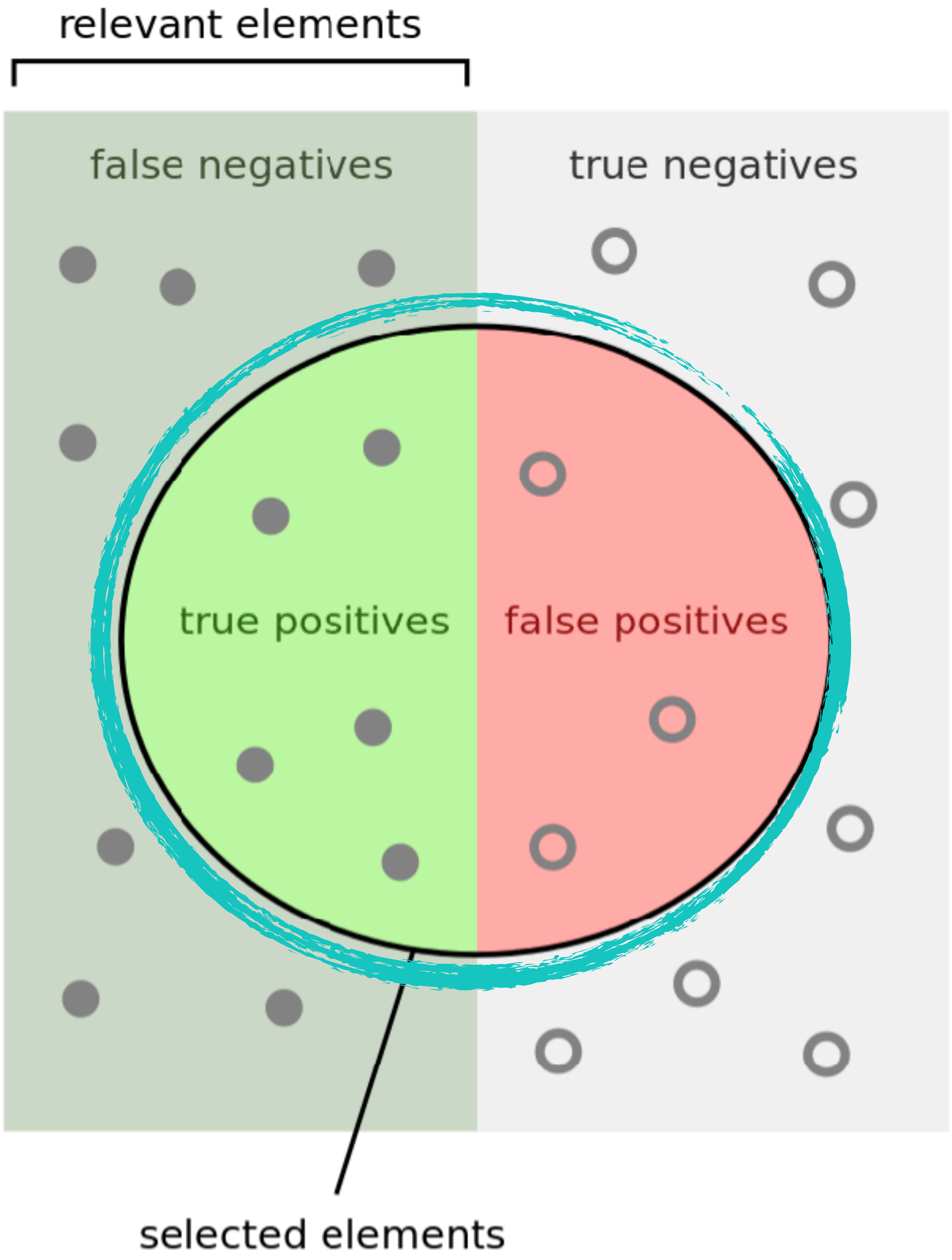
[Wikipedia]

Agreement



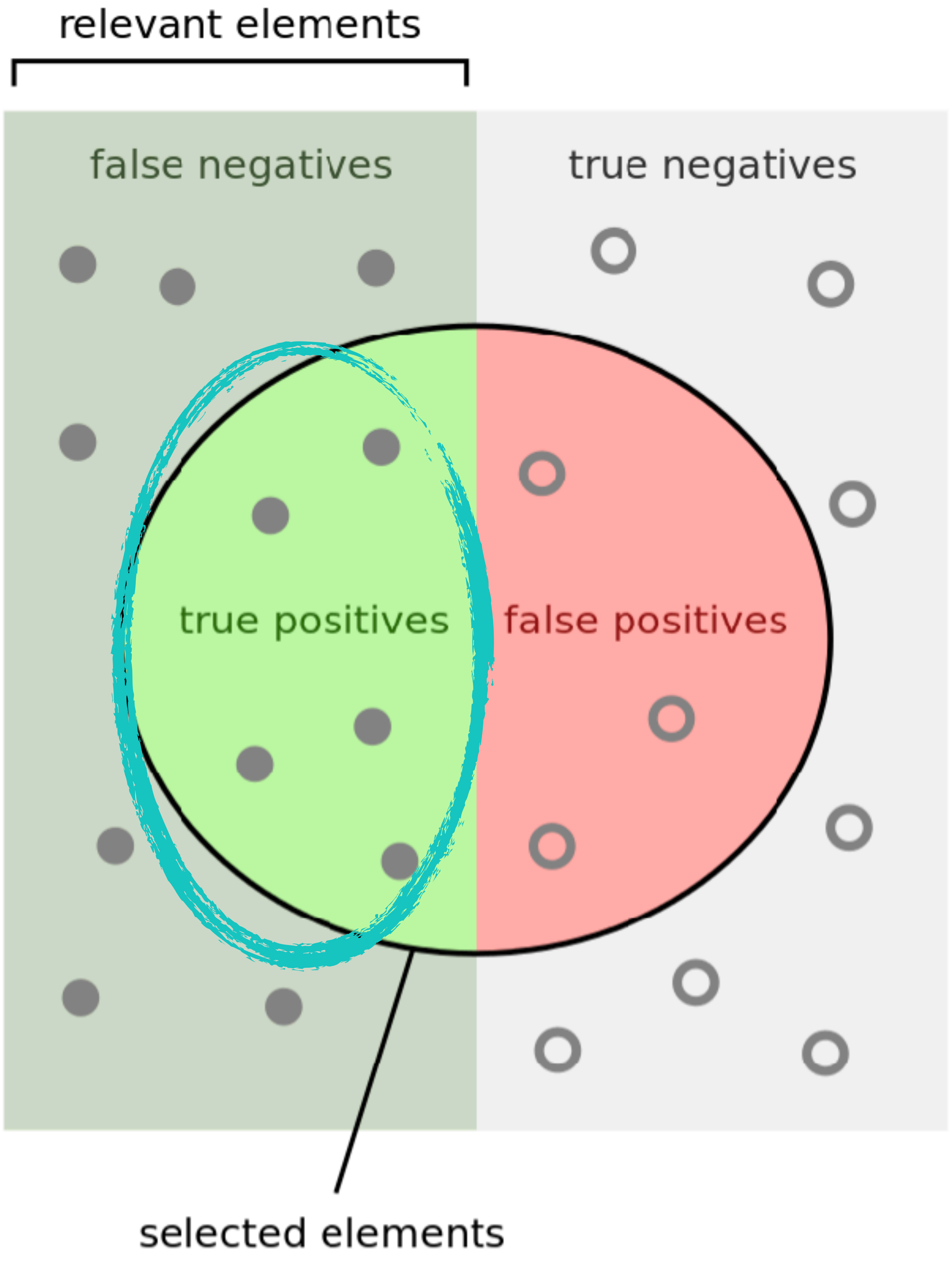
[Wikipedia]

Agreement



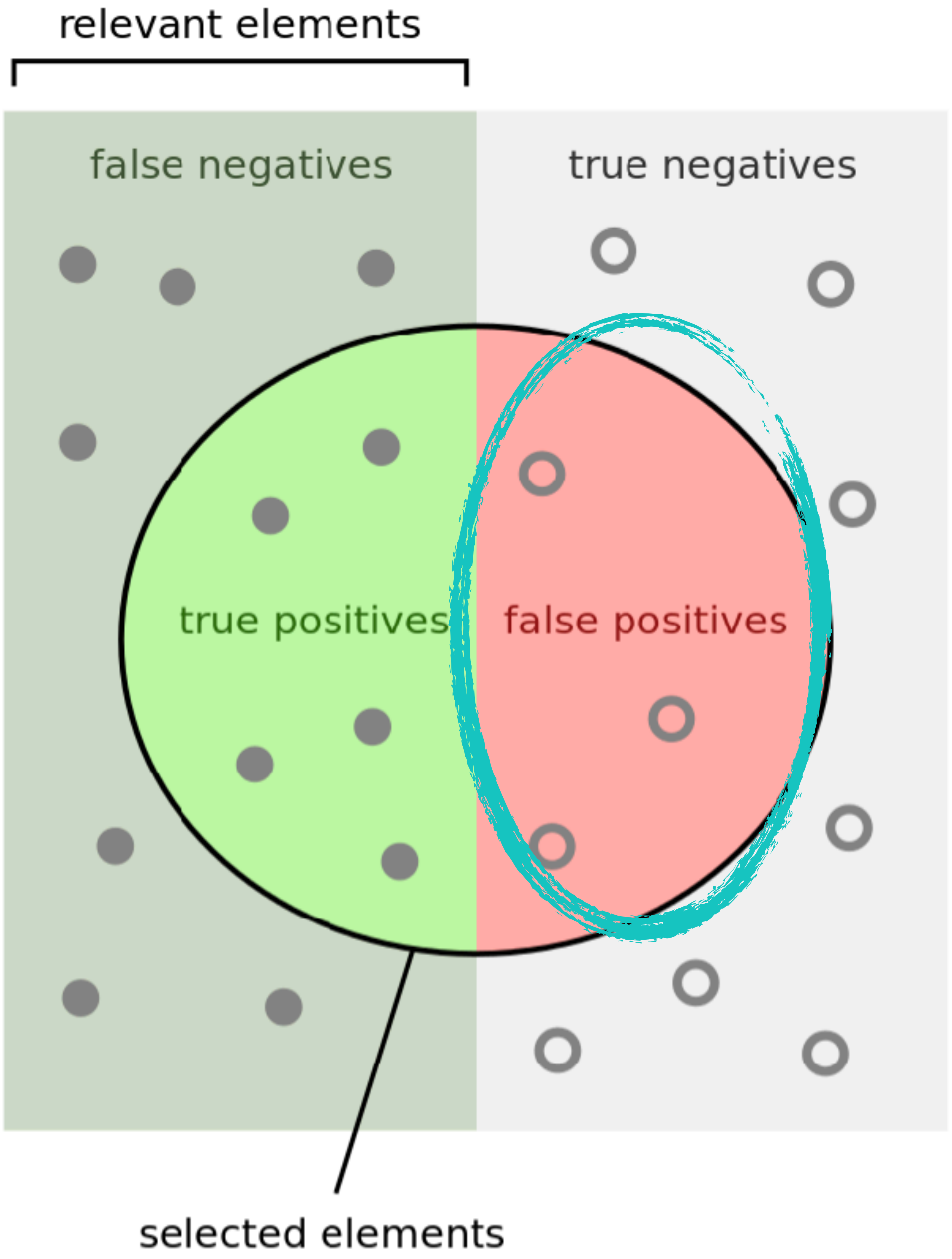
[Wikipedia]

Agreement



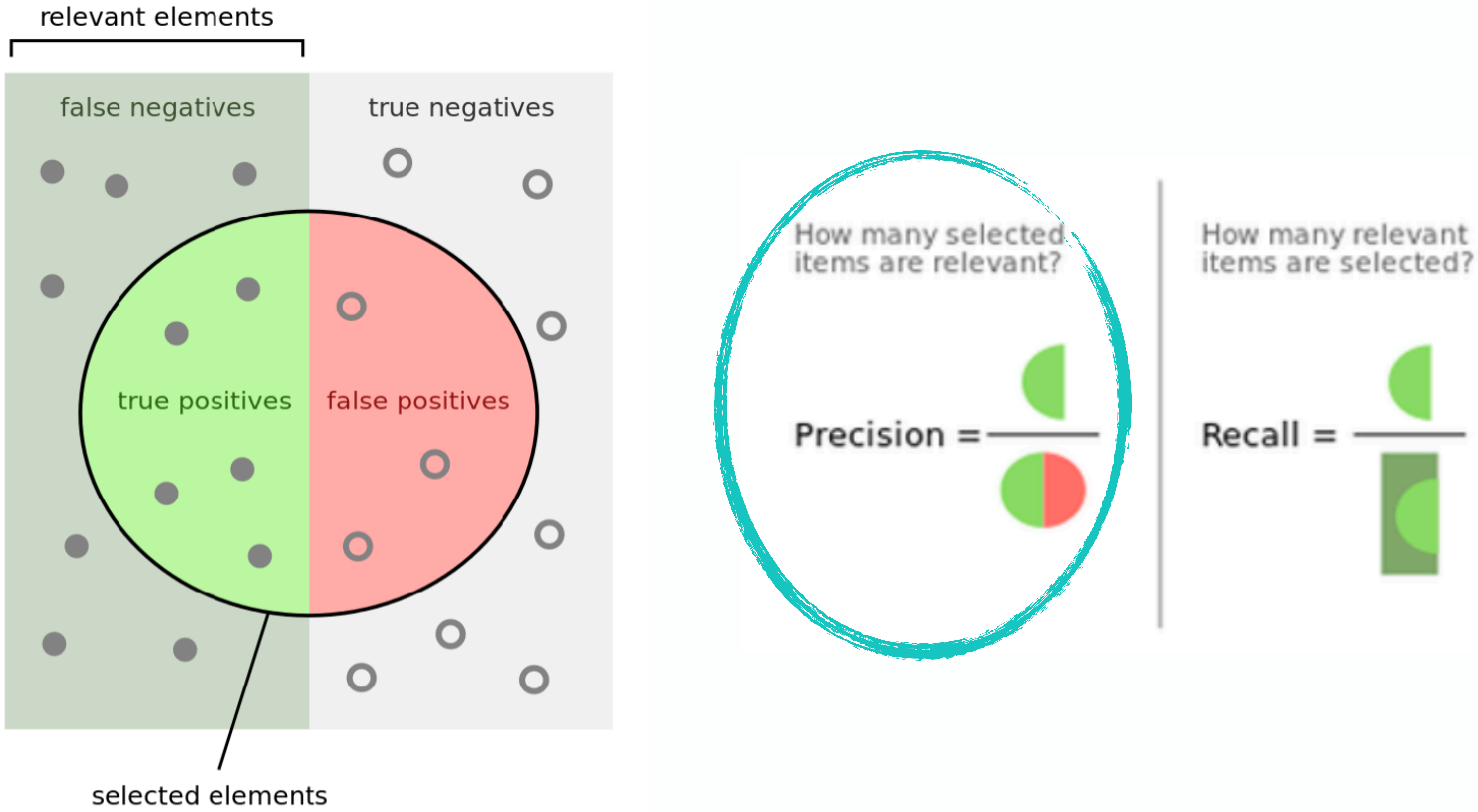
[Wikipedia]

