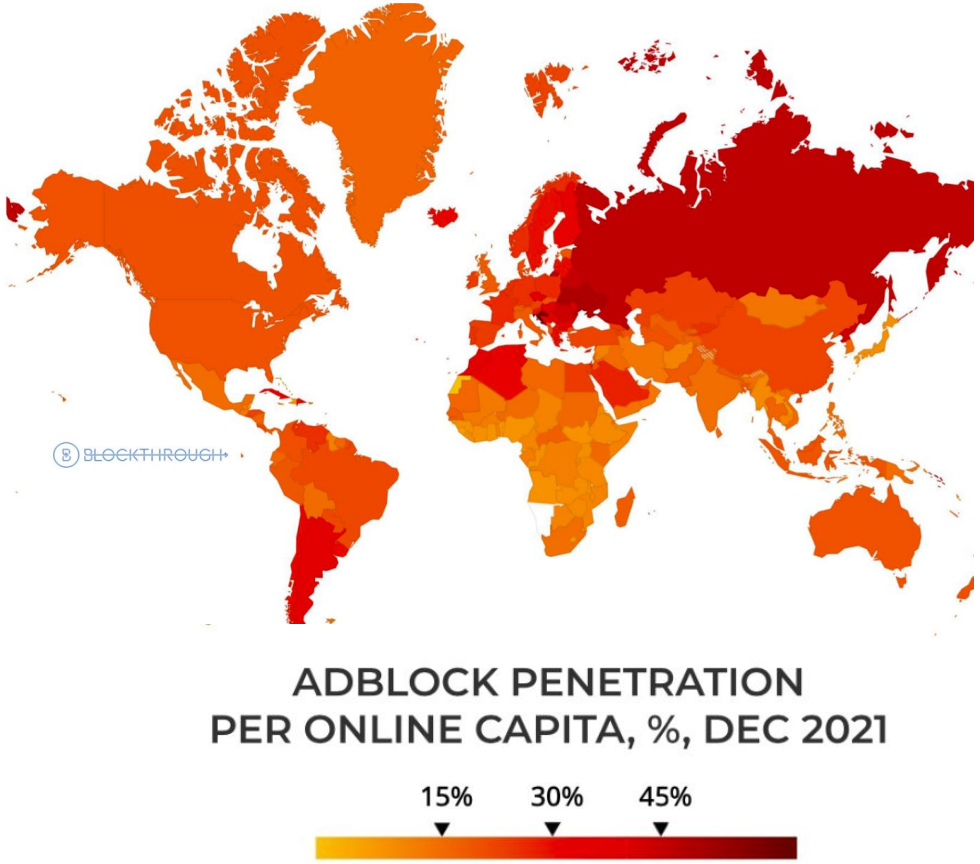
AutoFR: Automated Filter Rule Generation for Adblocking

Hieu Le*, Salma Elmalaki*, Athina Markopoulou*, Zubair Shafiq†

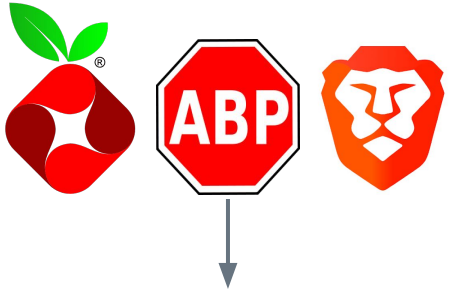
*UCI †UCDAVIS ProperData

Adblockers



[2022 PageFair Adblock Report]

Filter Rules



Filter Rules

Type	Filter Rule	Description
hostname	ad.com	Block network requests
eSLD	||ad.com^	
FQDN	||img.ad.com^	
With Path	||img.ad.com/banners	
Element Hiding	||example.com##.ad	Hide elements with class name “ad”
Stop JS Execution	||example.com#$#abort-on-property-read AdLoader	Stop JS execution if it reads from AdLoader

Filter List Authors

Filter Rules

EasyList (ads)

Filter List Authors

```
dvitohrrmq.xyz^          ##.ad_cont_footer          /rightad.
dviwjuggl.com^           ##.ad_contain              /rightad/*
dvnmyltt.xyz^            ##.ad_container            /rightAd1.
dvoyy3qexr.com^          ##.ad_container_300x250    /rightAd2.
dvtnrwppxk.com^          ##.ad_container_5          /rightads.$domain=~rightads.co.uk
dvucdnv.com^             ##.ad_container_6          /rightbanner/*
dvwvqeddi.com^           ##.ad_container_7          /rightnavads.
dvypar.com^              ##.ad_container_728x90     /rightnavadsanswer.
dwarfsubmerge.com^       ##.ad_container_8          /righttopads.
dwaterver.xyz^           ##.ad_container_9          /rollad.
dwepmhhyigfb.com^        ##.ad_container__sidebar   /rolloverads/*
dwetwdstom1020.com^      ##.ad_container__top       /rolloverbannerad.
dwxrrdaqtm.top^          ##.ad_container_body       /root_ad.
dxpxlt.com^              ##.ad_container_bottom     /rotad/*
dxrzzmhwdp.com^          ##.ad_content              /rotads/*
dycej.com^               ##.ad_content_below        /rotateads.
dydjxvazbj.com^          ##.ad_content_img          /rotatedads1.
dyecowwhy.com^           ##.ad_content_wide         /rotatedads13.
dyingdesignscharming.com^##.ad_content_wrapper      /rotatedads2.
dyino.com^               ##.ad_contents             /rotatingad.
dynamicadx.com^          ##.ad_crown                /rotatingpeels.
dynamicdn.com^           ##.ad_custombanner         /rotatingtextad.
dynamitedata.com^        ##.ad_db                   /rotation/banner
dynamitedepressionweapons.##.ad_default             /rotationad.
dynpaa.com^              ##.ad_deferrable           /rotatorad300x250.
dynpuqtsjpgqjv.com^      ##.ad_description         /rotatoradbottom.
dynspt.com^              ##.ad_descriptor           /roturl.js
dynsrvbaa.com^           ##.ad_desktop              /rpc/ad/*
dynsrvdea.com^           ##.ad_disclaimer           /rpgetad.
dynsrvtbg.com^           ##.ad_div                  /rsads.js
dynsrvtyu.com^           ##.ad_div_banner           /rsads/*
dynssp.com^              ##.ad_div_box              /rsc_ad_
dyoixbyp.xyz^            ##.ad_div_box2             /rss/ads/*
dyoncdgy.com^            ##.ad_element              /s_ad.aspx?
dyshigichu.com^          ##.ad_embed                /sadasds.js
dz4ad.com^               ##.ad_eniro                /safead/*
dzhjmp.com^              ##.ad_entry_title_under    /sailthru.js
dzienkudrow.com^         ##.ad_entrylists_end       /salesad/*
dzubavstal.com           ##.ad_event_mast_wrapper   /samplead1.
e-find.co^               ##.ad_external             /samsung_ad.
e-v-e-n.me^              ##.ad_eyebrow              /sas/ads/*
e0663490cca0296f7.com^   ##.ad_fadein              /satnetads.
e0a42e1a21669b.com^      ##.ad_feature              /satnetgoogleads.
e13085e58935e6.com^      ##.ad_filler               /savvyads.
                         ##.ad_flash                /sb-relevance.js
                         ##.ad_flat-boxright10      /sbnr.ads?
                         ##.ad_flat-boxright6       /sc-tagmanager/*
                         ##.ad_flat-boxright9       /scanscout.
```

Filter List Authors

Filter Rules

EasyList (ads)

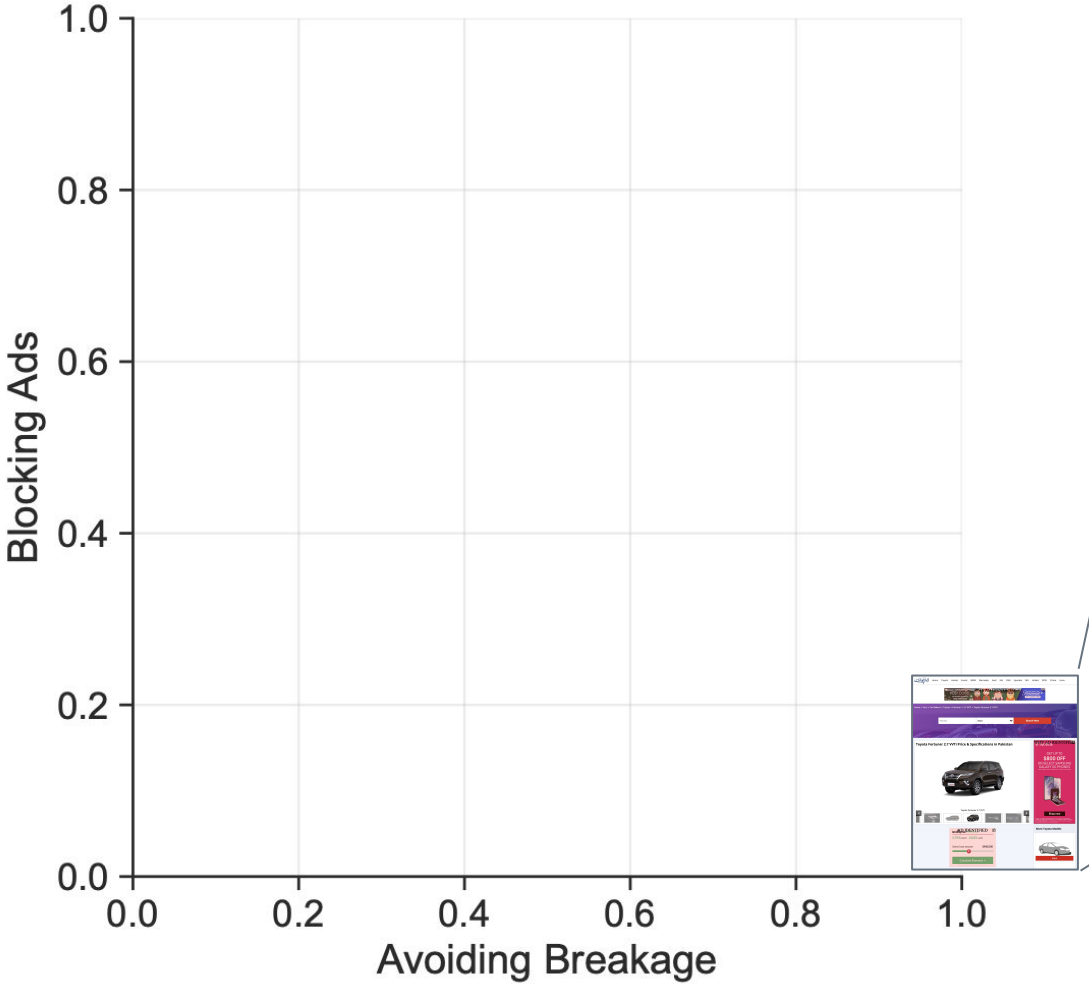
Filter List Authors

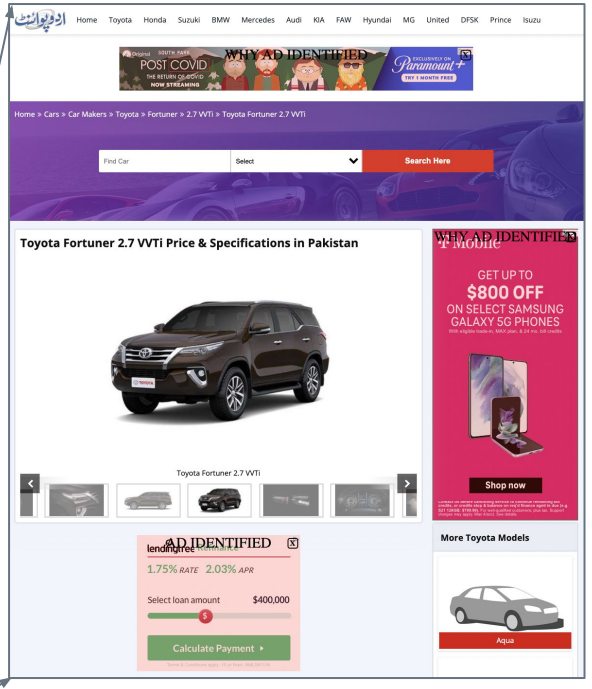
Type	Filter Rule	Description
hostname	ad.com	
eSLD	||ad.com^	Block network requests
FQDN	||img.ad.com^	
With Path	||img.ad.com/banners	
Element Hiding	||example.com##.ad	Hide elements with class name “ad”
Stop JS Execution	||example.com#$#abort-on-property-read AdLoader	Stop JS execution if it reads from AdLoader

What makes filter rule generation challenging?

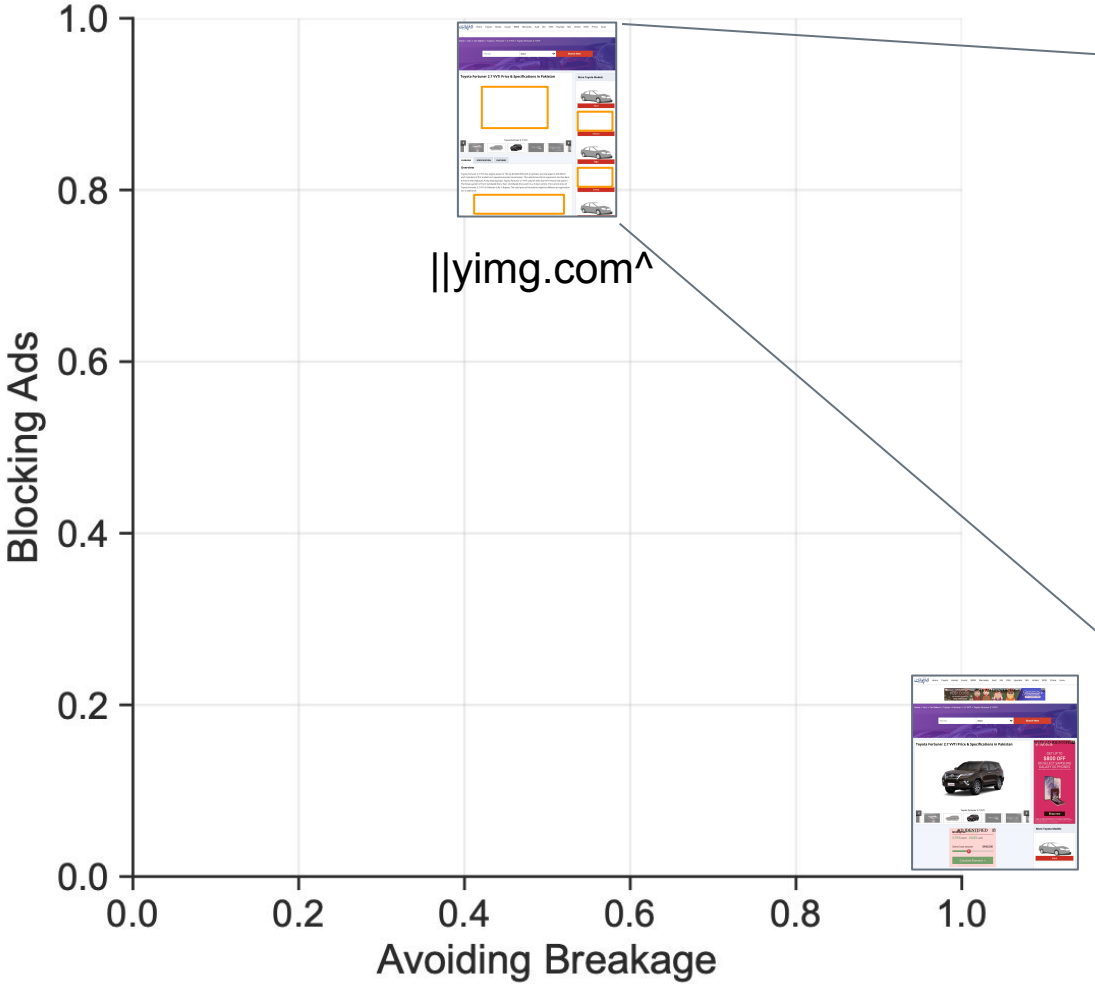
Blocking ads vs.
Avoiding breakage

This trade-off causes the need to have different rule granularities.