Agreement



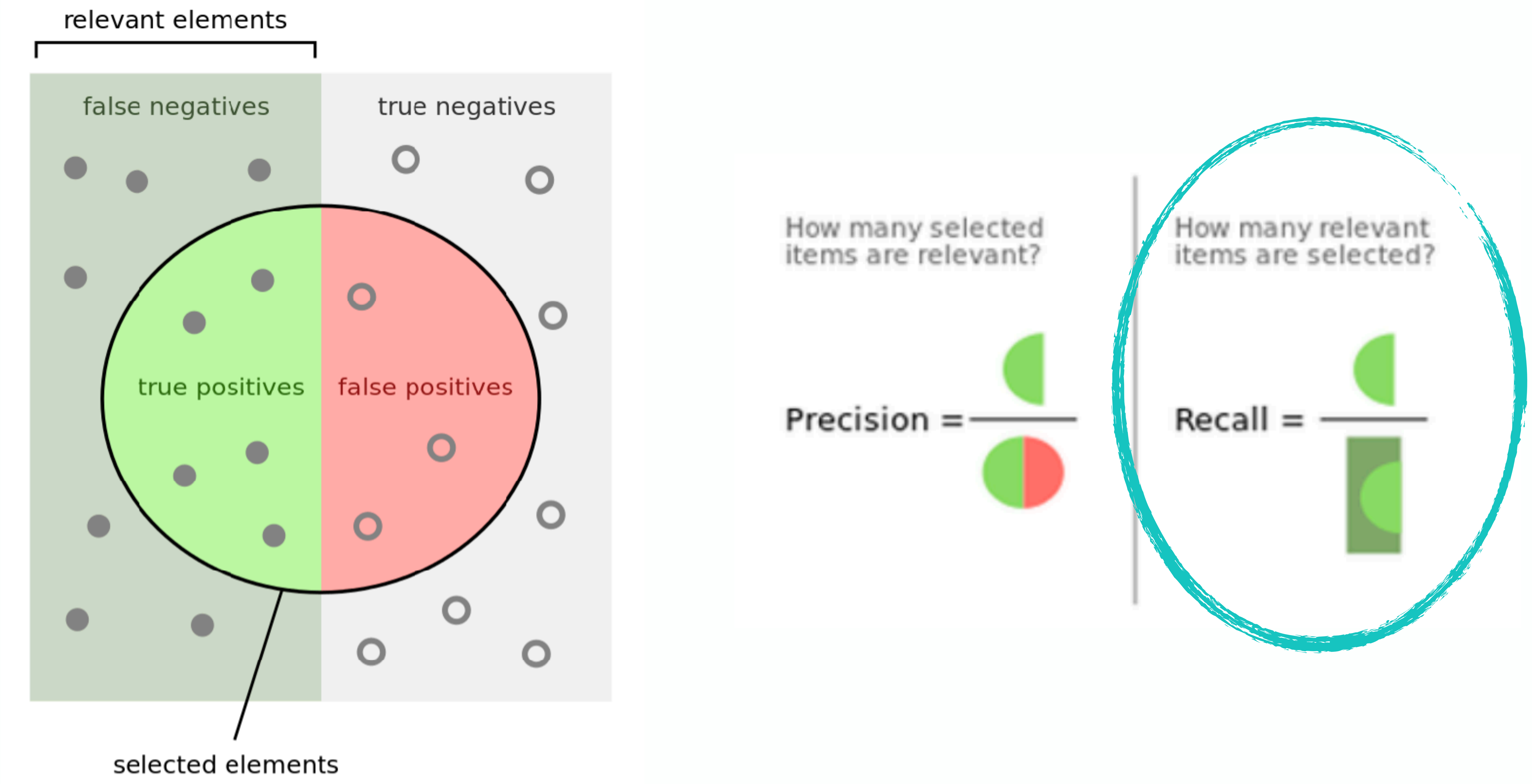
[Wikipedia]

Agreement



[Wikipedia]

Agreement



[Wikipedia]

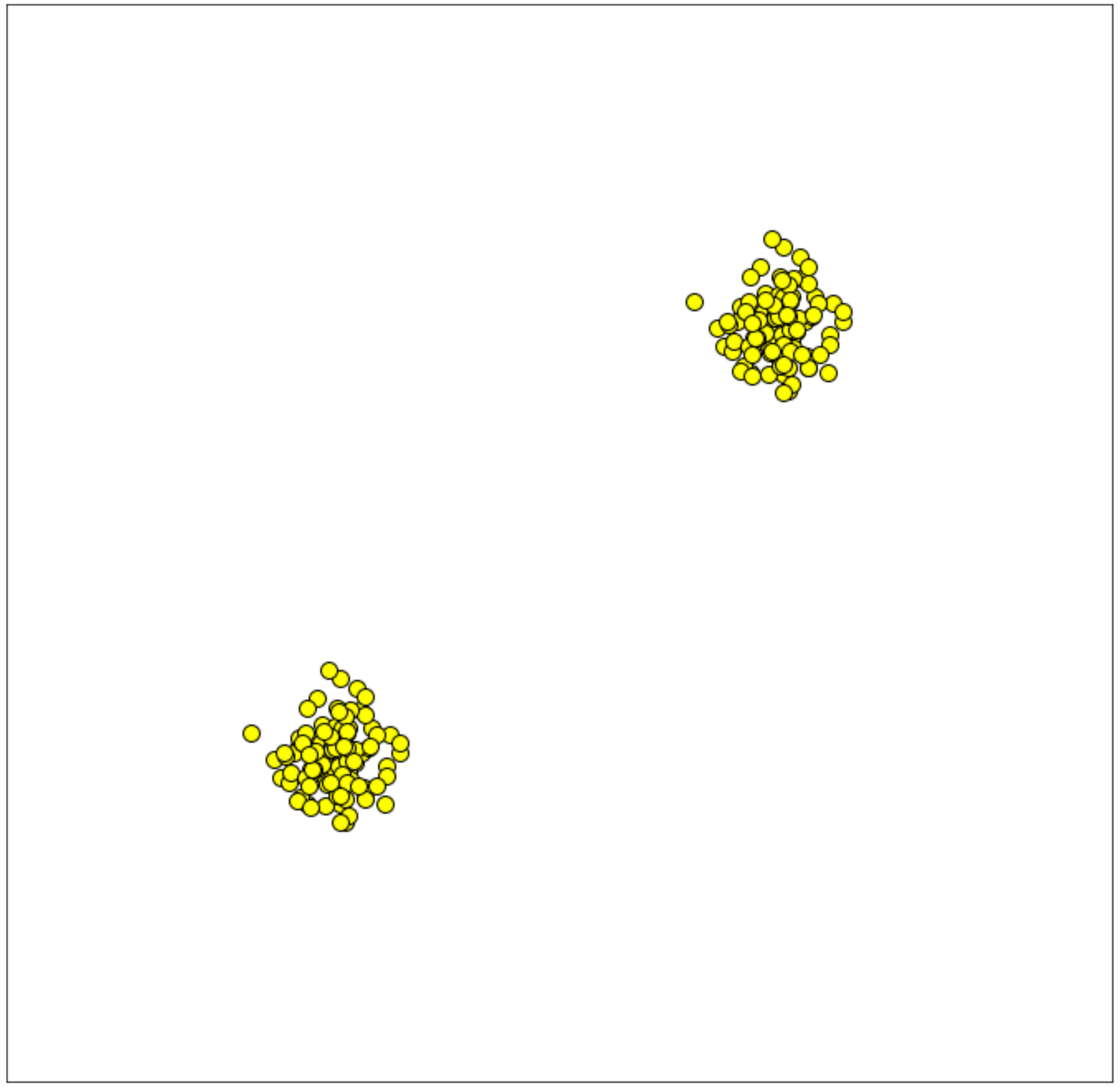
Agreement

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

[Wikipedia]

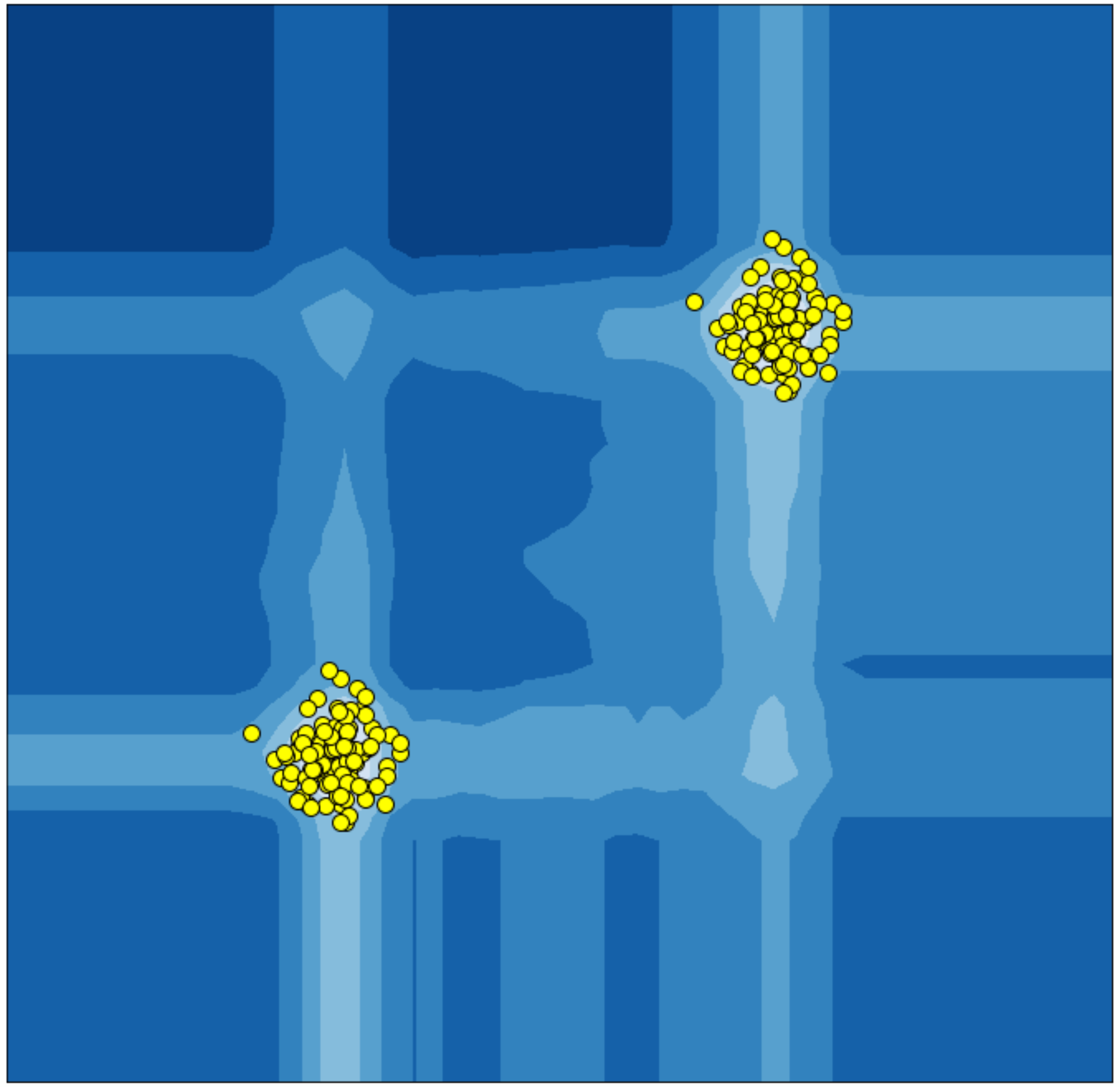
Similarity

Similarity



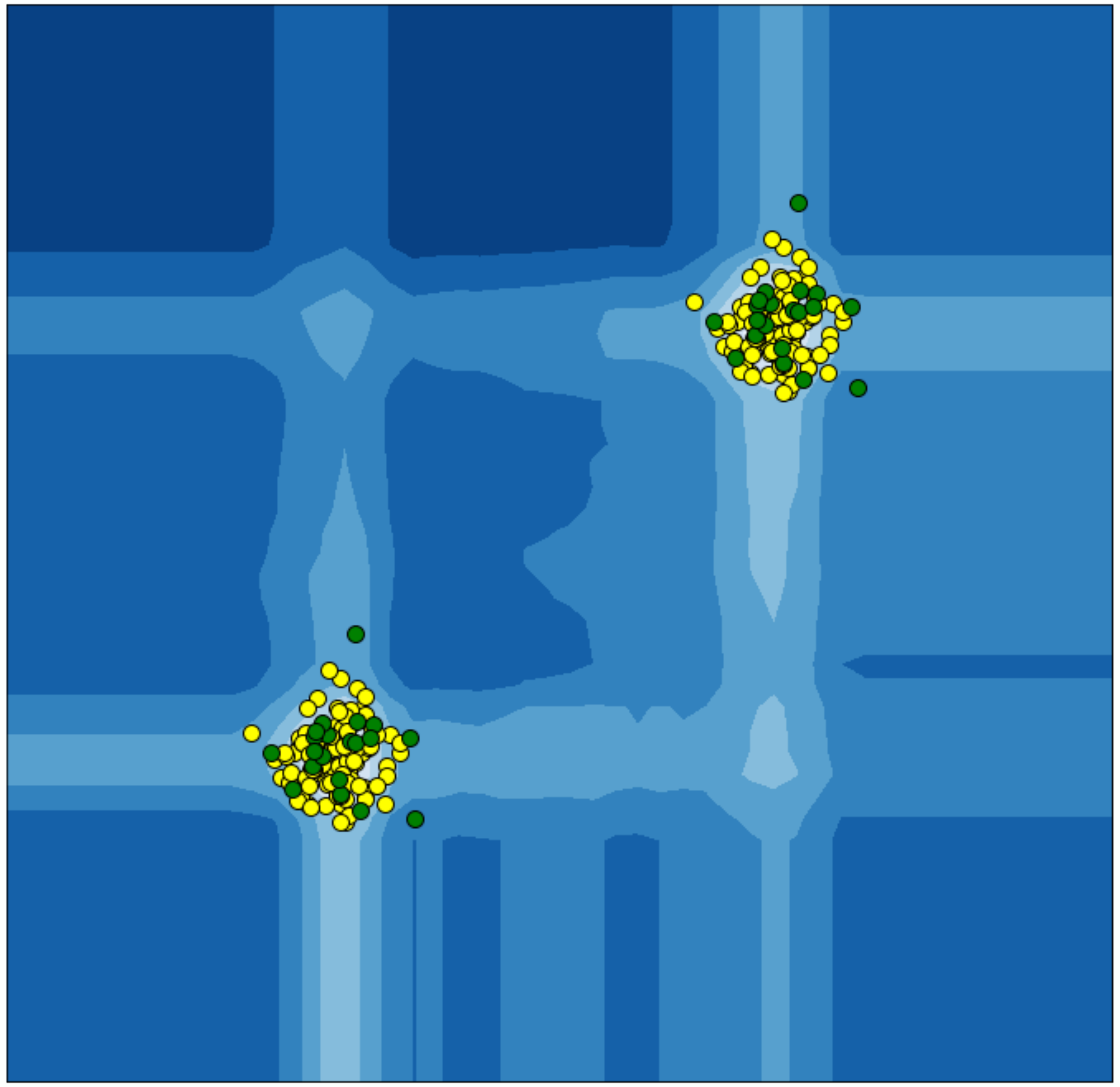
[Scikit-learn]