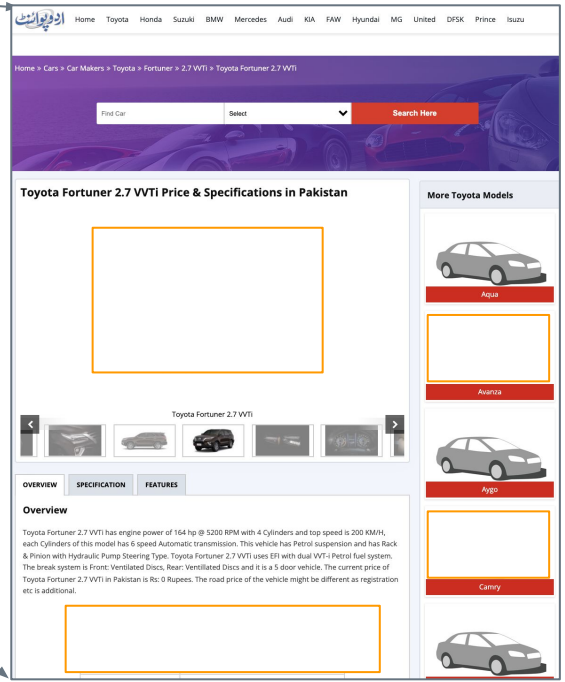




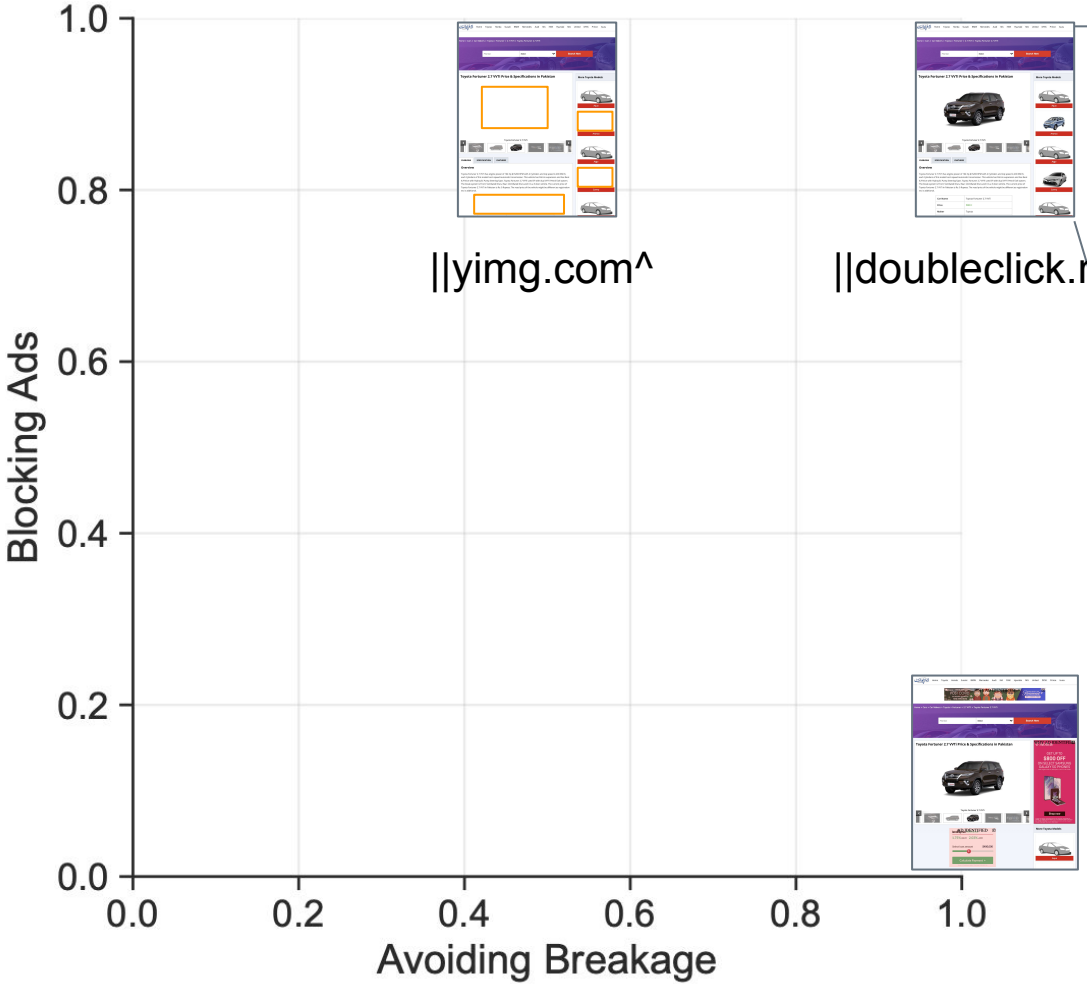
How does the site behave normally?

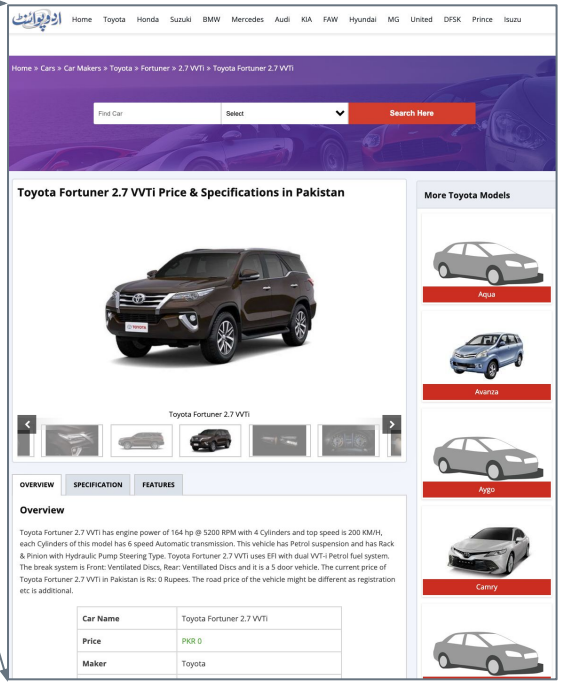


||yimg.com^

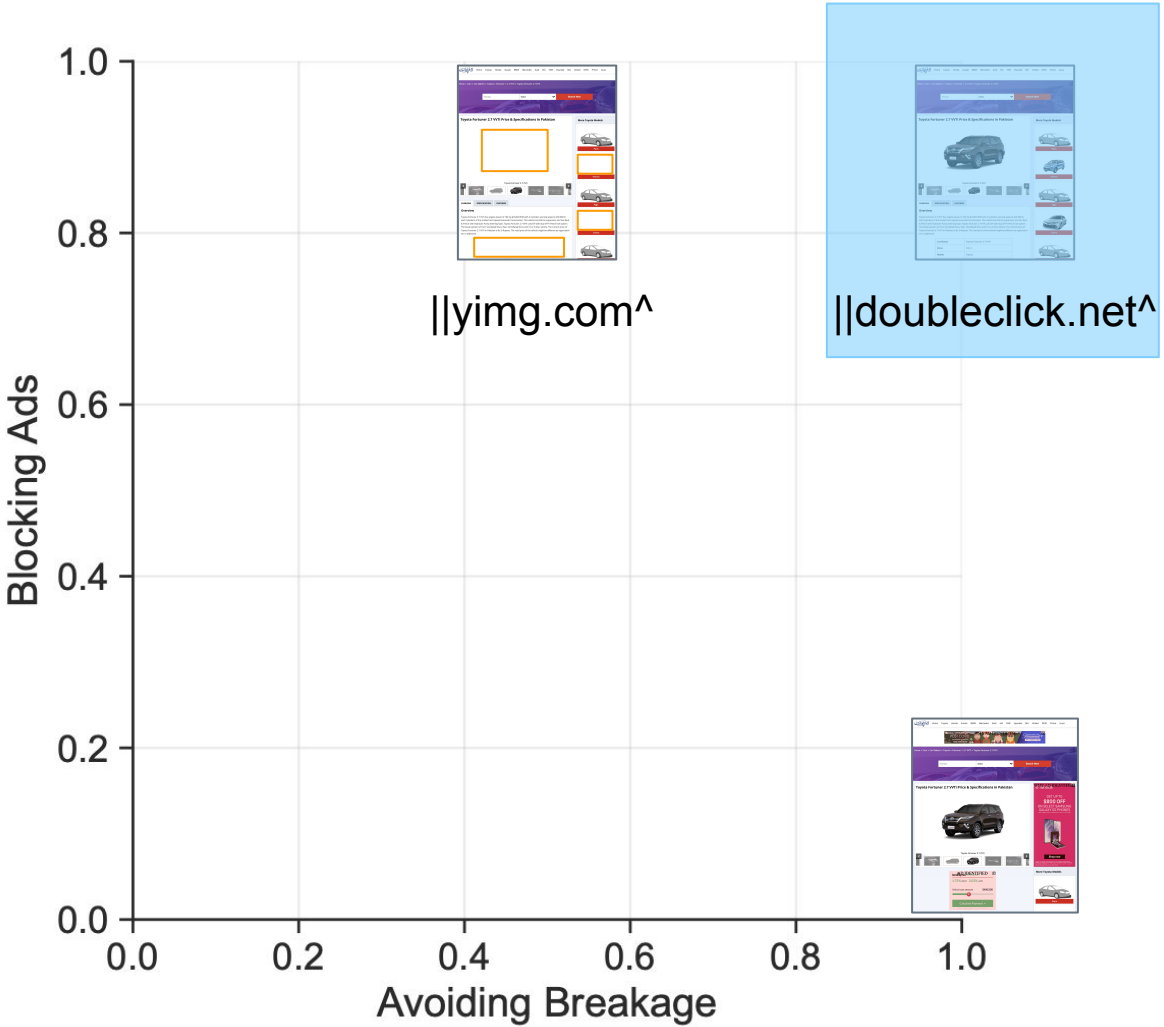


Are ads blocked with some breakage?

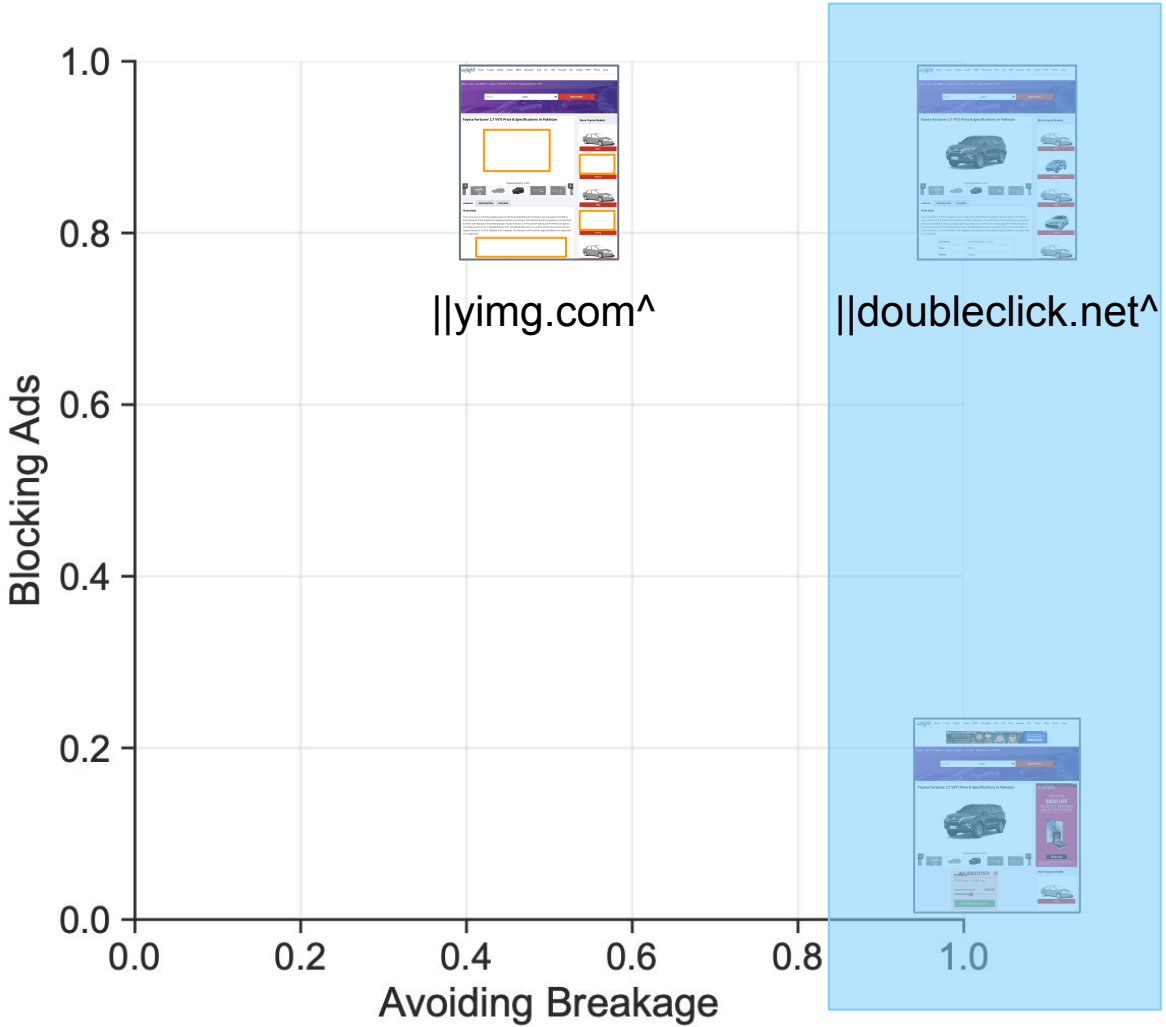




Are ads blocked with minimal breakage?



Ideal Region:
Filter rules that block **all** ads with minimal breakage



Acceptable Region:
Filter rules that block **some** ads with minimal breakage

Filter Rules

Preferred but may overblock → breakage

Avoid breakage more easily → easier to evade

Type	Filter Rule	Description
eSLD	||ad.com^	Block network requests
FQDN	||img.ad.com^	
With Path	||img.ad.com/banners	

What makes filter rule generation challenging?

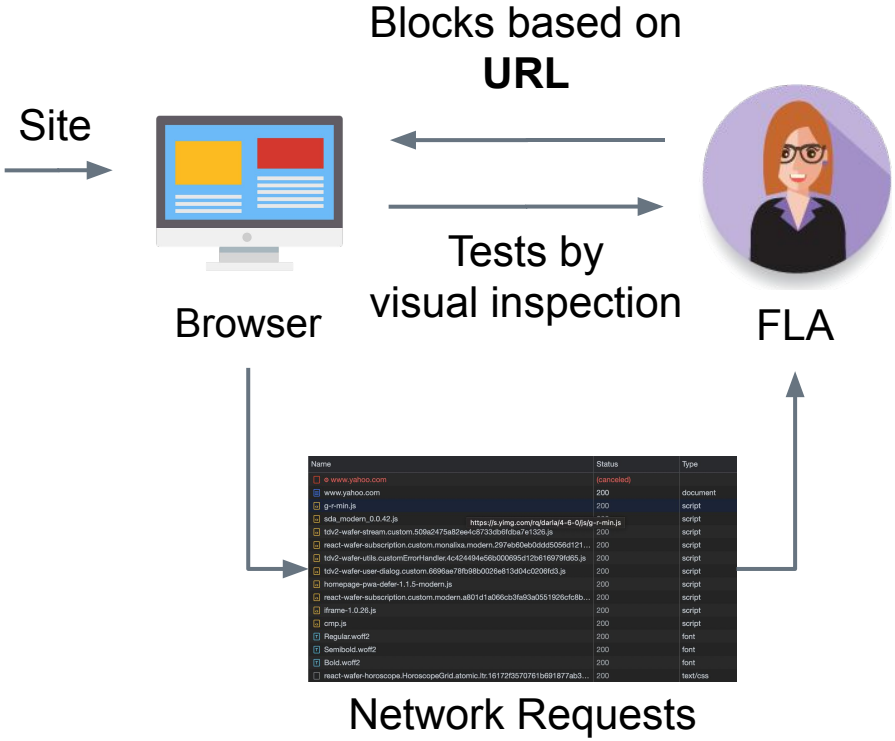
Blocking ads vs. Avoiding breakage

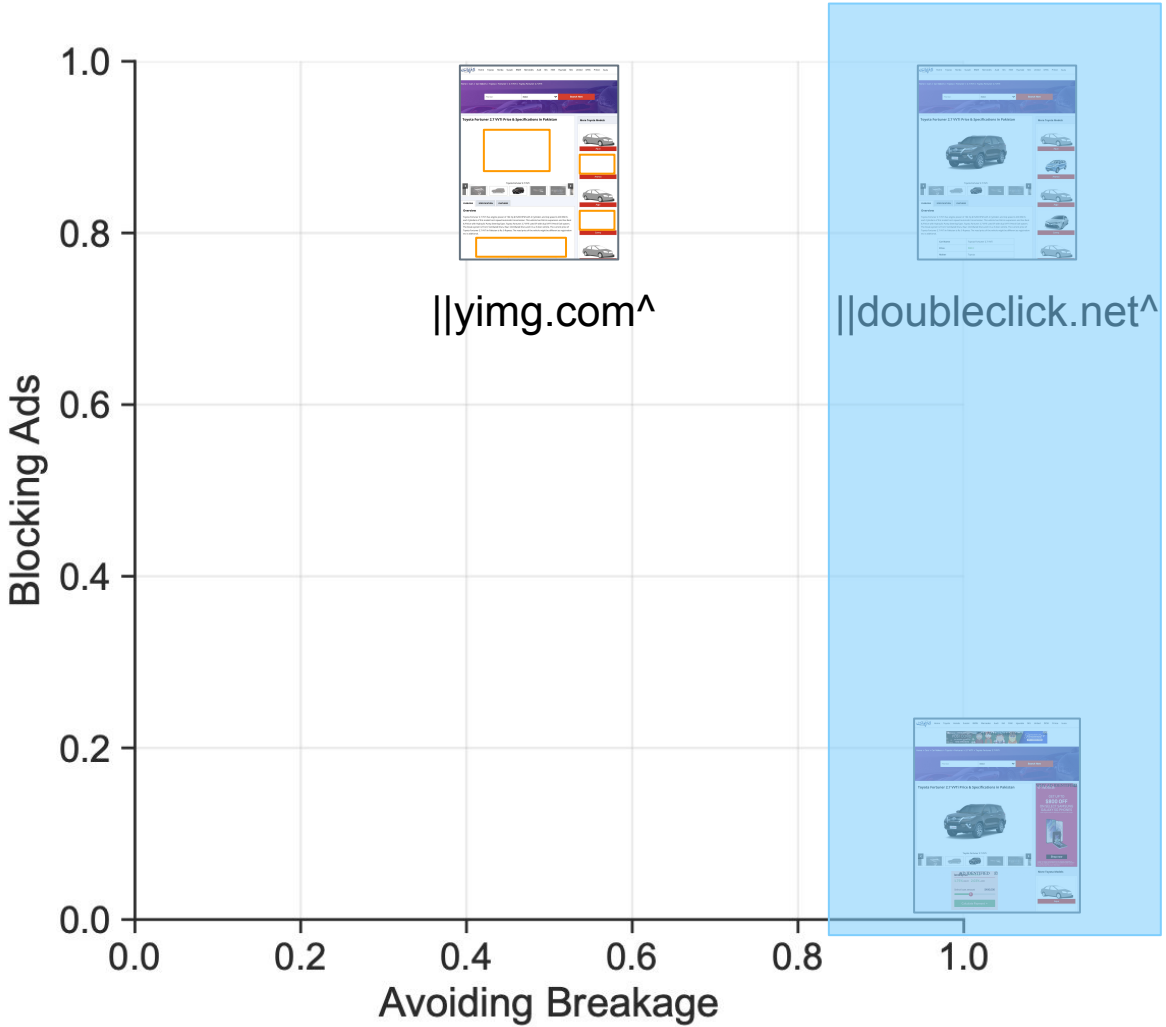
This trade-off causes the need to have different rule granularities.

High Human Effort

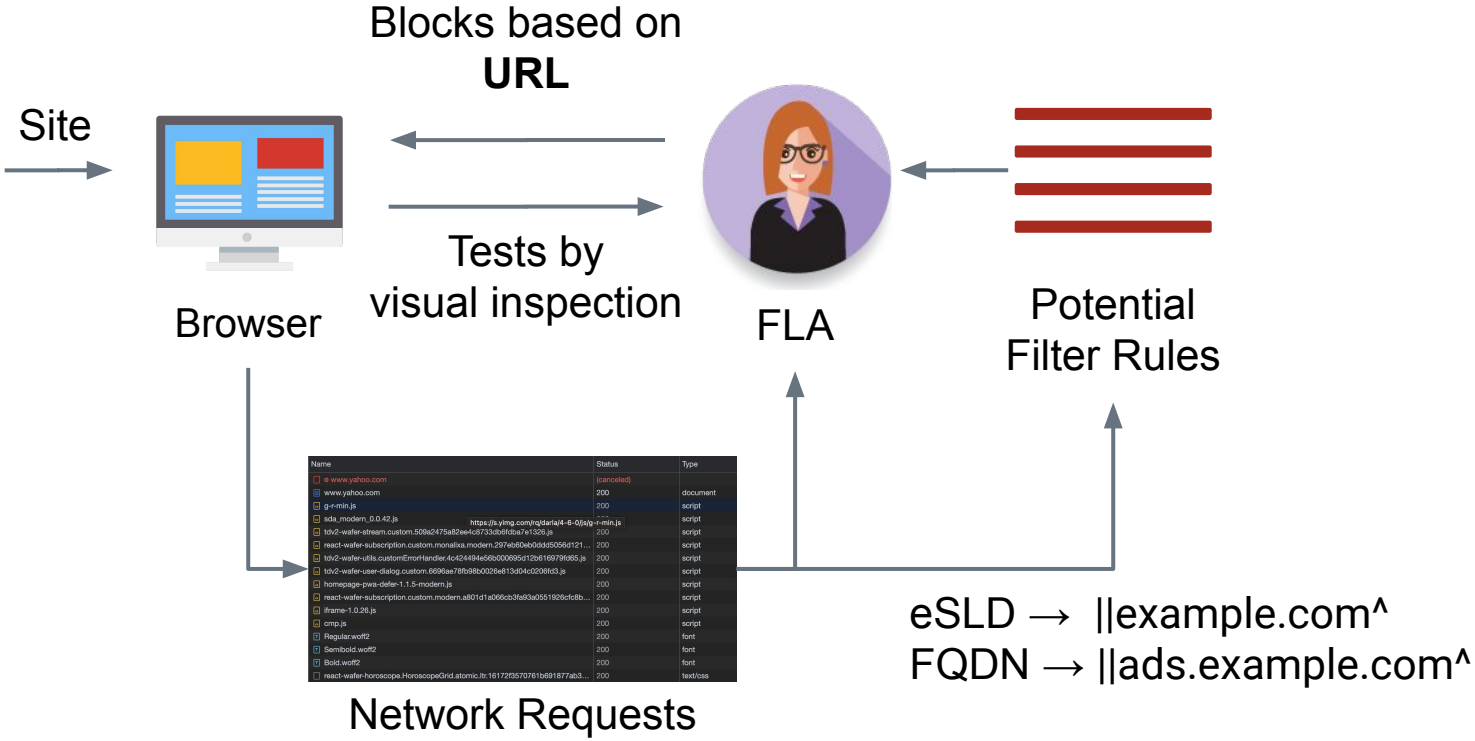
Affects scalability of creating and maintaining rules across the web and over time.

Filter List Author (FLA) Workflow

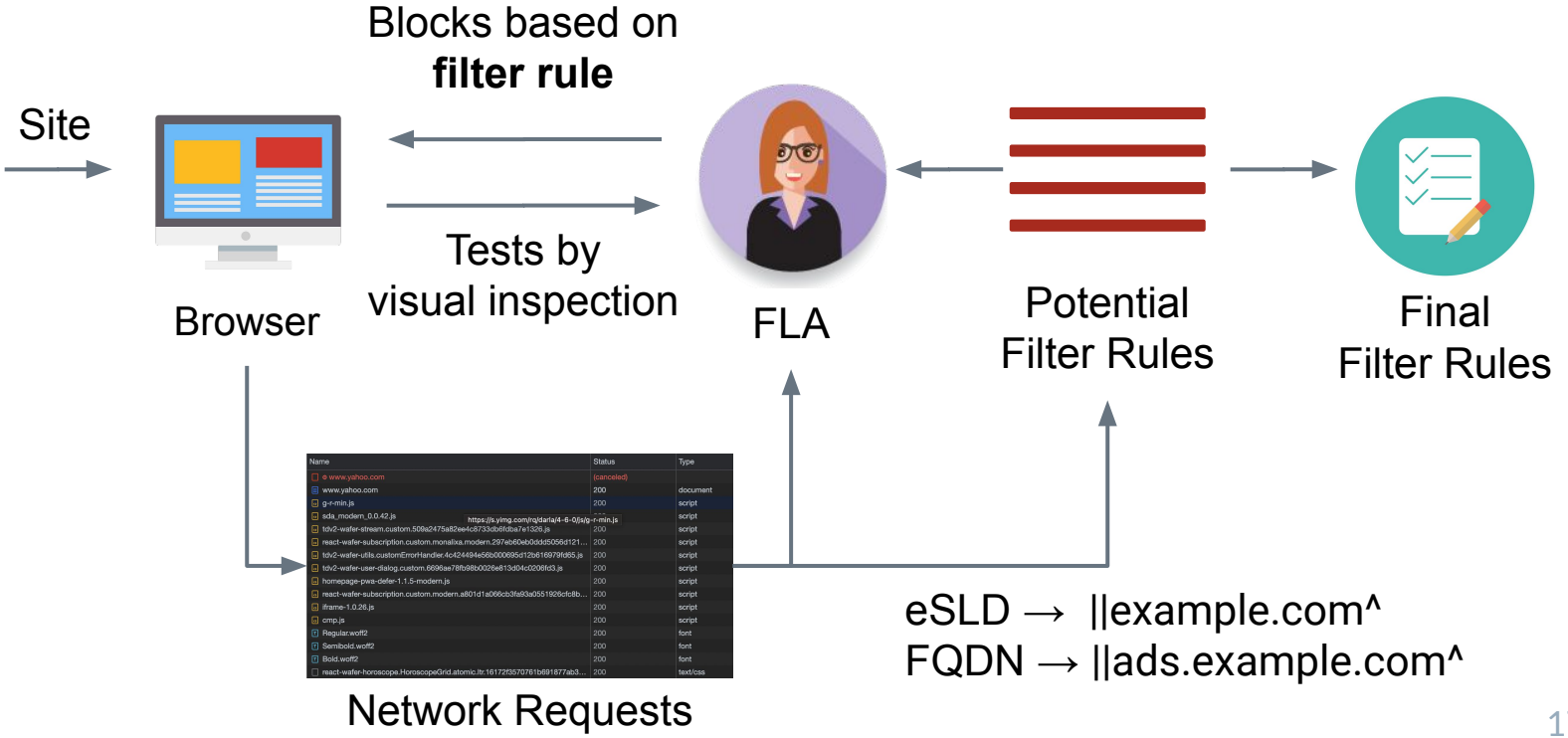



1.0
0.8
0.6
0.4
0.2
0.0
Blocking Ads
0.0
0.2
0.4
0.6
0.8
1.0
Avoiding Breakage
||yimg.com^
||doubleclick.net^

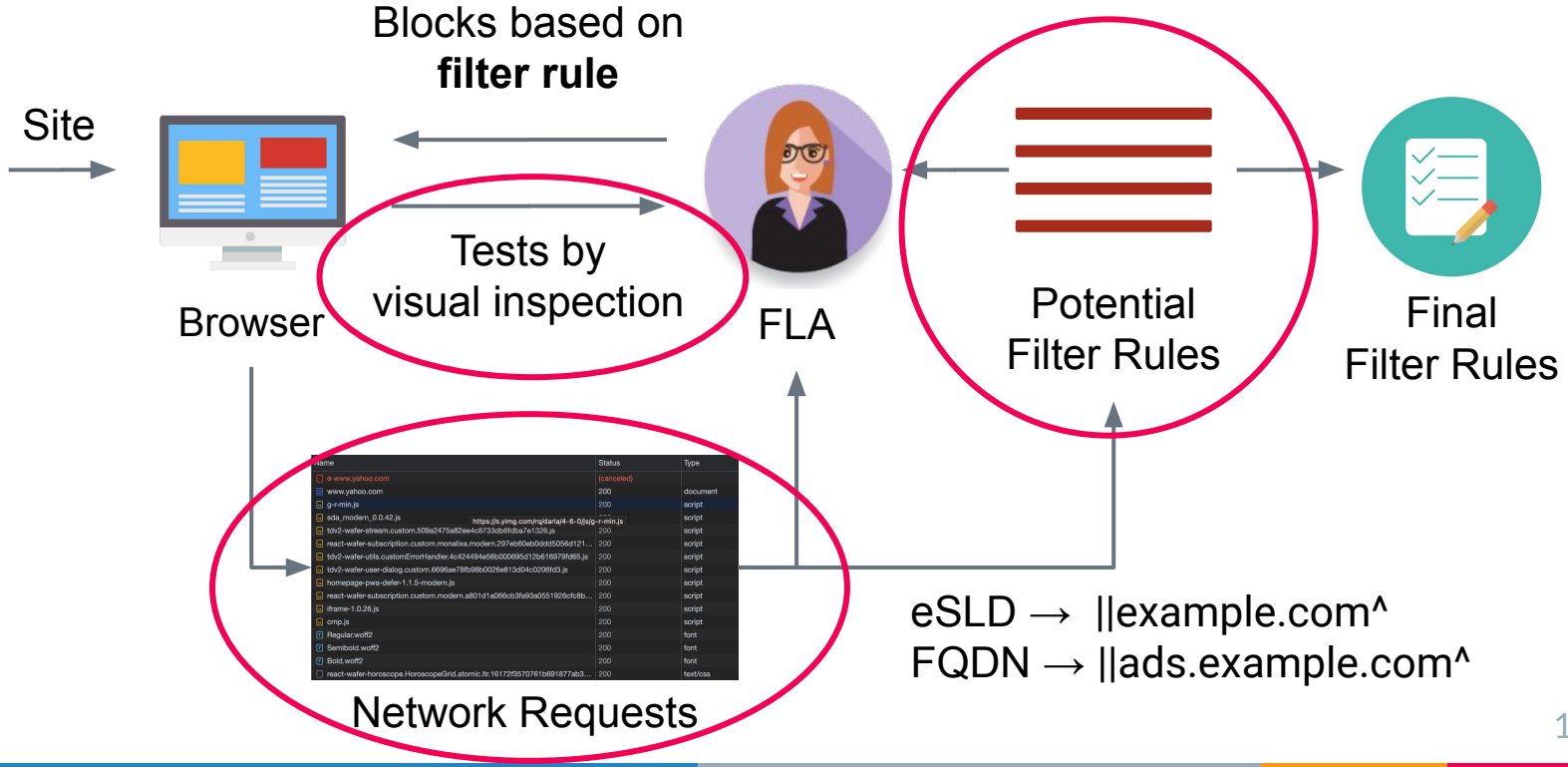
Filter List Author (FLA) Workflow



Filter List Author (FLA) Workflow



High Human Effort



AutoFR

What makes filter rule generation challenging?