Similarity



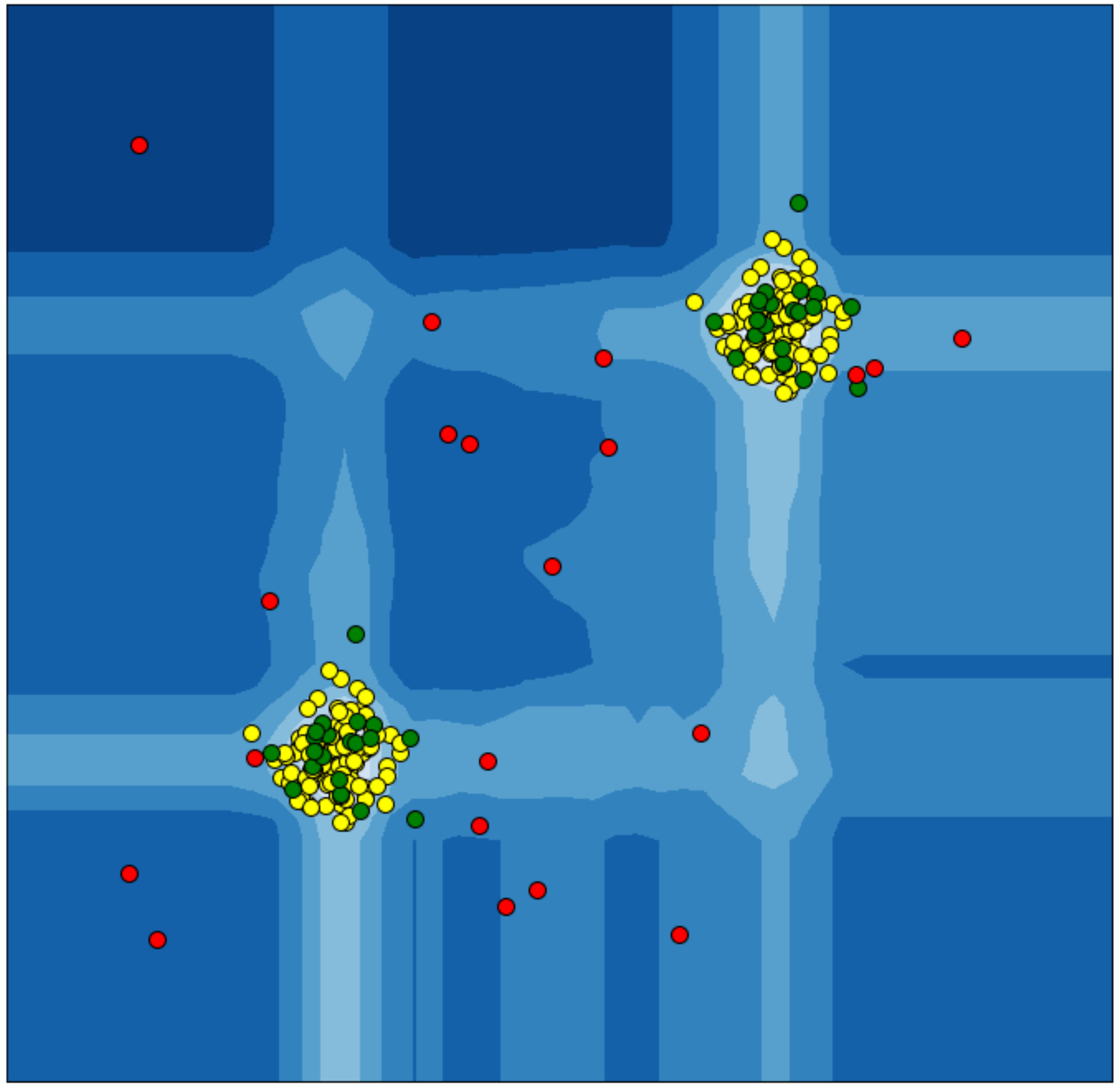
[Scikit-learn]

Similarity



[Scikit-learn]

Similarity



[Scikit-learn]

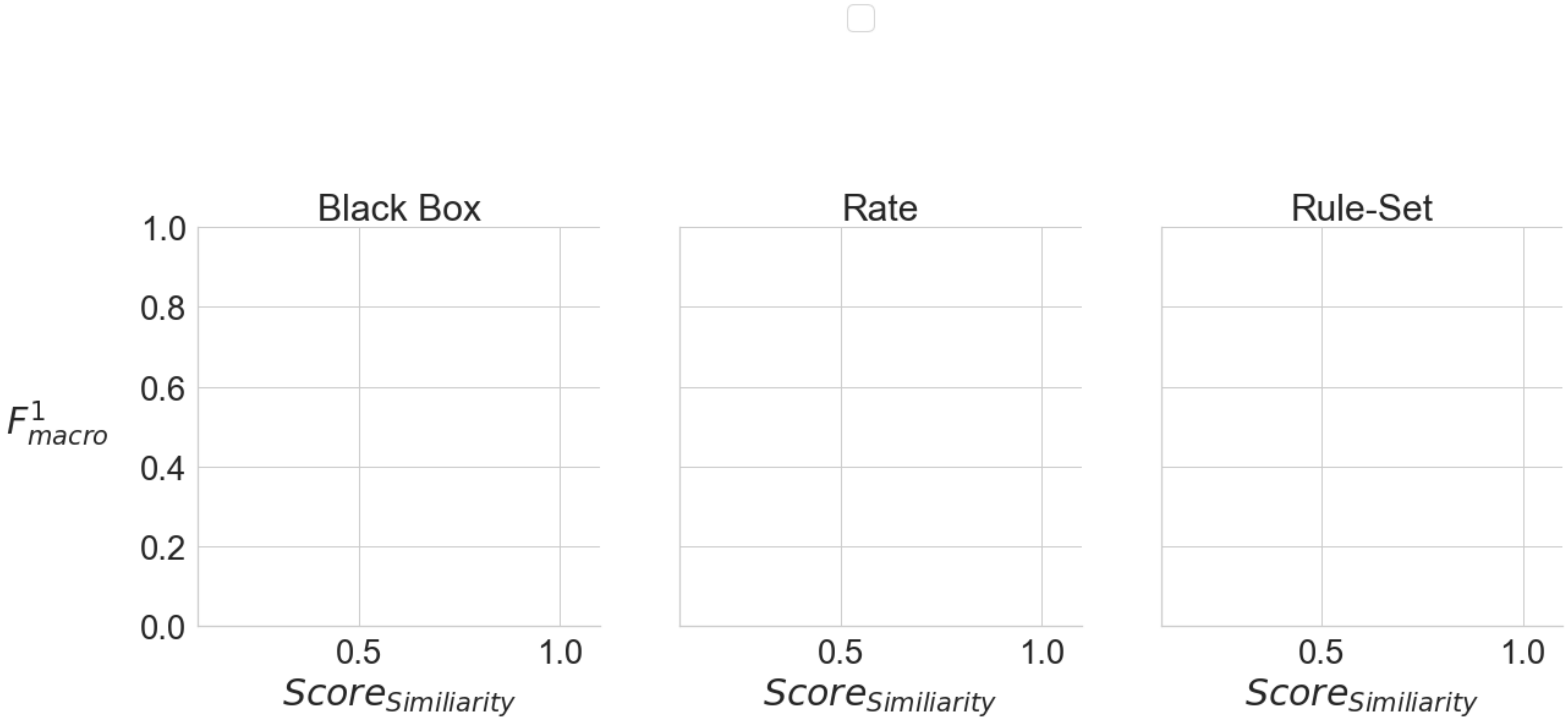
Similarity

$$Score_{Similarity} = \mathscr{P}(\text{Decision from reference distribution})$$

[Scikit-learn]

General metric analysis

General metric analysis

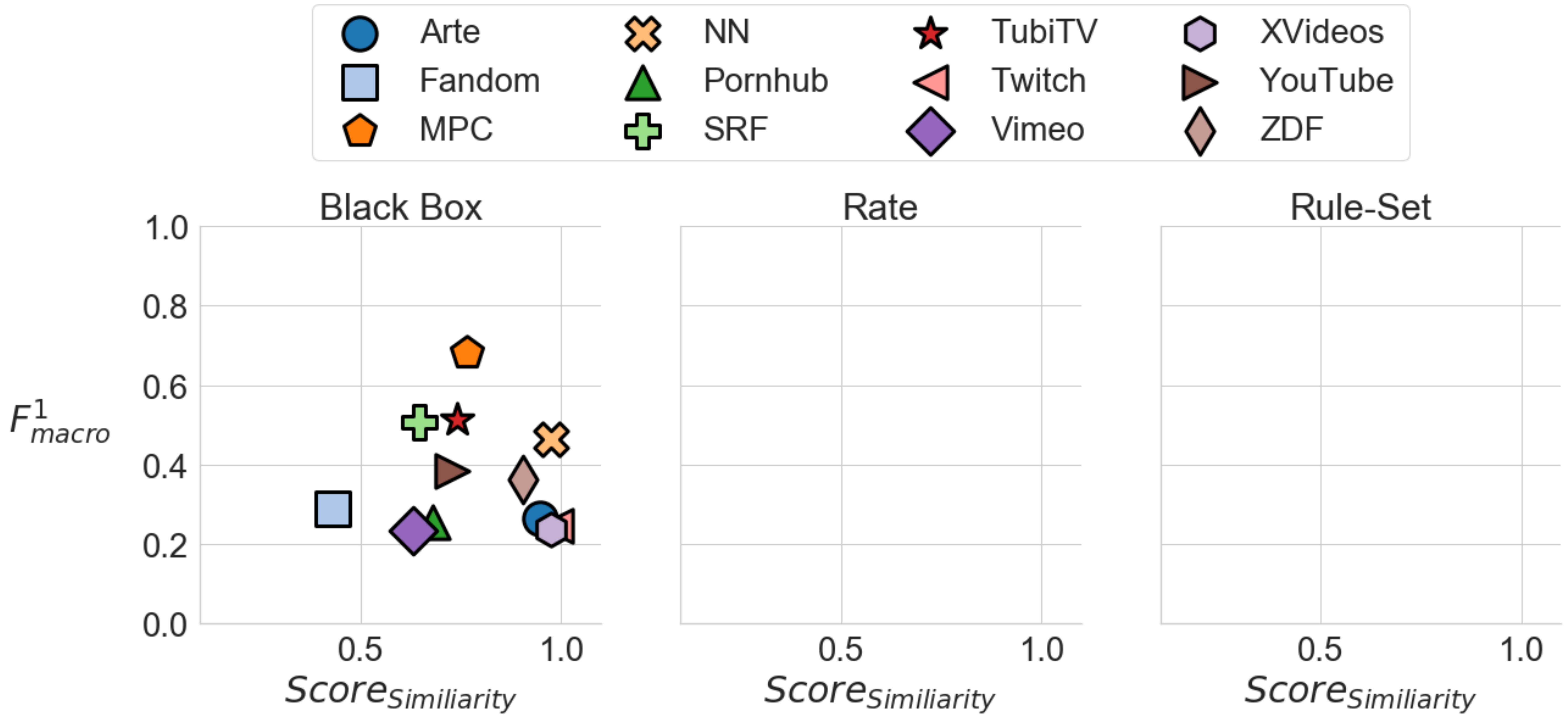


General metric analysis

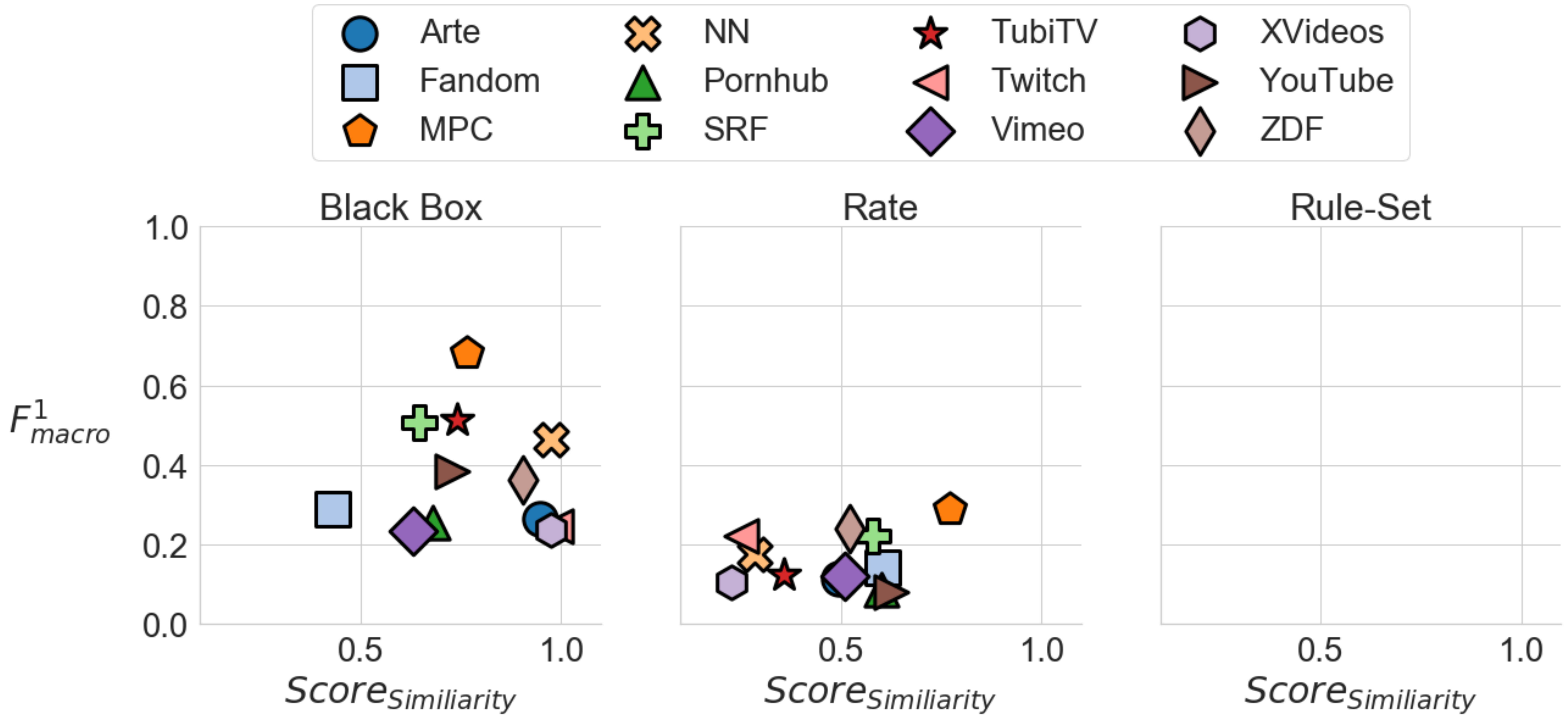
General metric analysis

□
Black Box
Rate
Rule-Set
Best
Worst
F^1_{macro}
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
$Score_{Similiarity}$

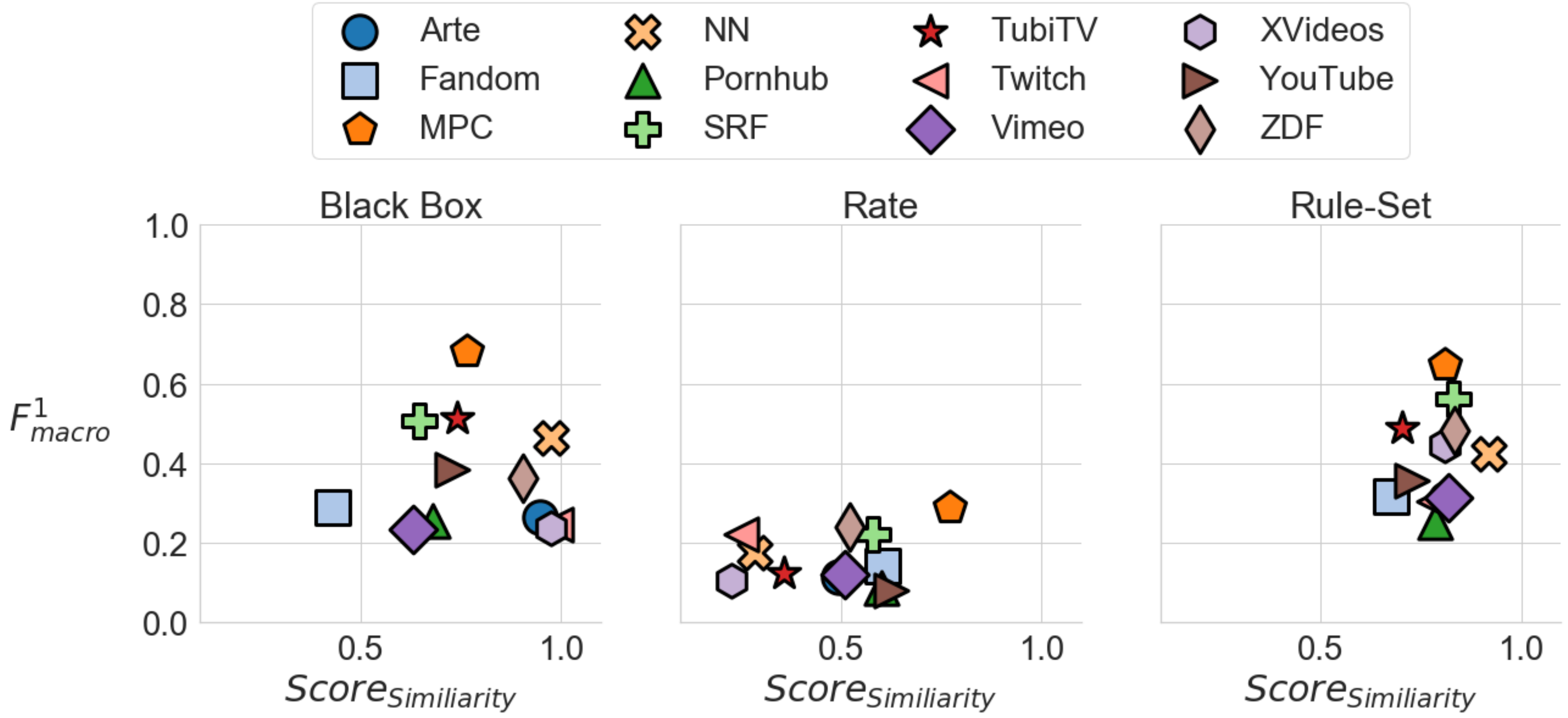
General metric analysis



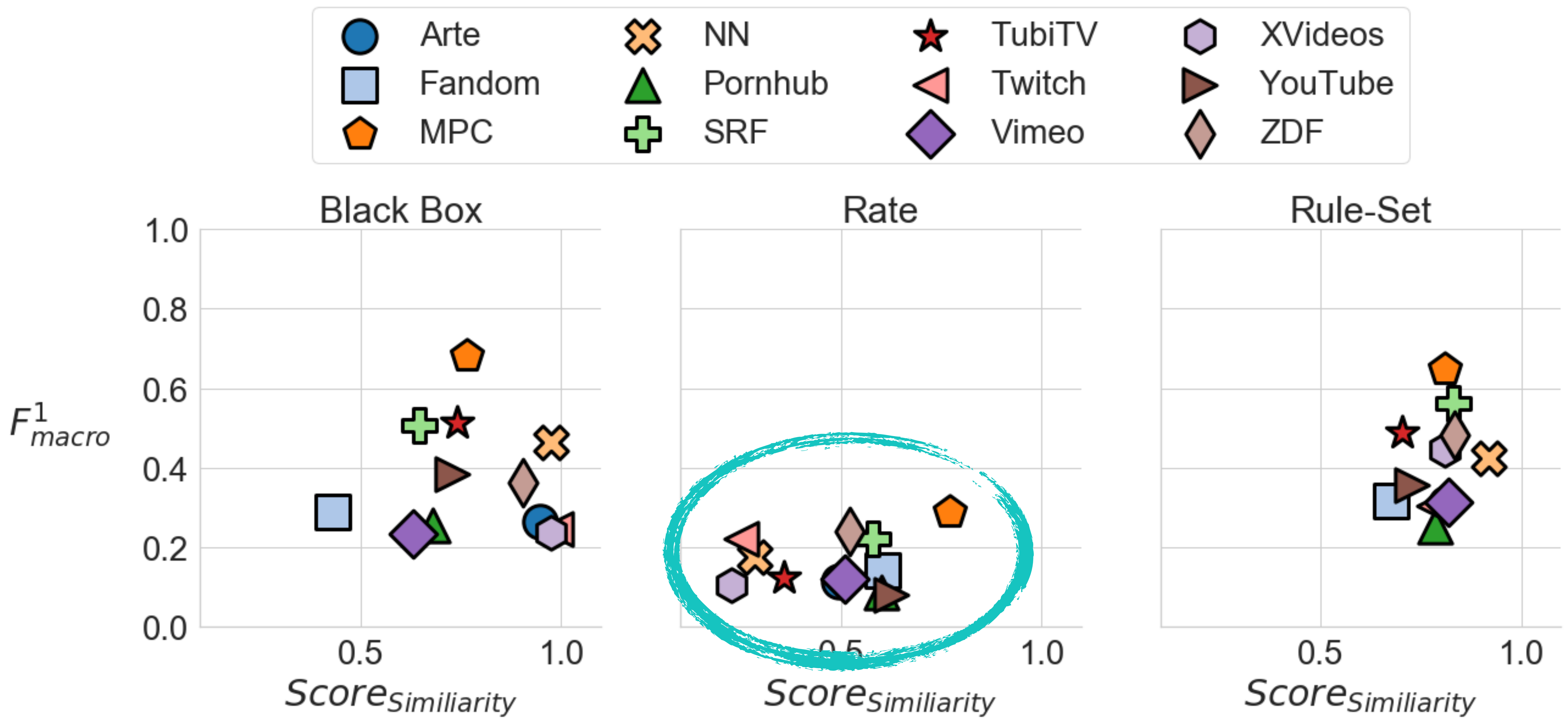
General metric analysis



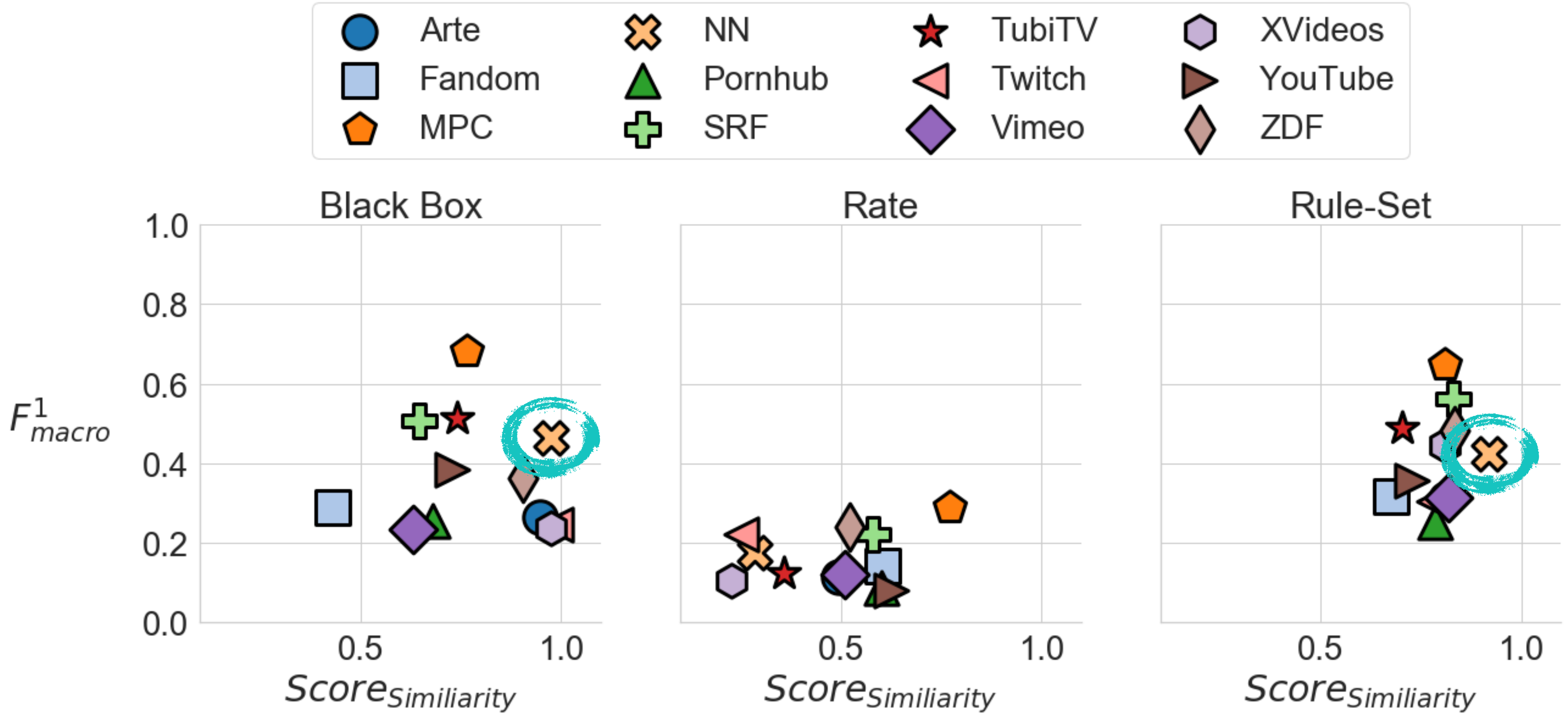
General metric analysis

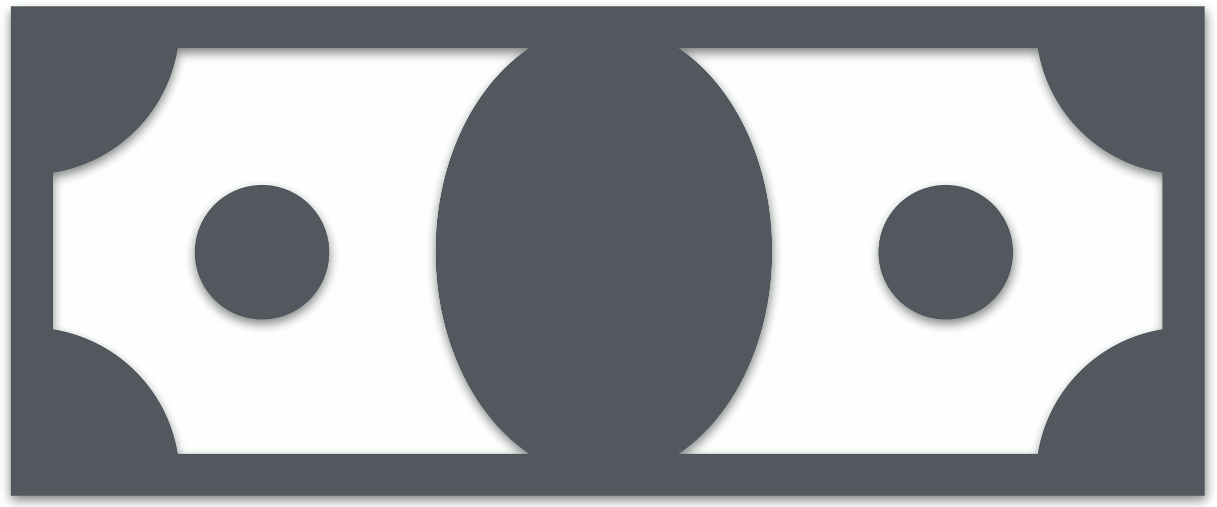


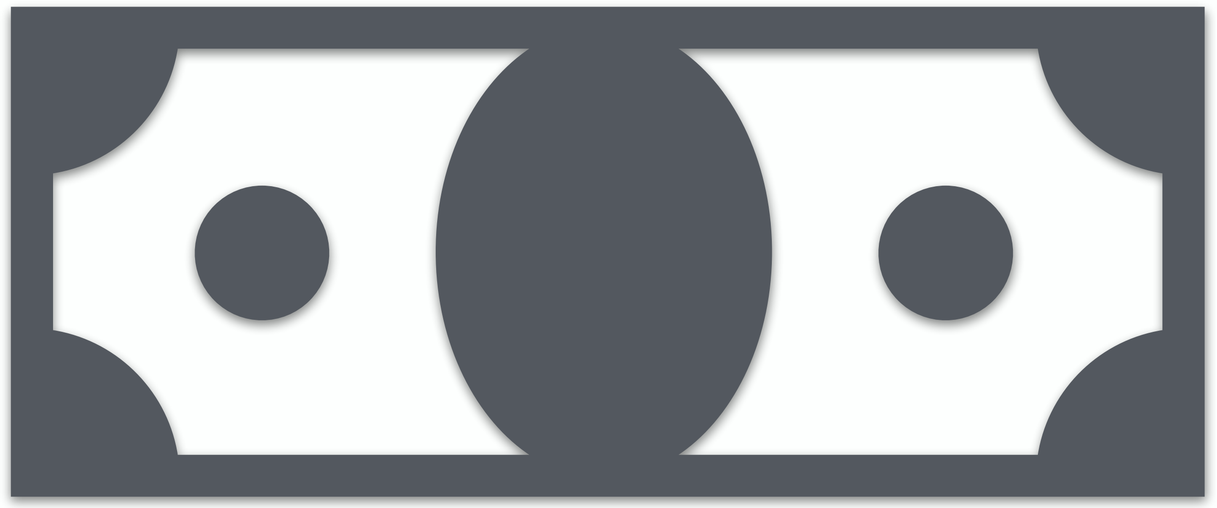
General metric analysis

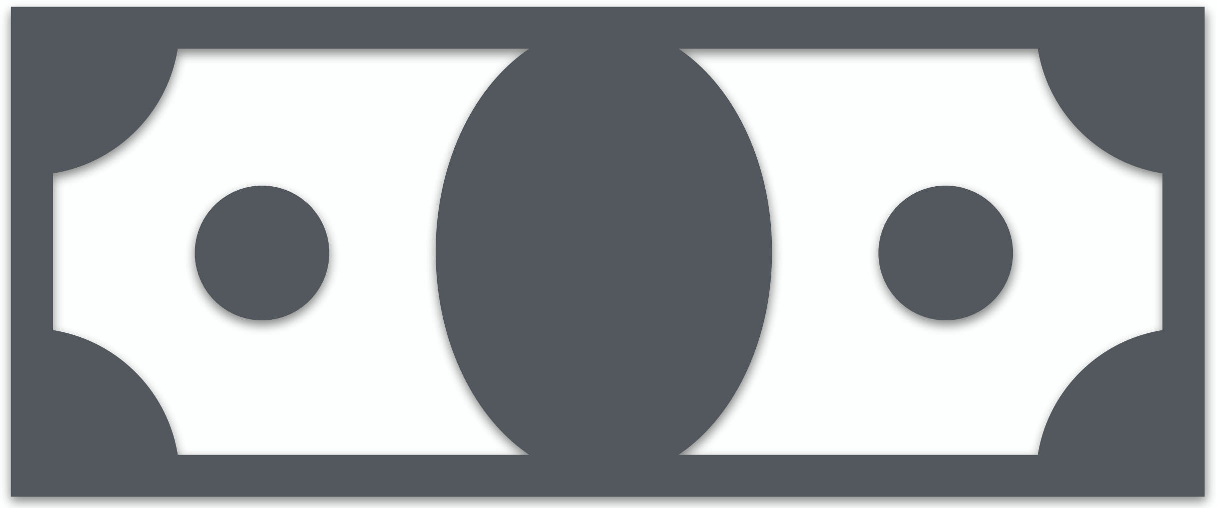


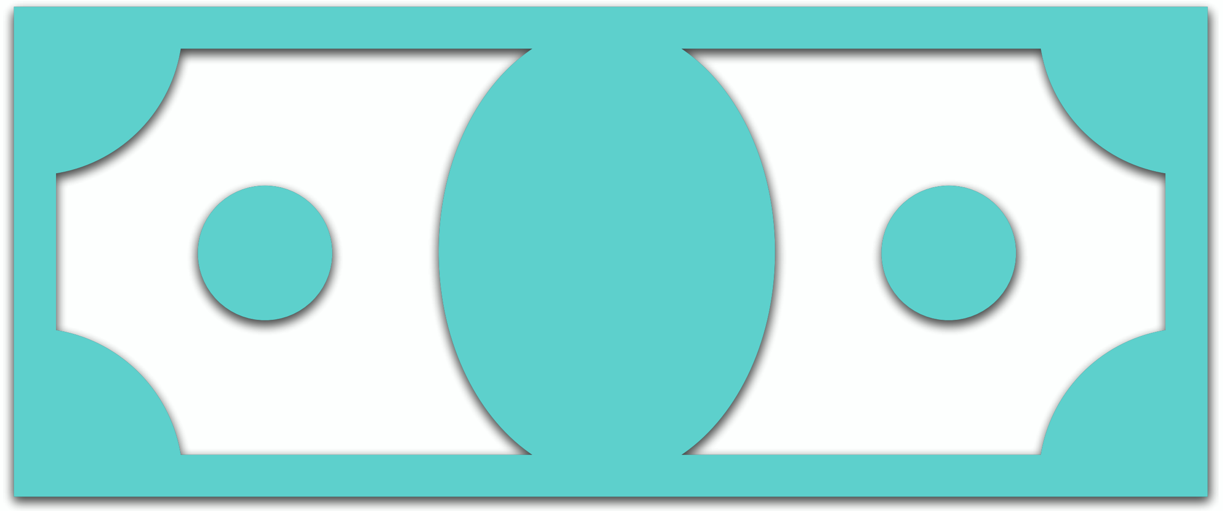
General metric analysis





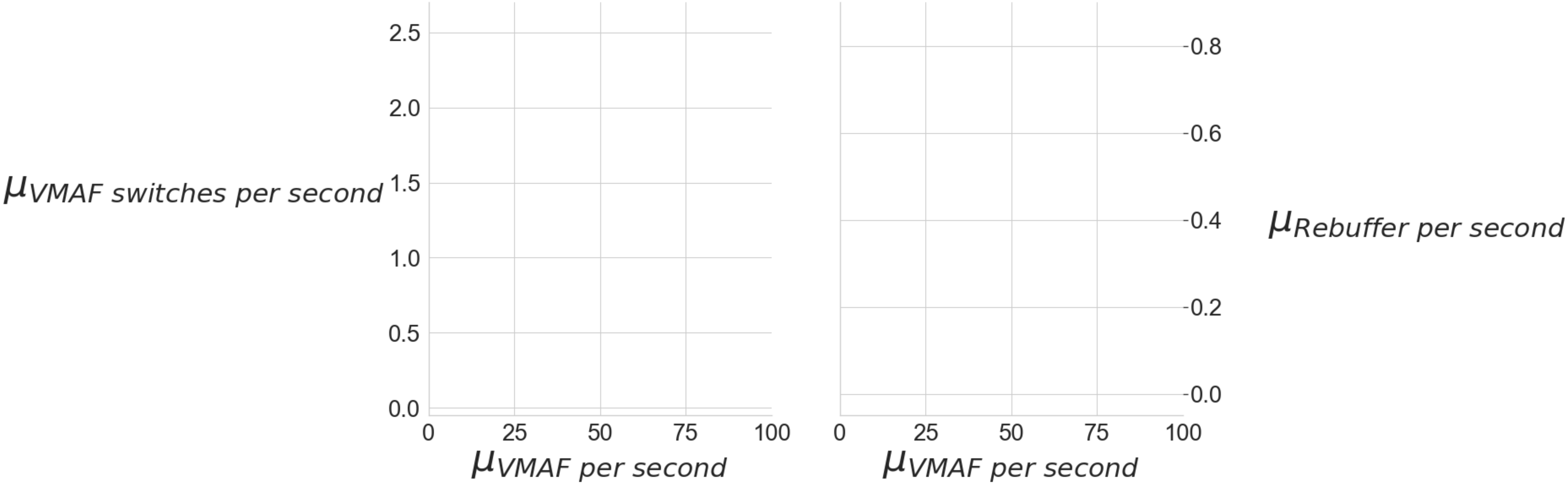




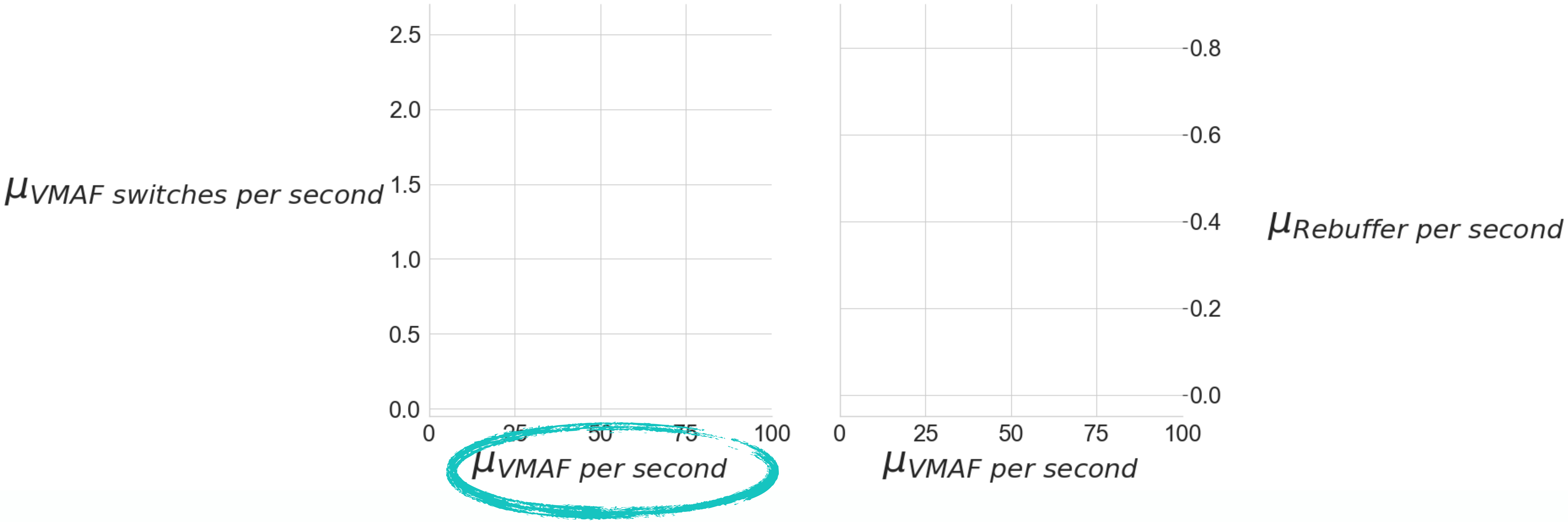


Perceptual metric analysis

Perceptual metric analysis