Blocking ads vs. Avoiding breakage

This trade-off causes the need to have different rule granularities.

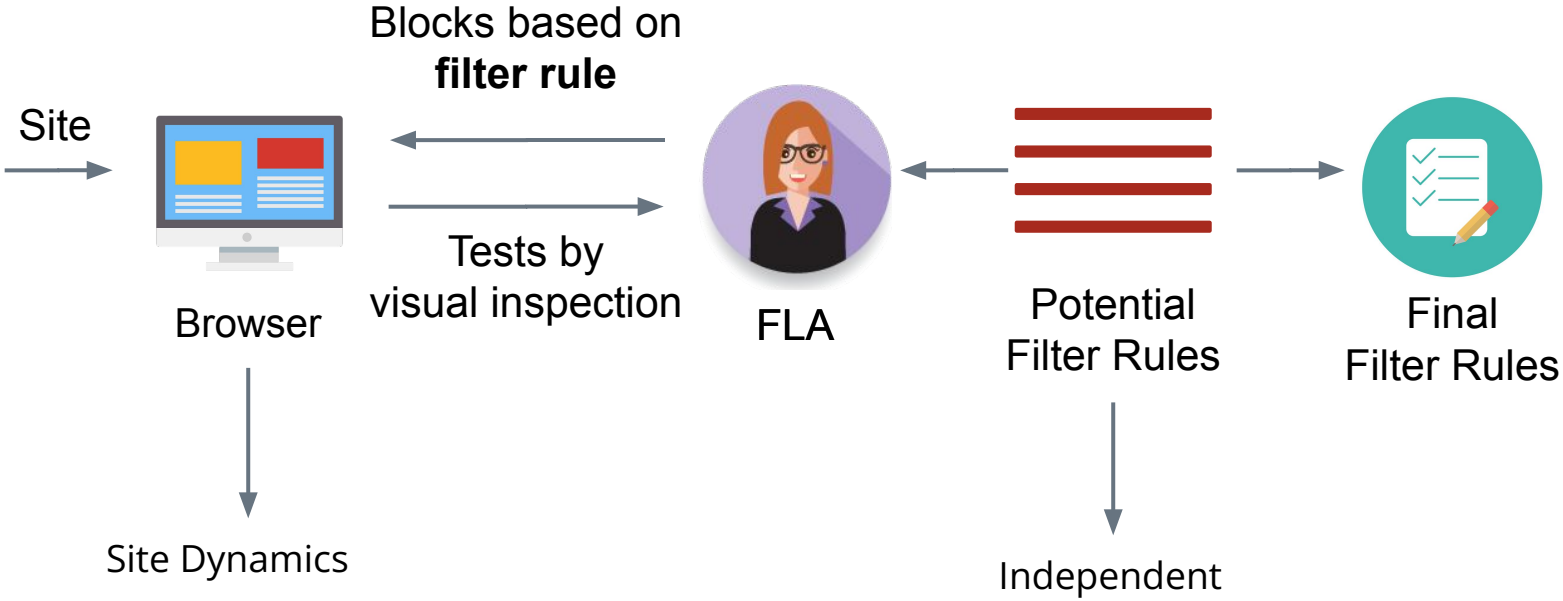
High Human Effort

Affects scalability of creating and maintaining rules across the web and over time.

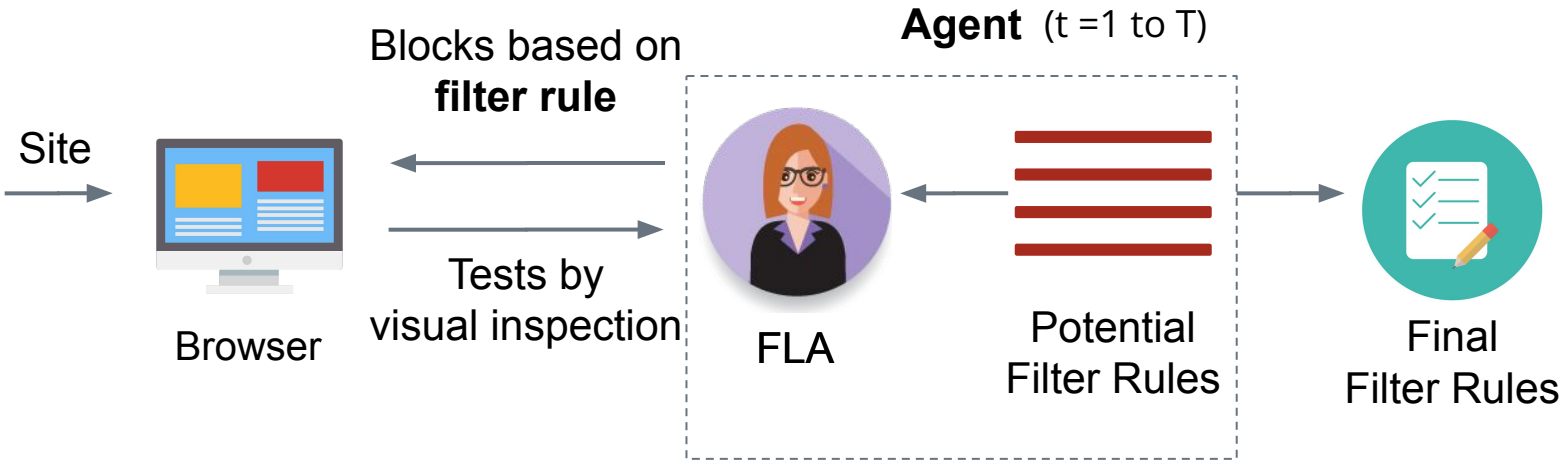
Outline

- Challenges to Filter Rule Generation
- **AutoFR: Formulation using Reinforcement Learning**
- AutoFR: Implementation for Scalability
- Evaluation
- Future Directions

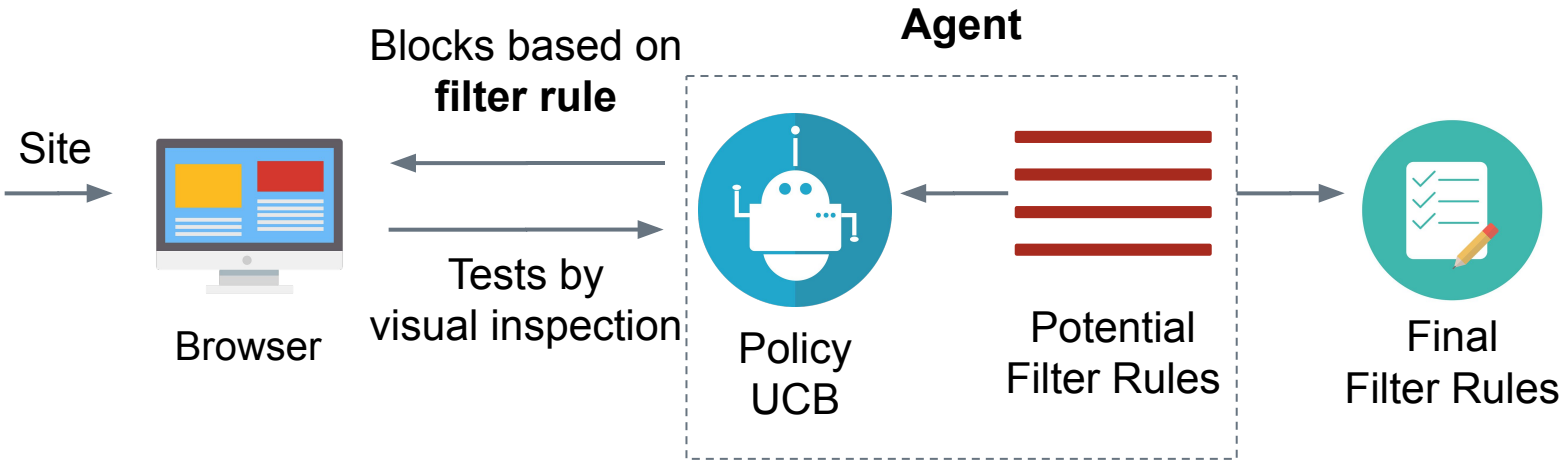
AutoFR: Formulation as a Multi-Arm Bandit