Perceptual metric analysis



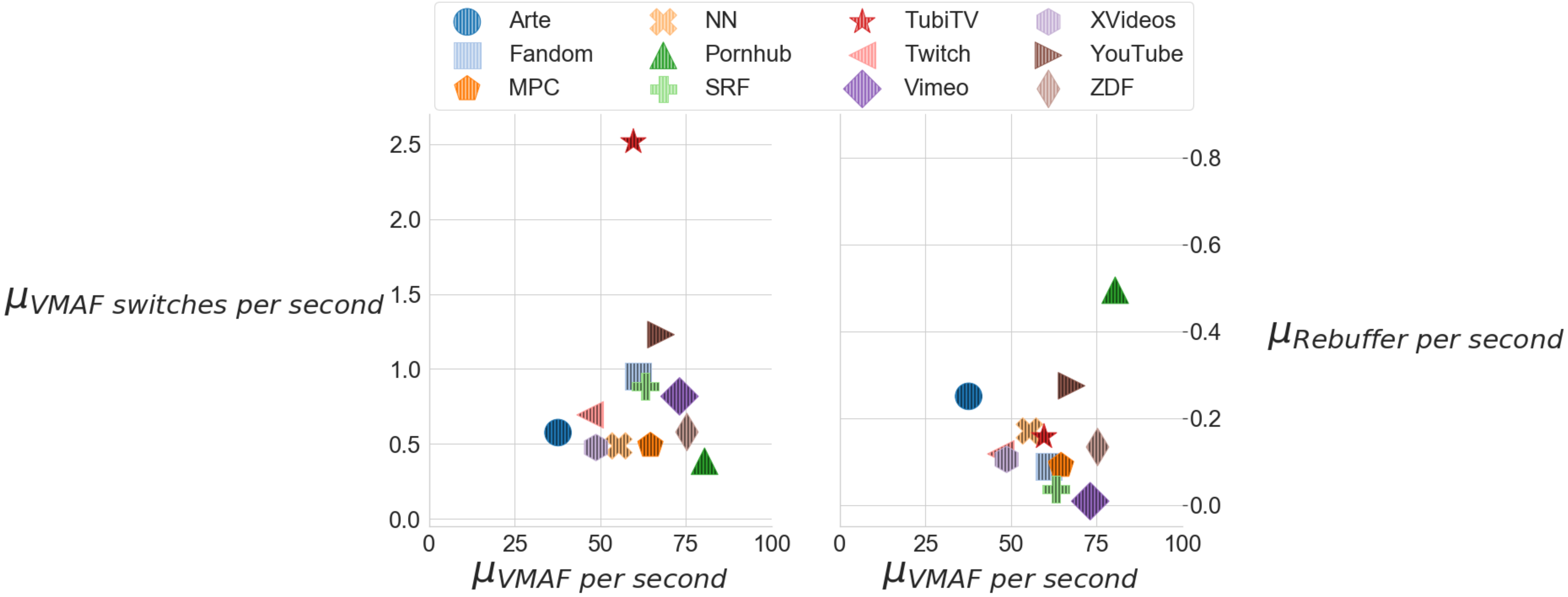
Perceptual metric analysis

$\mu_{VMAF\ switches\ per\ second}$
2.5
2.0
1.5
1.0
0.5
0.0
0
25
50
75
100
$\mu_{VMAF\ per\ second}$
0.8
0.6
0.4
0.2
0.0
0
25
50
75
100
$\mu_{VMAF\ per\ second}$
$\mu_{Rebuffer\ per\ second}$

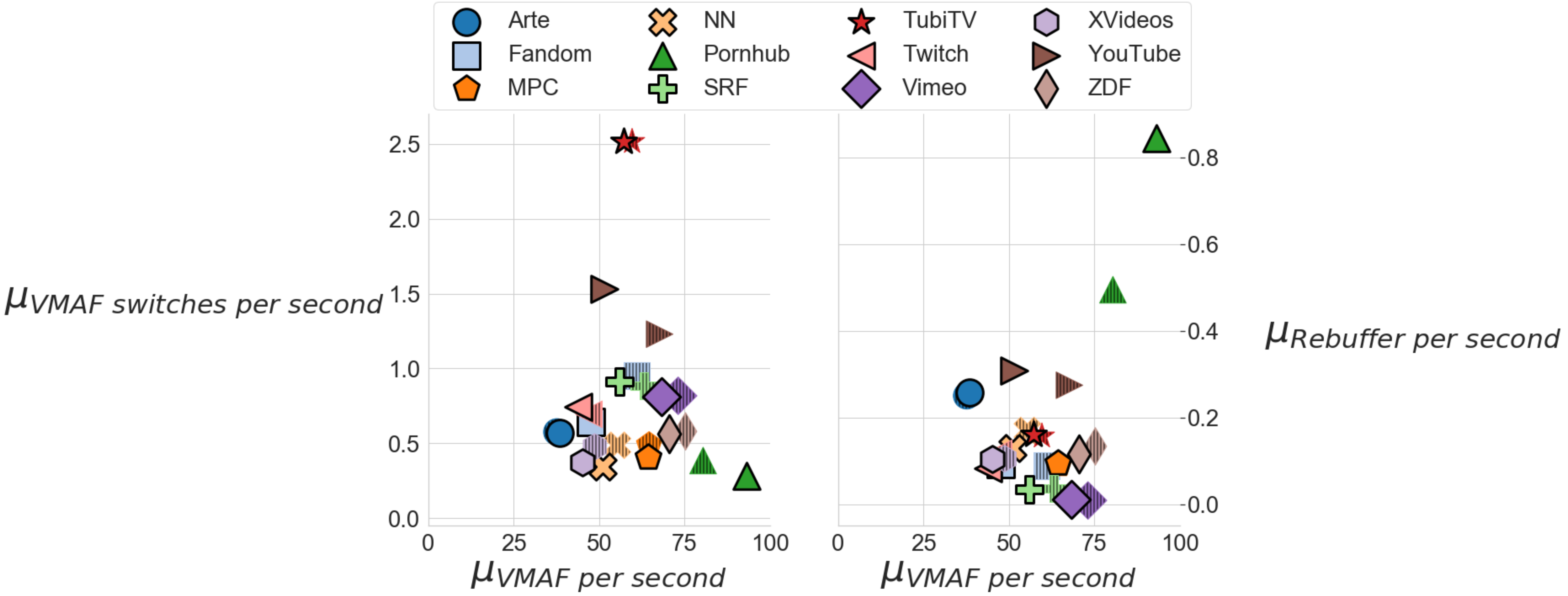
Perceptual metric analysis

$\mu_{VMAF\ switches\ per\ second}$
2.5
2.0
1.5
1.0
0.5
0.0
0
25
50
75
100
$\mu_{VMAF\ per\ second}$
0.8
0.6
0.4
0.2
0.0
0
25
50
75
100
$\mu_{VMAF\ per\ second}$
$\mu_{Rebuffer\ per\ second}$

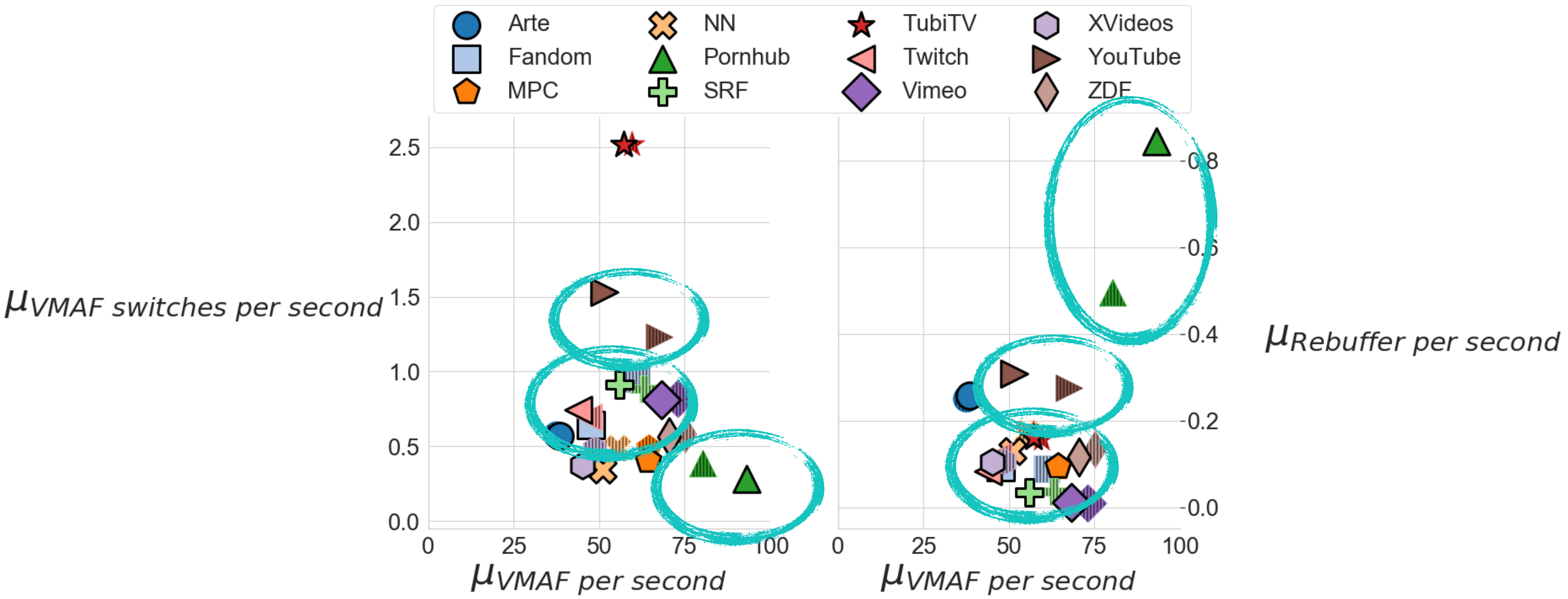
Perceptual metric analysis



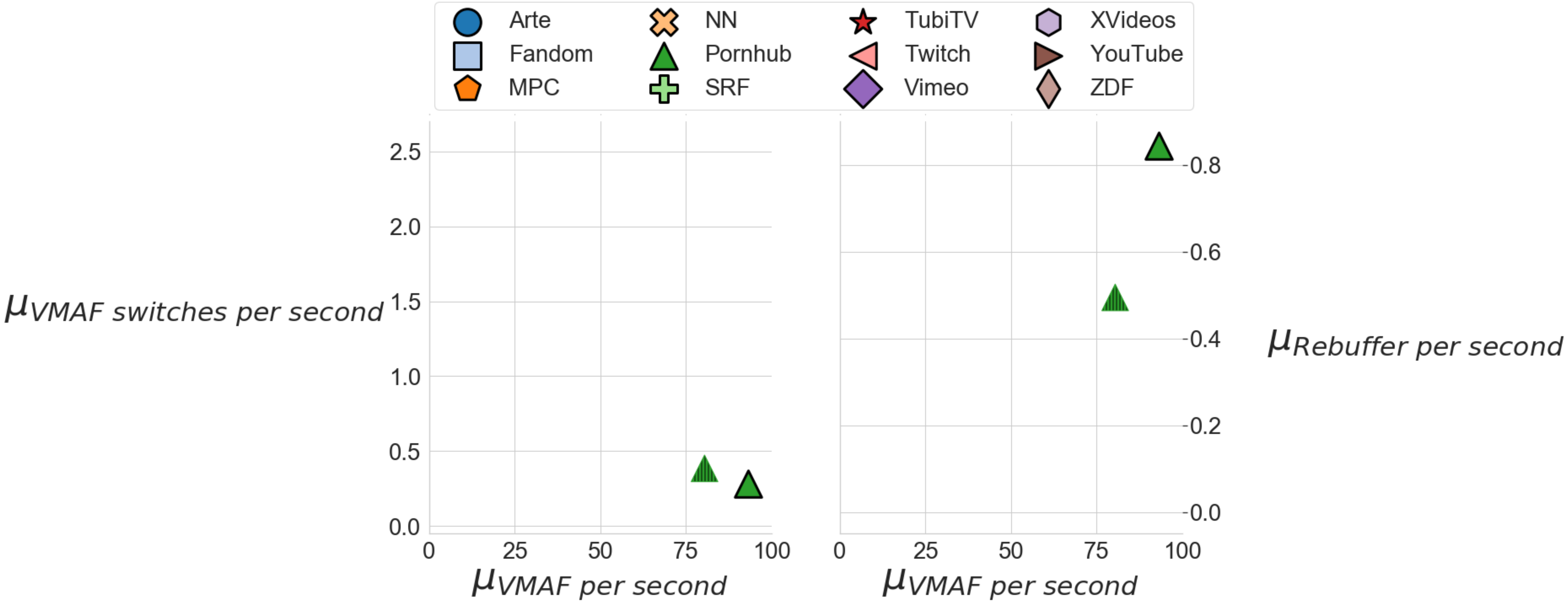
Perceptual metric analysis



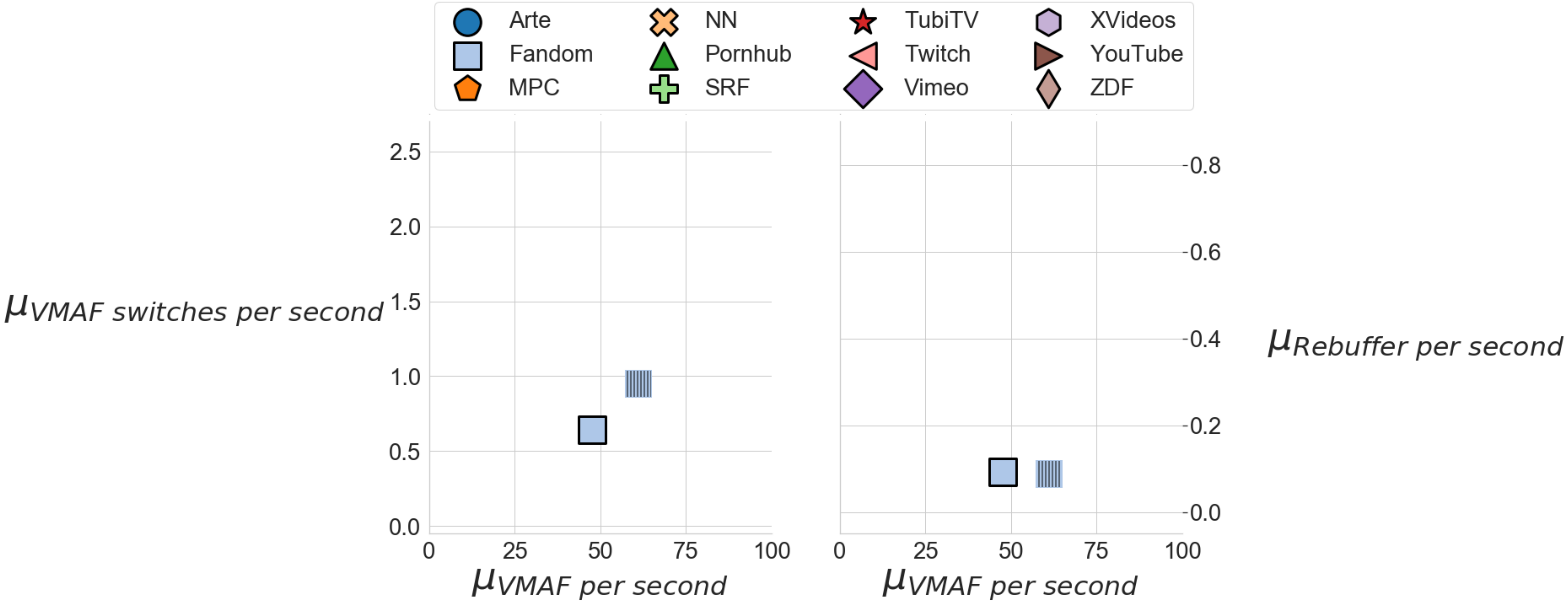
Perceptual metric analysis



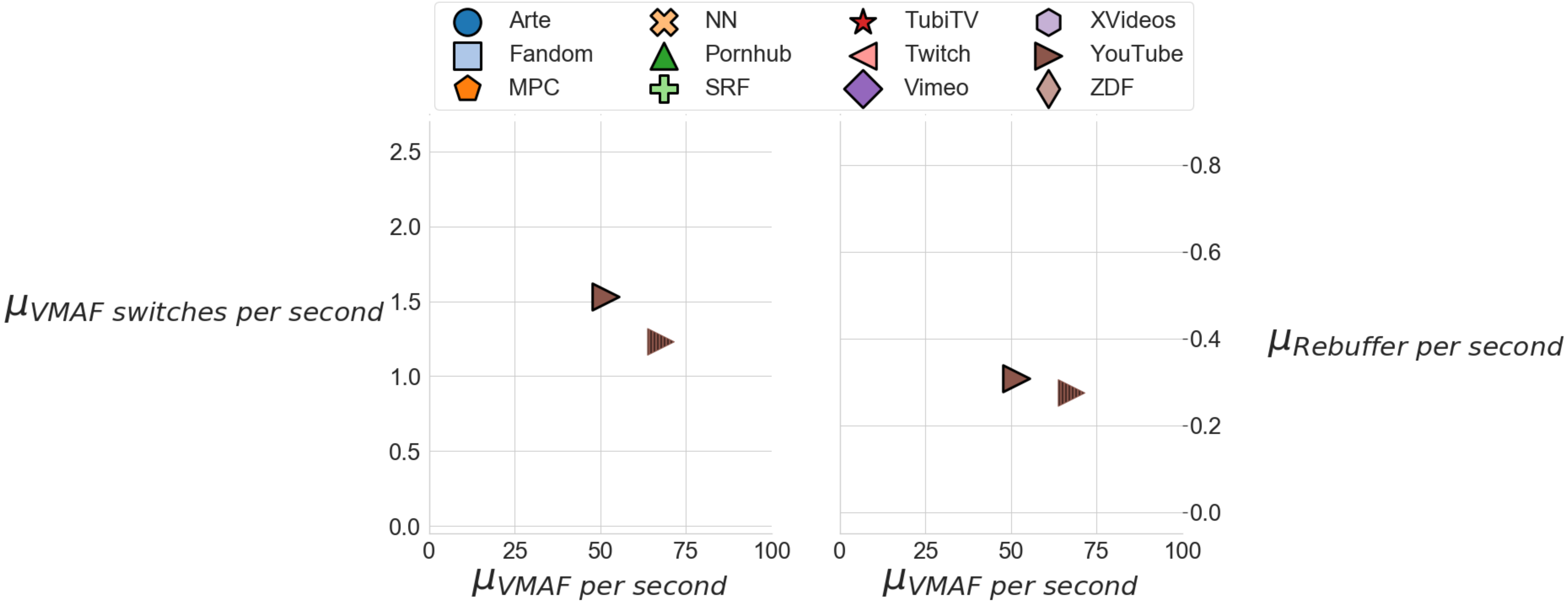
Perceptual metric analysis

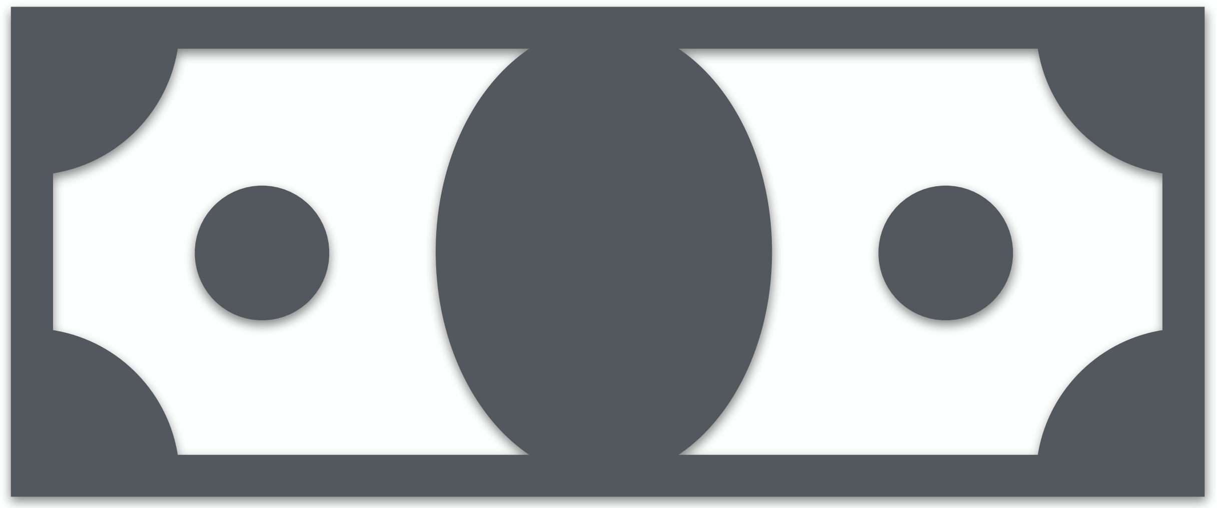


Perceptual metric analysis

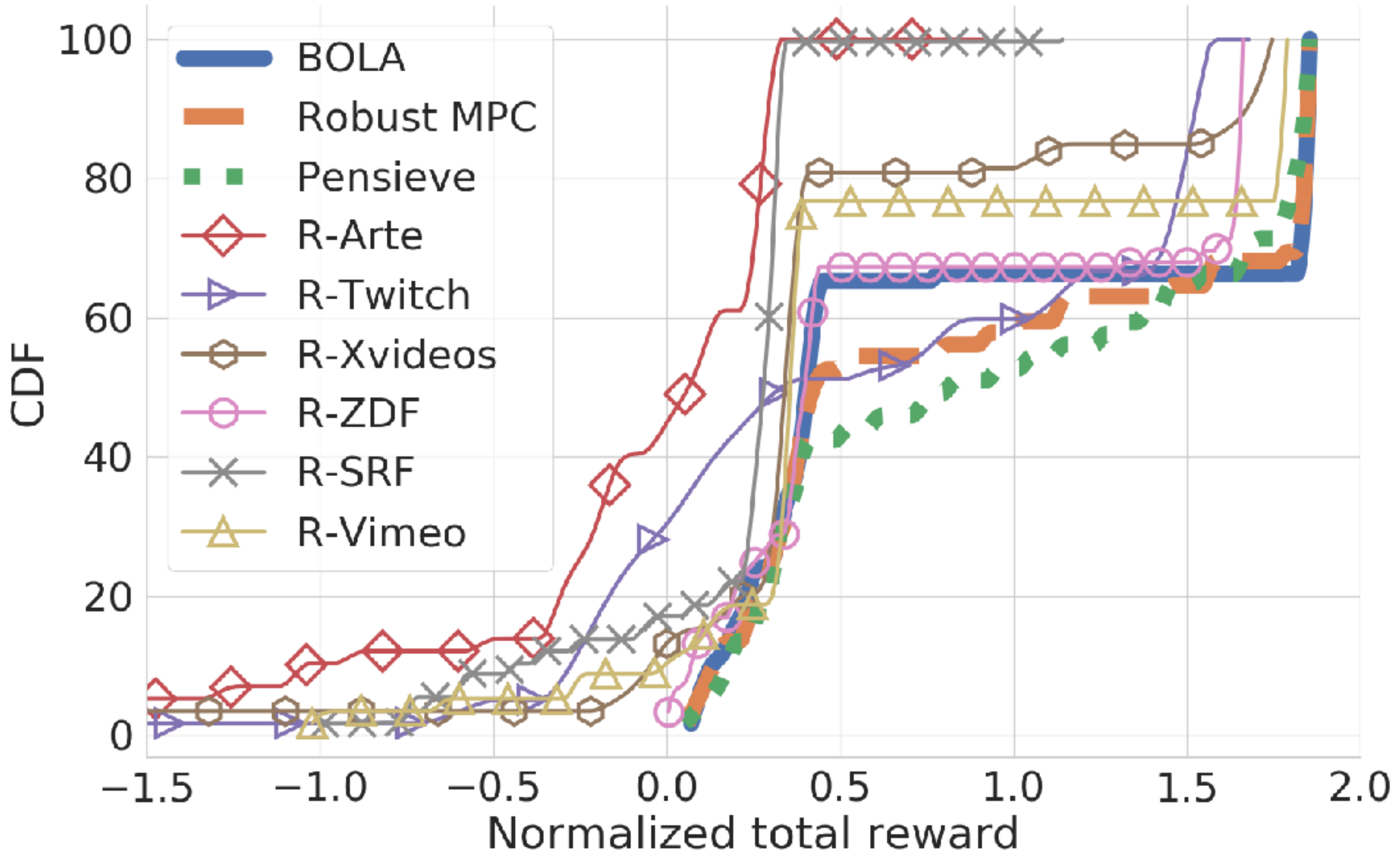


Perceptual metric analysis

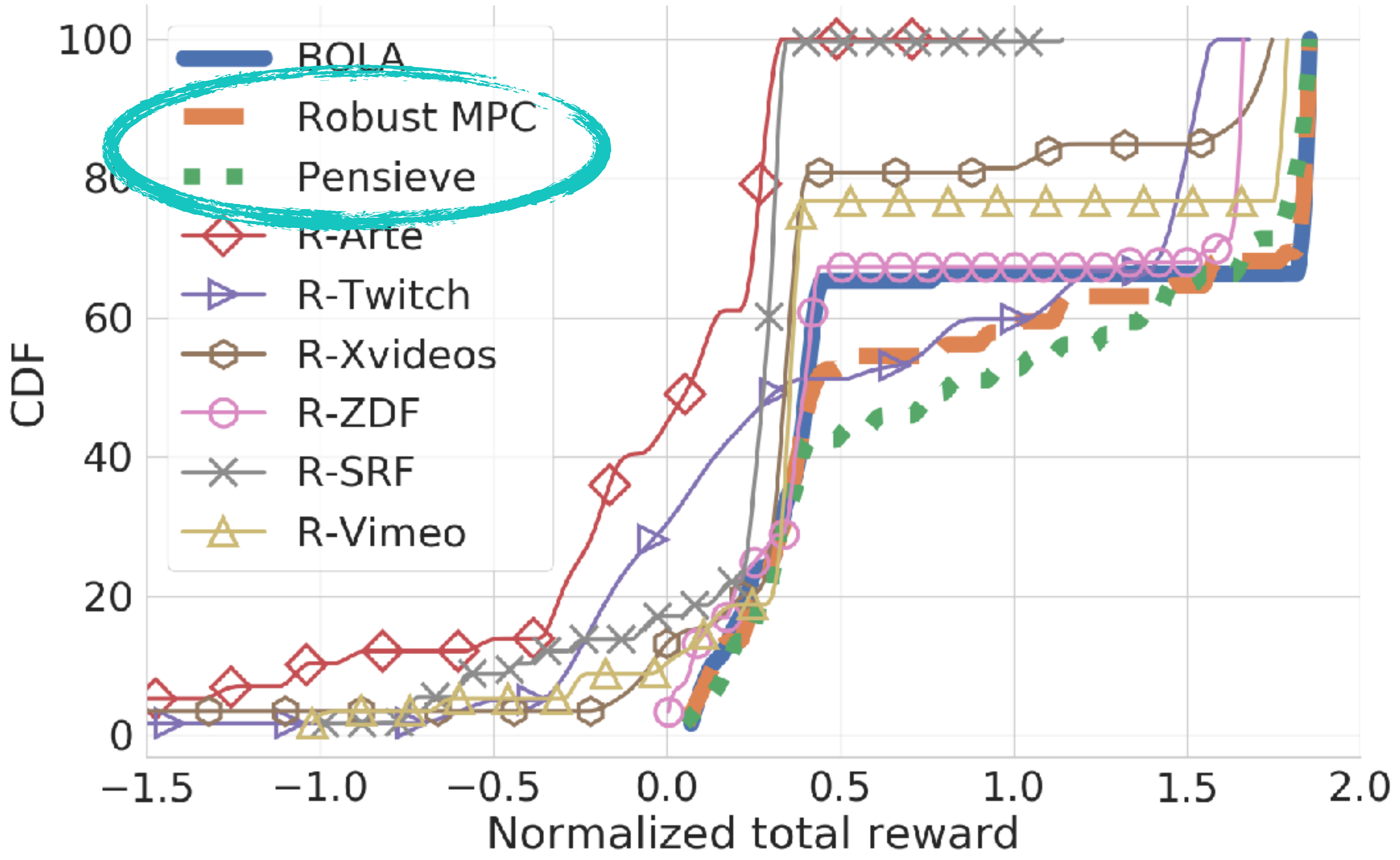