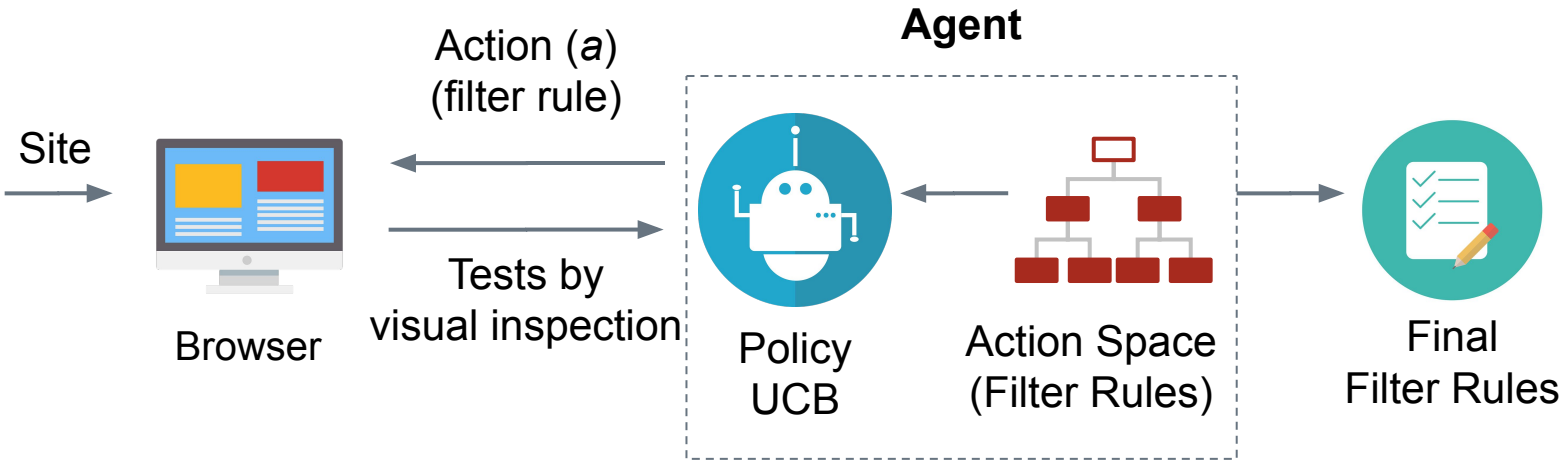
AutoFR: Agent



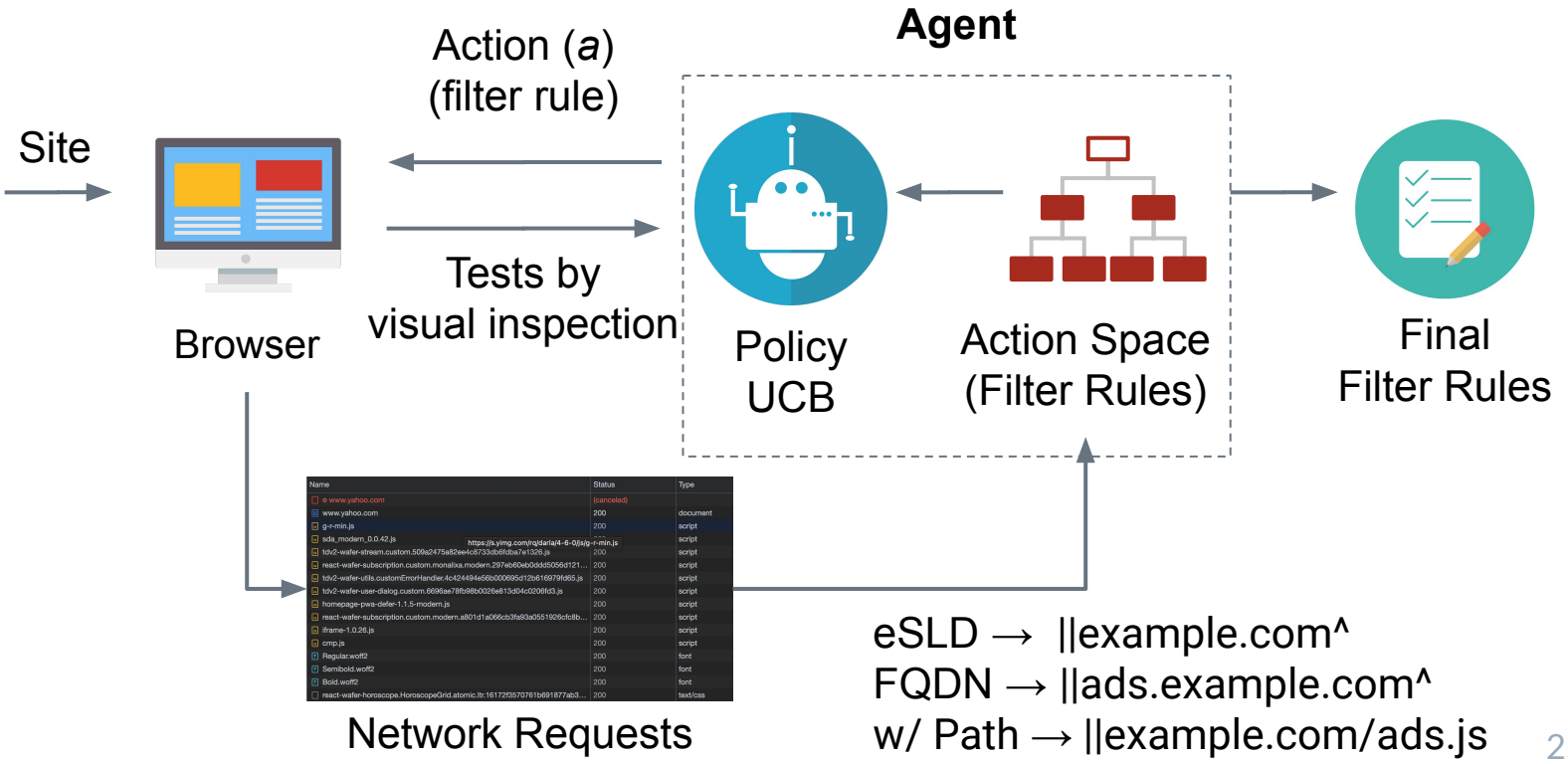
AutoFR: Policy



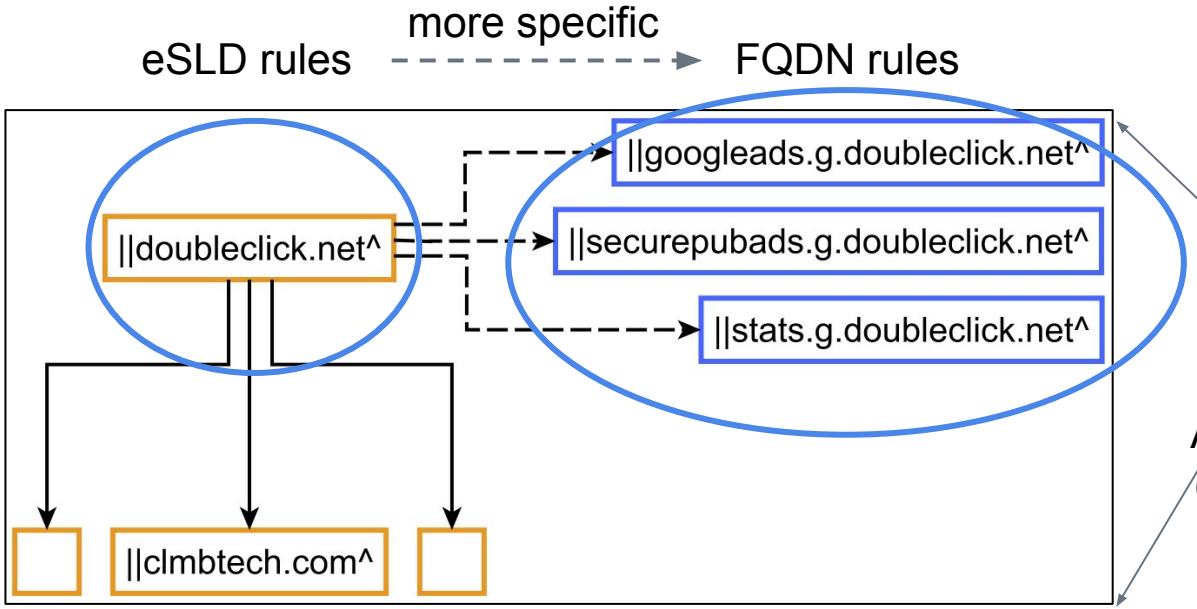
AutoFR: Actions

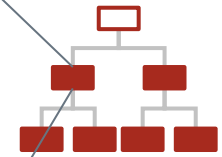


AutoFR: Deriving Actions



AutoFR: Hierarchical Action Space



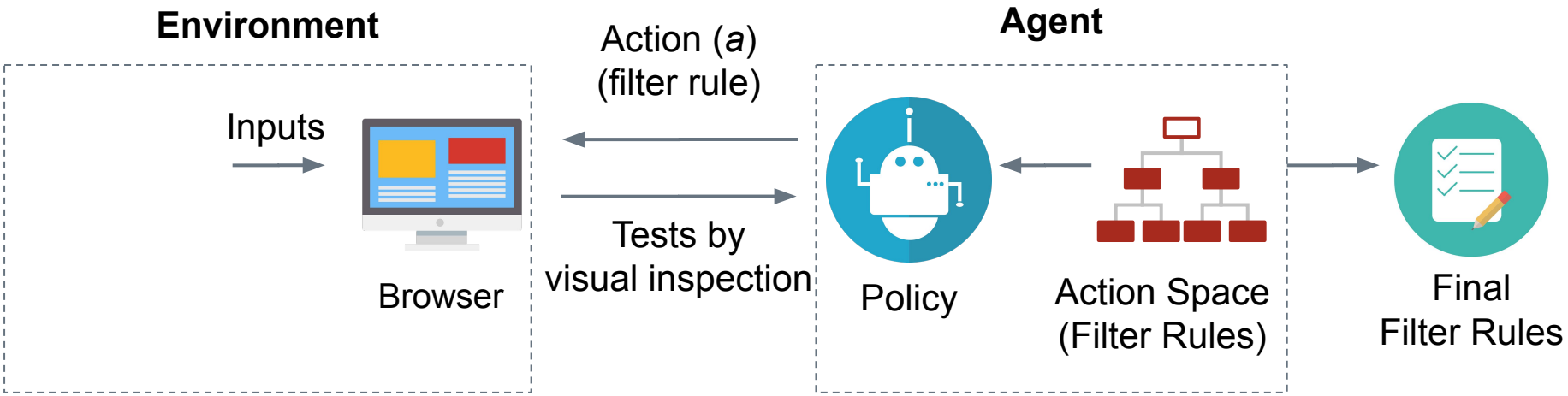


Action Space
(Filter Rules)

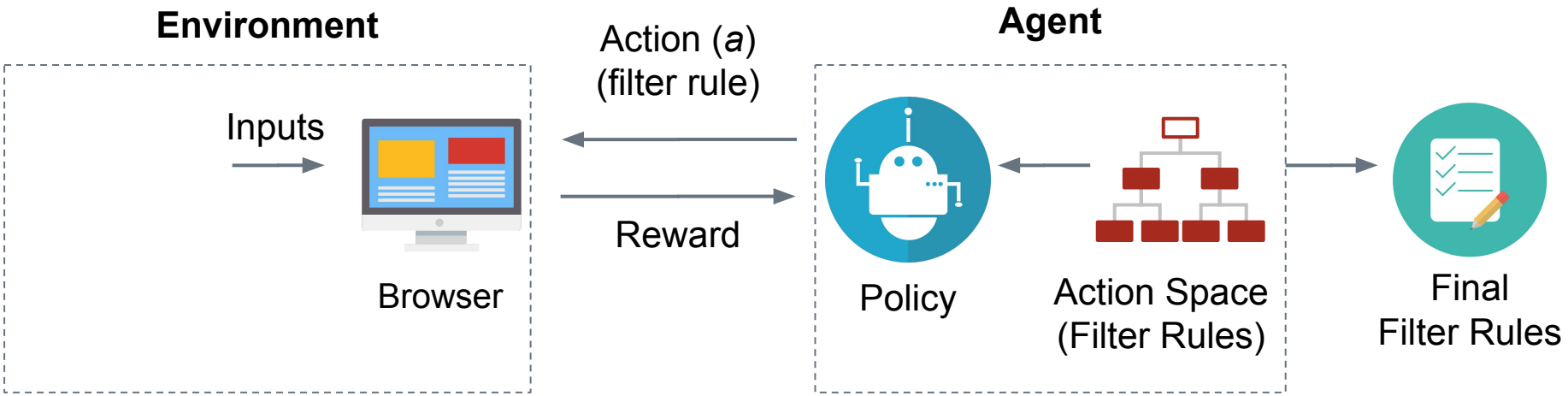
Intuition: try coarser grain filter rules first. If it causes too much breakage, then try more specific rules.

Hierarchical Structure: Reduces the number of actions that the agent needs to explore.

AutoFR: Environment



AutoFR: Reward



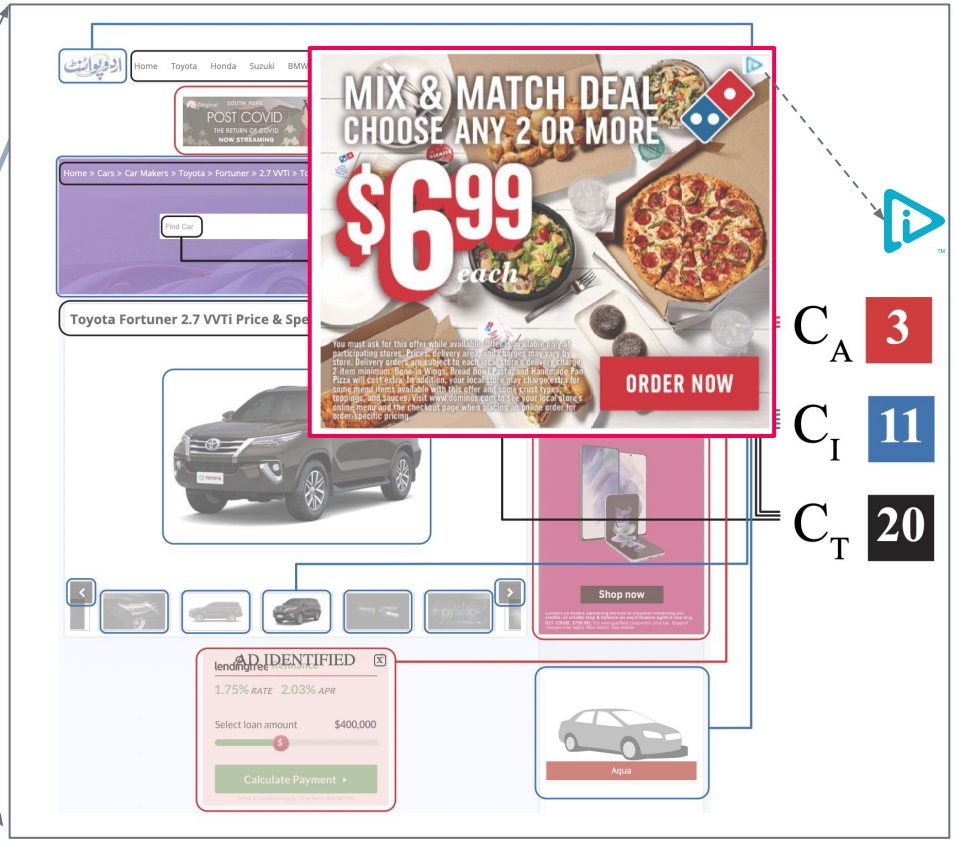
Reward considers trade-off blocking ads vs. avoiding breakage

AutoFR: Rew

Environment

Inputs

Browser



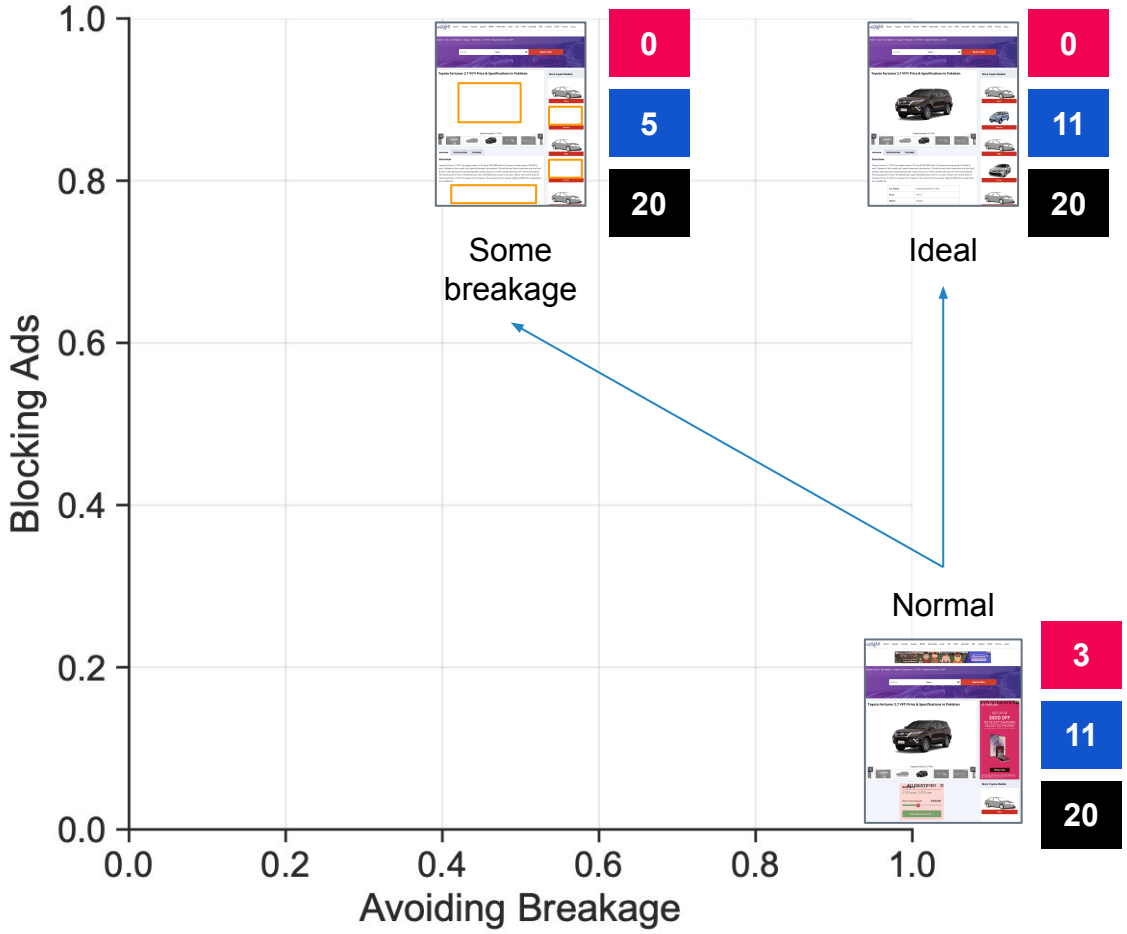
AdHighlighter

C_A 3 **Ads**

C_I 11 **Images**

C_T 20 **Text**

JS Injection

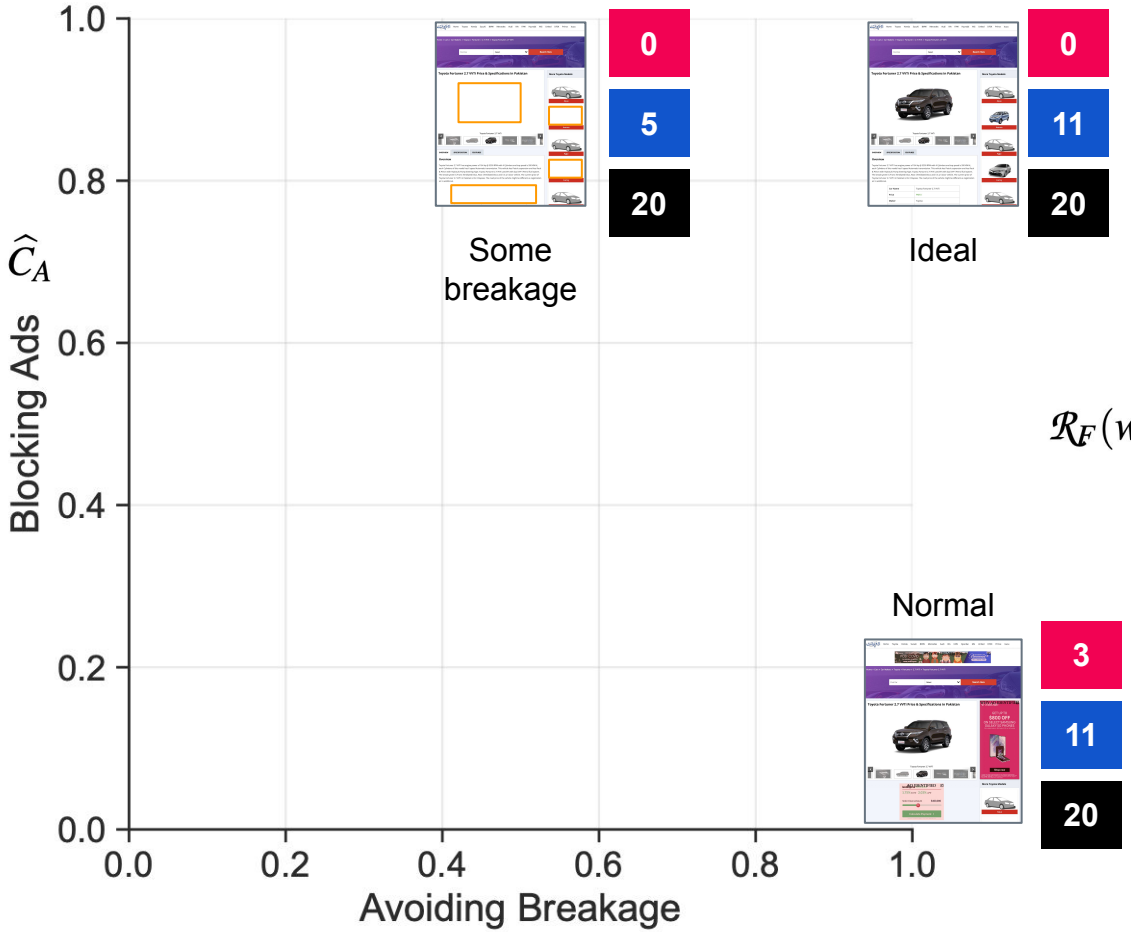


C_A Ads

C_I Images

C_T Text

Difference:
Normal behavior vs.
when rules are applied

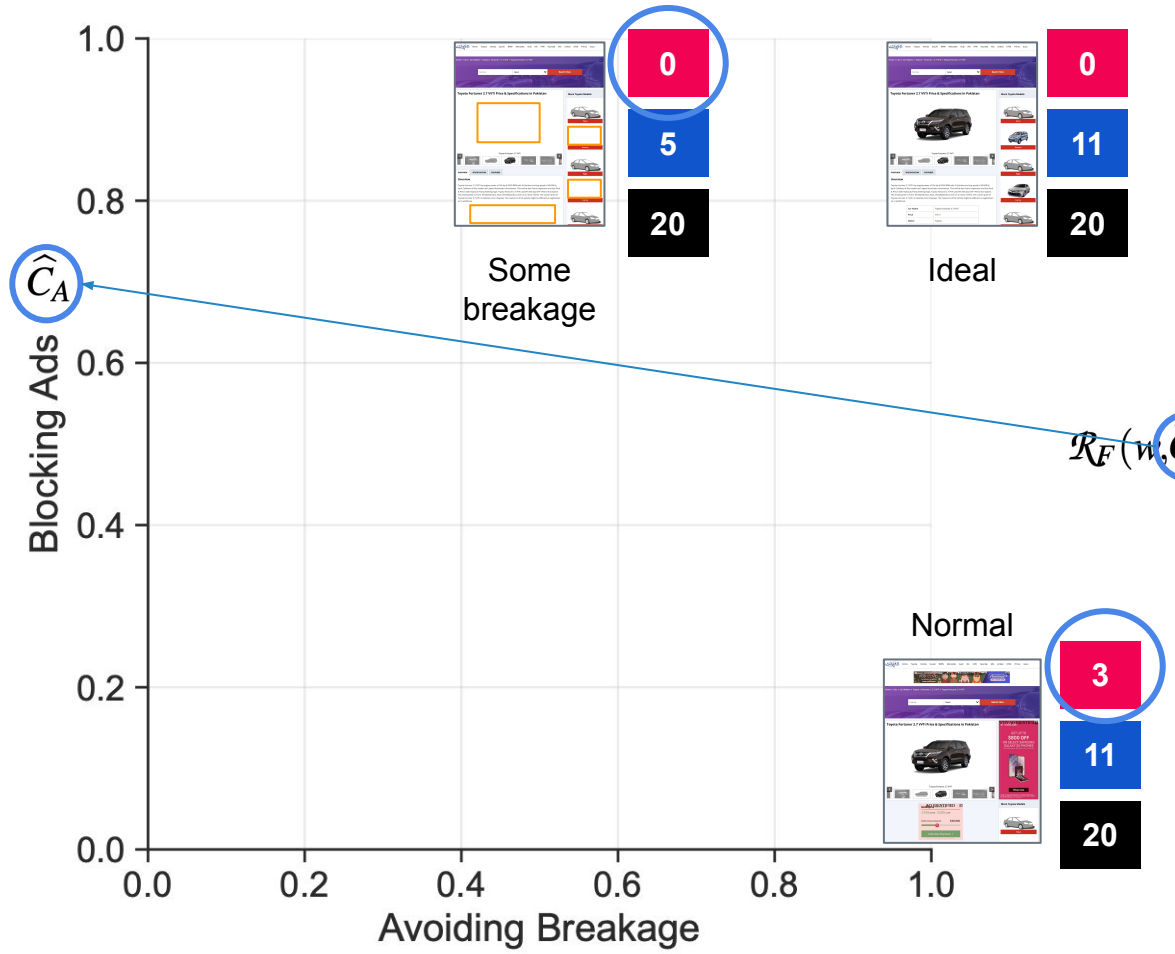


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

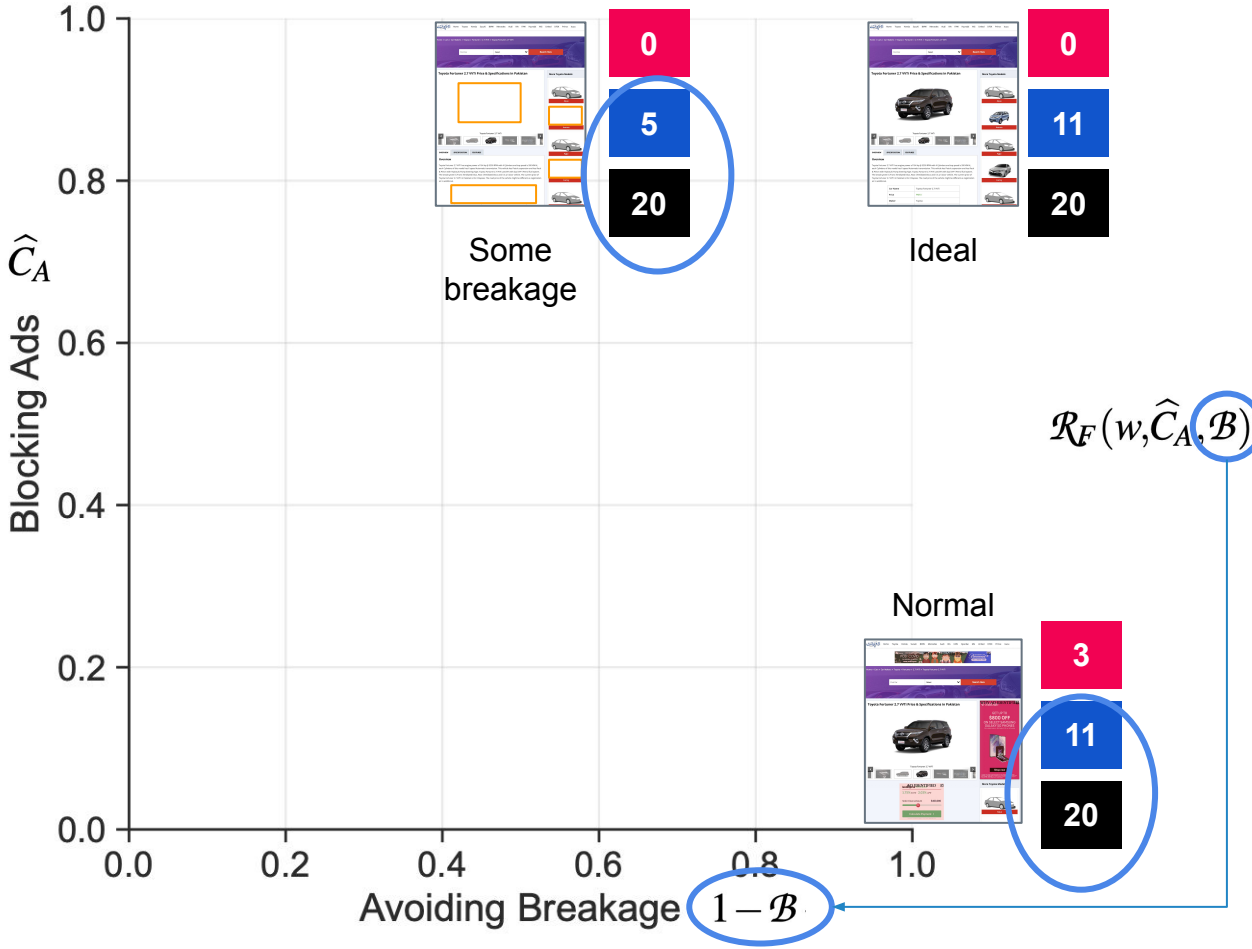


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w, \widehat{C}_A, \mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