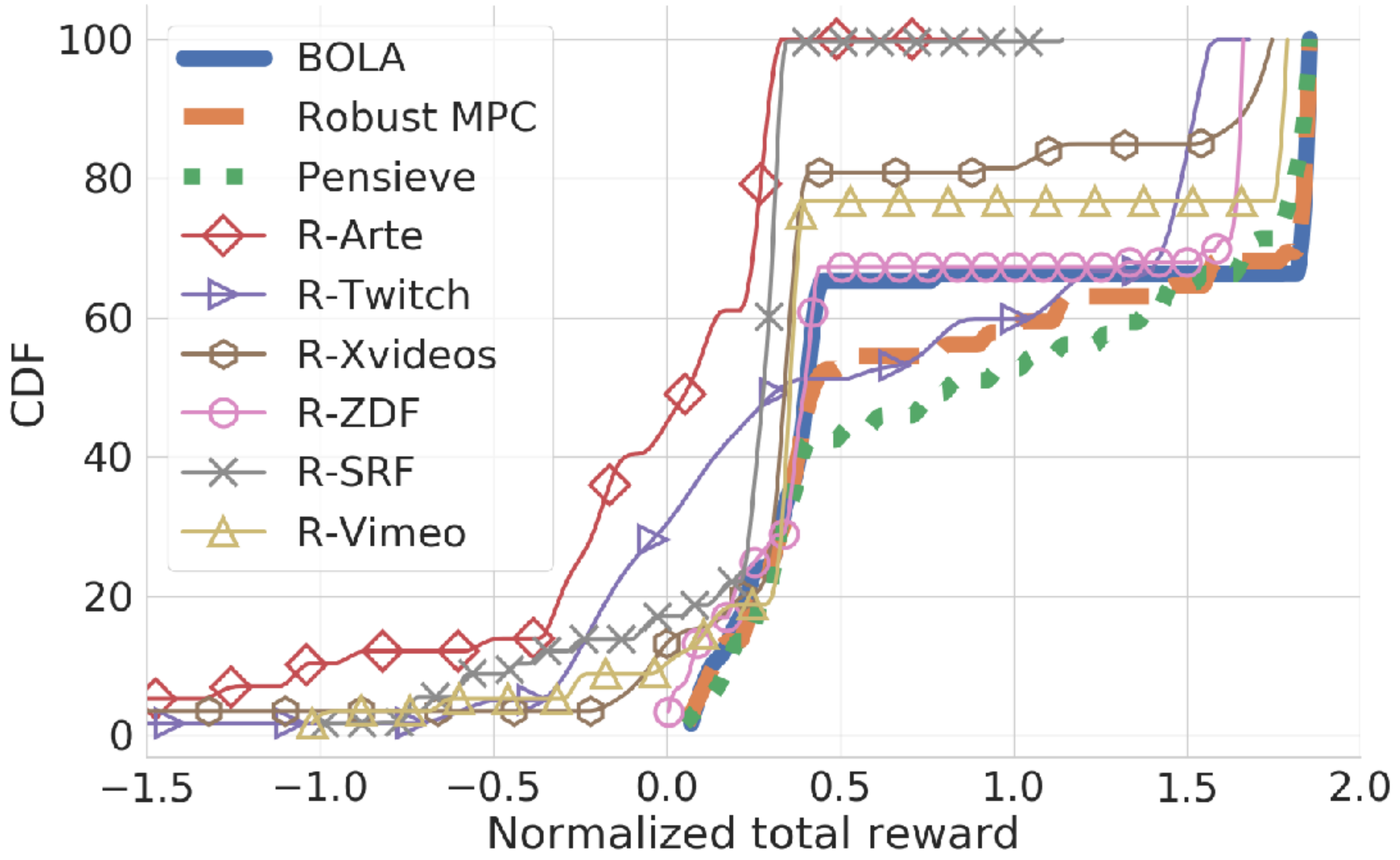
Comparing industry with academia



Comparing industry with academia



Comparing industry with academia

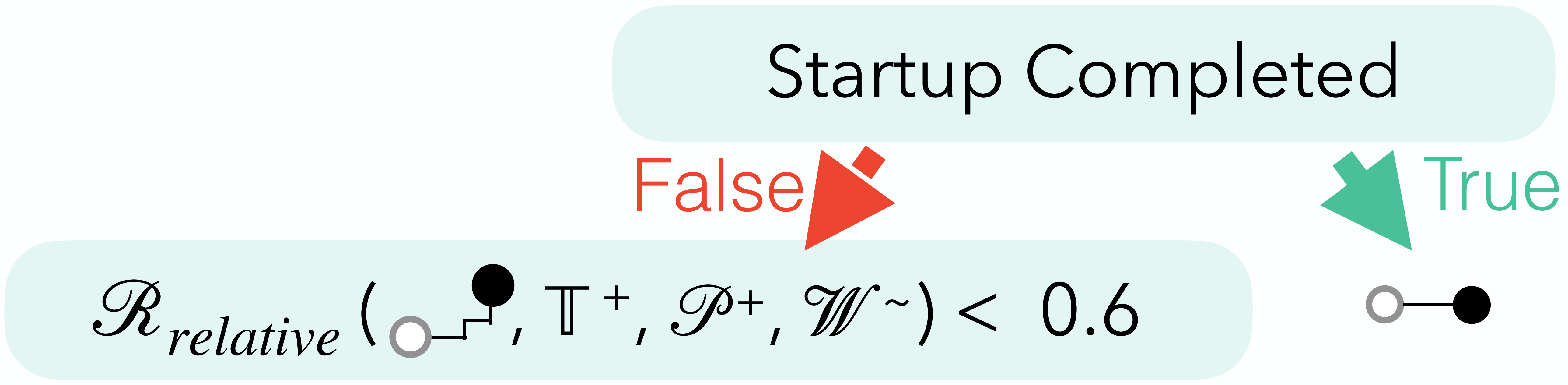


Analysing the **simplified** SRF tree

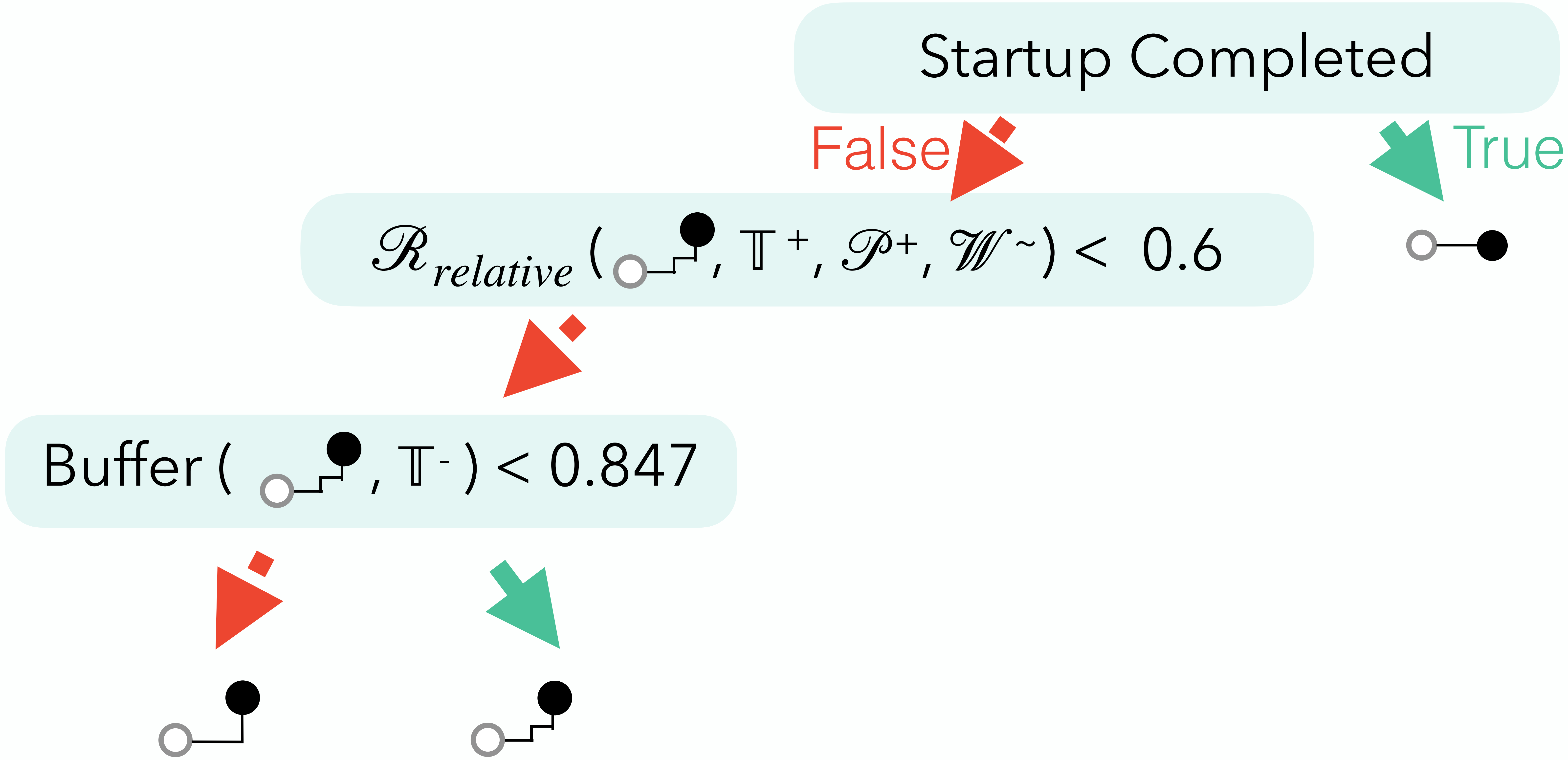
Analysing the **simplified** SRF tree

Startup Completed

Analysing the simplified SRF tree


Startup Completed
False
True
$\mathscr{R}_{relative}(\;, \mathbb{T}^{+}, \mathscr{P}^{+}, \mathscr{W}^{\sim}) < 0.6$

Analysing the **simplified** SRF tree

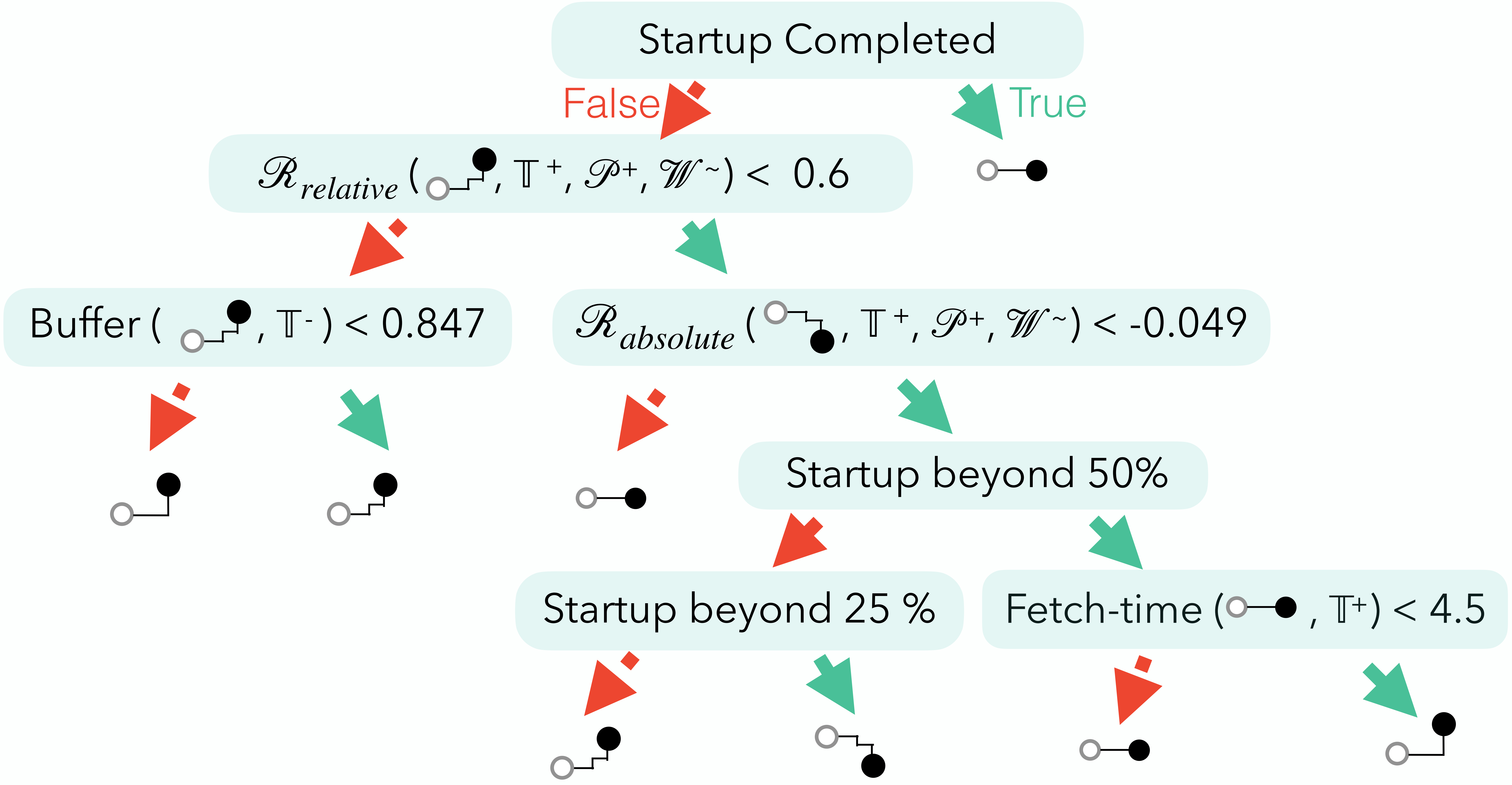


Startup Completed

$\mathscr{R}_{relative}(\;, \mathbb{T}^{+}, \mathscr{P}^{+}, \mathscr{W}^{\sim}) < 0.6$

Buffer ($, \mathbb{T}^{-}) < 0.847$

Analysing the simplified SRF tree



Startup Completed

False

True

$\mathscr{R}_{relative}(\ , \mathbb{T}^{+}, \mathscr{P}^{+}, \mathscr{W}^{\sim}) < 0.6$

Buffer ($, \mathbb{T}^{-}) < 0.847$

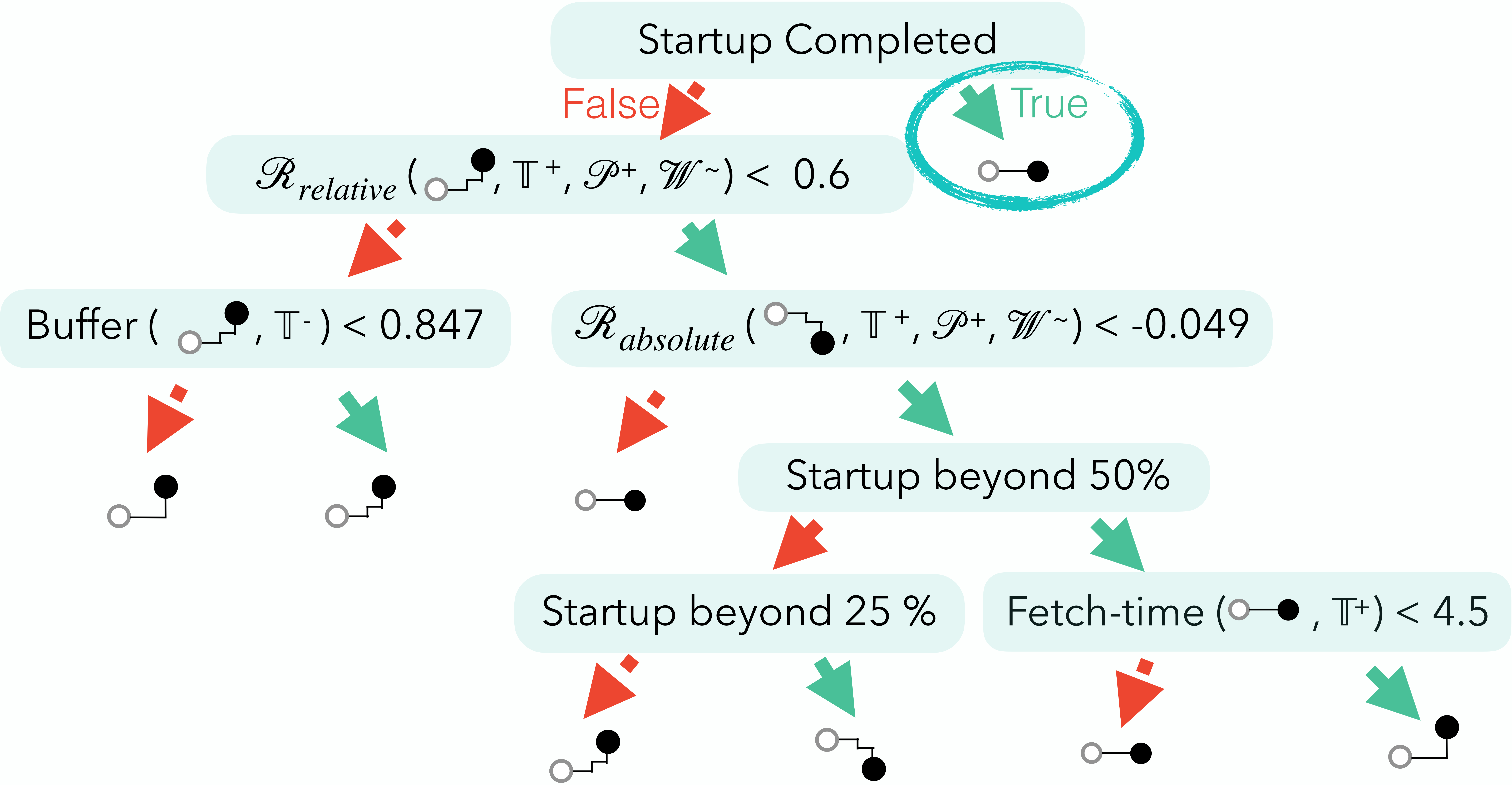
$\mathscr{R}_{absolute}(\ , \mathbb{T}^{+}, \mathscr{P}^{+}, \mathscr{W}^{\sim}) < -0.049$

Startup beyond 50%

Startup beyond 25 %

Fetch-time ($, \mathbb{T}^{+}) < 4.5$

Analysing the simplified SRF tree



Startup Completed

False

True

$\mathscr{R}_{relative}(\; , \mathbb{T}^{+}, \mathscr{P}^{+}, \mathscr{W}^{\sim}) < 0.6$

Buffer (, $\mathbb{T}^{-}$) < 0.847

$\mathscr{R}_{absolute}(\; , \mathbb{T}^{+}, \mathscr{P}^{+}, \mathscr{W}^{\sim}) < -0.049$

Startup beyond 50%

Startup beyond 25 %

Fetch-time (, $\mathbb{T}^{+}$) < 4.5

What usage has interpretability

What usage has interpretability

- Identify potential issues

What usage has interpretability

- Identify potential issues
- Tracing the input-output mapping

Limitations & Future Work

Limitations & Future Work

- Primitives

Limitations & Future Work

- Primitives
- Decision space

Limitations & Future Work

- Primitives
- Decision space
- Interpretability is subjective

Limitations & Future Work

- Primitives
- Decision space
- Interpretability is subjective
- Region

Takeaways

Takeaways

- Decision trees are a great versatile tool

Takeaways

- Decision trees are a great versatile tool
- Interpretability is useful

Takeaways

- Decision trees are a great versatile tool
- Interpretability is useful
- Interpretability is subjective

Takeaways

- Decision trees are a great versatile tool
- Interpretability is useful
- Interpretability is subjective
- Limited by domain knowledge

Melissa Licciardello

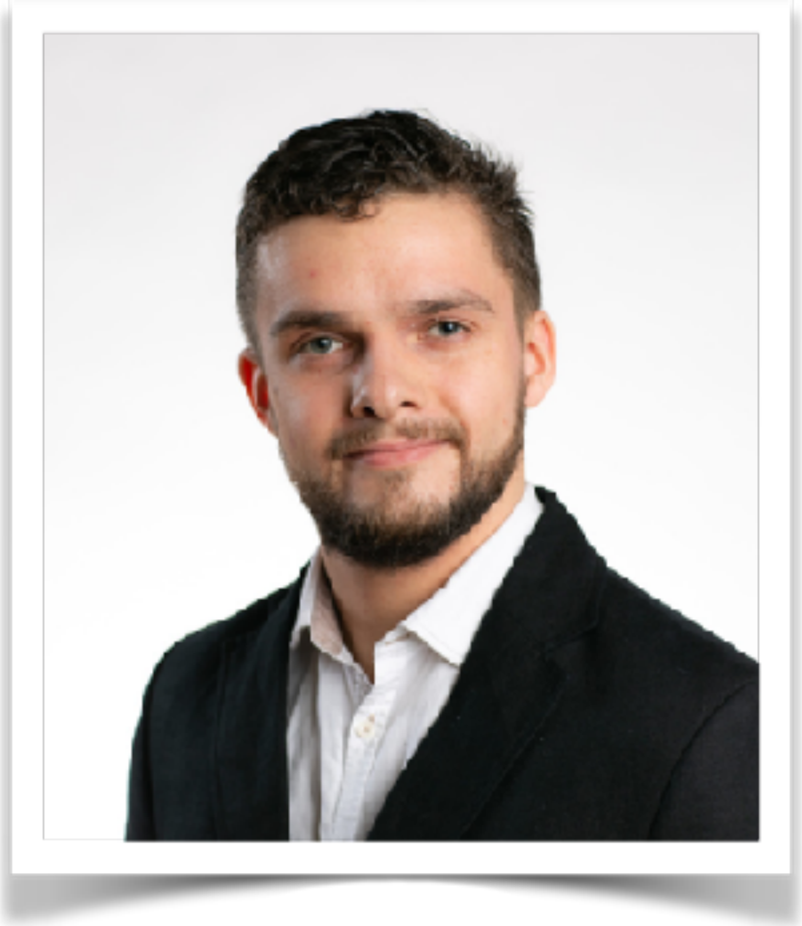
melissa.licciardello@inf.ethz.ch

Melissa Licciardello

melissa.licciardello@inf.ethz.ch

Ankit Singla

ankit.singla@inf.ethz.ch



Maximilian Grüner

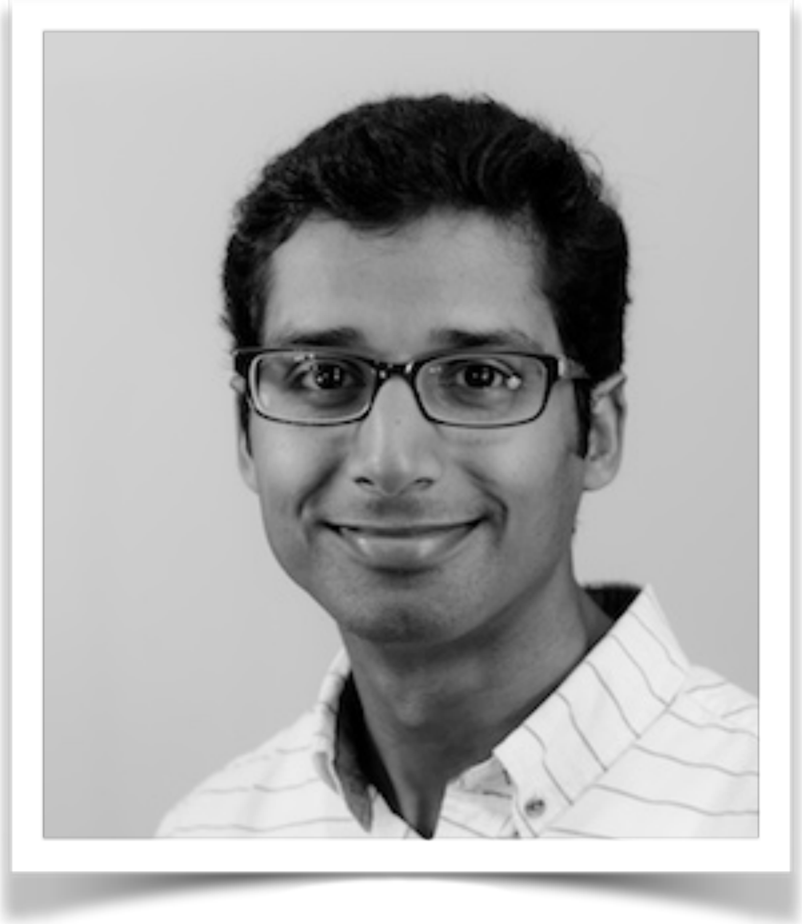
mgruener@inf.ethz.ch

Melissa Licciardello

melissa.licciardello@inf.ethz.ch

Ankit Singla

ankit.singla@inf.ethz.ch



Maximilian Grüner

mgruener@inf.ethz.ch

Melissa Licciardello

melissa.licciardello@inf.ethz.ch

Ankit Singla

ankit.singla@inf.ethz.ch

https://github.com/magruener/reconstructing-proprietary-video-streaming-algorithms