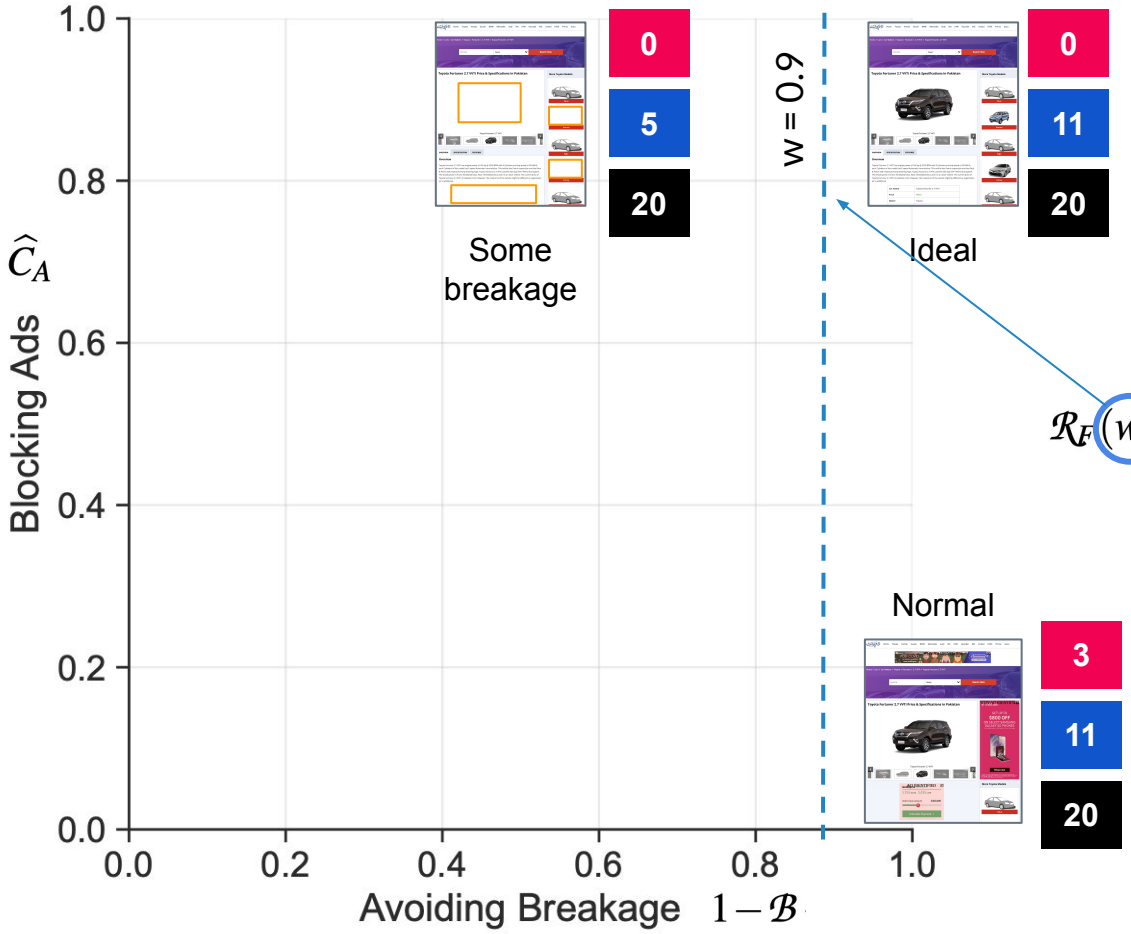


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

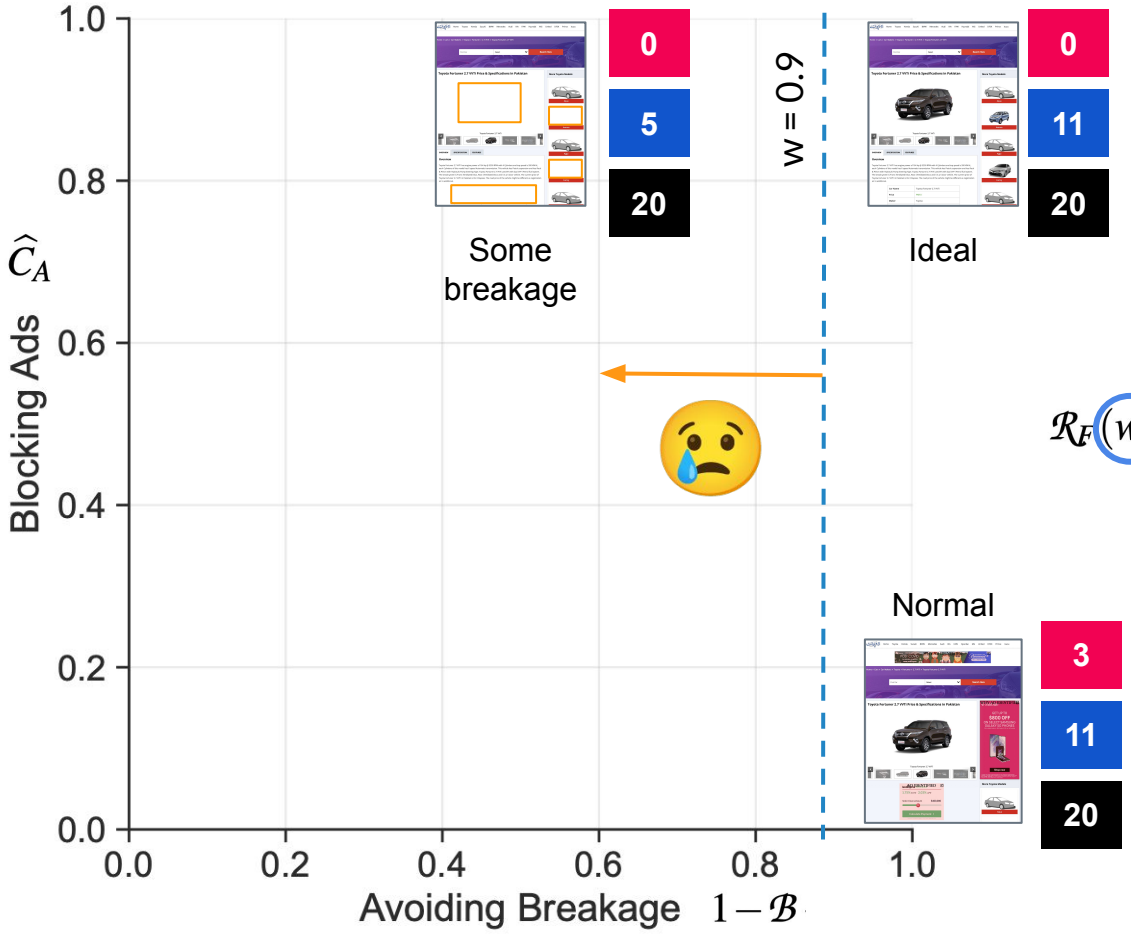


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

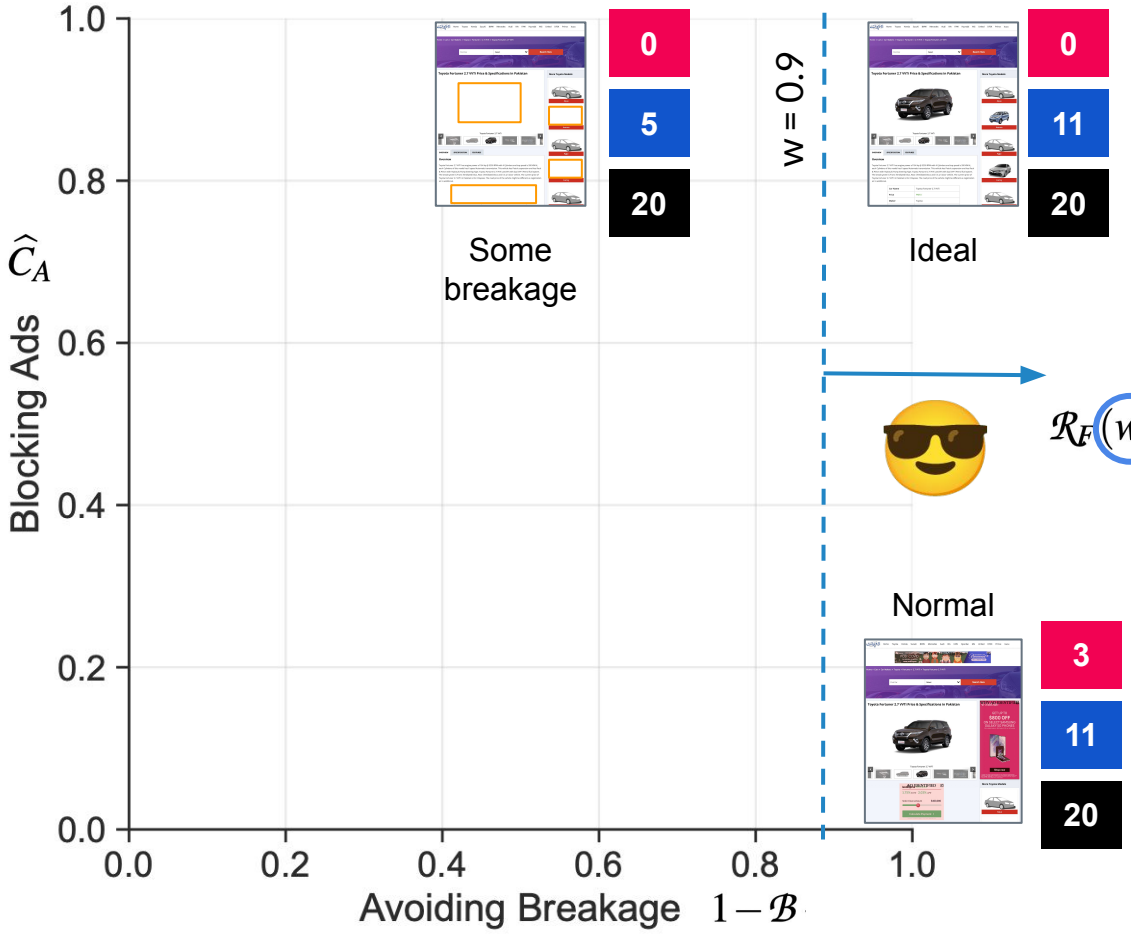


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

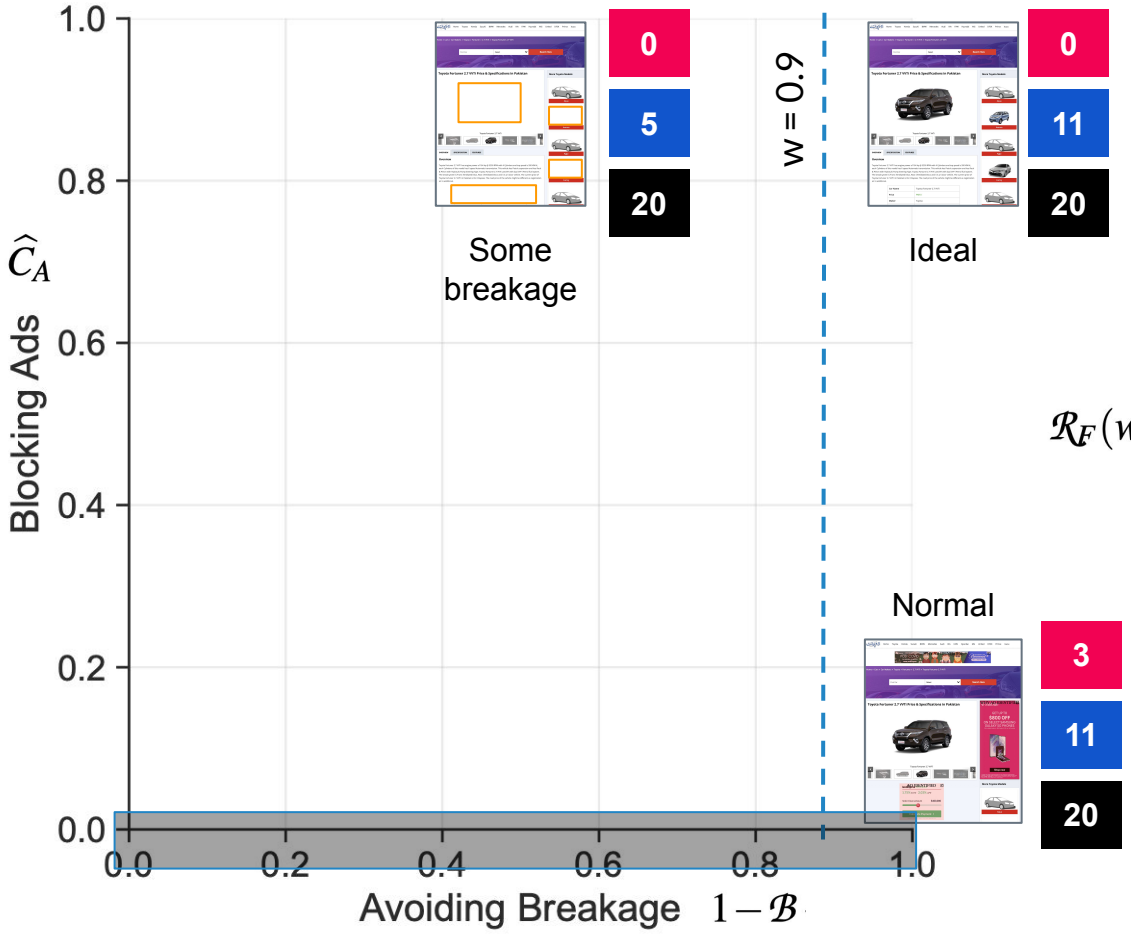


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

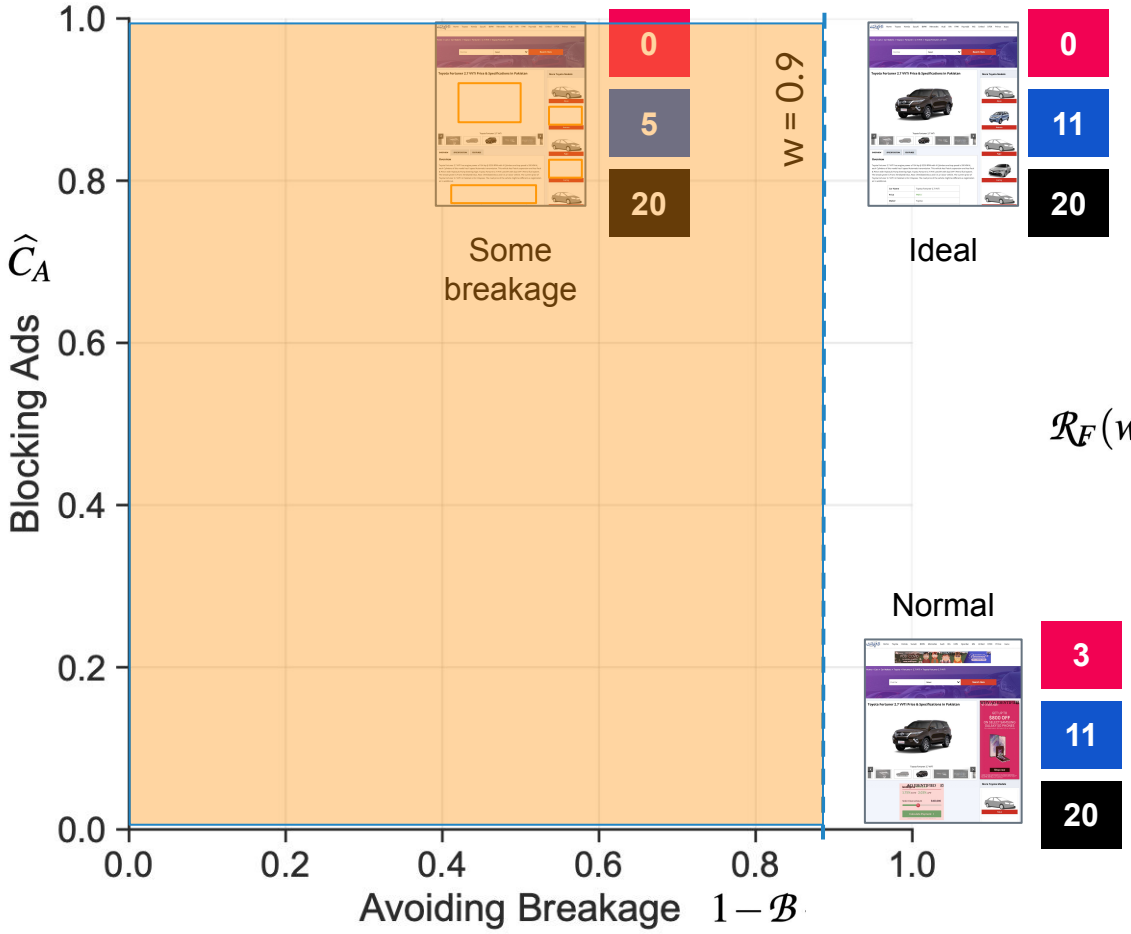


C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

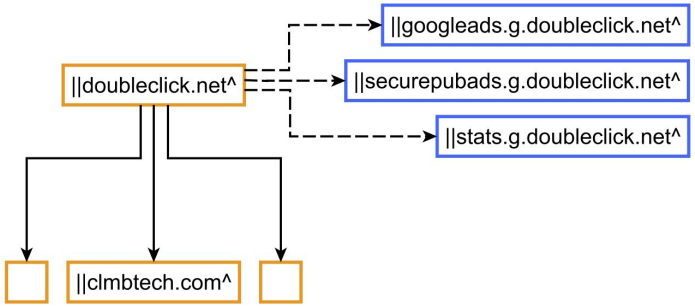


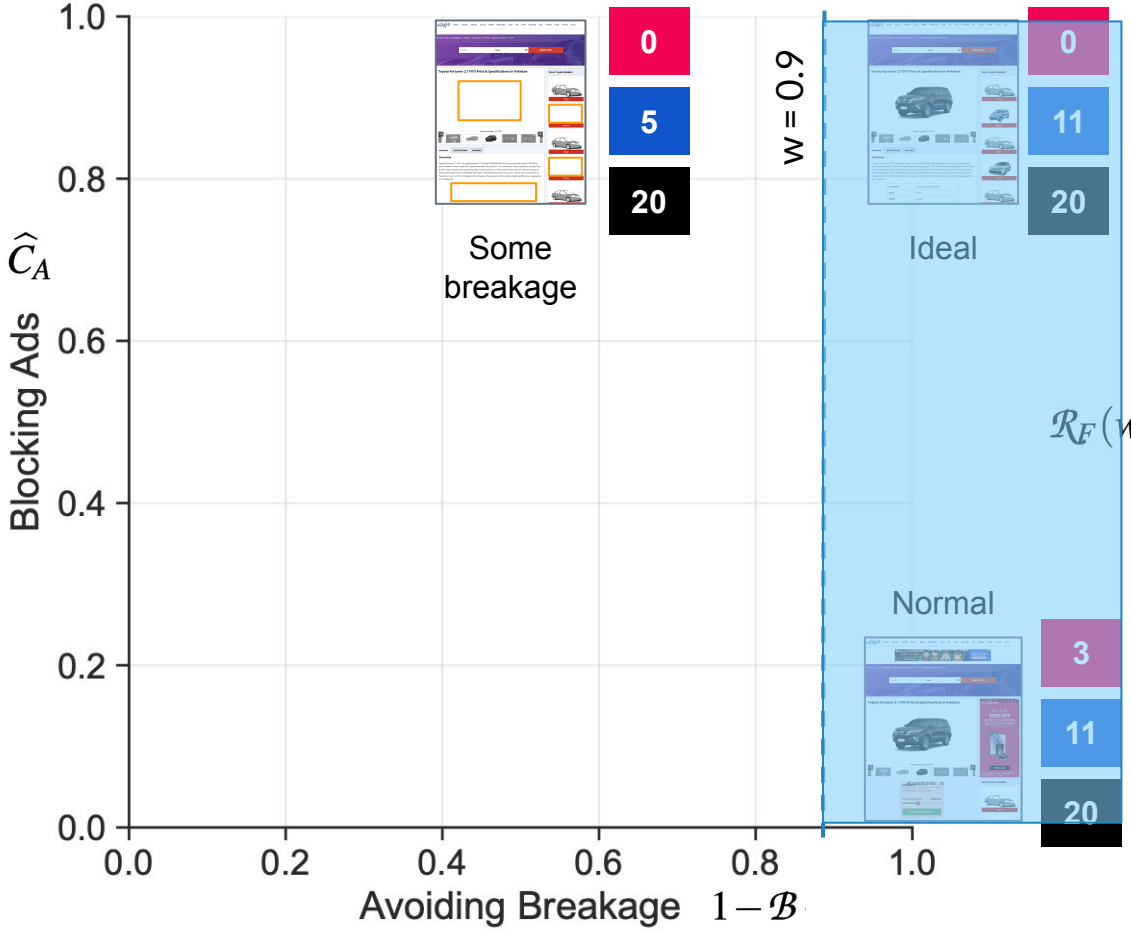
C_A Ads

C_I Images

C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$





C_A Ads

C_I Images

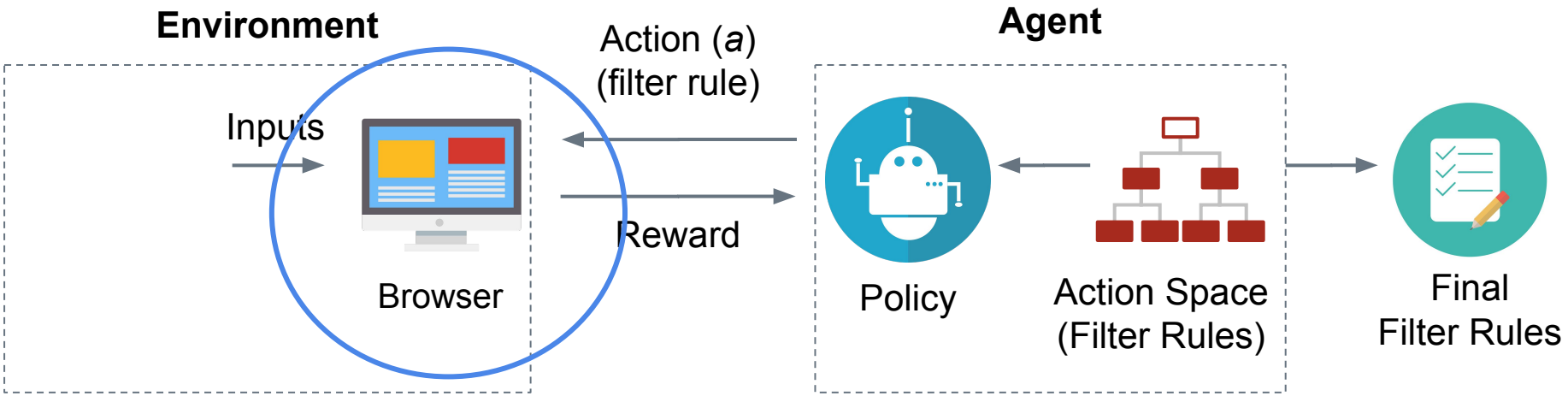
C_T Text

$$\mathcal{R}_F(w,\widehat{C}_A,\mathcal{B}) = \begin{cases} -1 & \text{if } \widehat{C}_A = 0 \quad (3a) \\ 0 & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} < w \quad (3b) \\ \widehat{C}_A & \text{if } \widehat{C}_A > 0\,,\ 1-\mathcal{B} \geq w \quad (3c) \end{cases}$$

Outline

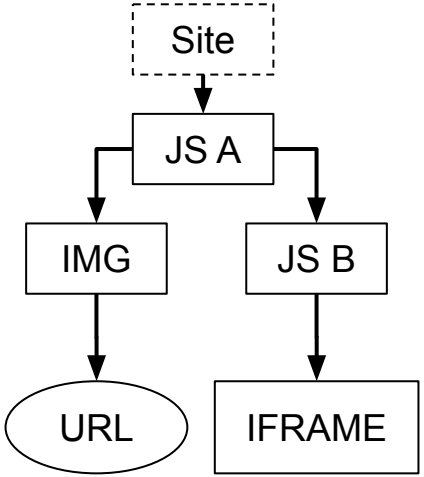
- Challenges to Filter Rule Generation
- AutoFR: Formulation using Reinforcement Learning
- **AutoFR: Implementation for Scalability**
- Evaluation
- Future Directions

AutoFR: Real-world Bottleneck



RL applied in the real world → x100-1K visits: → take hours per-site

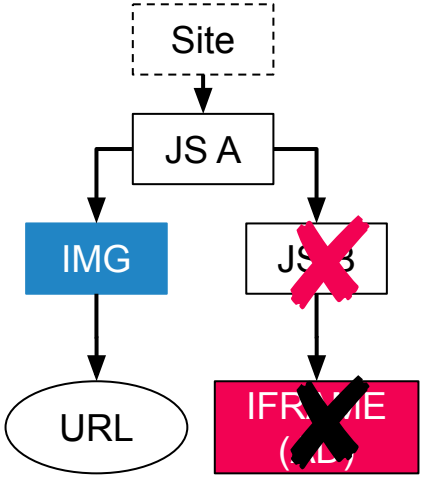
AutoFR: Site Snapshot



Example

AdGraph: Iqbal *et al.* IEEE S&P 2020

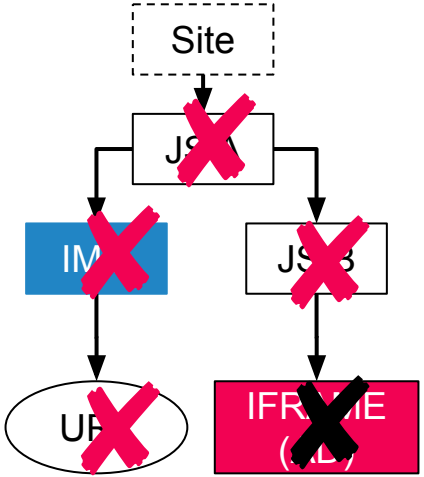
AutoFR: Site Snapshot



Example

AdGraph: Iqbal *et al.* IEEE S&P 2020

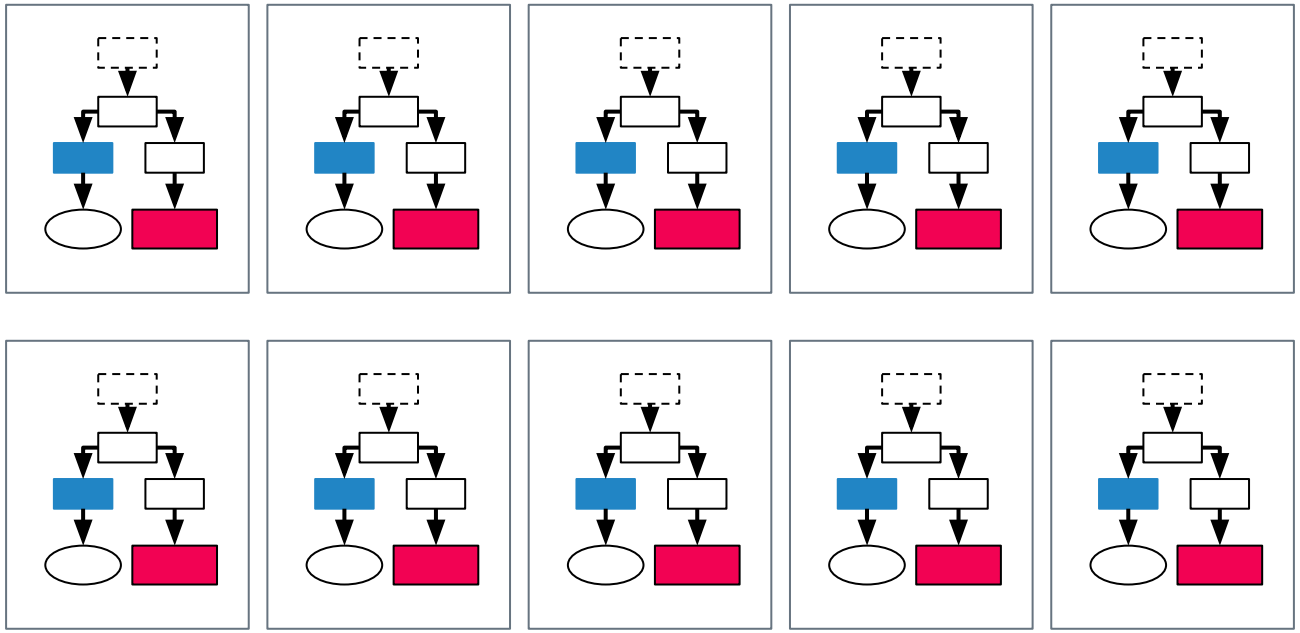
AutoFR: Site Snapshot



Example

AdGraph: Iqbal *et al.* IEEE S&P 2020

Collect, Annotate, and Reuse



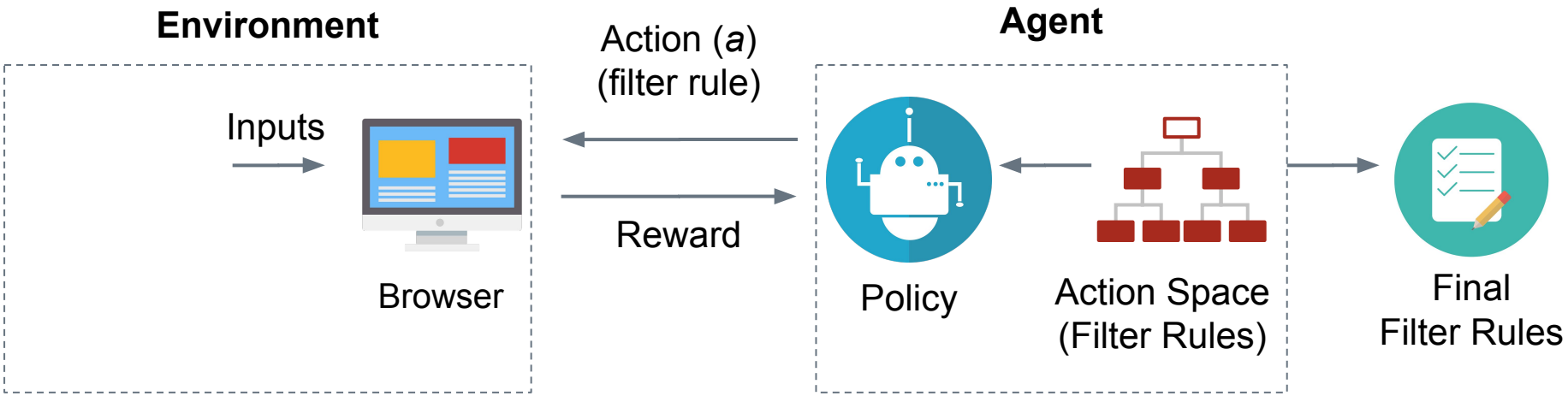
Capture Site Dynamics:

1. Collect: 10 raw graphs, each represents a visit to the site (no rules applied)

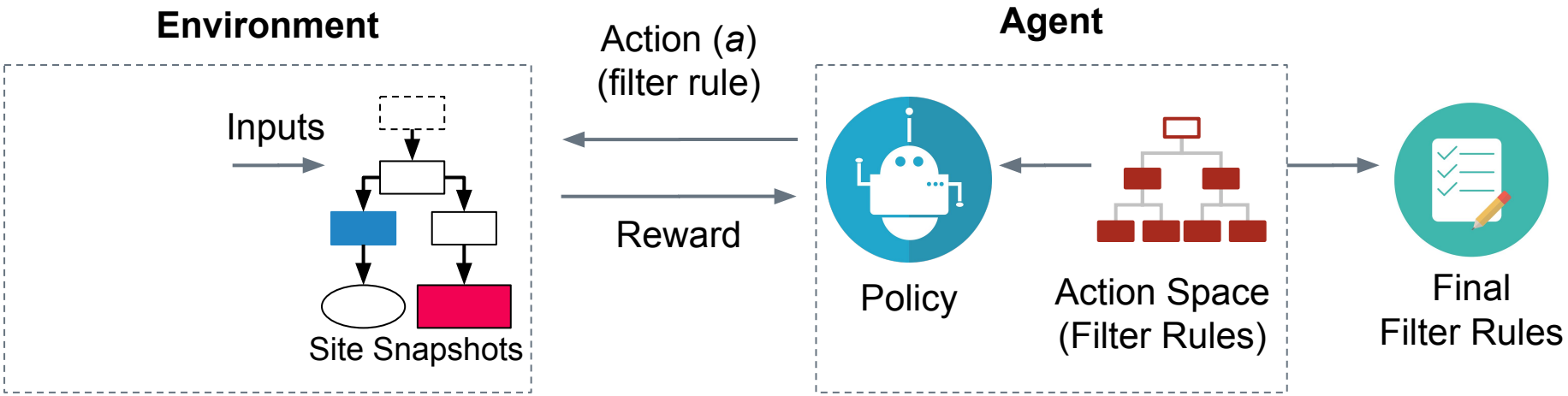
2. Annotate them and save locally as .graphml files

3. Reuse them

AutoFR: Reuse Site Snapshots



AutoFR: Reuse Site Snapshots



For each action (filter rule), randomly select a site snapshot, apply rule

AutoFR → efficient and scalable

Outline

- Challenges to Filter Rule Generation
- AutoFR: Formulation using Reinforcement Learning
- AutoFR: Implementation for Scalability
- **Evaluation**
- Future Directions

AutoFR: Efficiency

13 hours

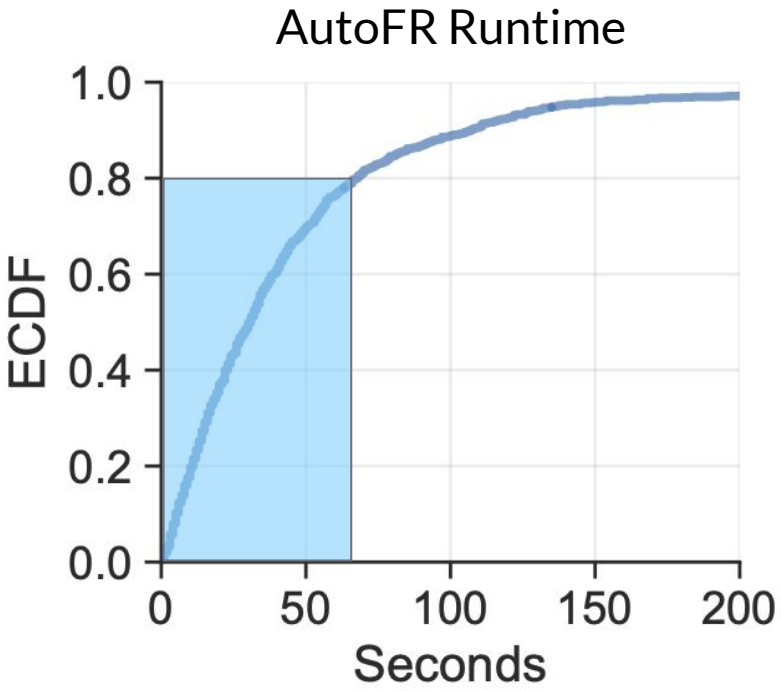
One site, 10 rules to explore, 1K visits to site (toy example)

(47 sec / visit * 1K visits)

1.6 minutes

One site, collect and use site snapshots (averaged on Top–5K sites)

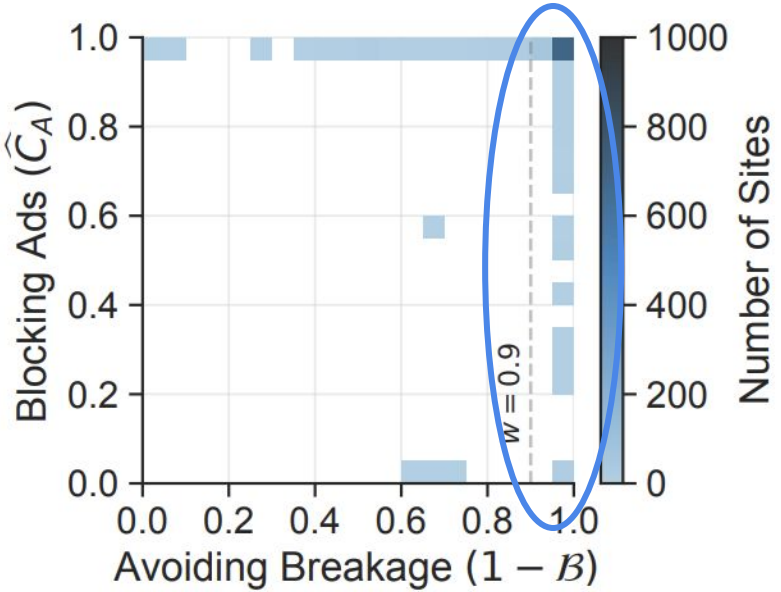
AutoFR: Efficiency



1.6 minutes

One site, collect and use site snapshots (averaged on Top–5K sites)

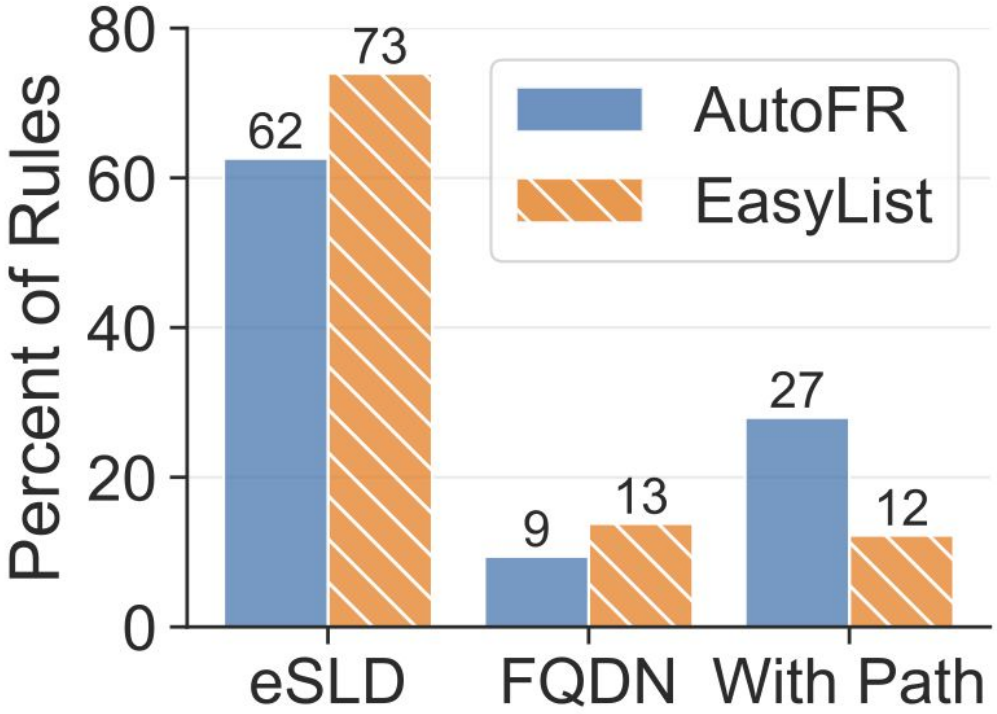
AutoFR: Performance (Top–5K)



(b) AutoFR (In the Wild)

86%

AutoFR: Rule Granularity (Top–5K)



AutoFR generates rules with similar type distribution as EasyList on the Top-5K.

(Extensive) Evaluation

- **Efficiency**
- **Performance vs. EasyList**
- **Rules vs. EasyList**
  - **Rule Type Distribution**
  - Differences in Rules
- Per-site to Global Rules
- Longitudinal Analysis
  - Robustness after 6 months
  - Robustness every four-days

Future Directions

AutoFR

Different Rewards

Explore different rewards to generate rules

AutoFR for Element Hiding

Create rules that hide ads

AutoFR for Tracking

Create rules to block tracking

AutoFR for Mobile

Create rules for other platforms

AutoFR

A framework and tool that generate URL-based filter rules to block ads while avoiding visual breakage automatically

Paper and Artifacts

H. Le, S. Elmalaki, A. Markopoulou, Z. Shafiq, "AutoFR: Automated Filter Rule Generation for Adblocking", In Proc. of the 32nd USENIX Conference on Security Symposium (SEC 2023). Anaheim CA, August 2023.

Github: https://github.com/UCI-Networking-Group/AutoFR
Dataset: athinagroup.eng.uci.edu/projects/ats-on-the-web/autofr-dataset/

UCI UCDAVIS ProperData

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash