Privid: Practical, Privacy-Preserving Video Analytics Queries

Frank Cangialosi, Neil Agarwal, Venkat Arun,
Srinivas Narayan, Junchen Jiang, Anand Sarwate, Ravi Netravali

MIT CSAIL
Princeton University
The University of Chicago
Rutgers

Two recent trends

1. Increasingly **pervasive** cameras
2. Increasingly **accurate & fast** computer vision

Manhattan, NYC

Source: NYTimes, NYC DOT

CITY	COUNTRY	CCTV CAMERAS	CCTV CAMERAS PER 1 KM2
Chennai	INDIA	280,000	657.28
Hyderabad	INDIA	300,000	480
Harbin	CHINA	250,000	410.51
London	ENGLAND (UK)	627,707	399.27
Xiamen	CHINA	150,000	385.13

Two recent trends

1. Increasingly **pervasive** cameras
2. Increasingly **accurate & fast** computer vision

Increasingly feasible to **track** objects across network of cameras

Manhattan, NYC

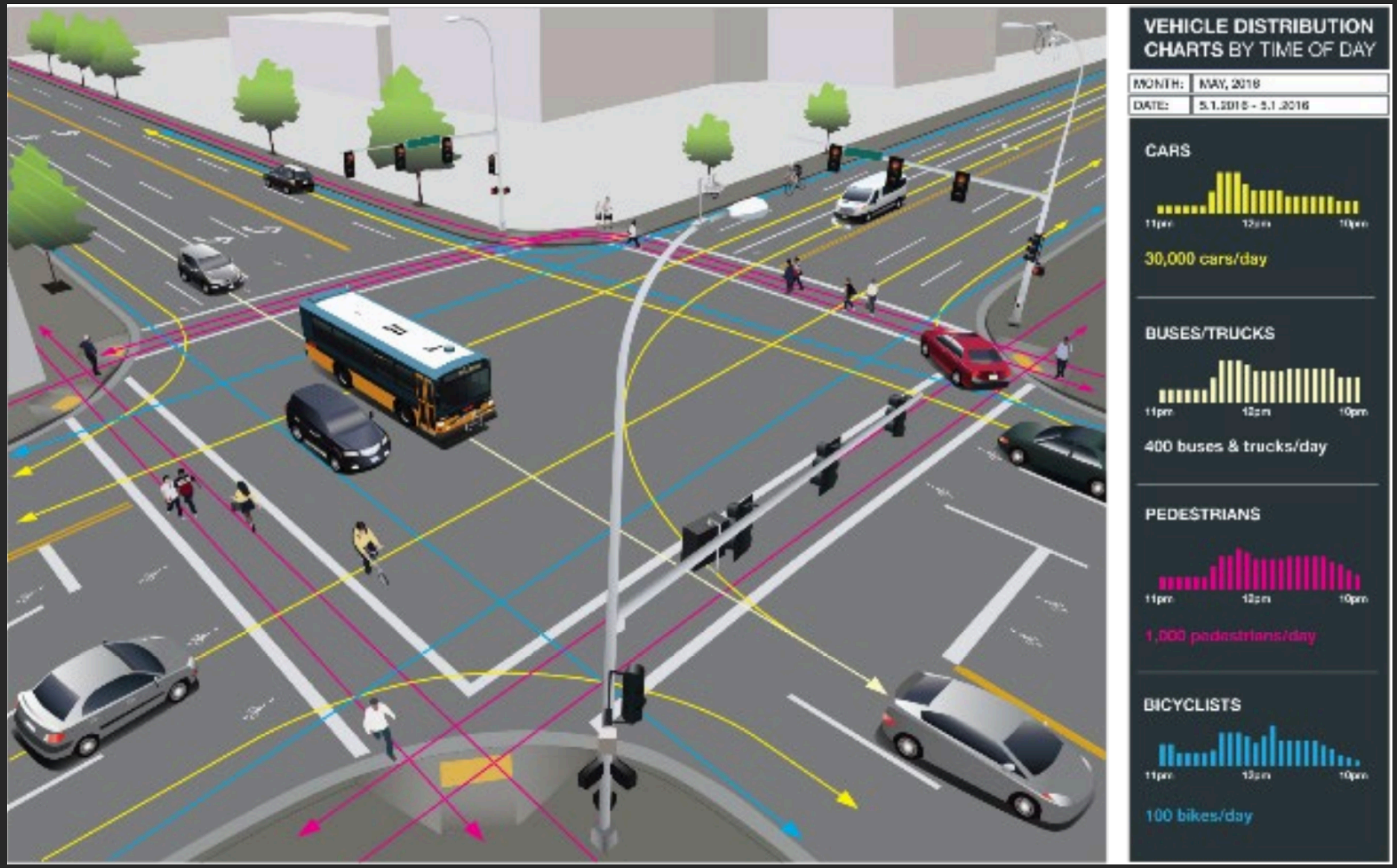
Source: NYTimes, NYC DOT

CITY	COUNTRY	CCTV CAMERAS	CCTV CAMERAS PER 1 KM2
Chennai	INDIA	280,000	657.28
Hyderabad	INDIA	300,000	480
Harbin	CHINA	250,000	410.51
London	ENGLAND (UK)	627,707	399.27
Xiamen	CHINA	150,000	385.13

Increasingly feasible to **track** objects across network of cameras

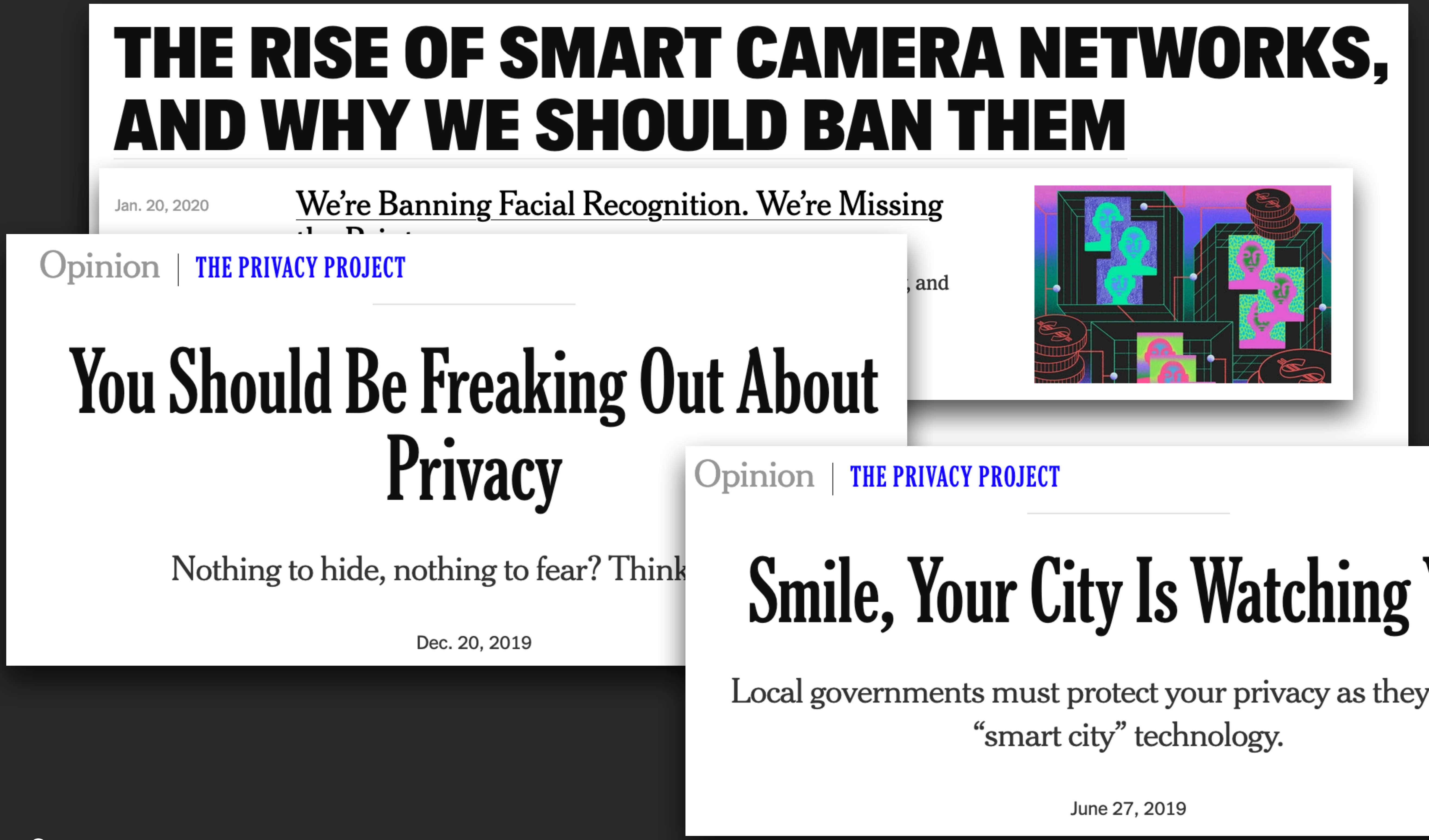
Beneficial Applications

Urban planning, smart cities, business analysis, public health



Source: PSU TREC

Privacy Concerns


THE RISE OF SMART CAMERA NETWORKS, AND WHY WE SHOULD BAN THEM

Jan. 20, 2020 We're Banning Facial Recognition. We're Missing

Opinion | THE PRIVACY PROJECT

You Should Be Freaking Out About Privacy

Nothing to hide, nothing to fear? Think

Dec. 20, 2019

Opinion | THE PRIVACY PROJECT

Smile, Your City Is Watching

Local governments must protect your privacy as they "smart city" technology.

June 27, 2019

Problem: How to tradeoff privacy & utility?

Don’t release video at all

Release video **publicly**

“Best” Privacy

“Best” Utility

Problem: How to tradeoff privacy & utility?

Don't release video at all

Release video **publicly**

?

"Best" Privacy

"Best" Utility

Can we extract useful information from video *without* giving up privacy?

Threat Model

Protect privacy

Do not trust

Individuals

(Citizens)

Video Owner (VO)

(Organizational Entity)

Analysts

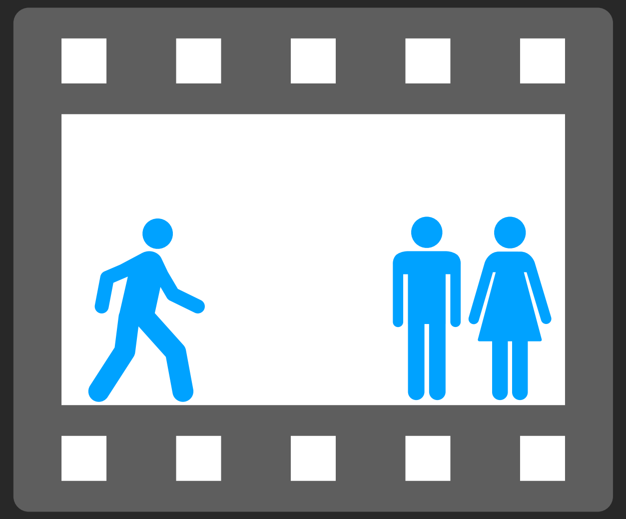
(Employees OR 3rd parties)

Benevolent Query:
"Count **people** per hour"

Malicious Query:
"When did **Frank** pass this camera?"

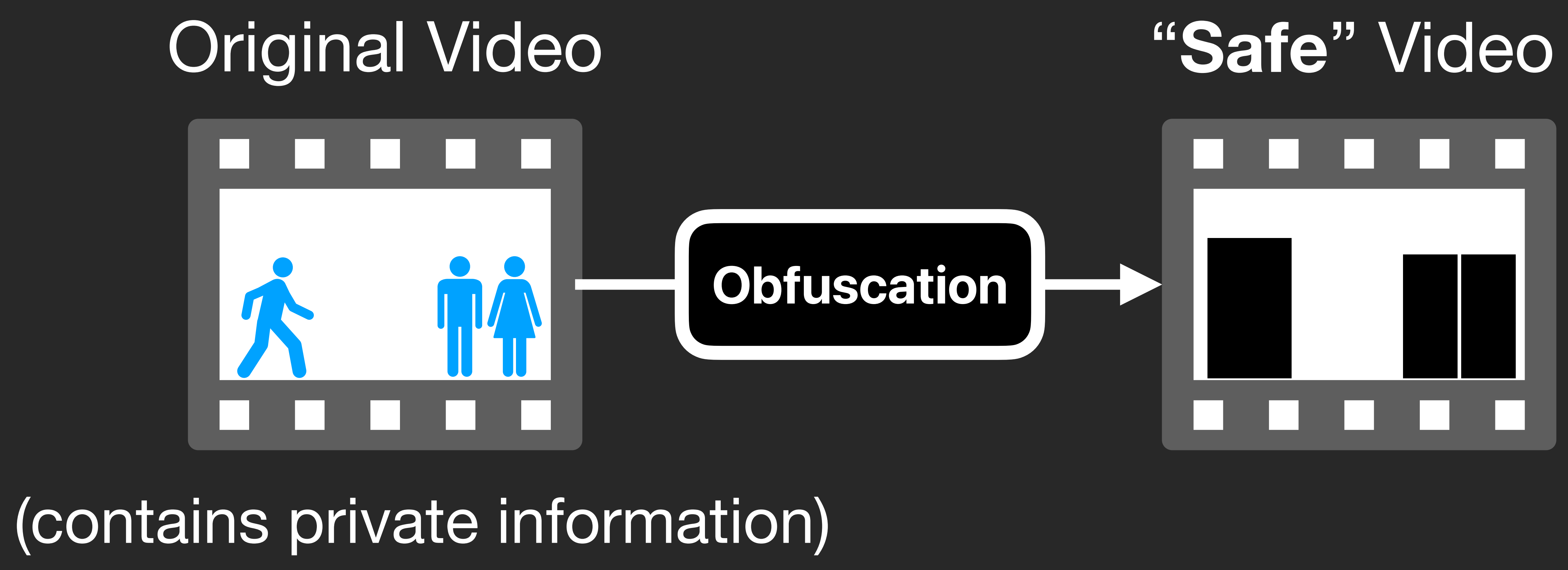
Obfuscation

Original Video

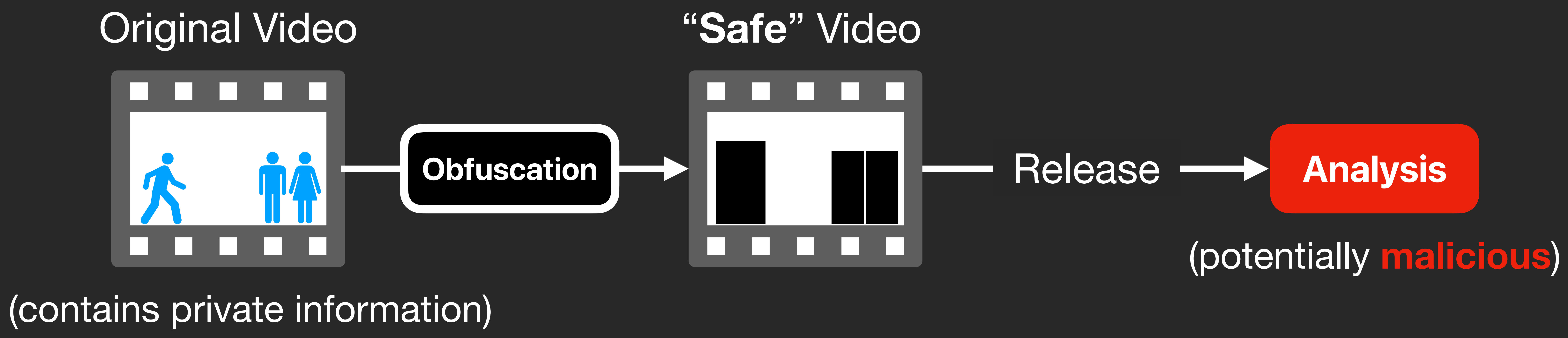


(contains private information)

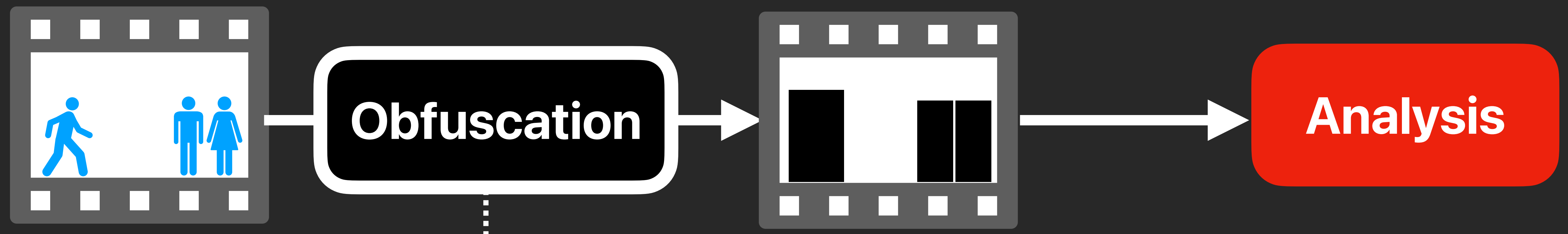
Obfuscation



Obfuscation



Obfuscation



1. **Find** all “private” information
2. **Remove** it

Obfuscation

1 **Find** all “private” information

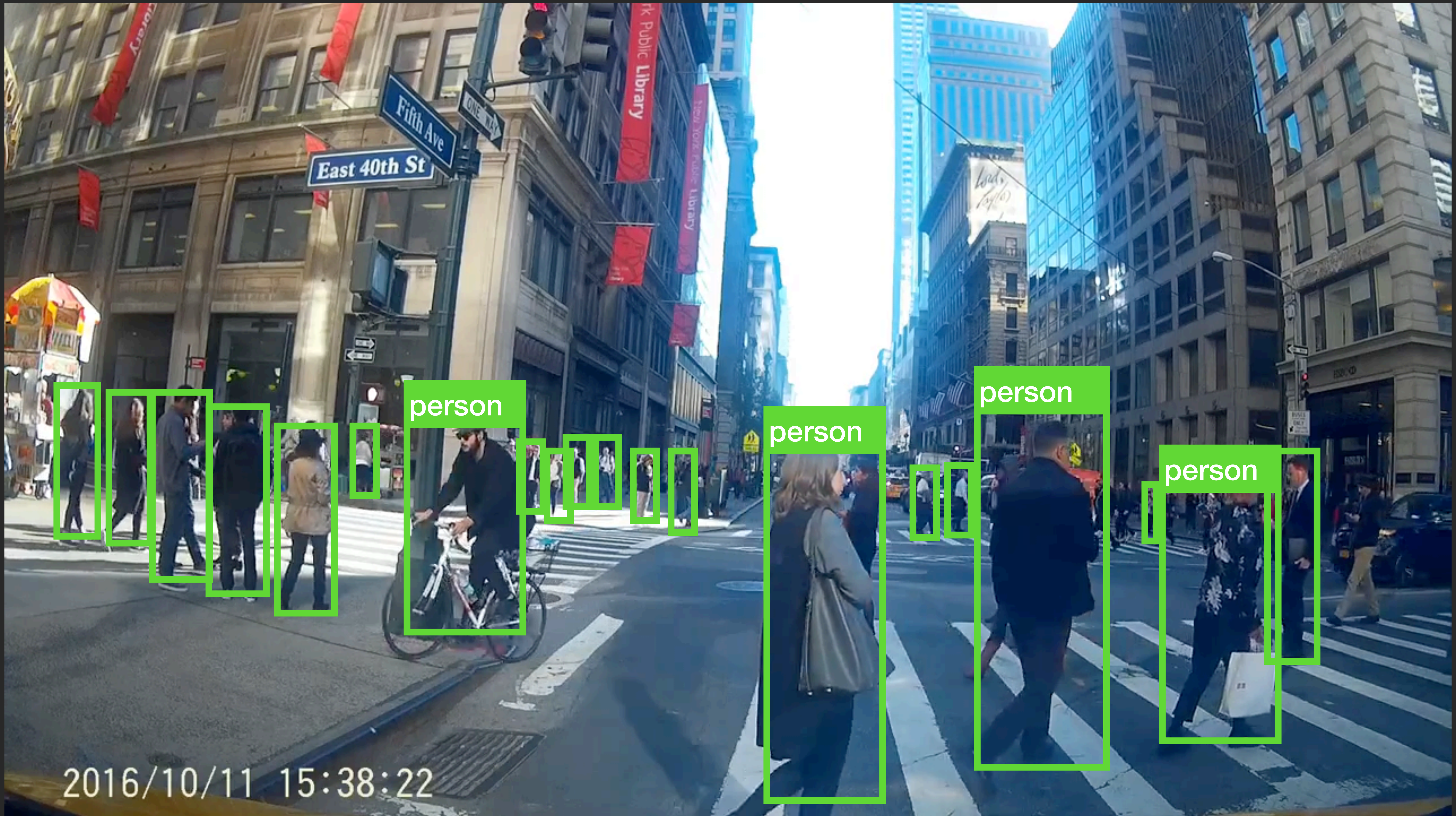
2 **Remove** it

Obfuscation

1 **Find** all “private” information   2 **Remove** it

Obfuscation

1 **Find** all "private" information   2 **Remove** it



Obfuscation

1 **Find** all “private” information

2 **Remove** it

Obfuscation

1 **Find** all “private” information

2 **Remove** it

Obfuscation

1 **Find** all "private" information   2 **Remove** it

Obfuscation

1 **Find** all “private” information

2 **Remove** it

Obfuscation

1 **Find** all “private” information

2 **Remove** it

If **missed,** can be tracked!

Can’t guarantee privacy

Obfuscation

1 **Find** all “private” information

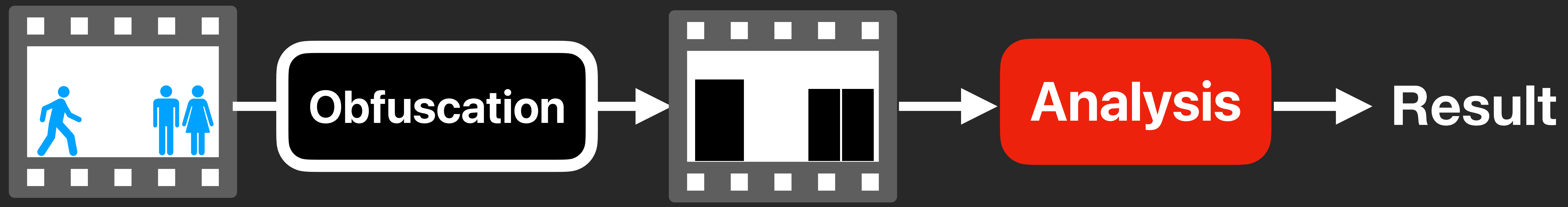
2 **Remove** it

If **missed,** can be tracked!

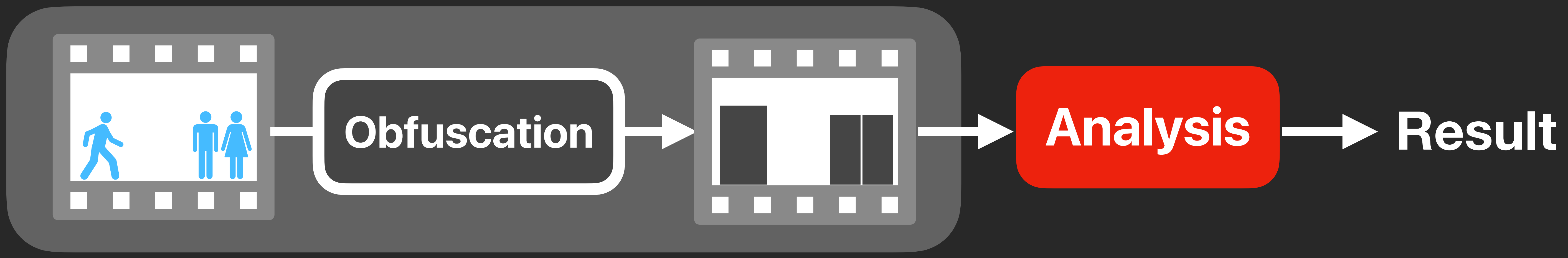
Can’t guarantee privacy

Removing **precludes** queries that don’t identify

Obfuscation



Obfuscation



Differential Privacy

Differential Privacy: Motivating Example

Name	Balance
…	
Alice	x1
Bob	x2
Charlie	x3
David	x4
Eve	x5
Frank	**x6**
Greg	x7
…	

Differential Privacy: Motivating Example

Name	Balance
...	
Alice	x1
Bob	x2
Charlie	x3
David	x4
Eve	x5
Frank	**x6**
Greg	x7
...	

→ `mean(balance)`

Differential Privacy: Motivating Example

Name	Balance
...	
Alice	x1
Bob	x2
Charlie	x3
David	x4
Eve	x5
Frank	**x6**
Greg	x7
...	

→ `mean(balance)` **Unsafe!**

what if analyst knows *all* balances except **Frank's**?

Differential Privacy: Motivating Example

Name	Balance
...	
Alice	x1
Bob	x2
Charlie	x3
David	x4
Eve	x5
Frank	**x6**
Greg	x7
...	

→ `mean(balance)` **Unsafe!**

what if analyst knows *all* balances except **Frank's**?

```
SELECT mean(balance) FROM bank;
```

Differential Privacy: Motivating Example

Name	Balance
...	
Alice	x1
Bob	x2
Charlie	x3
David	x4
Eve	x5
Frank	**x6**
Greg	x7
...	

→ `mean(balance)` **Unsafe!**

what if analyst knows *all* balances except **Frank's**?

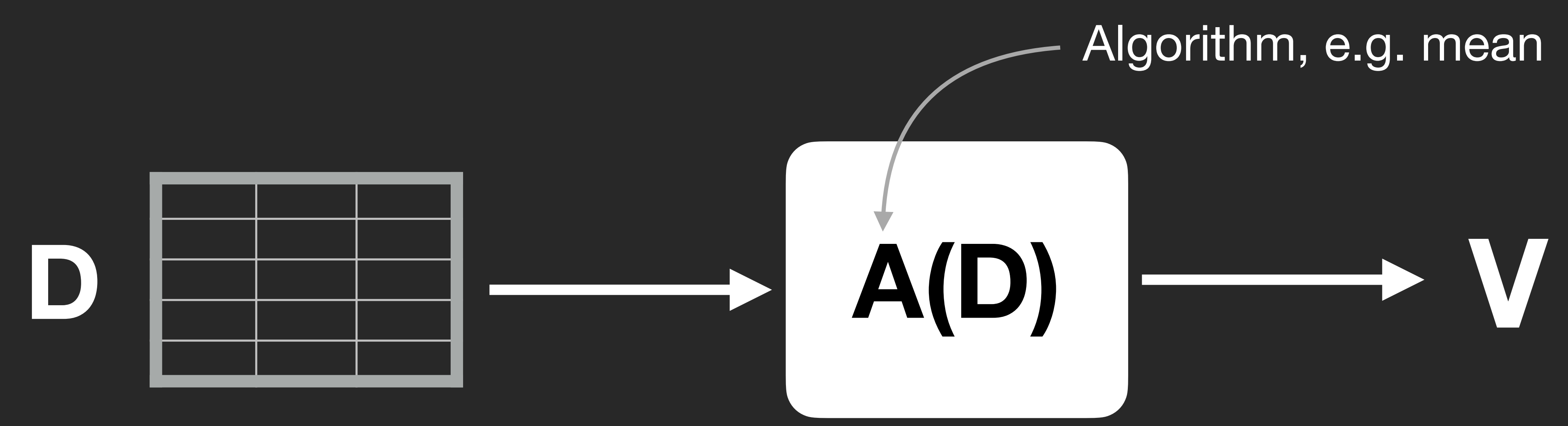
```
SELECT mean(balance) FROM bank;
SELECT mean(balance) FROM bank WHERE name!="Frank";
```

Differential Privacy: Intuition

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset

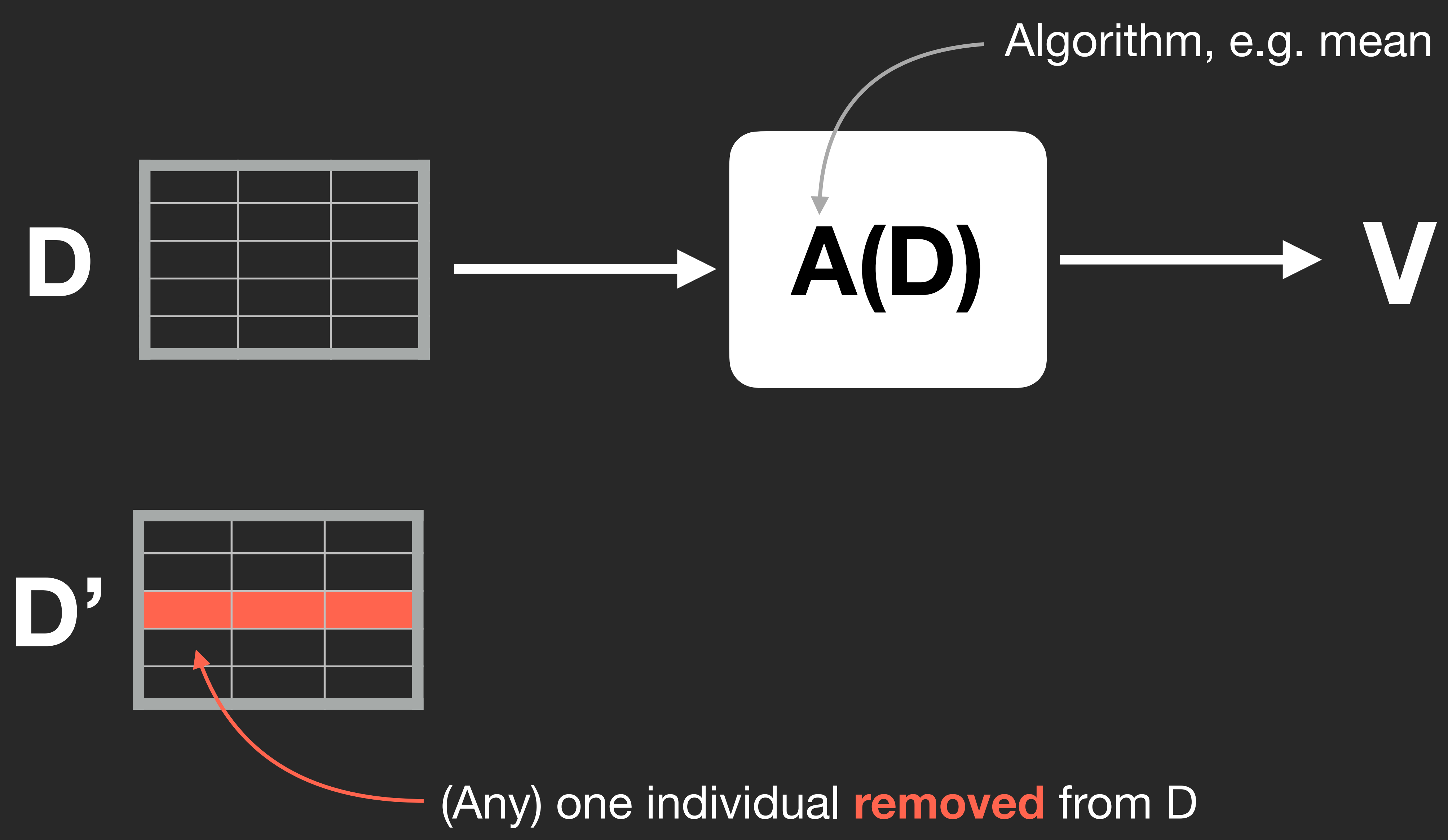
Differential Privacy: Intuition

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



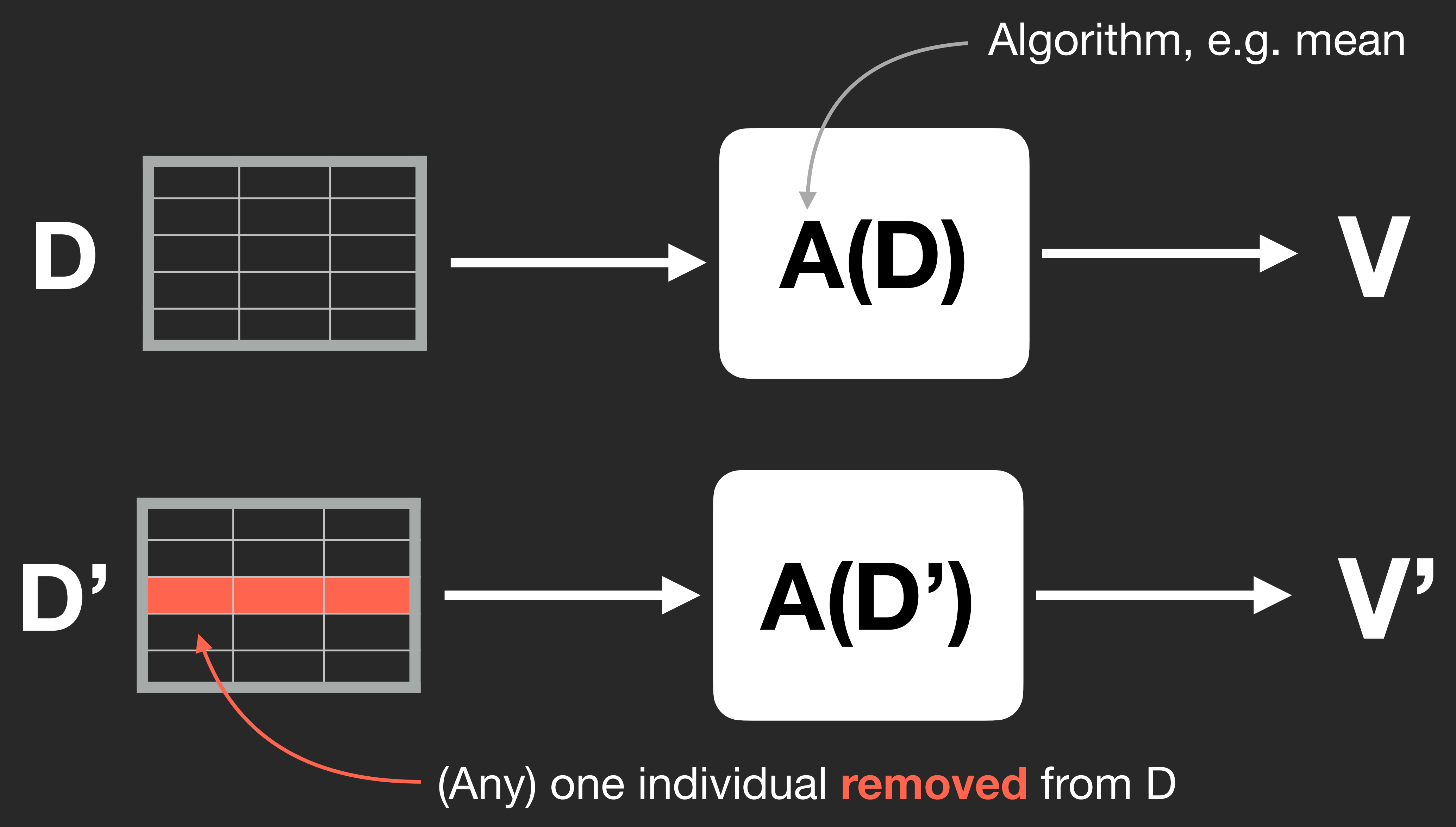
Differential Privacy: Intuition

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



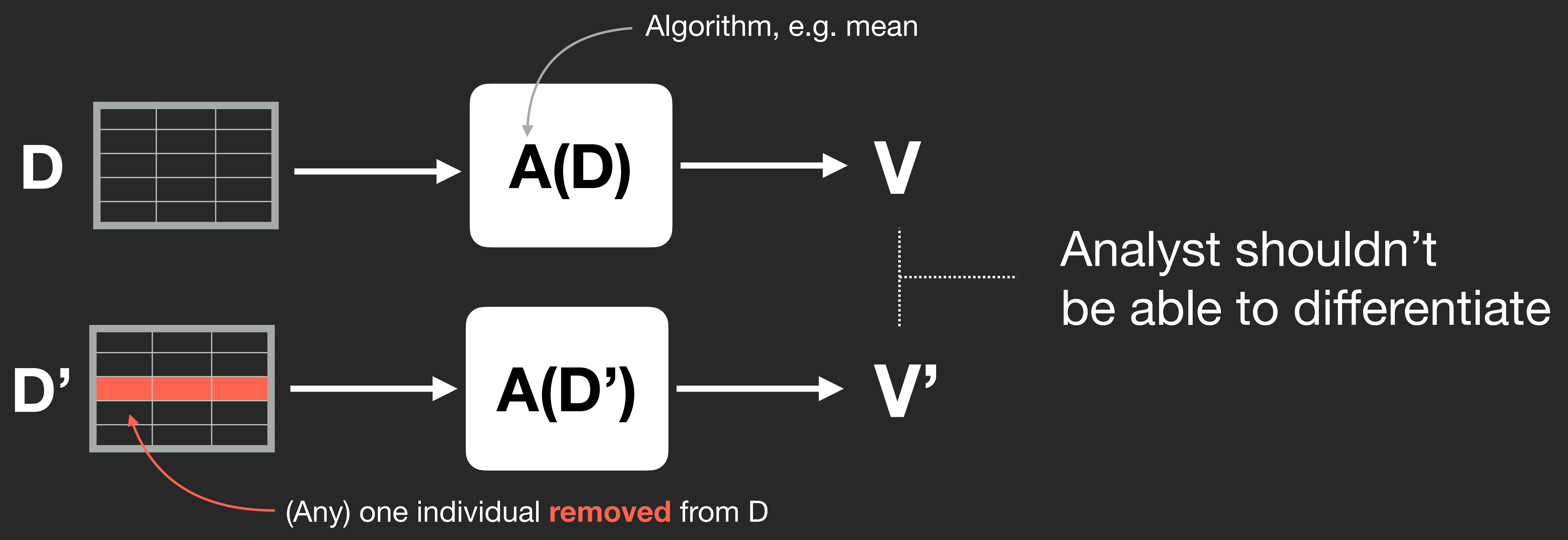
Differential Privacy: Intuition

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



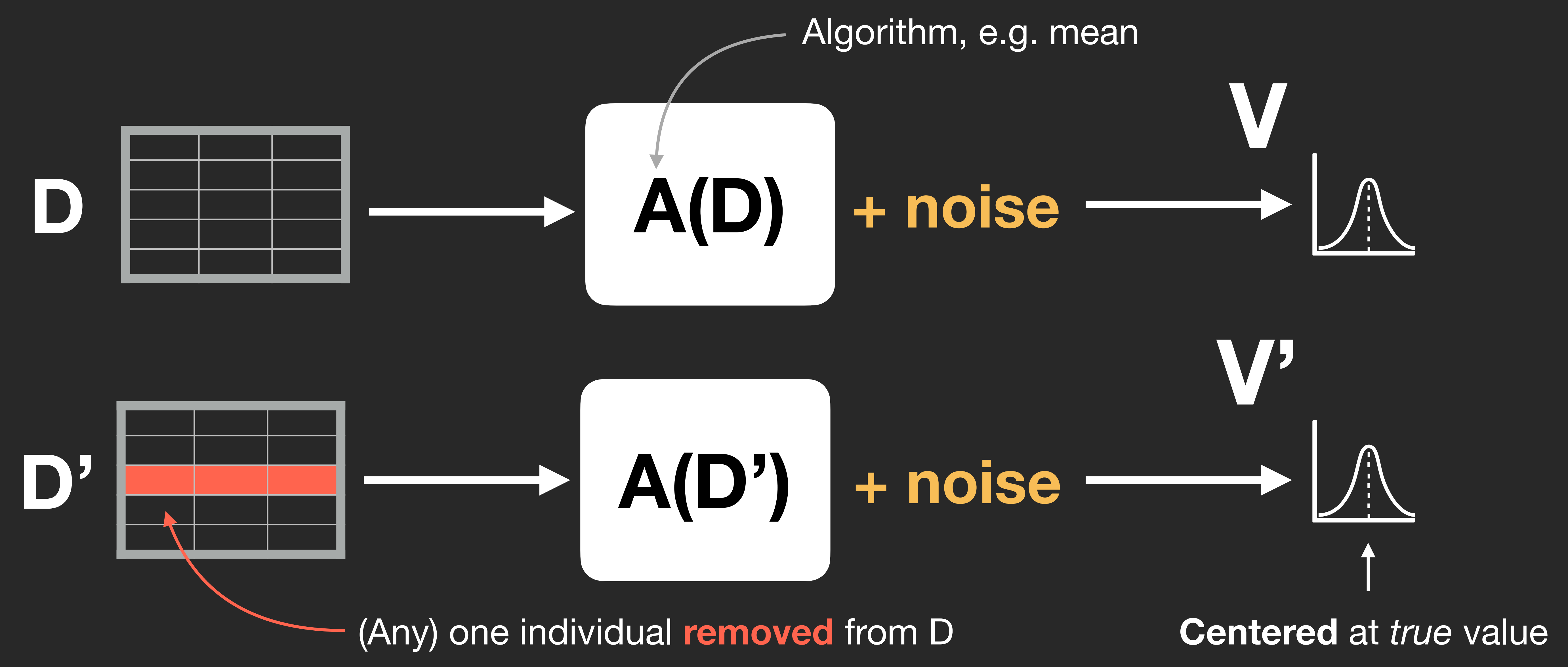
Differential Privacy: Intuition

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



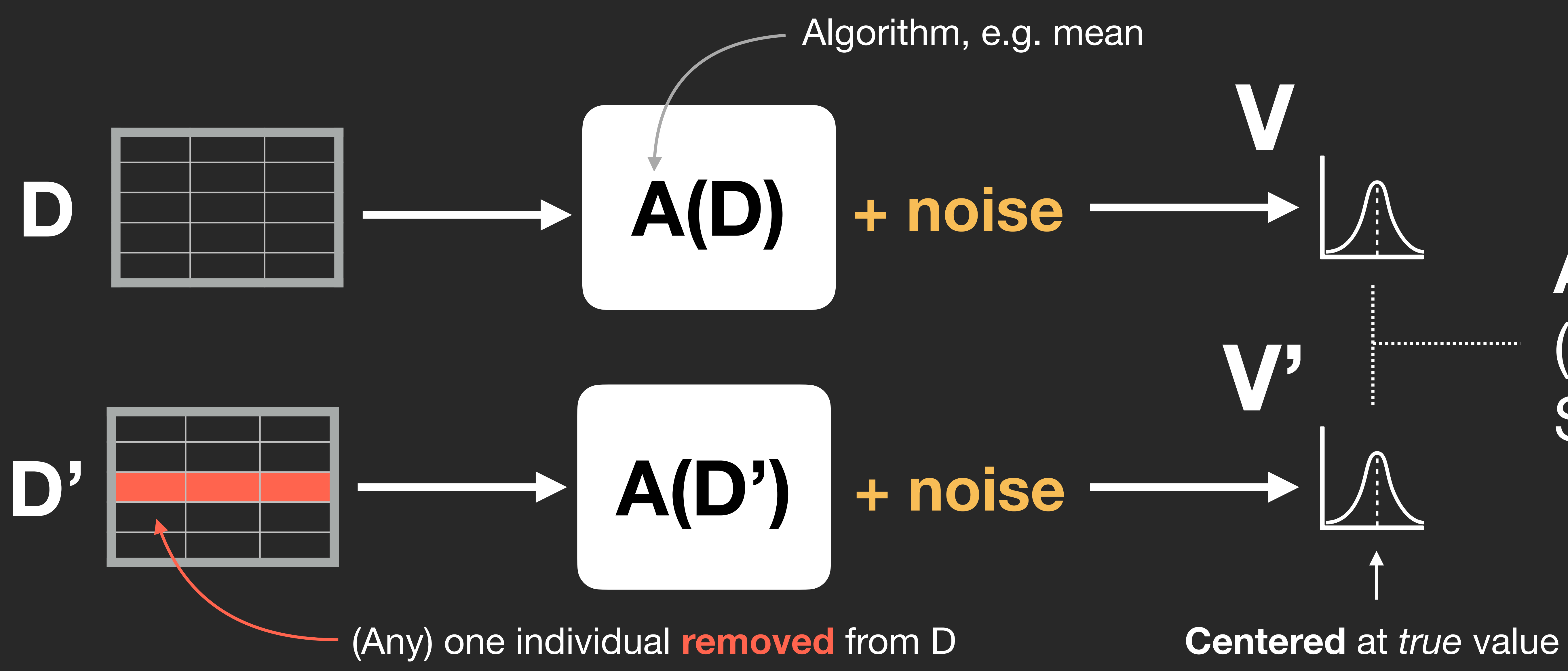
Differential Privacy: Just add noise!

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



Differential Privacy: Just add noise!

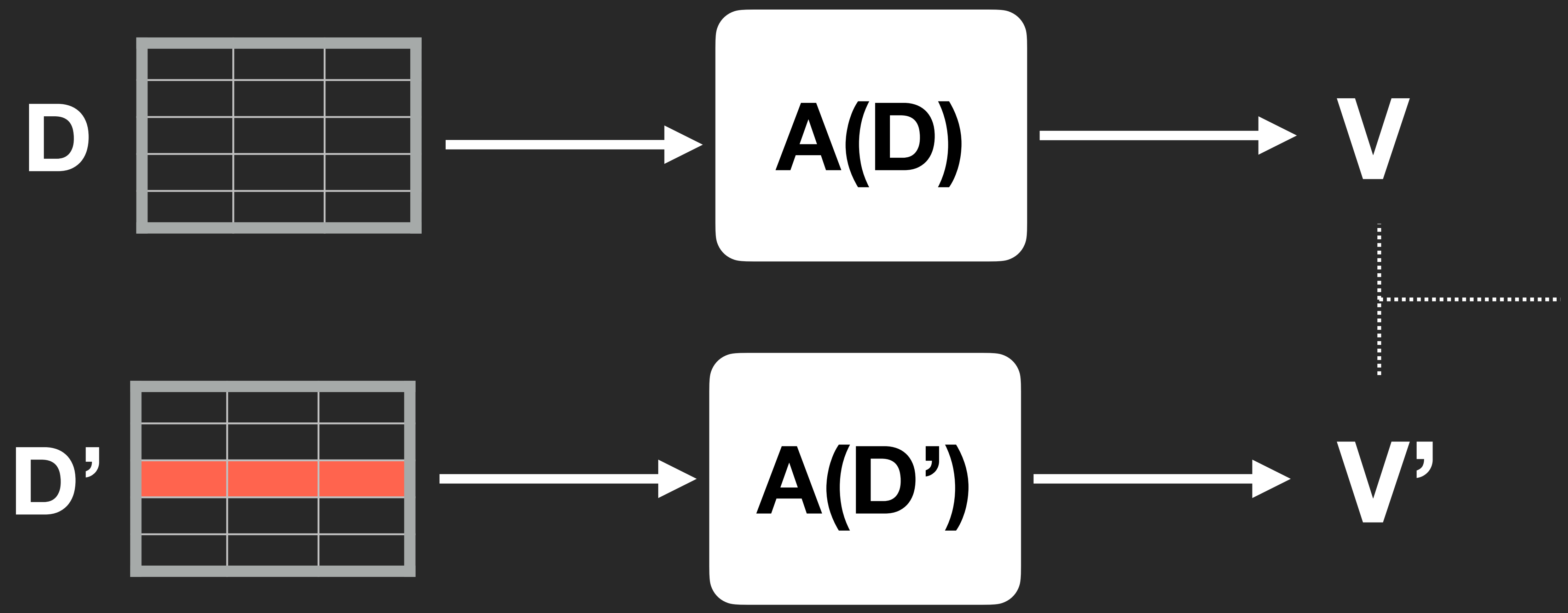
GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



Analyst observes one query result (sample from distribution)
Sample equally likely to be from V *or* V'

Differential Privacy: how much noise…?

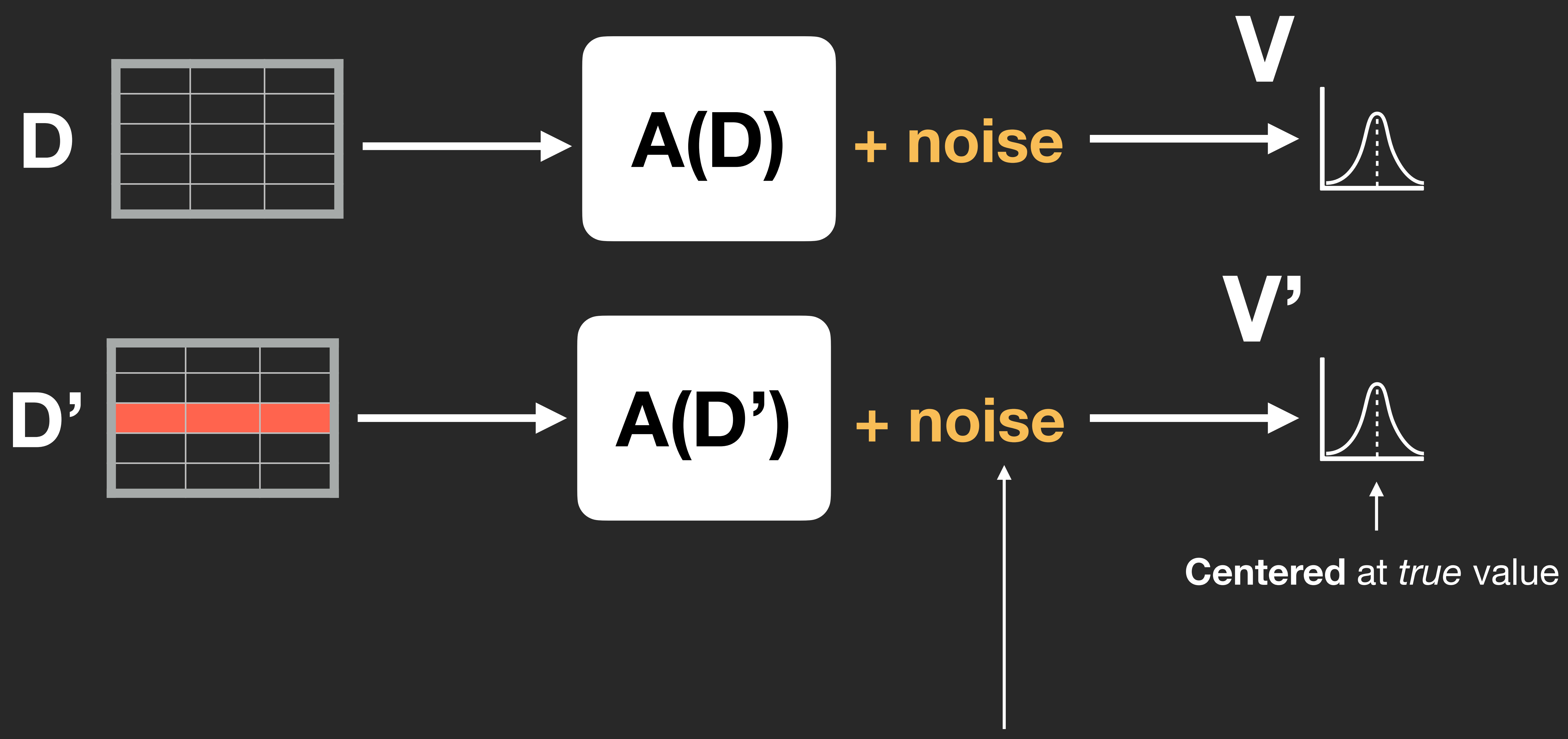
GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



Sensitivity: max difference between V and V’ for *any* (D,D’)

Differential Privacy: how much noise…?

GOAL: *Regardless* of prior information, analyst cannot identify *anyone* in dataset



Add noise *proportional* to **sensitivity**

Differential Privacy: Sensitivity Example

Balance $\in$ **[$0, $1000]**

Name	Balance
...	
Alice	x1
Bob	x2
Charlie	x3
David	x4
Eve	x5
Frank	**x6**
Greg	x7
...	

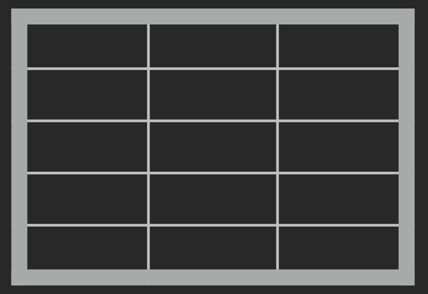
10,000 total customers

Could change by *at most* **$1,000**

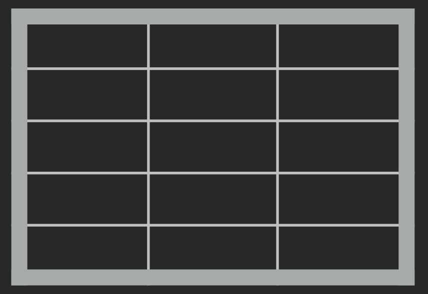
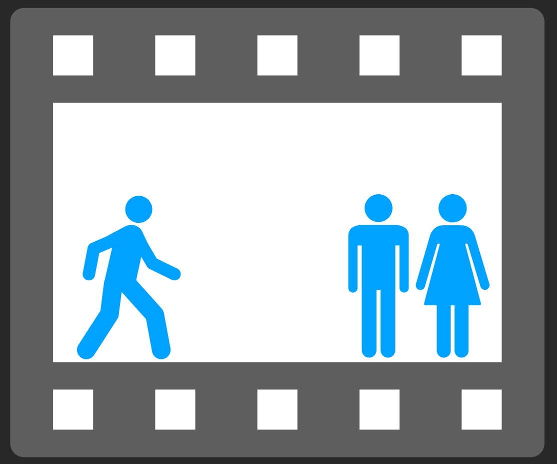
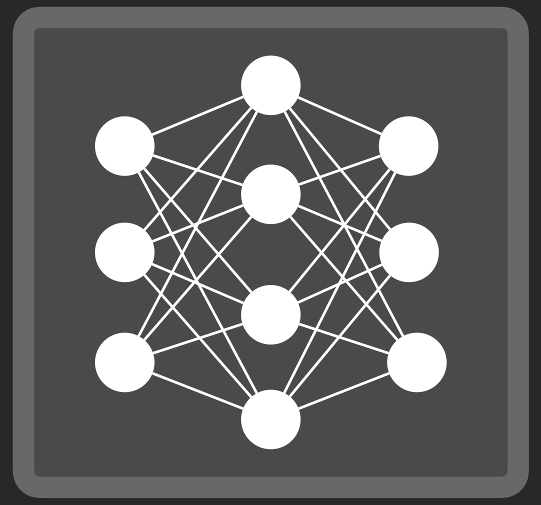
$$\text{mean} = \frac{\text{sum of balances}}{\text{num customers}}$$

$$\Delta(\text{mean(balance)}) \le \frac{|1000 - 0|}{10{,}000} \le 0.1$$

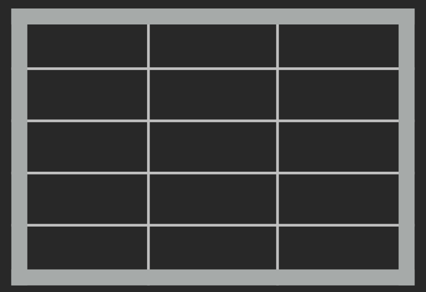
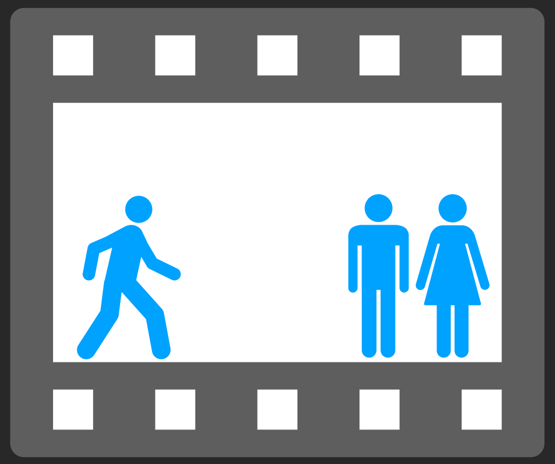
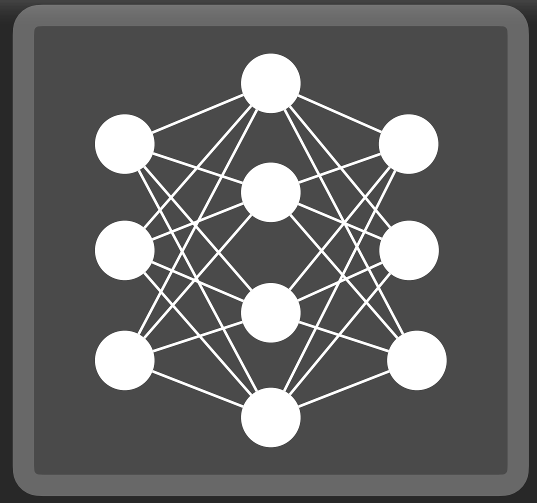
Challenge: how to satisfy DP over arbitrary NNs?

	Input	Algorithm	Satisfy DP
Classic DP Setting		Standard statistics (“white box”) μ	🧐 Manually derive bound

Challenge: how to satisfy DP over arbitrary NNs?

	Input	Algorithm	Satisfy DP
Classic DP Setting		Standard statistics ("white box") μ	Manually derive bound
Video Analytics		**Arbitrary** NNs ("black box") 	?

Challenge: how to satisfy DP over arbitrary NNs?

	Input		Algorithm	Satisfy DP
Classic DP Setting		→	Standard statistics (“white box”) μ	Manually derive bound
Video Analytics		→	**Arbitrary** NNs (“black box”) 	?

Challenge: how to satisfy DP over arbitrary NNs?

How can we *bound* the amount [someone's appearance] could contribute to the [output of a black box]?

Motivating Observation

Source: YouTube, Auburn University (speed 30x)

Motivating Observation

Source: YouTube, Auburn University (speed 30x)

Average Car:

~5s

Average Person:

~60s

Motivating Observation

Source: Newark, NJ Citizen Virtual Patrol (speed 30x)

Motivating Observation

Source: Newark, NJ Citizen Virtual Patrol (speed 30x)

Average Car:

~3s

Average Person:

~40s*

*ignoring parked cars, will come back to this at the end of the talk!

Motivating Observation

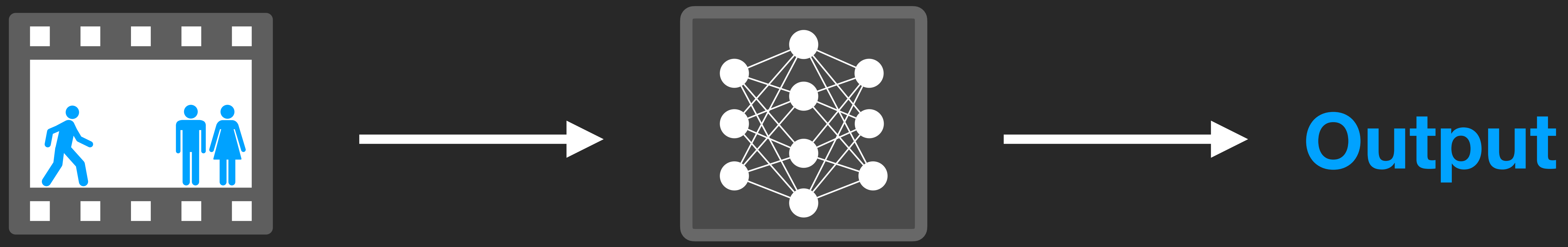
seconds

Objects tend to be visible for short duration relative to granularity of many useful queries

hours or days

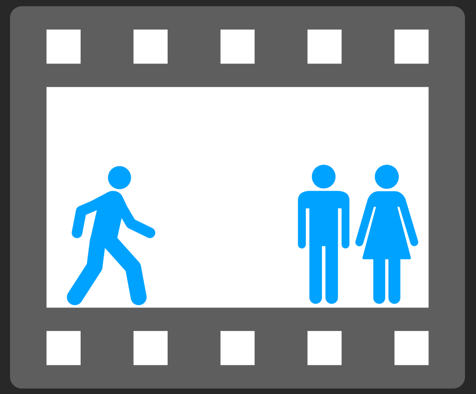
Key Insight: implicit structure of query

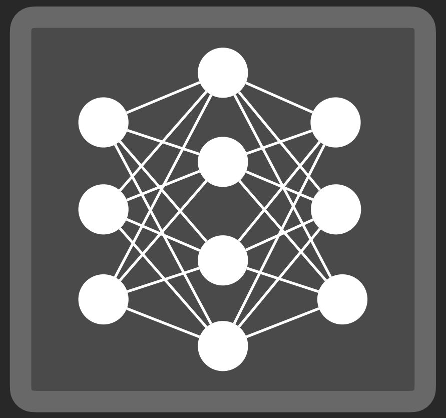
“Count **people** per hour”



Key Insight: implicit structure of query

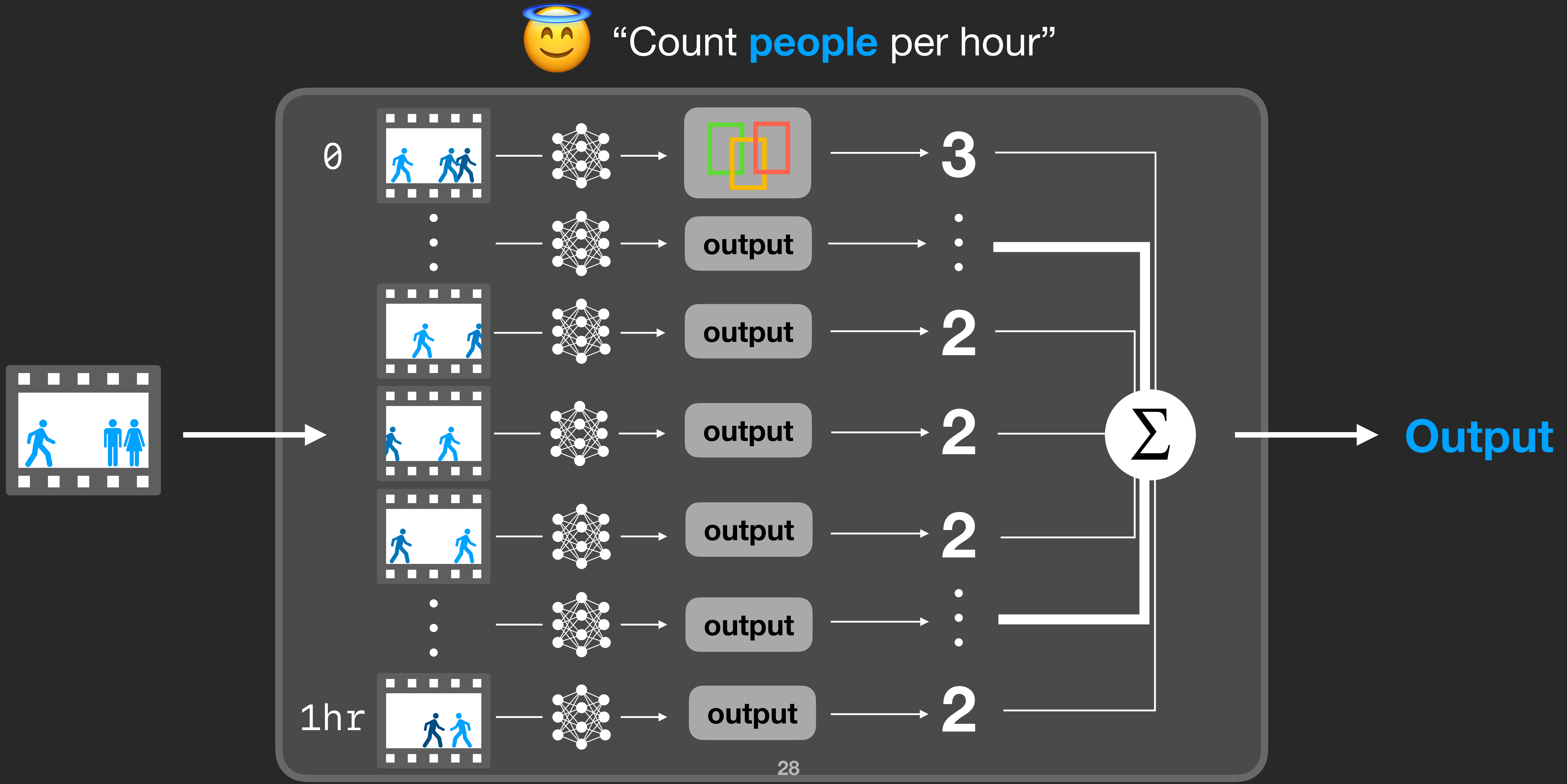
"Count **people** per hour"





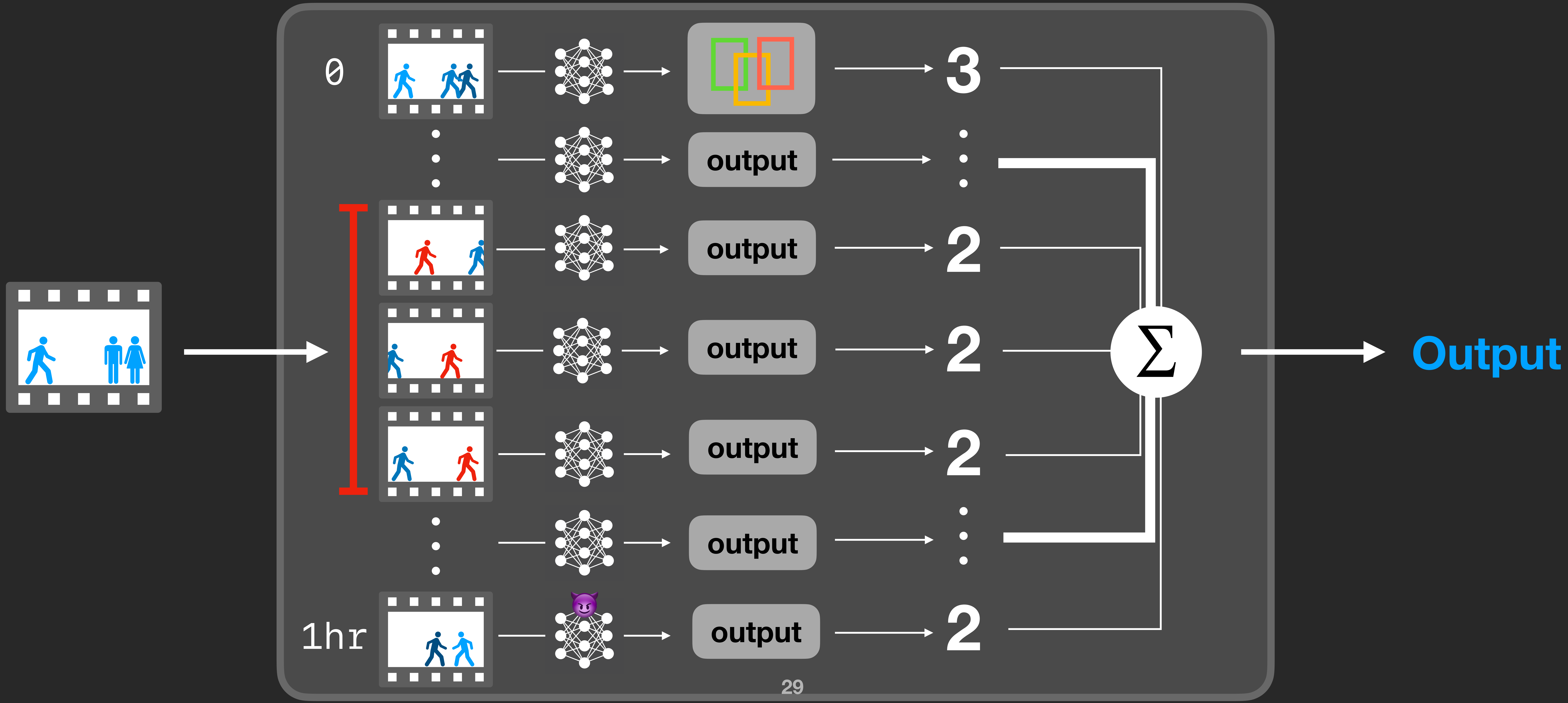
Output

Key Insight: implicit structure of query



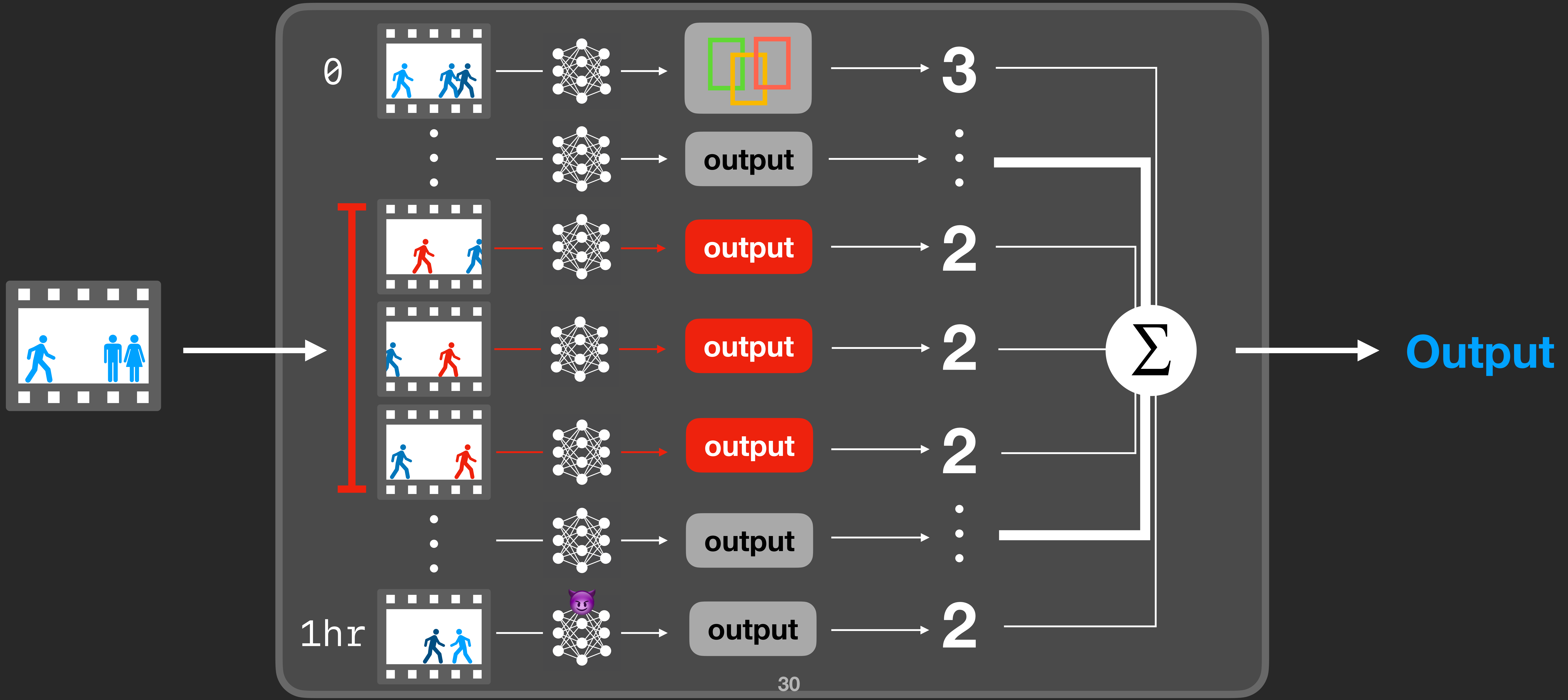
Key Insight: implicit structure of query

"When did **Frank** pass this camera?"

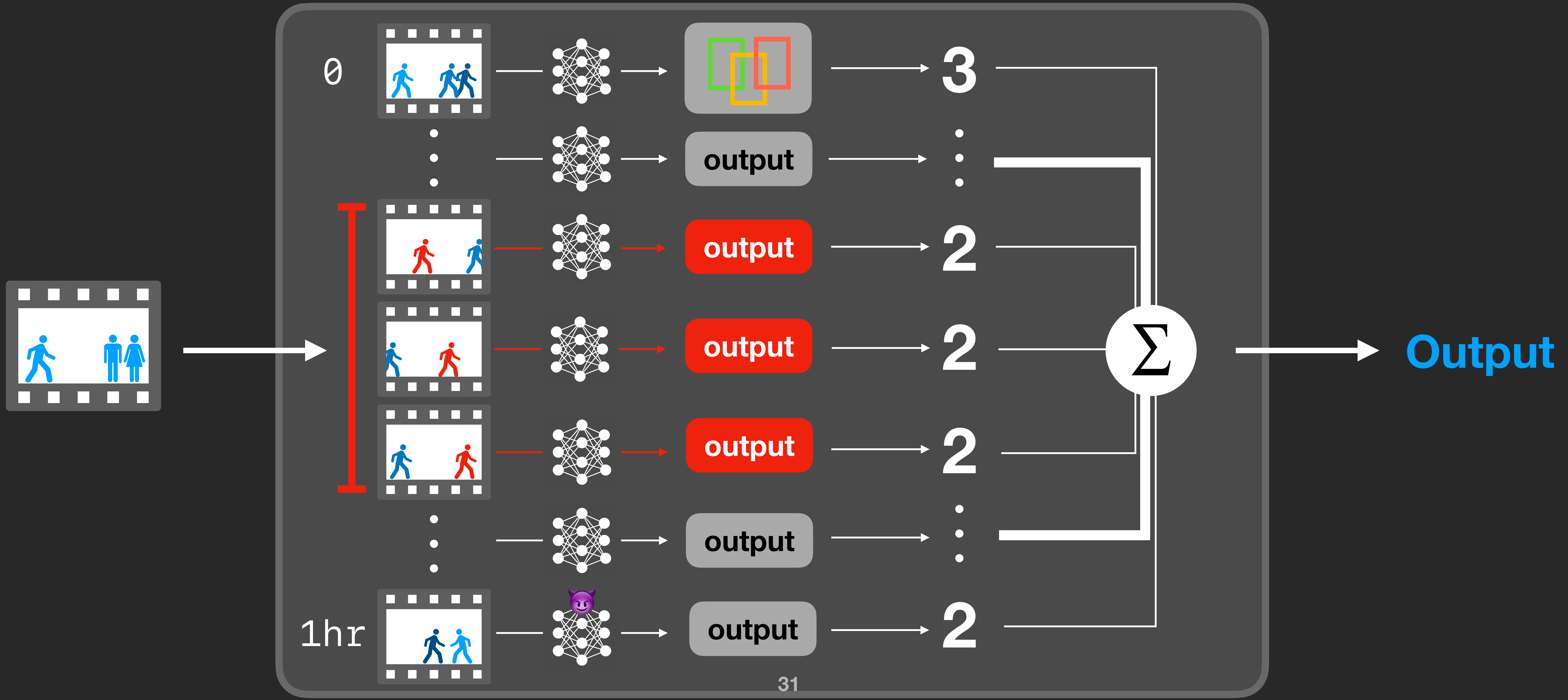


Key Insight: implicit structure of query

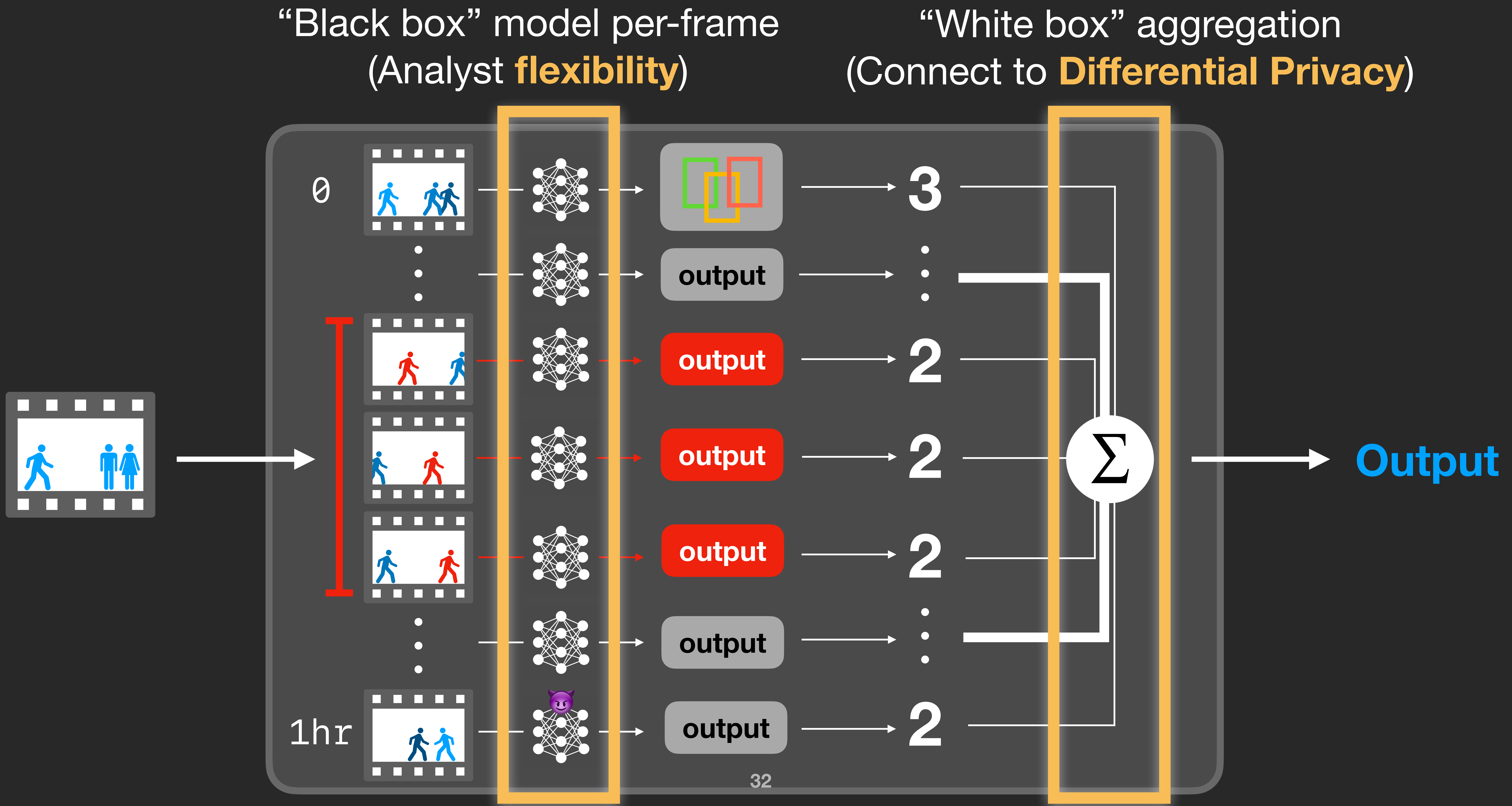
"When did **Frank** pass this camera?"



Key Insight: implicit structure of query



Key Insight: implicit structure of query



Our Contributions

1 New definition: **(ρ)-Event-Duration Privacy**

Does *not* require locating all individuals to protect them

2 System that satisfies definition: **Privid**

Analysts provide their own (*untrusted*) video processing code

(ρ)-Event-Duration Privacy

Classic DP: "analyst cannot detect presence of any one person"

Our Definition: "analyst cannot detect *anything* visible for $\leq \rho$ seconds"

(e.g., 30 seconds)

1. Simple and expressive interface
2. Gracefully handles CV imperfections
3. Enables safe aggregation queries
4. Predictable impact on utility

Formally: $\forall$ possible videos (V, V') that differ by $\leq \rho$ seconds, and all output subsets S,

$$\frac{\Pr[Q(V) \in S]}{\Pr[Q(V') \in S]} \leq e^{\epsilon}$$

(**Small** constant)

(ρ)-Event-Duration Privacy

Classic DP: “analyst cannot detect presence of any one person”

Our Definition: “analyst cannot detect *anything* visible for $\leq \rho$ seconds”

(e.g., 30 seconds)

1. Simple and expressive interface
2. Gracefully handles CV imperfections
3. Enables safe aggregation queries
4. Predictable impact on utility

Formally: $\forall$ possible videos (V, V') that differ by $\leq \rho$ seconds, and all output subsets S,

$$\frac{\Pr[Q(V) \in S]}{\Pr[Q(V') \in S]} \leq e^{\epsilon}$$

(**Small** constant)

(ρ)-Event-Duration Privacy

Classic DP: "analyst cannot detect presence of any one person"

Our Definition: "analyst cannot detect *anything* visible for $\leq \rho$ seconds"

(e.g., 30 seconds)

1. Simple and expressive interface
2. Gracefully handles CV imperfections
3. Enables safe aggregation queries
4. Predictable impact on utility

Formally: $\forall$ possible videos (V, V') that differ by $\leq \rho$ seconds, and all output subsets S,

$$\frac{\Pr[Q(V) \in S]}{\Pr[Q(V') \in S]} \leq e^{\epsilon}$$

(**Small** constant)

(ρ)-Event-Duration Privacy

Video	Maximum Duration		% Objects CV Missed
	Ground Truth	CV Estimate	
campus	81 sec	83 sec	29%
highway*	316 sec	439 sec	5%
urban	270 sec	354 sec	76%

(ρ)-Event-Duration Privacy

Classic DP: "analyst cannot detect presence of any one person"

Our Definition: "analyst cannot detect *anything* visible for $\le \rho$ seconds"

(e.g., 30 seconds)

1. Simple and expressive interface
2. Gracefully handles CV imperfections
3. Enables safe aggregation queries
4. Predictable impact on utility

Formally: $\forall$ possible videos (V, V') that differ by $\le \rho$ seconds, and all output subsets S,

🥱

$$\frac{\Pr[Q(V) \in S]}{\Pr[Q(V') \in S]} \le e^{\epsilon}$$

(**Small** constant)

(ρ)-Event-Duration Privacy

Classic DP: "analyst cannot detect presence of any one person"

Our Definition: "analyst cannot detect *anything* visible for $\leq \rho$ seconds"
(e.g., 30 seconds)

1. Simple and expressive interface
2. Gracefully handles CV imperfections
3. Enables safe aggregation queries
4. Predictable impact on utility

Formally: $\forall$ possible videos (V, V') that differ by $\leq \rho$ seconds, and all output subsets S,

🥱

$$\frac{\Pr[Q(V) \in S]}{\Pr[Q(V') \in S]} \leq e^{\epsilon}$$

(**Small** constant)

(ρ)-Event-Duration Privacy

Classic DP: “analyst cannot detect presence of any one person”

Our Definition: “analyst cannot detect *anything* visible for $\leq \rho$ seconds”

(e.g., 30 seconds)

1. Simple and expressive interface
2. Gracefully handles CV imperfections
3. Enables safe aggregation queries
4. Predictable impact on utility

Formally: $\forall$ possible videos (V, V') that differ by $\leq \rho$ seconds, and all output subsets S,

🥱

$$\frac{\Pr[Q(V) \in S]}{\Pr[Q(V') \in S]} \leq e^{\epsilon}$$

(**Small** constant)

Privid: Overview

Registration

VO

Choose per-camera guarantee

cam1: ρ=30s, cam2: …

Privid

Analytics

Analyst

NN + Query

Result

Privid

Guarantee: satisfies (ρ)-event-duration privacy

Privid: Query Interface

SPLIT	PROCESS	SELECT
into chunks	using NN, output rows	aggregation over table

Privid: Query Interface

SPLIT

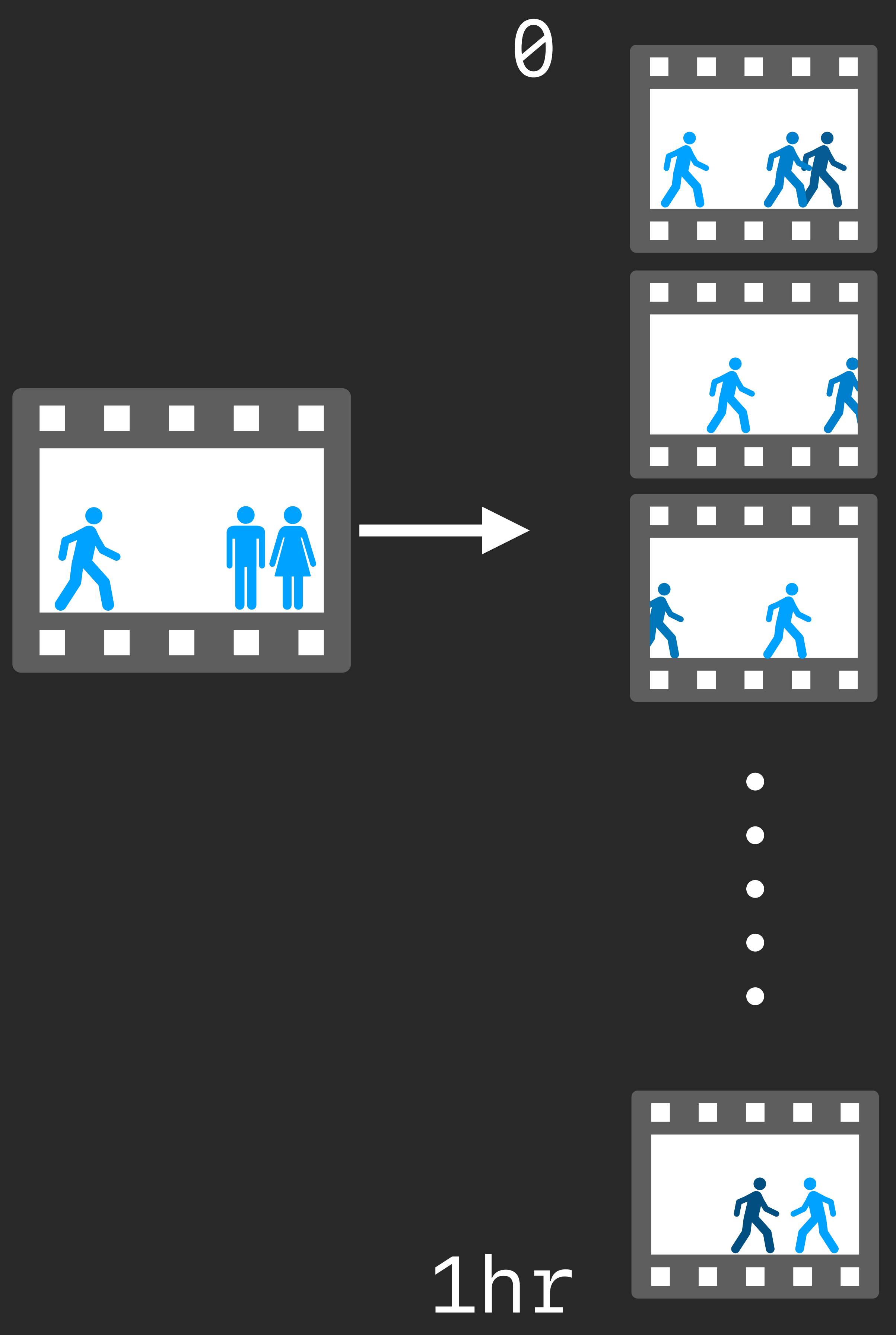
into chunks

PROCESS

using NN, output rows

SELECT

aggregation over table



Privid: Query Interface

SPLIT

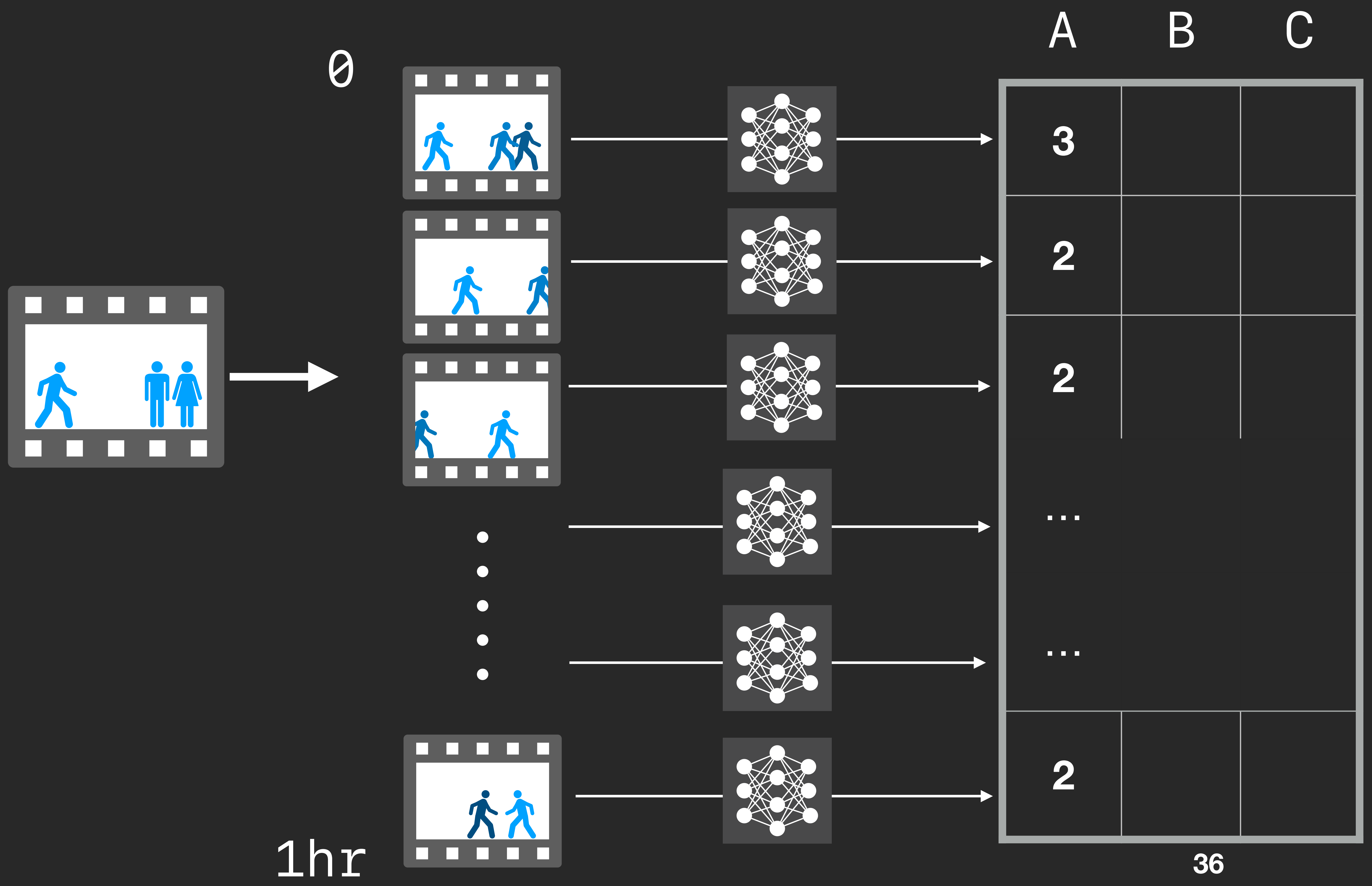
into chunks

PROCESS

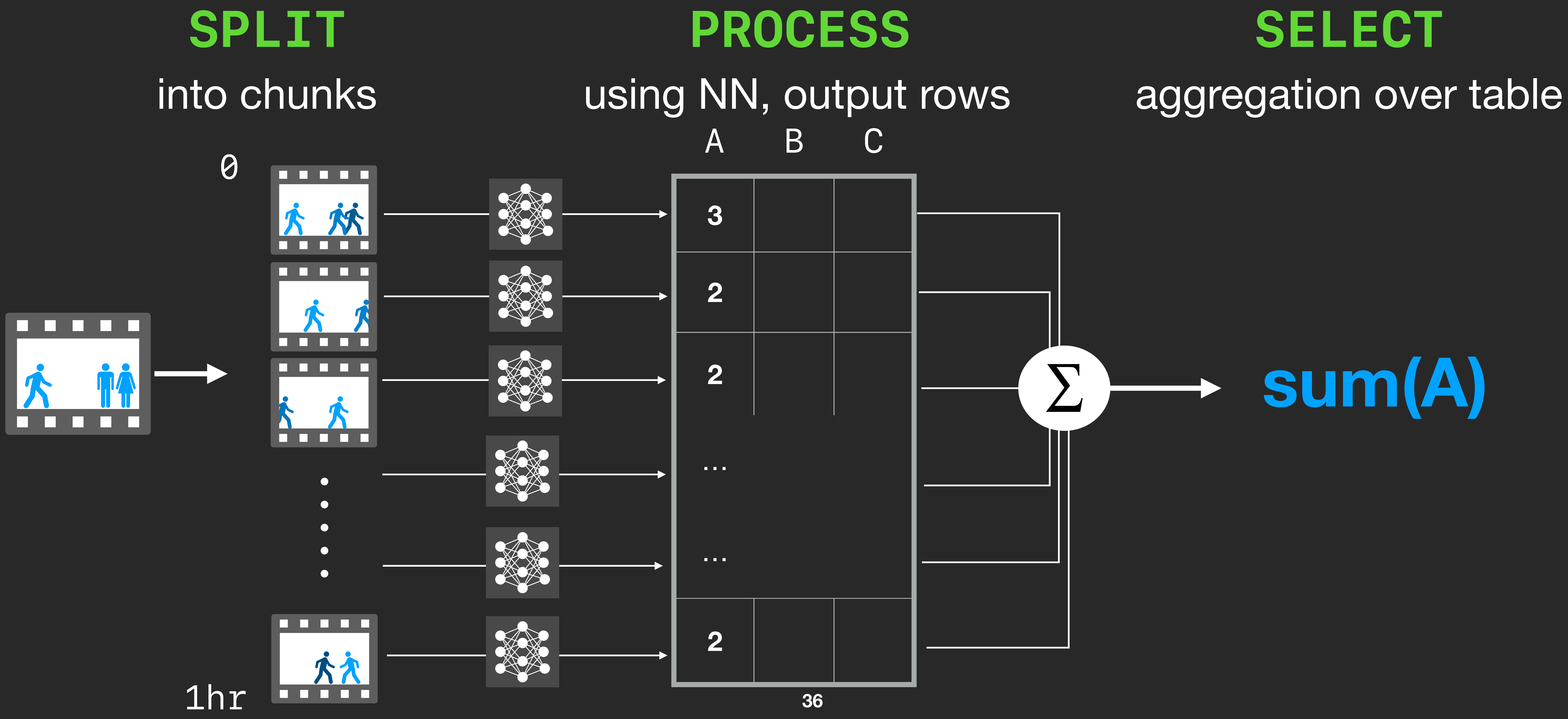
using NN, output rows

SELECT

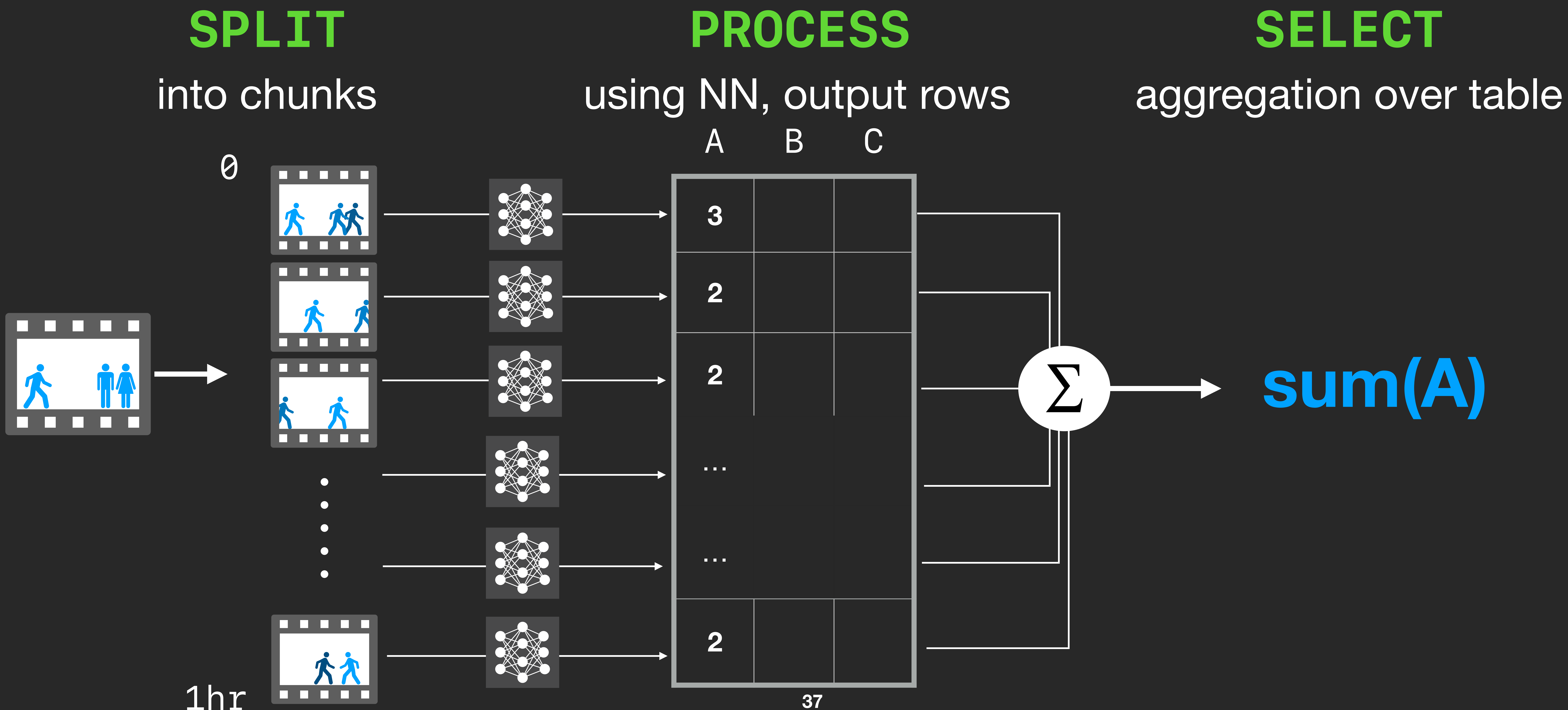
aggregation over table



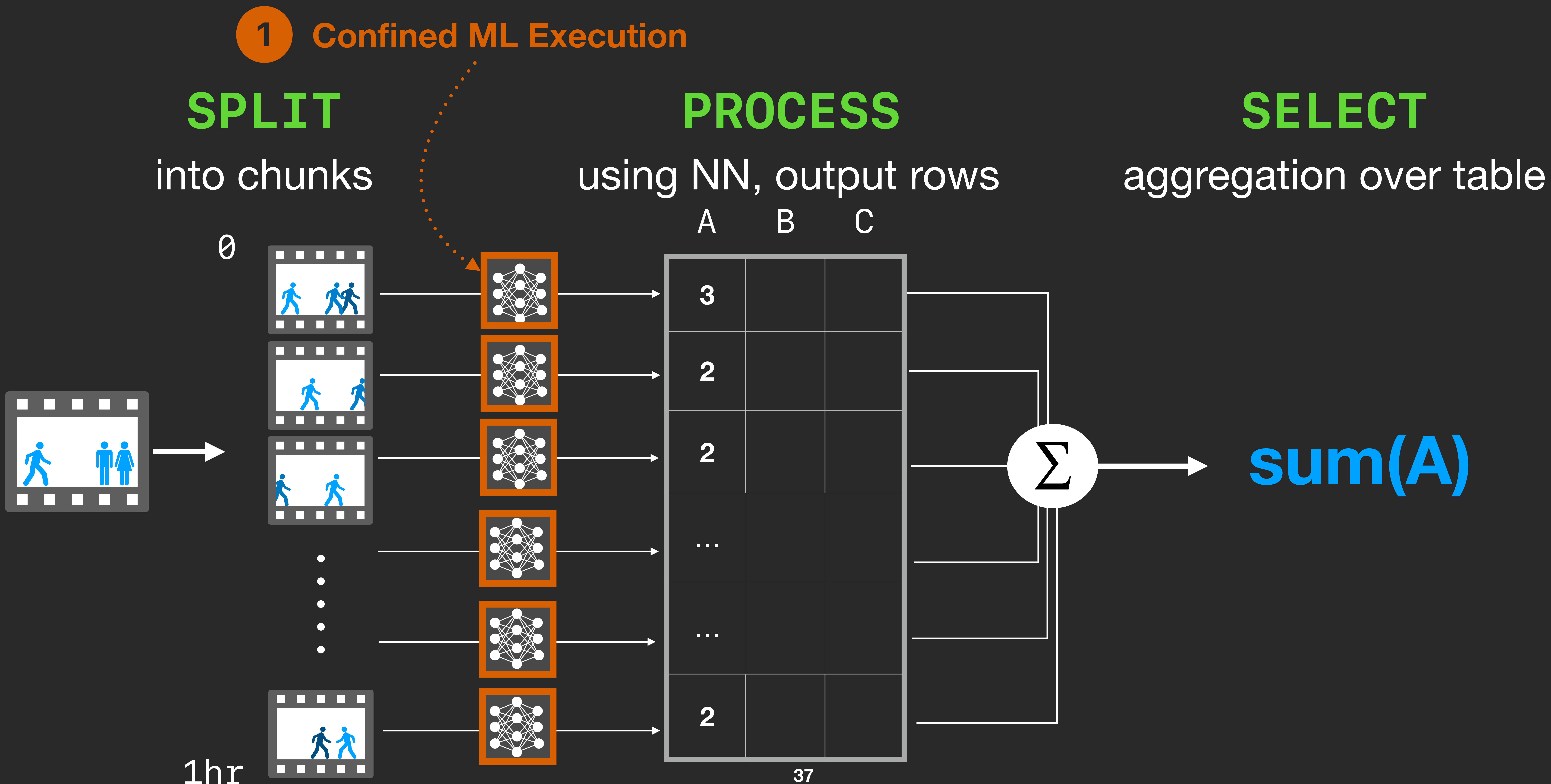
Privid: Query Interface



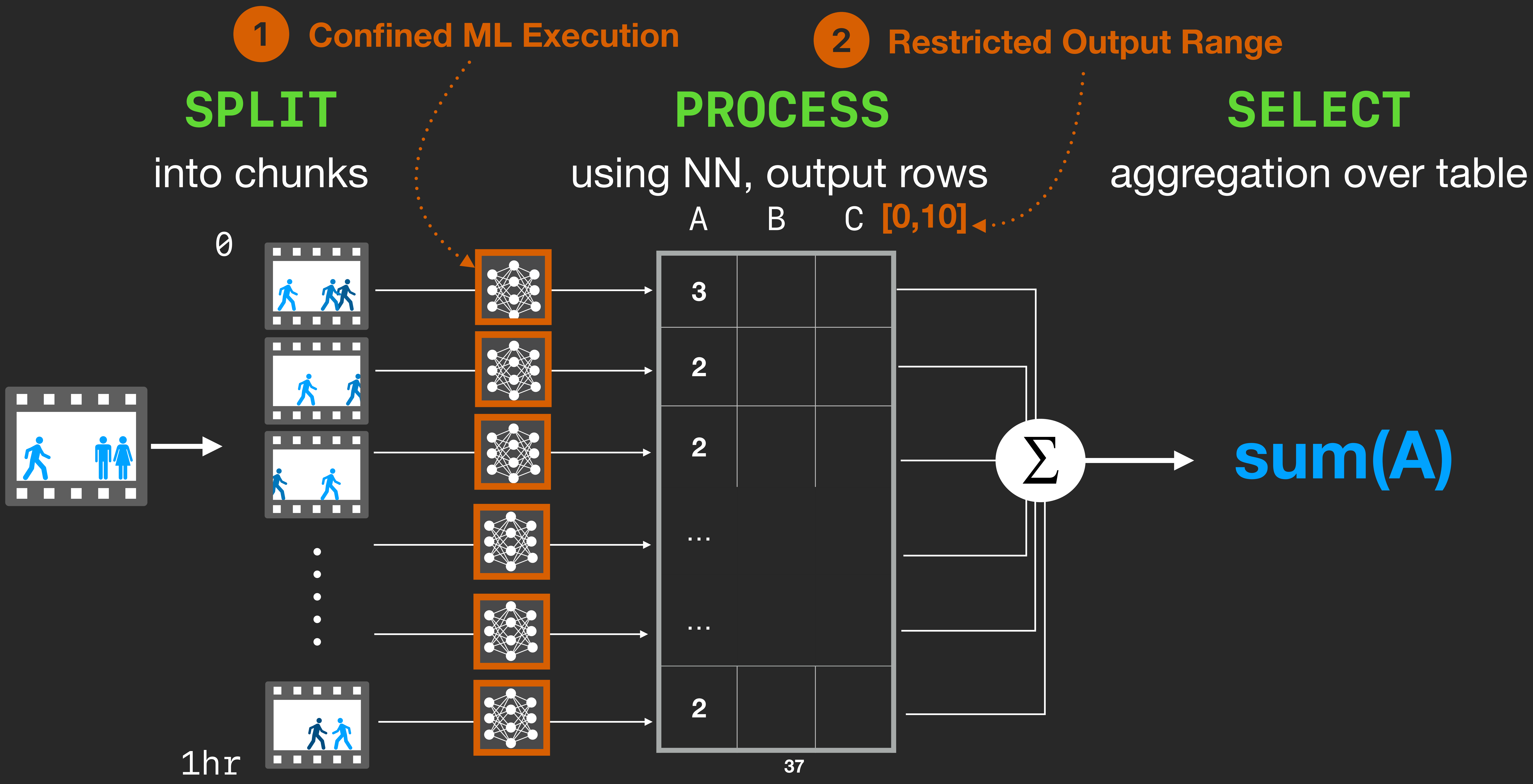
Privid: Query Interface



Privid: Query Interface



Privid: Query Interface

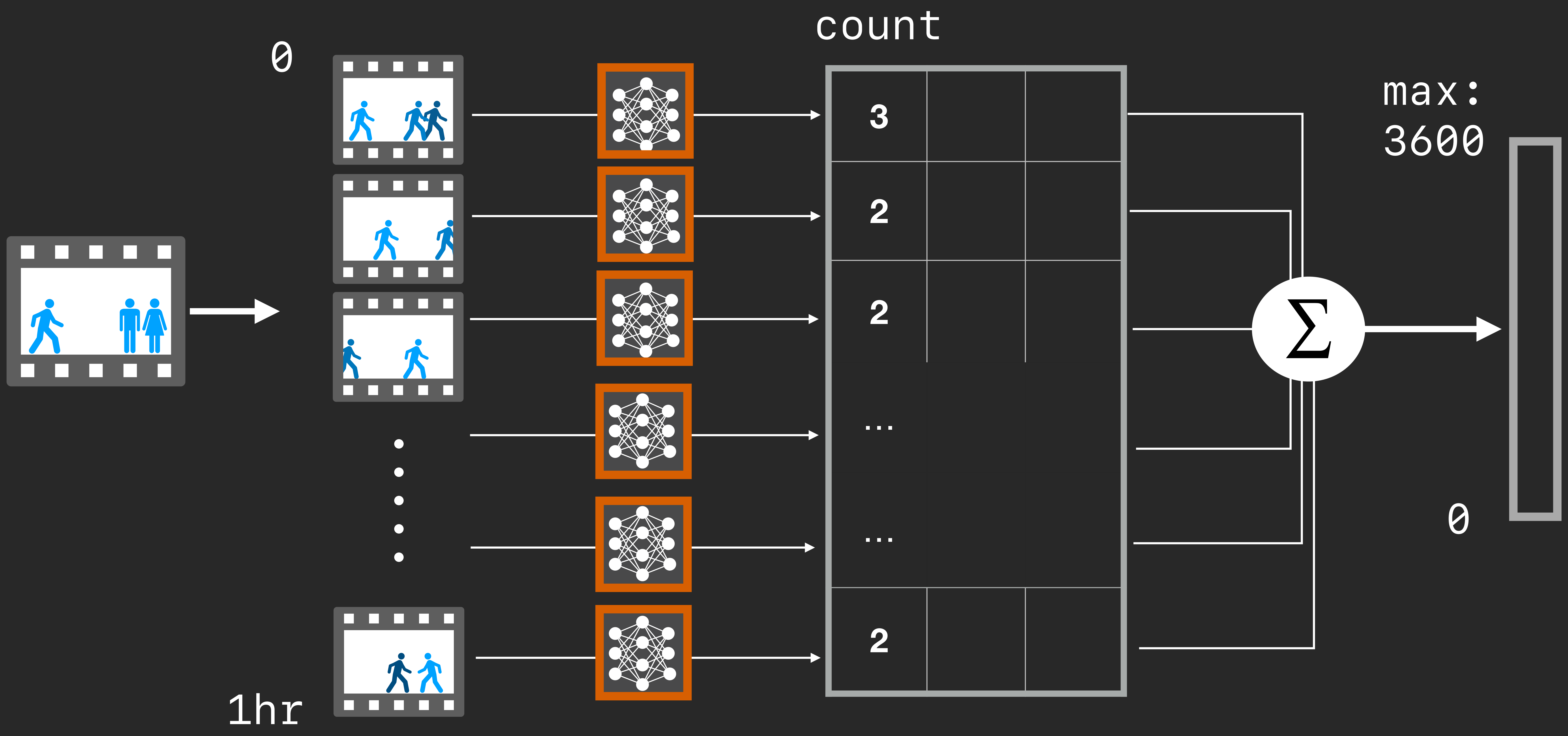


(Query intent unknown)

SPLIT
10s chunks

PROCESS
count: [0,10]

SELECT
SUM(count)




(Query intent unknown)
SPLIT
10s chunks
PROCESS
count: [0,10]
SELECT
SUM(count)
count
0
ρ=20s
≤ 3 chunks
1hr
3
2
2
...
...
2
max:
3600
0


(Query intent unknown)
SPLIT
10s chunks
PROCESS
count: [0,10]
SELECT
SUM(count)
count
0
ρ=20s
≤ 3 chunks
10
10
10
...
...
0
1hr
40
max:
3600
0
Σ

(Query intent unknown)
SPLIT
10s chunks
PROCESS
count: [0,10]
SELECT
SUM(count)
Sensitivity ≤
3 chunks x
| [0,10] |
= 30
count
0
ρ=20s
≤ 3 chunks
10
10
10
...
...
0
1hr
40
max:
3600
0
+
noise

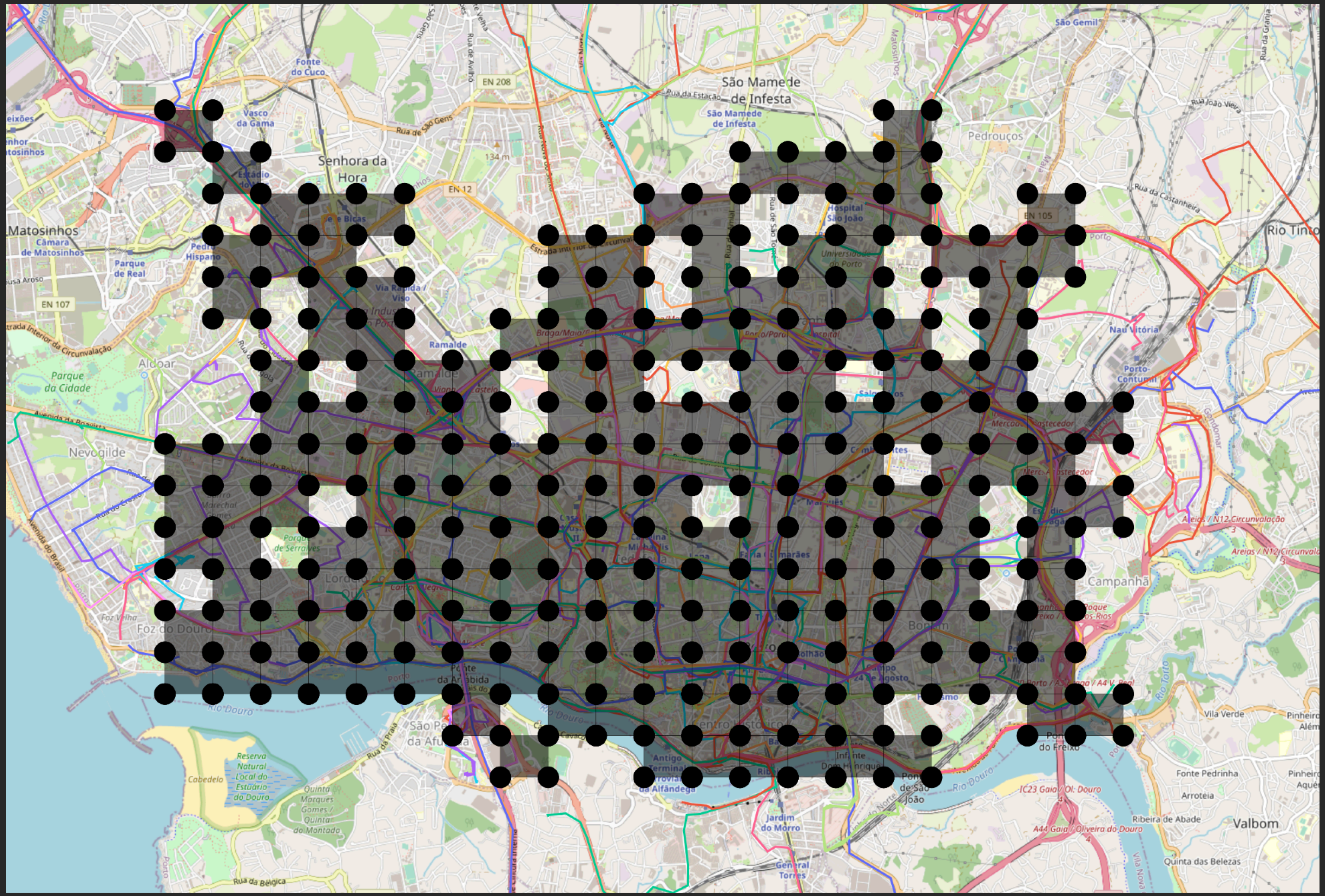
“When did Frank pass this camera?”
SPLIT
10s chunks
PROCESS
count: [0,10]
SELECT
SUM(count)
Sensitivity ≤
3 chunks x
| [0,10] |
= 30
count
0
ρ=20s
≤ 3 chunks
1hr
10
10
10
…
…
0
41
max:
3600
0
+
noise
Equally likely!
Could be result of
presence OR noise

"Count people per hour"
SPLIT
10s chunks
PROCESS
count: [0,10]
SELECT
SUM(count)
Sensitivity ≤
3 chunks x
| [0,10] |
= 30
count
0
ρ=20s
≤ 3 chunks
1hr
3
2
2
6
8
2
max:
3600
0
+
noise

"Count people per hour"
SPLIT
10s chunks
PROCESS
count: [0,10]
SELECT
SUM(count)
Sensitivity ≤
3 chunks x
| [0,10] |
= 30
count
0
ρ=20s
≤ 3 chunks
3
2
2
6
8
2
42
1hr
max:
3600
0
+
noise
+ [-30,30]

Concrete Example (Q4)

Location of **127 Cameras** across Porto, Portugal



Taxi Trajectory Data: EKML/PKDD 2015

"Average working hours of taxis in Porto during 2013-2014?"

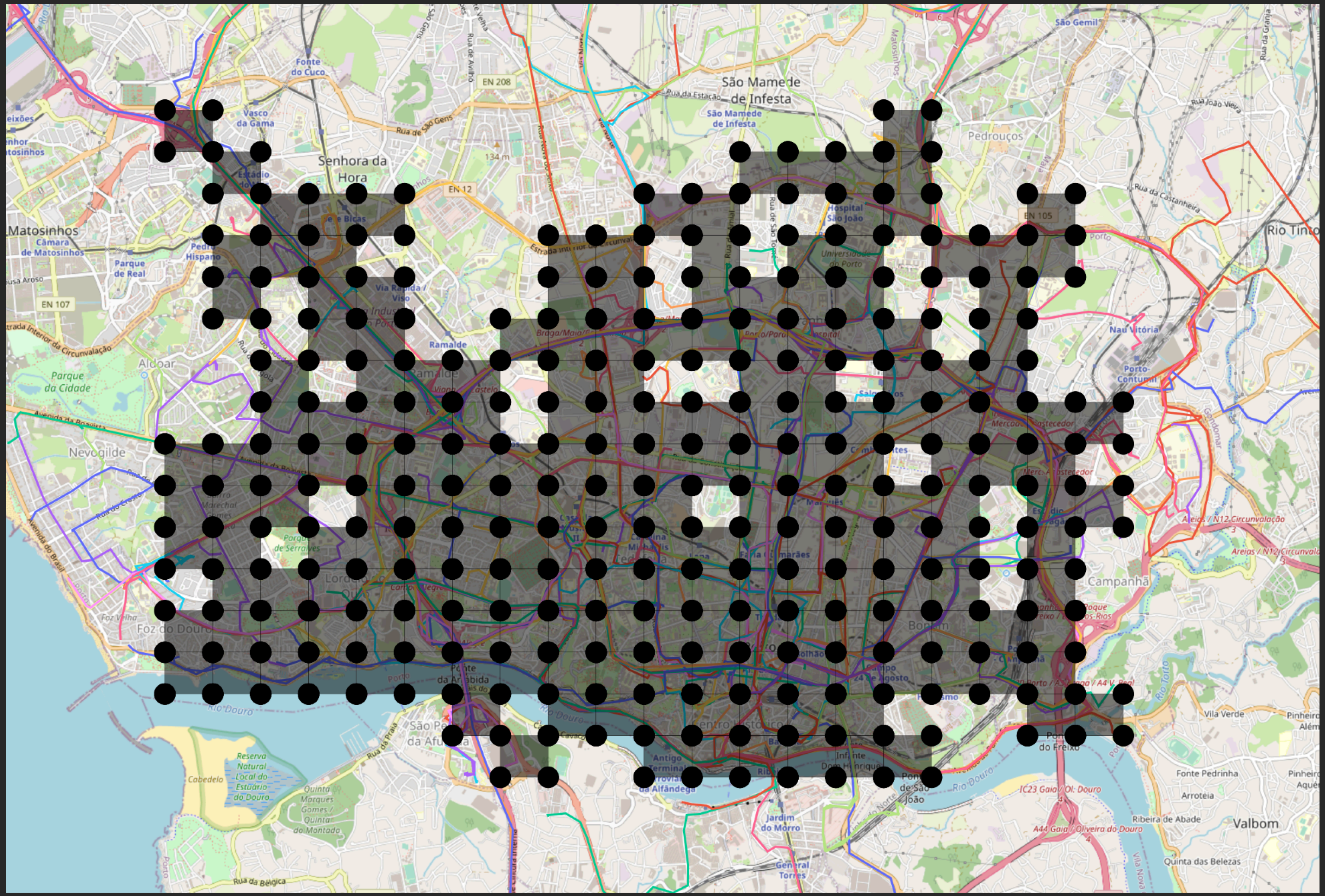
```
-- Repeat for portoCam1...portoCam127:
SPLIT portoCam1
   BEGIN 07-01-2013/12:00am END 07-01-2014/12:00am
   BY TIME 15sec STRIDE 0sec
   INTO chunks1;
-- Repeat for chunks1...chunks127:
PROCESS chunks1 USING porto.py TIMEOUT 1sec
   PRODUCING 3 ROWS
   WITH SCHEMA (plate:STRING="")
   INTO table1;

-- Query 4: Average Taxi Working Hours
SELECT avg(avg_shift) FROM
   (SELECT plate,avg(RANGE(shift, [0,16])) FROM
      (SELECT plate,day,(max(chunk)-min(chunk) as shift) FROM
         table10 UNION table27 GROUP BY plate,day(chunk))
   GROUP BY plate LIMIT 300)
CONSUMING eps=0.33;
```

5.87 hours

(error: **13 mins** with **99.9% confidence**)

Concrete Example (Q4)

Location of **127 Cameras** across Porto, Portugal



Taxi Trajectory Data: EKML/PKDD 2015

"Average working hours of taxis in Porto during 2013-2014?"

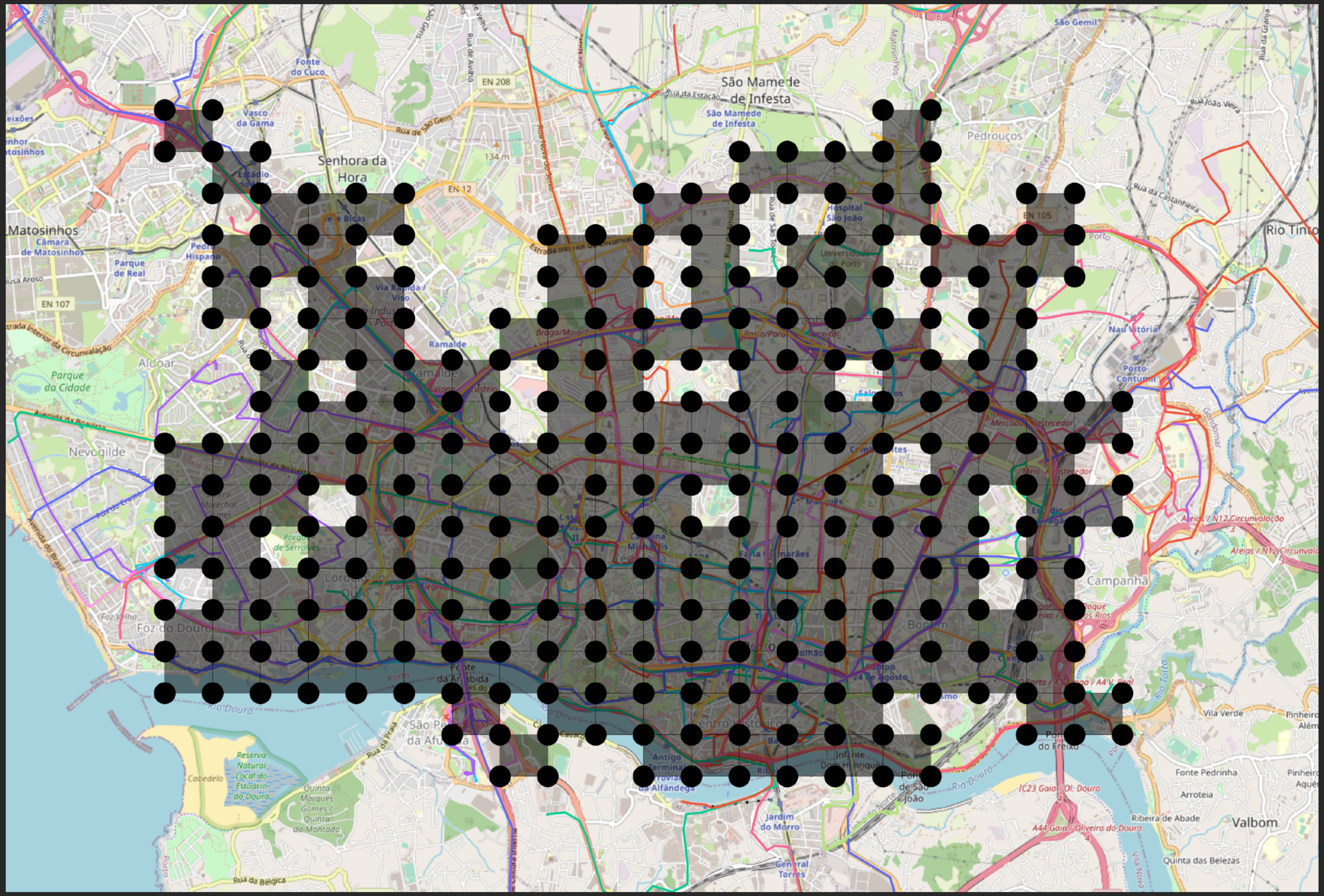
```
-- Repeat for portoCam1...portoCam127:
SPLIT portoCam1
    BEGIN 07-01-2013/12:00am END 07-01-2014/12:00am
    BY TIME 15sec STRIDE 0sec
    INTO chunks1;
-- Repeat for chunks1...chunks127:
PROCESS chunks1 USING porto.py TIMEOUT 1sec
    PRODUCING 3 ROWS
    WITH SCHEMA (plate:STRING="")
    INTO table1;

-- Query 4: Average Taxi Working Hours
SELECT avg(avg_shift) FROM
    (SELECT plate,avg(RANGE(shift, [0,16])) FROM
        (SELECT plate,day,(max(chunk)-min(chunk) as shift) FROM
            table10 UNION table27 GROUP BY plate,day(chunk))
    GROUP BY plate LIMIT 300)
CONSUMING eps=0.33;
```

5.87 hours

(error: 13 mins with 99.9% confidence)

Concrete Example (Q4)

Location of **127 Cameras** across Porto, Portugal



Taxi Trajectory Data: EKML/PKDD 2015

"Average working hours of taxis in Porto during 2013-2014?"

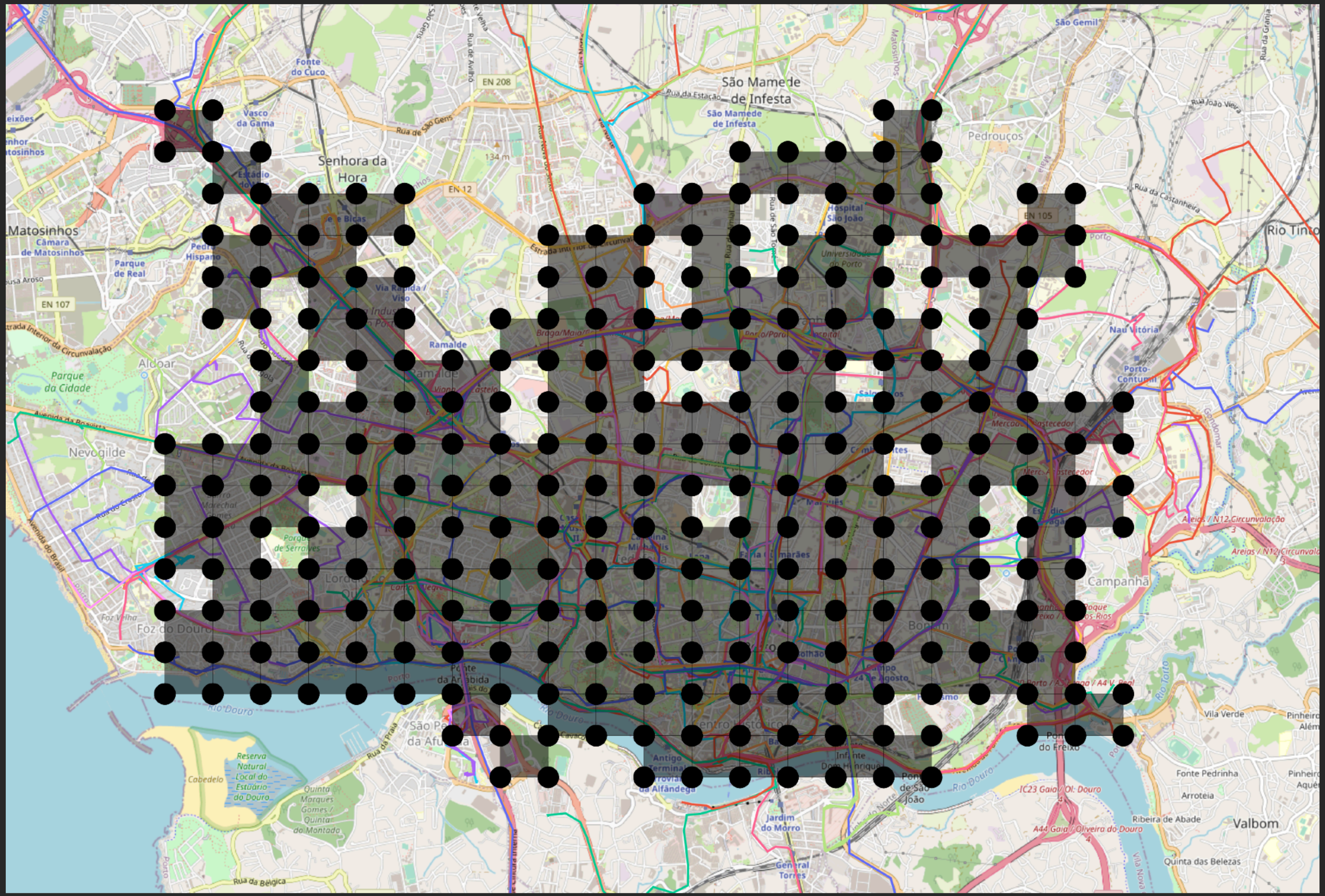
```
-- Repeat for portoCam1...portoCam127:
SPLIT portoCam1
   BEGIN 07-01-2013/12:00am END 07-01-2014/12:00am
   BY TIME 15sec STRIDE 0sec
   INTO chunks1;
-- Repeat for chunks1...chunks127:
PROCESS chunks1 USING porto.py TIMEOUT 1sec
   PRODUCING 3 ROWS
   WITH SCHEMA (plate:STRING="")
   INTO table1;

-- Query 4: Average Taxi Working Hours
SELECT avg(avg_shift) FROM
   (SELECT plate,avg(RANGE(shift, [0,16])) FROM
      (SELECT plate,day,(max(chunk)-min(chunk) as shift) FROM
         table10 UNION table27 GROUP BY plate,day(chunk))
   GROUP BY plate LIMIT 300)
CONSUMING eps=0.33;
```

5.87 hours

(error: 13 mins with 99.9% confidence)

Concrete Example (Q4)

Location of **127 Cameras** across Porto, Portugal



Taxi Trajectory Data: EKML/PKDD 2015

"Average working hours of taxis in Porto during 2013-2014?"

```
-- Repeat for portoCam1...portoCam127:
SPLIT portoCam1
   BEGIN 07-01-2013/12:00am END 07-01-2014/12:00am
   BY TIME 15sec STRIDE 0sec
   INTO chunks1;
-- Repeat for chunks1...chunks127:
PROCESS chunks1 USING porto.py TIMEOUT 1sec
   PRODUCING 3 ROWS
   WITH SCHEMA (plate:STRING="")
   INTO table1;

-- Query 4: Average Taxi Working Hours
SELECT avg(avg_shift) FROM
   (SELECT plate,avg(RANGE(shift, [0,16])) FROM
      (SELECT plate,day,(max(chunk)-min(chunk) as shift) FROM
        table10 UNION table27 GROUP BY plate,day(chunk))
   GROUP BY plate LIMIT 300)
CONSUMING eps=0.33;
```

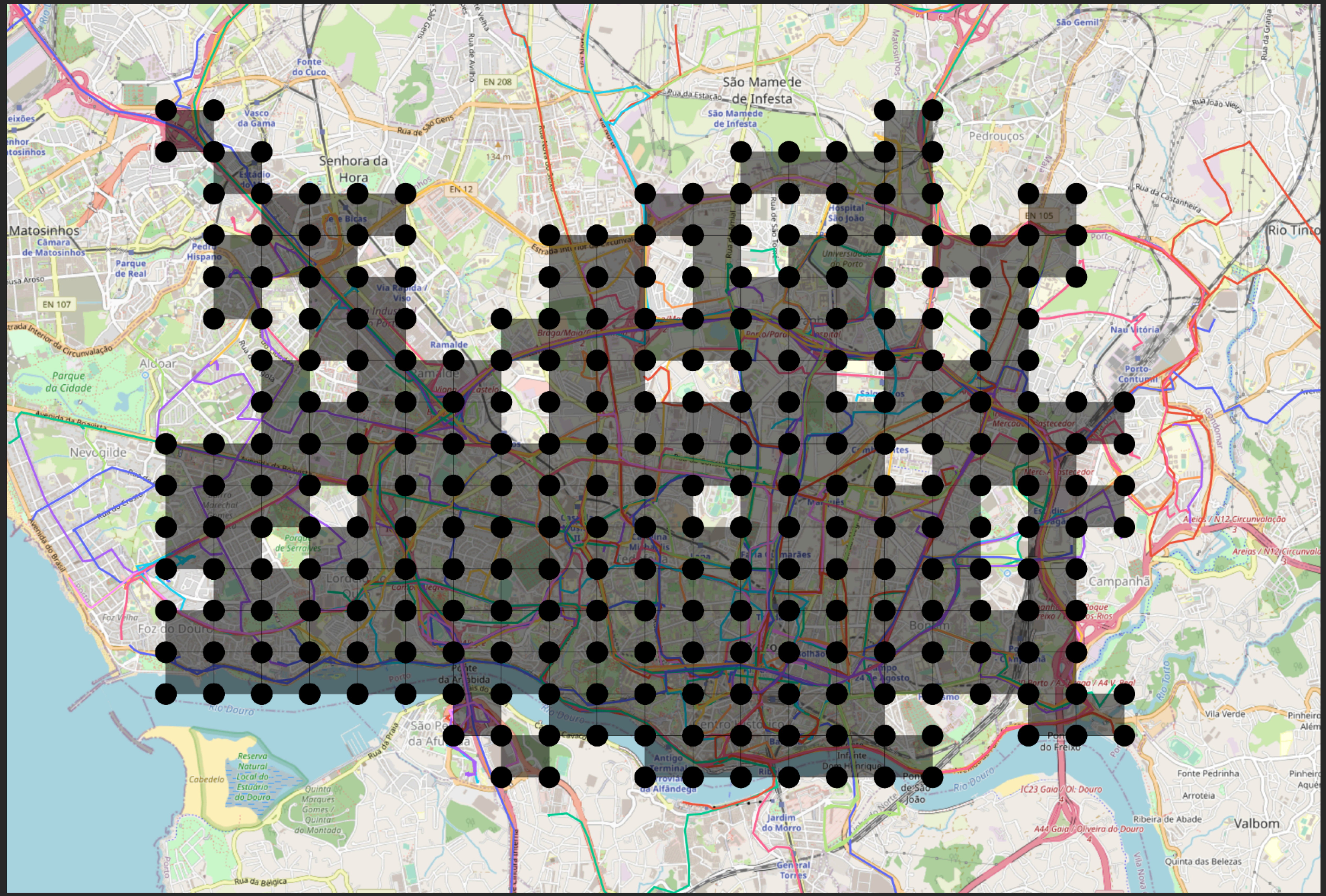
5.87 hours

(error: **13 mins** with **99.9% confidence**)

Concrete Example (Q4)

"Average working hours of taxis in Porto during 2013-2014?"

Location of **127 Cameras** across Porto, Portugal



Taxi Trajectory Data: EKML/PKDD 2015

```
-- Repeat for portoCam1...portoCam127:
SPLIT portoCam1
   BEGIN 07-01-2013/12:00am END 07-01-2014/12:00am
   BY TIME 15sec STRIDE 0sec
   INTO chunks1;
-- Repeat for chunks1...chunks127:
PROCESS chunks1 USING porto.py TIMEOUT 1sec
   PRODUCING 3 ROWS
   WITH SCHEMA (plate:STRING="")
   INTO table1;

-- Query 4: Average Taxi Working Hours
SELECT avg(avg_shift) FROM
   (SELECT plate,avg(RANGE(shift, [0,16])) FROM
      (SELECT plate,day,(max(chunk)-min(chunk) as shift) FROM
         table10 UNION table27 GROUP BY plate,day(chunk))
   GROUP BY plate LIMIT 300)
CONSUMING eps=0.33;
```

5.87 hours

(error: **13 mins** with **99.9% confidence**)

Sensitivity Calculation (Q4)

- Cam10: (p=45, K=1), Cam27: (p=195s,K=1)
- **SPLIT** 15s chunks, max 3 rows per chunk

$$\Delta(t_{10}) = \text{maxrows} \times K \times (1 + \lceil \frac{\rho}{c} \rceil)$$
$$= 3 \times 1 \times (\lceil \frac{45}{15} \rceil + 1) = 12$$

$$\Delta(t_{27}) = \ldots = 3 \times 1 \times (\lceil \frac{195}{15} \rceil + 1) = 42$$

- **UNION** t10 and t27:
$$\Delta(t_{10} \cup t_{27}) = 12 + 42 = 54$$
- **ASSUMPTIONS:** max 24-hour day, 300 total taxis
- **AVERAGE**
$$\Delta^{AVG} = \frac{\text{rows impacted} \times \text{impact per row}}{\text{total rows in table}} = \frac{\Delta(\cup) \times \text{range}}{\text{taxis} \times \text{days}} = \frac{54 \times 24}{300\text{plates} \times 365\text{days}} = 0.012$$

```
-- Repeat for portoCam1...portoCam127:
SPLIT portoCam1
   BEGIN 07-01-2013/12:00am END 07-01-2014/12:00am
   BY TIME 15sec STRIDE 0sec
   INTO chunks1;
-- Repeat for chunks1...chunks127:
PROCESS chunks1 USING porto.py TIMEOUT 1sec
   PRODUCING 3 ROWS
   WITH SCHEMA (plate:STRING="")
   INTO table1;

-- Query 4: Average Taxi Working Hours
SELECT avg(avg_shift) FROM
   (SELECT plate,avg(RANGE(shift, [0,16])) FROM
      (SELECT plate,day,(max(chunk)-min(chunk) as shift) FROM
         table10 UNION table27 GROUP BY plate,day(chunk))
   GROUP BY plate LIMIT 300)
CONSUMING eps=0.33;
```

If we scale distribution by **0.012**, noise sample will be **<=13** with **99.9% probability**

(Minutes!)

Privid Sensitivity Rules

NOTATION

$\mathcal{P}$	Privacy policy for each camera: $\{(\rho, K)_c \;\forall\; c \in \text{cameras}\}$
$\Delta_{\mathcal{P}}(R)$	Maximum number of rows in relation R that could differ by the addition or removal of any (ρ, K)-bounded event.
$\tilde{C_r}(R,a)$	Range constraint: range of attribute a in R
$\tilde{C_s}(R)$	Size constraint: upper bound on total number of rows in R
$\varnothing$	Indicates that a relational operator leaves a constraint unbound. If this constraint is required for the aggregation, it must be bound by a predecessor. If it is not required, it can be left unbound.

AGGREGATION FUNCTIONS

Function	Definition	Constraints	Sensitivity ($\Delta(Q)$)
Count	$Q := \Pi_{\text{count}(*)}(R)$	Δ	$1 \cdot \Delta(R)$
Sum	$Q := \Pi_{\text{sum}(a)}(R)$	$\Delta, \tilde{C_r}$	$\Delta(R) \cdot \tilde{C_r}(R,a)$
Average	$Q := \Pi_{\text{avg}(a)}(R)$	$\Delta, \tilde{C_r}, \tilde{C_s}$	$\frac{\Delta(R) \cdot \tilde{C_r}(R,a)}{\tilde{C_s}(R)}$
Std. Dev	$Q := \Pi_{\text{stddev}(a)}(R)$	$\Delta, \tilde{C_r}, \tilde{C_s}$	$\Delta(R) \cdot \tilde{C_r}(R,a) / \sqrt{\tilde{C_s}(R)}$
Argmax	$Q := \Pi_{\text{argmax}(a)}(R)$	$\Delta, a \in K$	$\max_{k \in K} \Delta(\sigma_{a=k}(R))$

RELATIONAL OPERATORS

Operator	Type	Definition	$\Delta_{\mathcal{P}}(R')$	$\tilde{C_r}(R', a_i)$	$\tilde{C_s}(R')$
Base Case	Base Table	R	$mr \cdot K \cdot (1 + \lceil \frac{\rho}{c} \rceil)$	$\varnothing$	$\varnothing$
Selection (σ)	Standard selection: rows from R that match WHERE condition	$R' := \sigma_{\text{WHERE}(\ldots)}(R)$	$\Delta_{\mathcal{P}}(R)$	$\tilde{C_r}(R, a_i)$	$\tilde{C_s}(R)$
	Limit: first x rows from R	$R' := \sigma_{\text{LIMIT}=x}(R)$	$\Delta_{\mathcal{P}}(R)$	$\tilde{C_r}(R, a_i)$	$\min(x, \tilde{C_s}(R))$
Projection (Π)	Standard projection: select attributes $a_i, \ldots$ from R	$R' := \Pi_{a_i, \ldots}$	$\Delta_{\mathcal{P}}(R)$	$\tilde{C_r}(R, a_i)$	$\tilde{C_s}(R)$
	Apply (user-provided, but stateless) f to column a_i	$R' := \Pi_{f(a_i), \ldots}$	$\Delta_{\mathcal{P}}(R)$	$\varnothing$	$\tilde{C_s}(R)$
	Add range constraint to column a_i	$R' := \Pi_{a_i \in [l_i, u_i], \ldots}$	$\Delta_{\mathcal{P}}(R)$	$[l_j, u_j]$ if $a_i \neq \varnothing$ $\tilde{C_r}(R, a_i)$ otherwise	$\tilde{C_s}(R)$
GroupBy (γ)	Group attribute(s) (g_j) are `chunk` (or binned `chunk`) or `region`	$R' := {}_{g_j, \ldots}\gamma_{\text{agg}(a_i), \ldots}$ $g_j := \text{chunk} \mid \text{bin}(\text{chunk})$	Equation 5.2	$\Delta(\text{agg}(a_i))$	$\frac{\tilde{C_s}(R)}{\text{(bin size)}}$
	Group attribute(s) (g_j) are *not* `chunk` or `region`	$R' := {}_{g_j, \ldots}\gamma_{\text{agg}(a_i), \ldots}$	$\Delta_{\mathcal{P}}(R)$	$\varnothing$	$\varnothing$
	... discrete set of keys provided for each group (constrains size)	$R' := {}_{g_j \in K_j, \ldots}\gamma_{\text{agg}(a_i), \ldots}$	...	...	$\Pi_j \lvert K_j \rvert$
	... aggregation constrains range: $agg(a_i) \in [l_i, u_i]$	$R' := {}_{g_j, \ldots}\gamma_{\text{agg}(a_i) \in [l_i, u_i], \ldots}$	...	$[l_j, u_j]$ if $a_i \neq \varnothing$ $\tilde{C_r}(R, a_i)$ otherwise	...
Joins* ($\bowtie$)	*When *immediately* preceeded by GroupBy *over the same key(s)* ... equijoin on g_j (intersection on g_j) ... outer join on g_j (union on g_j)	$R' := {}_g\gamma_{\text{agg}(a)}(R_1 \bowtie_g \ldots \bowtie_g R_n)$ $R' := {}_g\gamma_{\text{agg}(a)}(R_1 ⟗_g \ldots ⟗_g R_n)$	$\sum_{i=1}^{n} \Delta_{\mathcal{P}}(R_i)$	(GroupBy rules)	(GroupBy rules)

Practical Considerations

Video Owner (VO)

- What if some individuals are visible for longer than ρ?
- What if ρ is very large?

Analysts

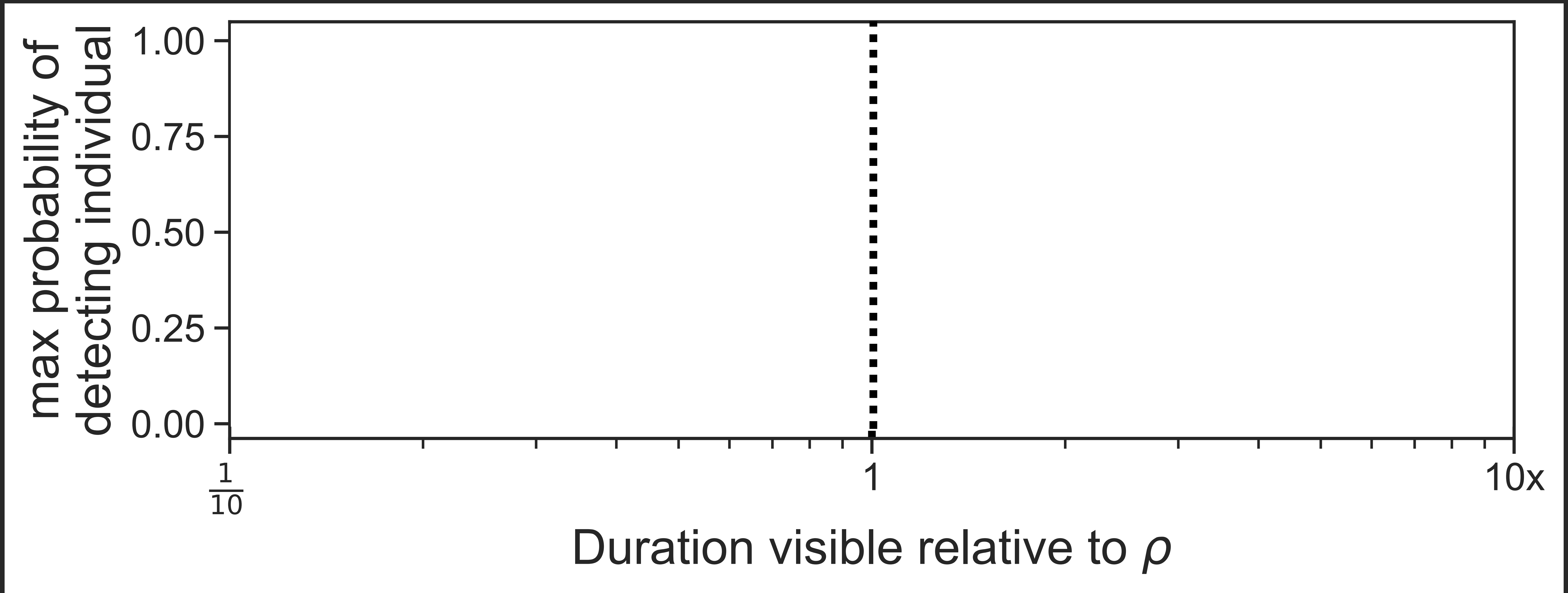
- How should I choose query parameters?
- How many queries can I execute? At what granularity?

But what if… **I'm visible for longer than ρ?**

Privacy **degrades gracefully**

Output was large,
very unlikely to be noise

Output was small,
could just be noise

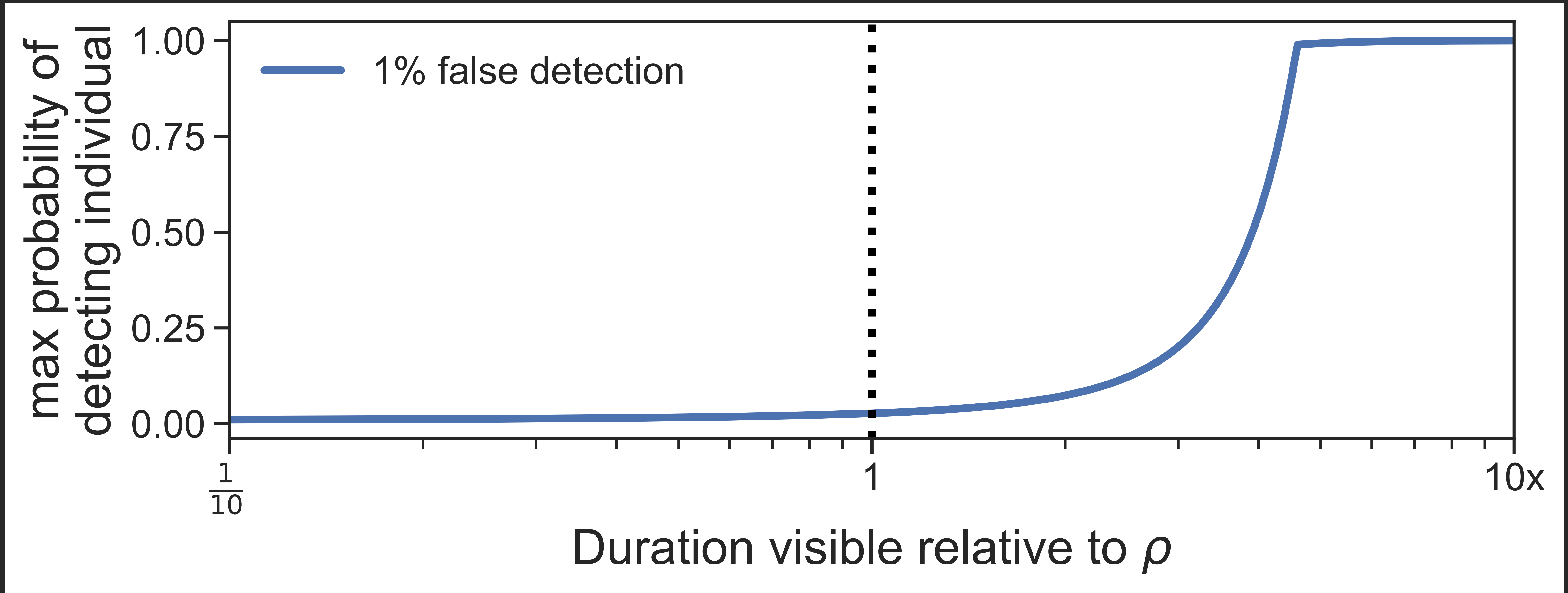


But what if… **I’m visible for longer than ρ?**

Privacy **degrades gracefully**

Output was large,
very unlikely to be noise

Output was small,
could just be noise



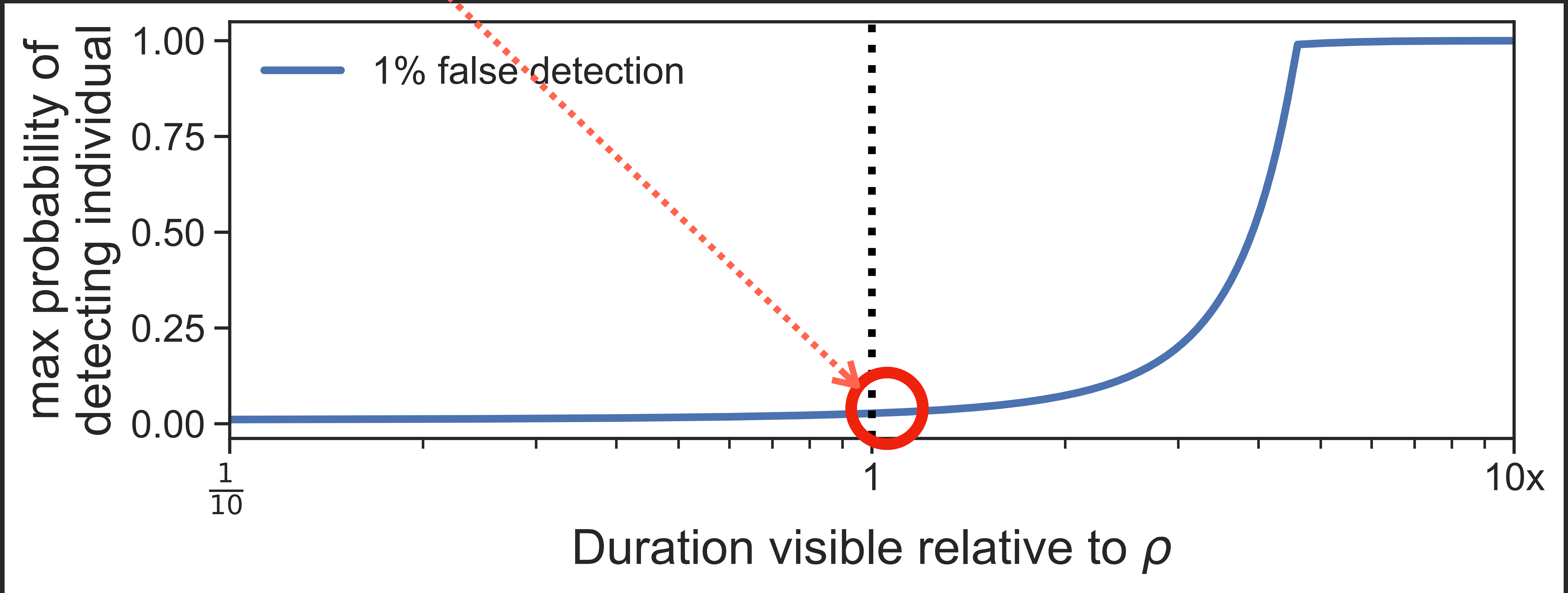
But what if… **I'm visible for longer than ρ?**

Privacy **degrades gracefully**

Privacy is roughly the same just beyond ρ

Output was large,
very unlikely to be noise

Output was small,
could just be noise



But what if… **I'm visible for longer than ρ?**

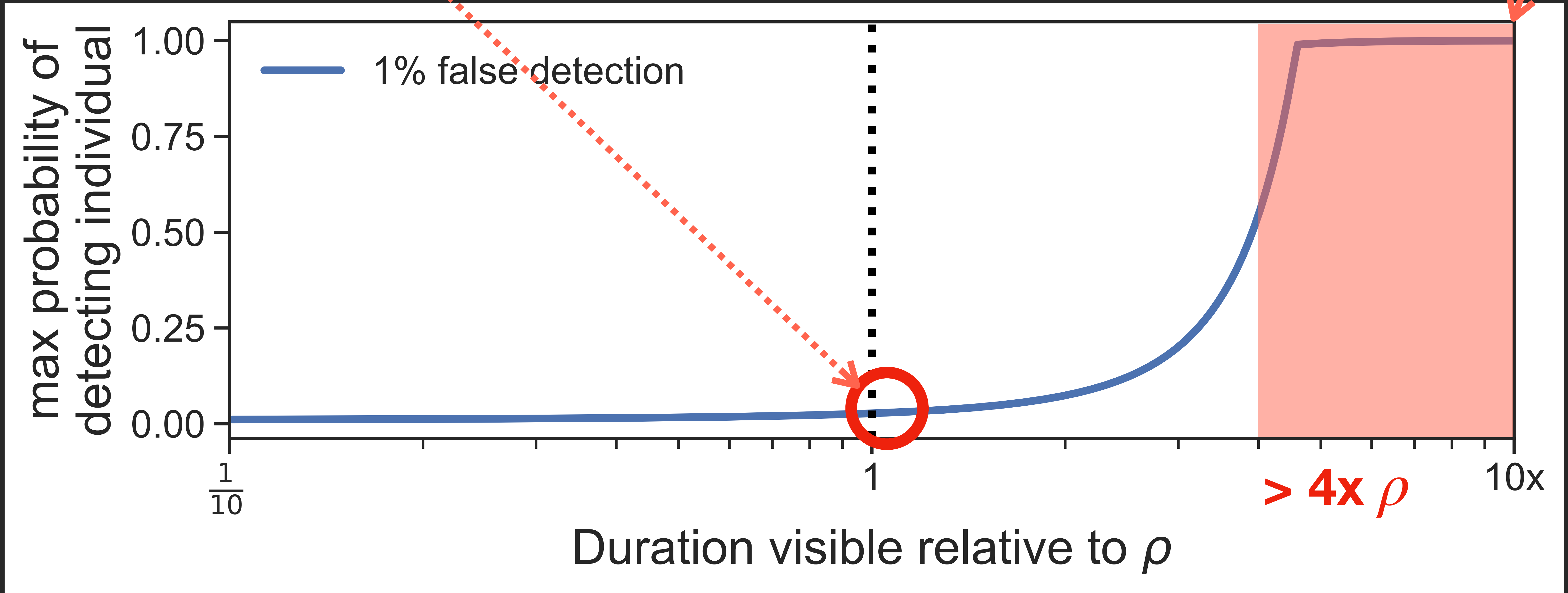
Privacy **degrades gracefully**

Privacy is roughly the same just beyond ρ

Must be visible for many multiples of ρ
to be detected…
but details still protected

Output was large,
very unlikely to be noise

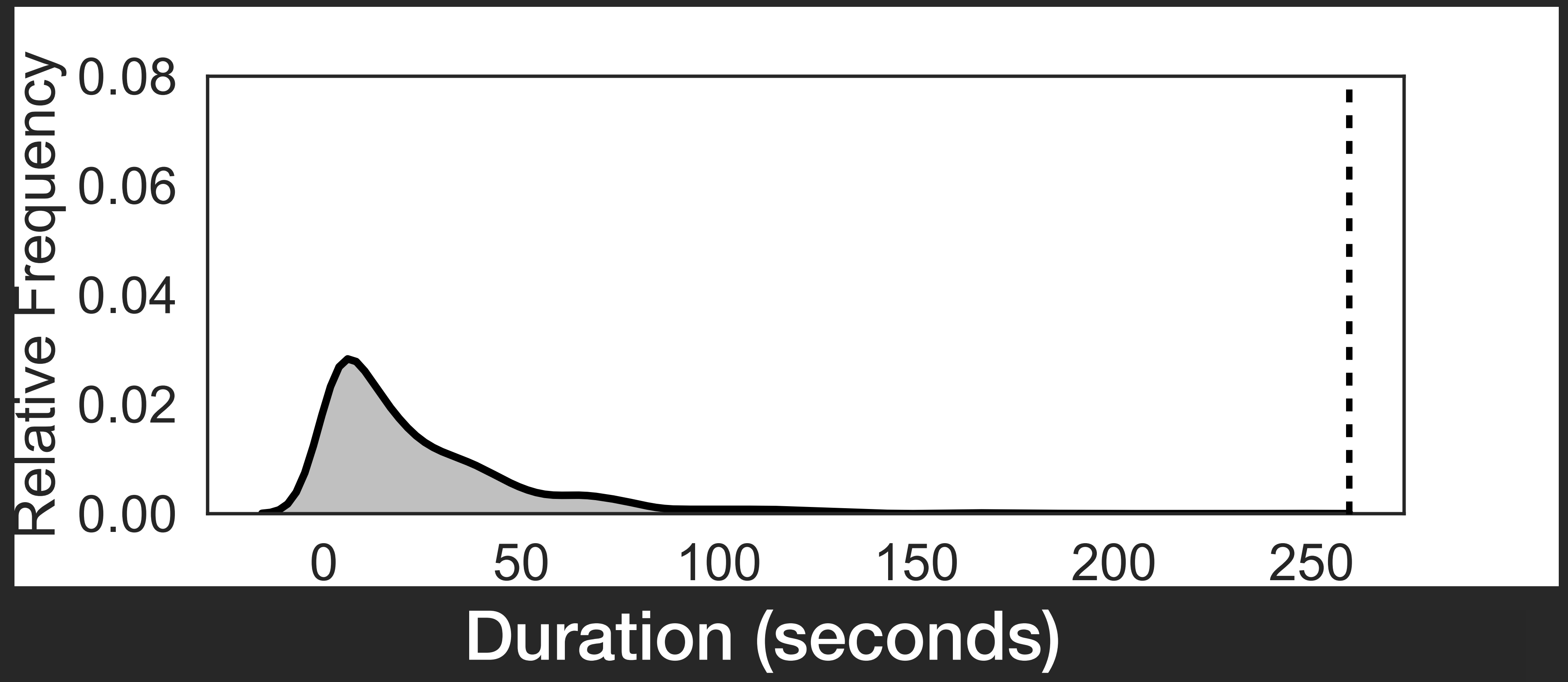
Output was small,
could just be noise



But what if… **longest duration is *too* long?**

Add static **frame mask**

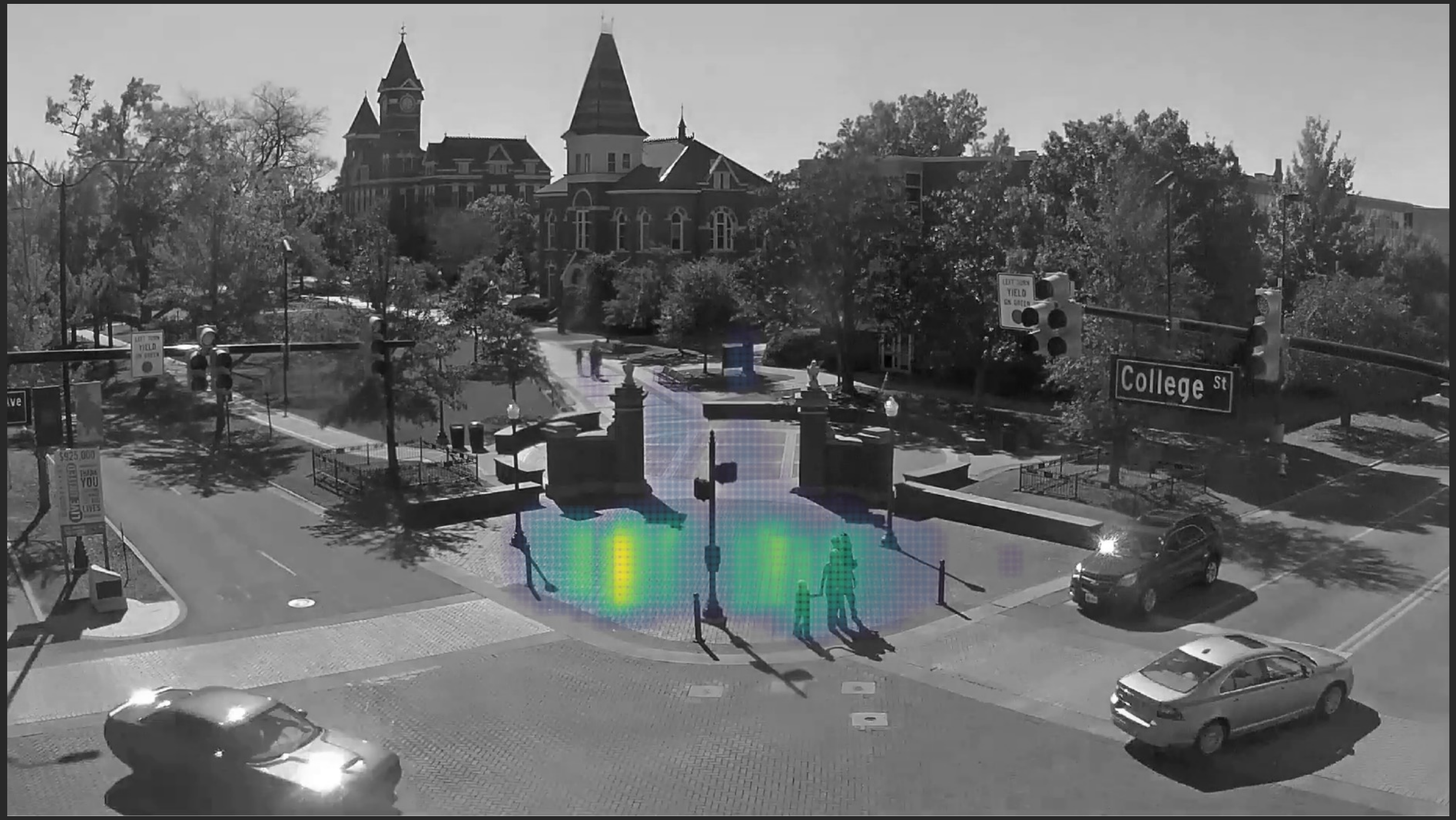
- Remove areas where people linger, reducing maximum duration
- Minimal impact on query accuracy

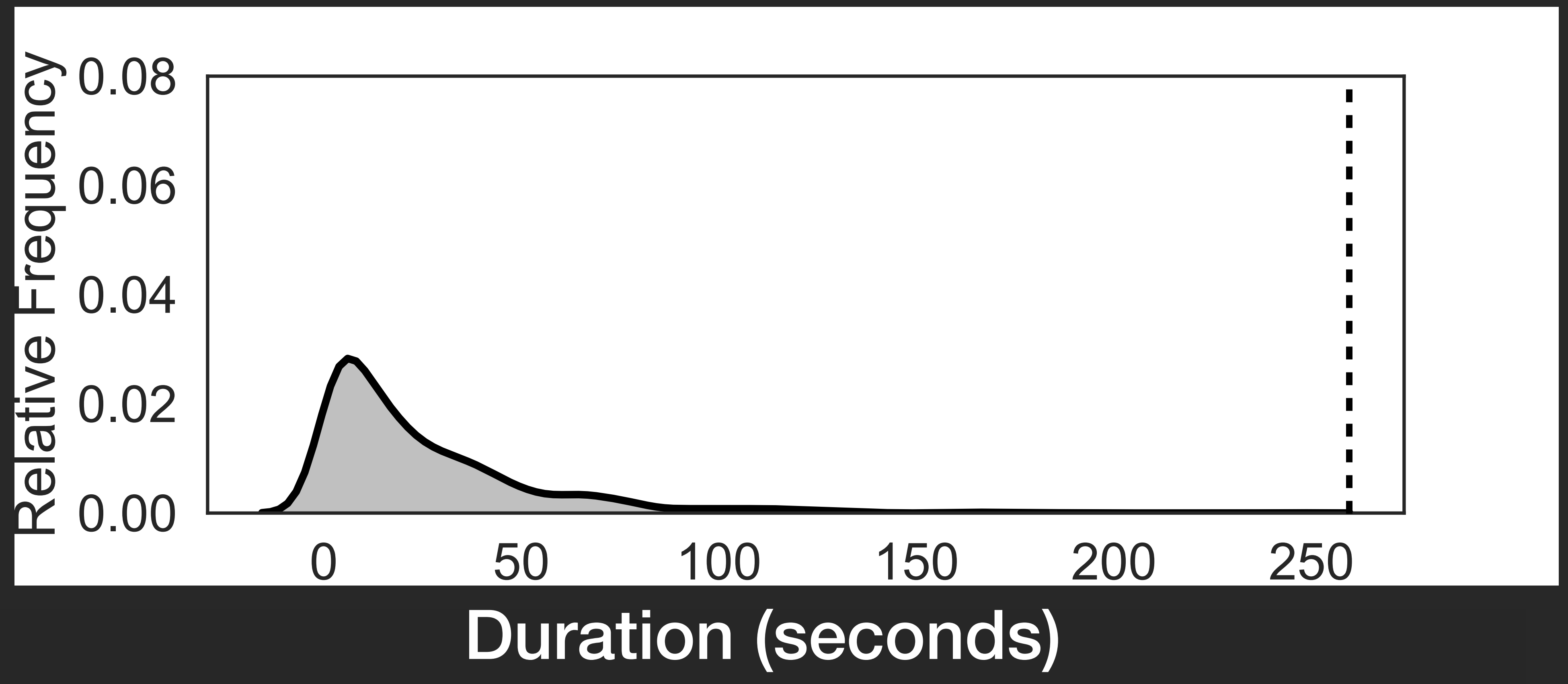


But what if… **longest duration is *too* long?**

Add static **frame mask**

- Remove areas where people linger, reducing maximum duration
- Minimal impact on query accuracy

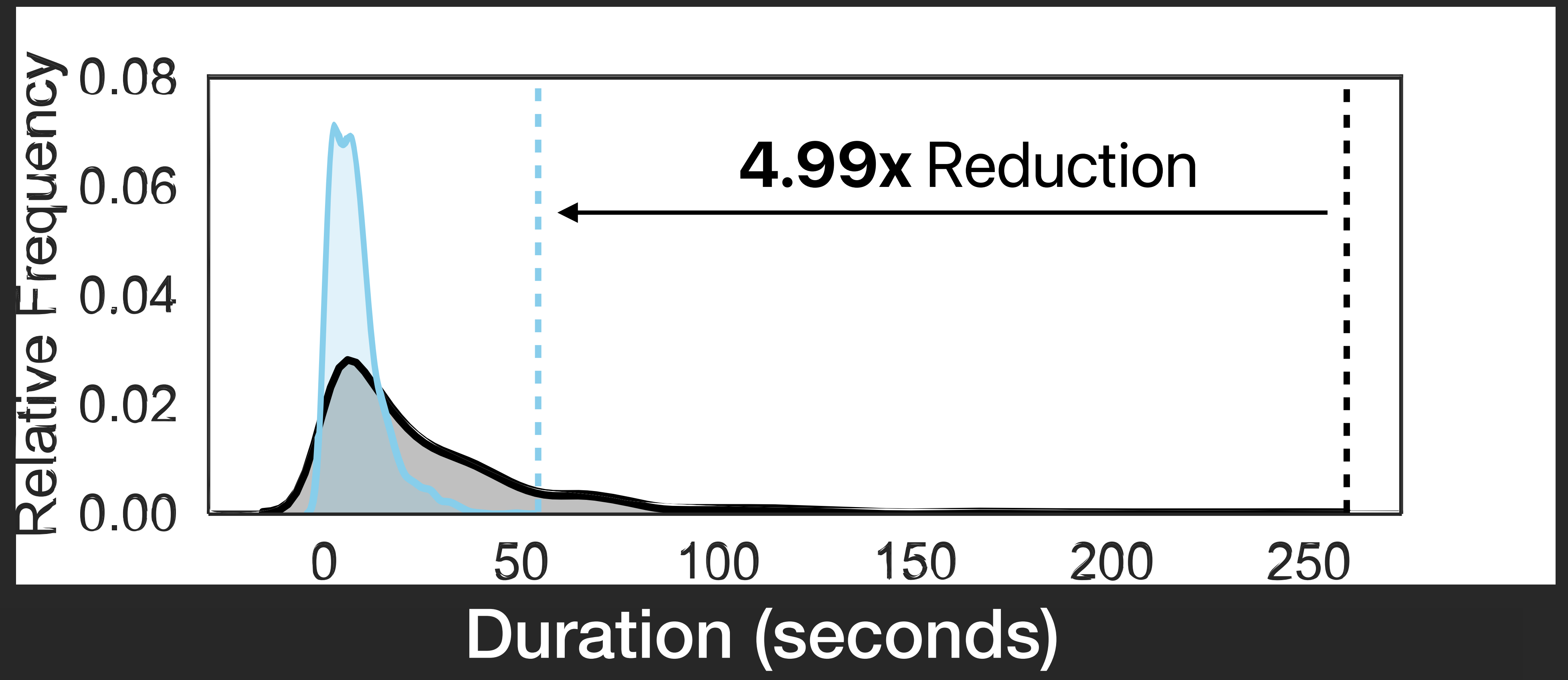




But what if… **longest duration is *too* long?**

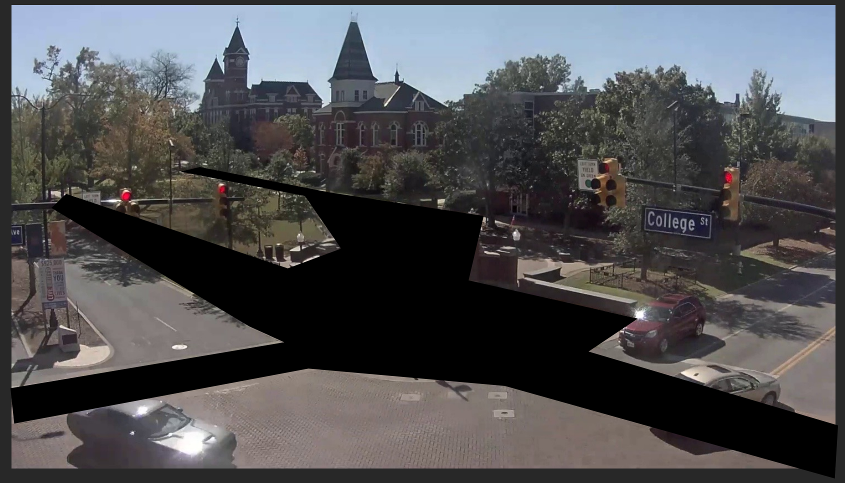
Add static **frame mask**

- Remove areas where people linger, reducing maximum duration
- Minimal impact on query accuracy



General privacy policies with masks

Analyst chooses mask at query time, gets corresponding threshold

No Mask		$\rho = 250s$	Baseline
Mask 1: Center		$\rho = 50s$	5X lower noise
Mask 2: Walkways		$\rho = 30s$	8X lower noise
...	...		

Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound

Analyst's Budget-Granularity Tradeoff

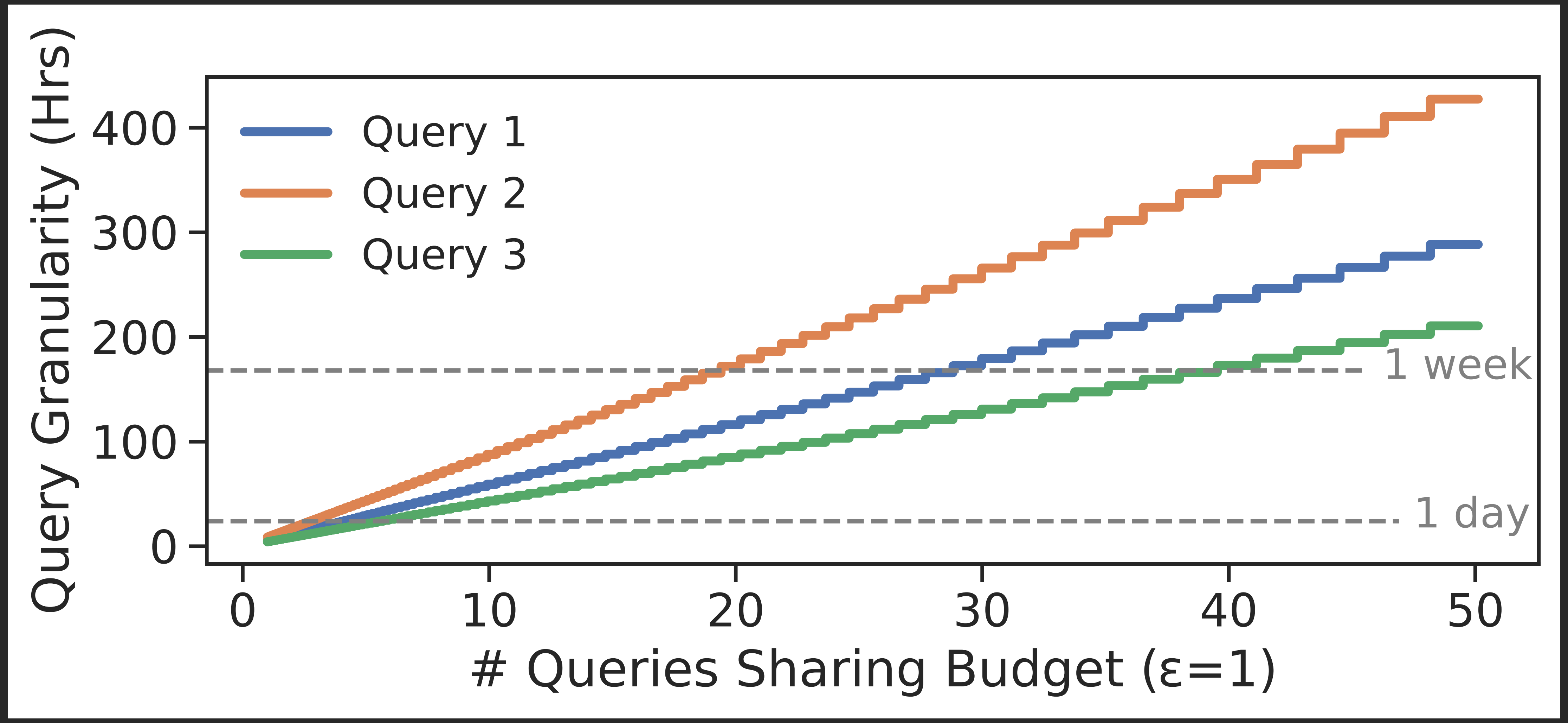
(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both

Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both

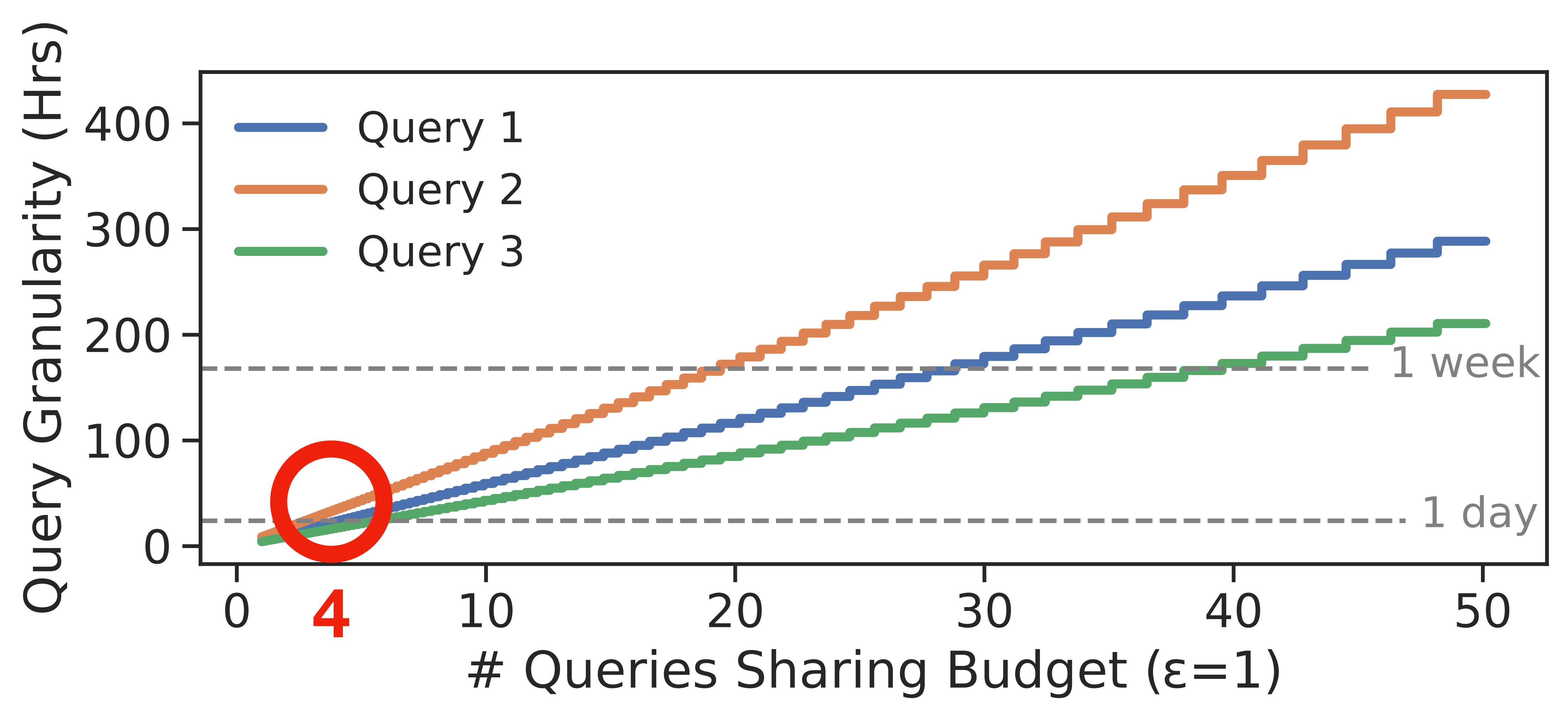


(Fixed 1% error goal)

Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both

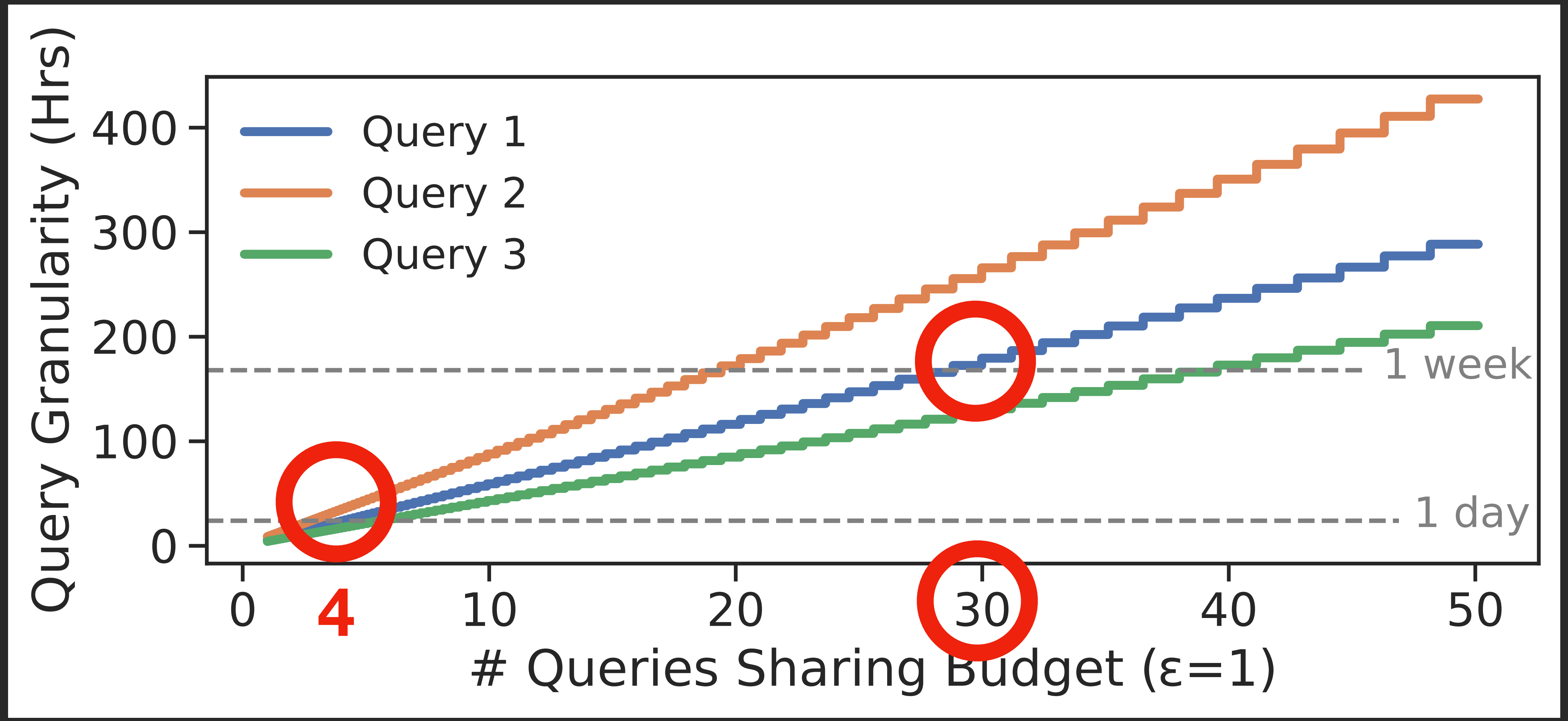


(Fixed 1% error goal)

Analyst’s Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both

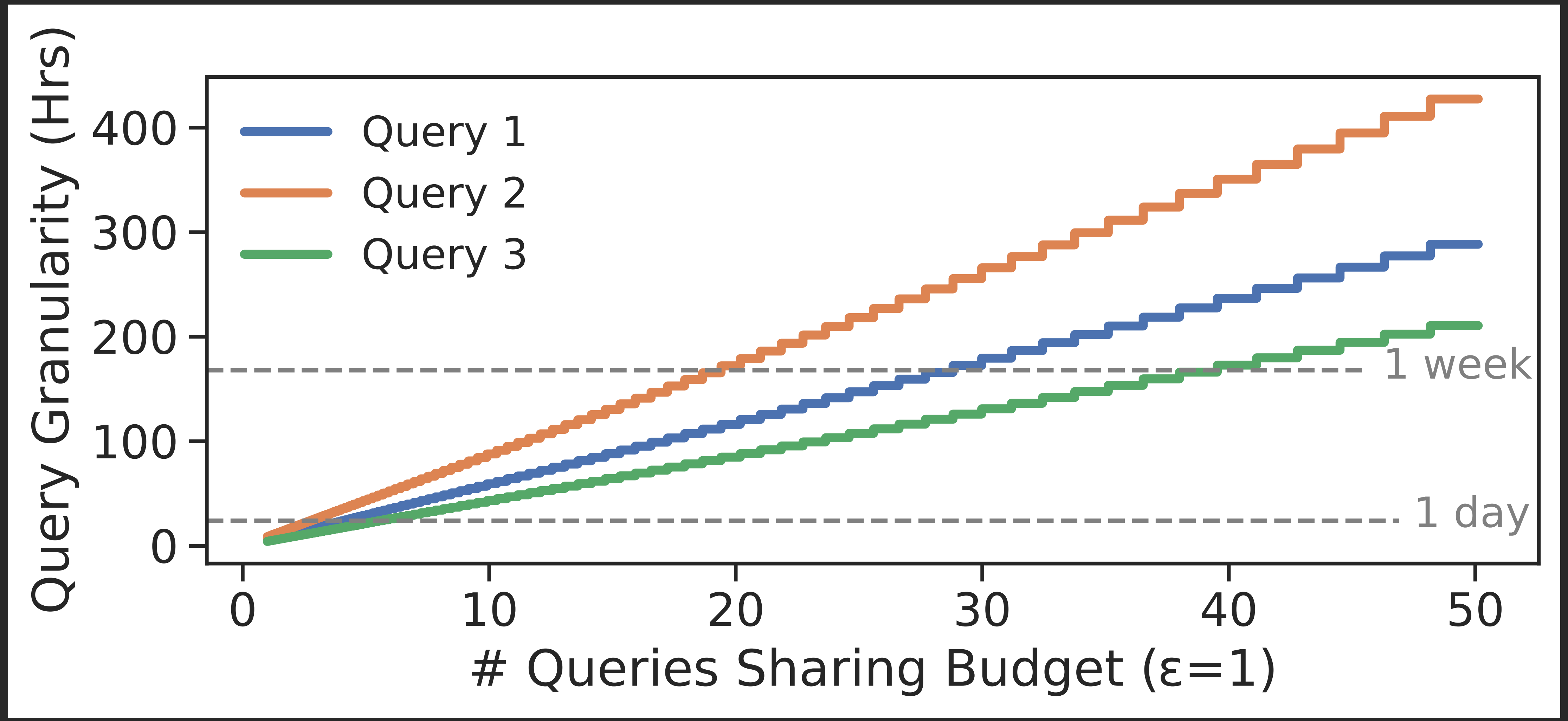


(Fixed 1% error goal)

Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both

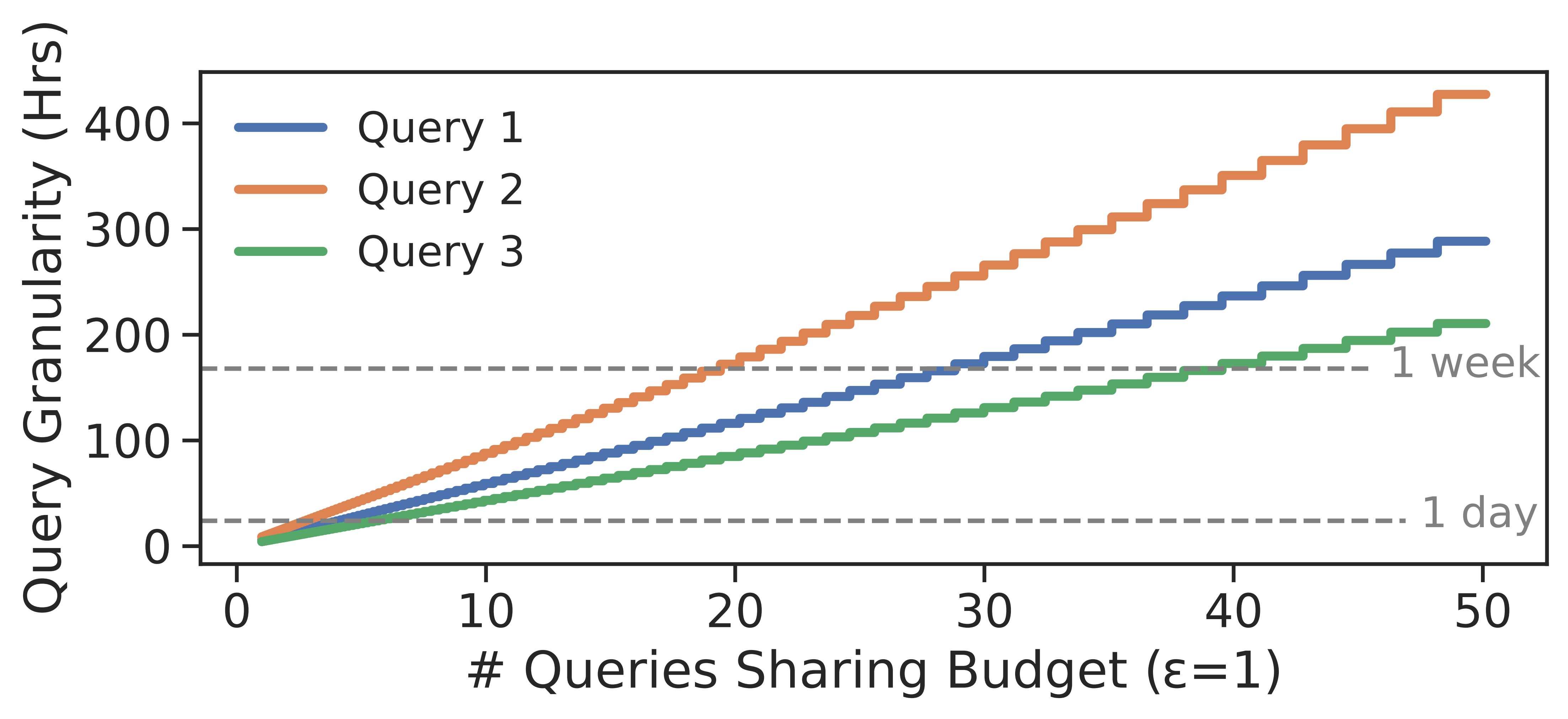


(Fixed 1% error goal)

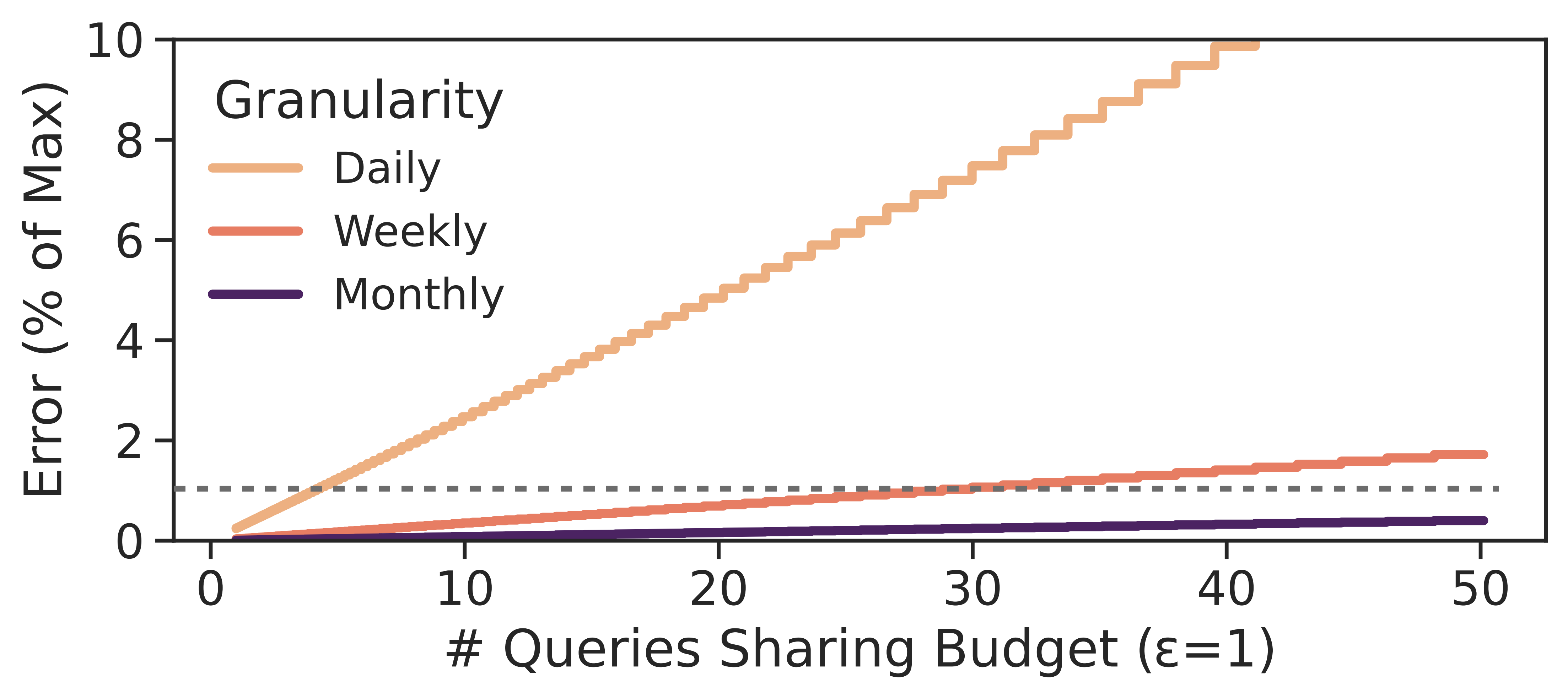
Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both



(Fixed 1% error goal)

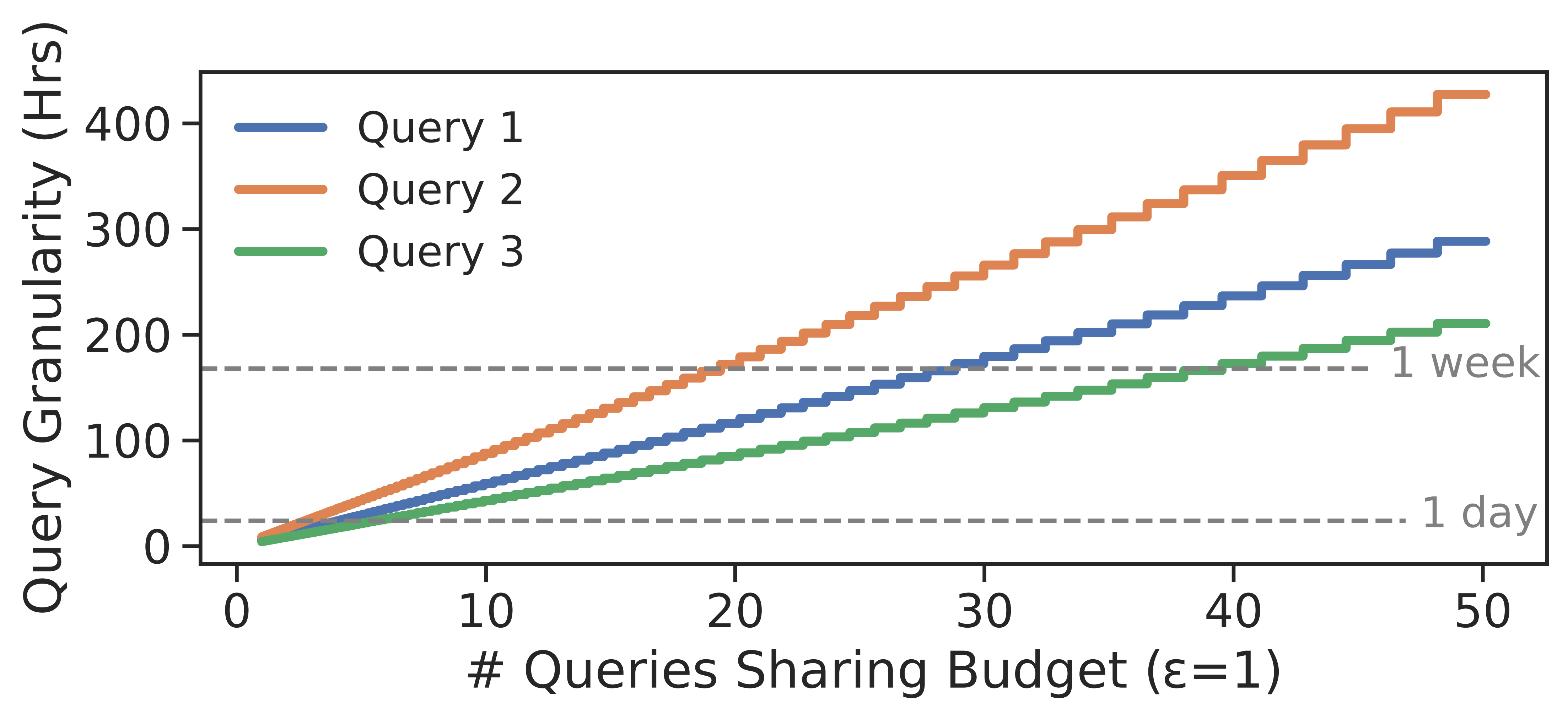


(Fixed Query 1)

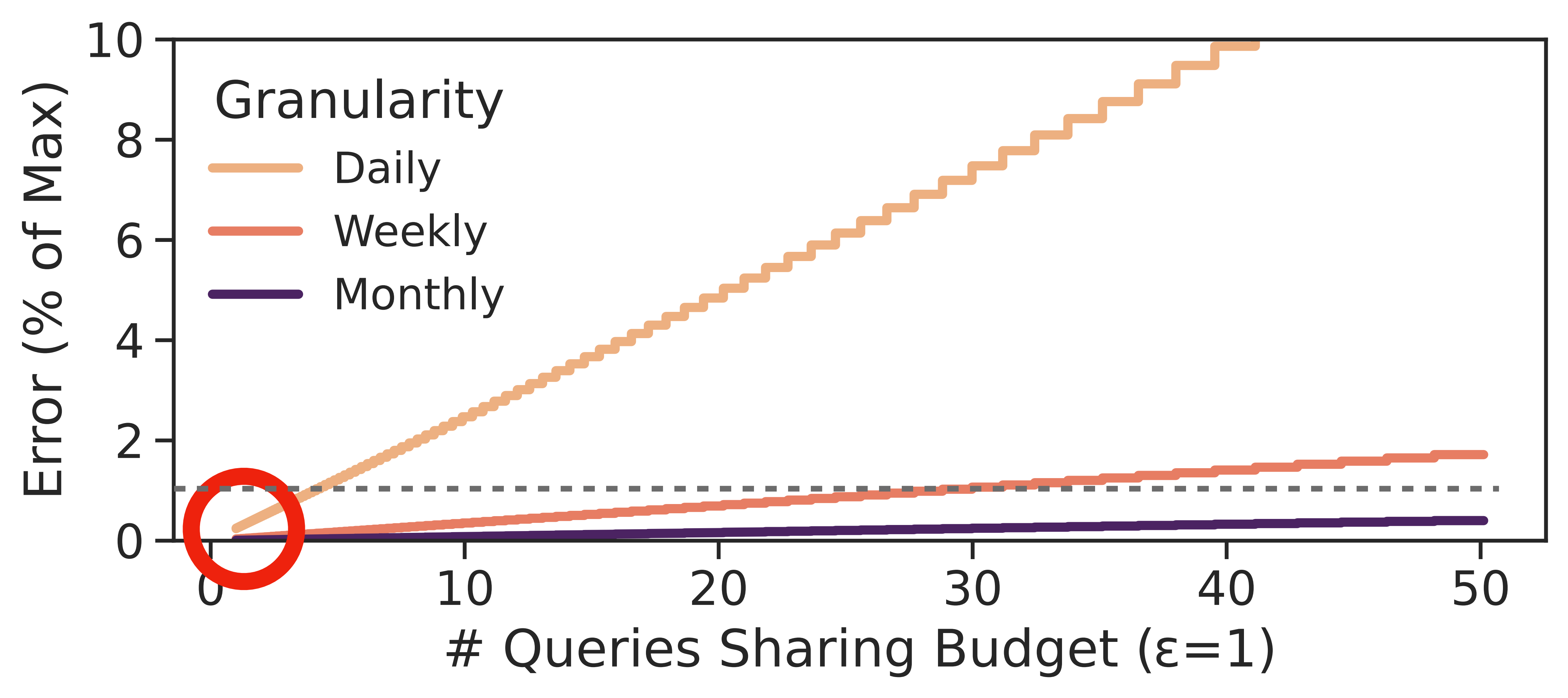
Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both



(Fixed 1% error goal)

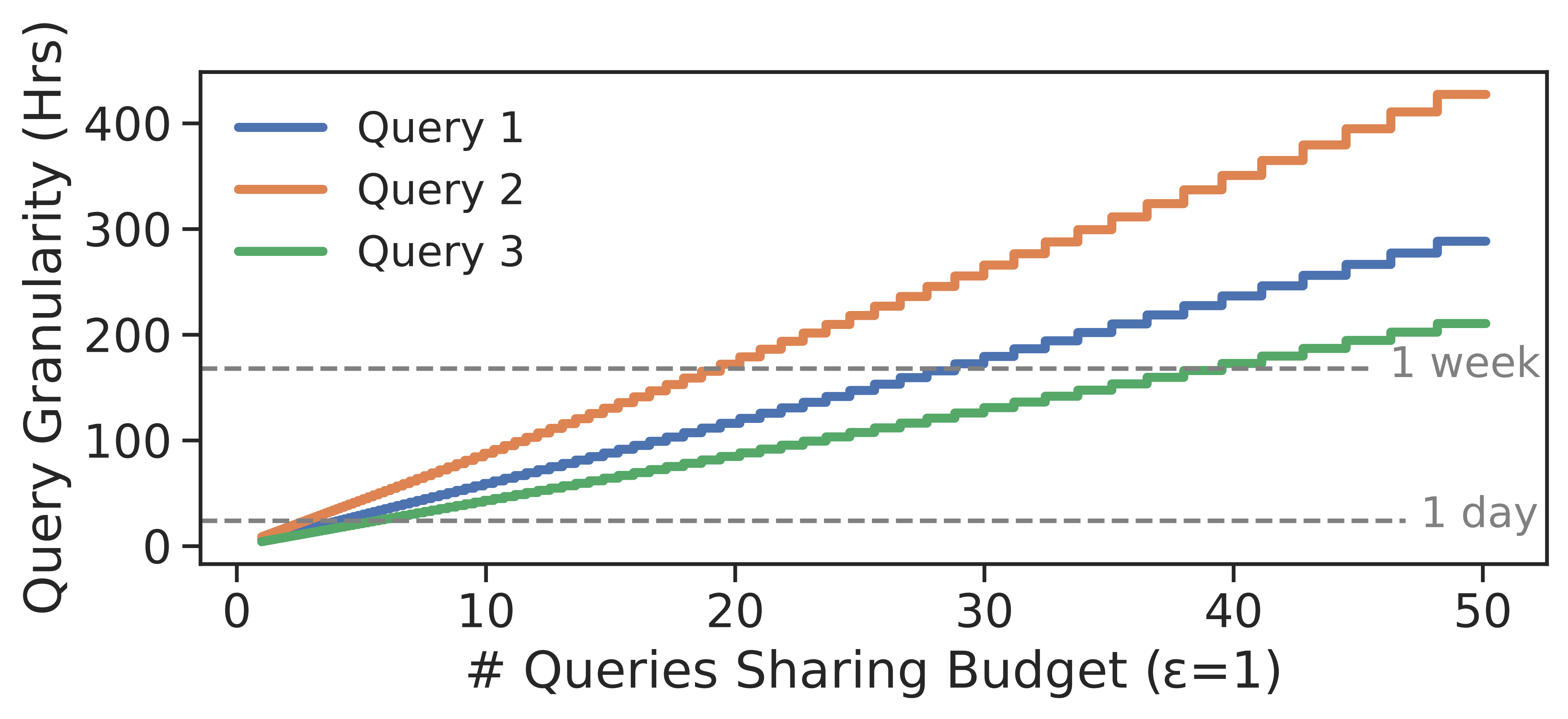


(Fixed Query 1)

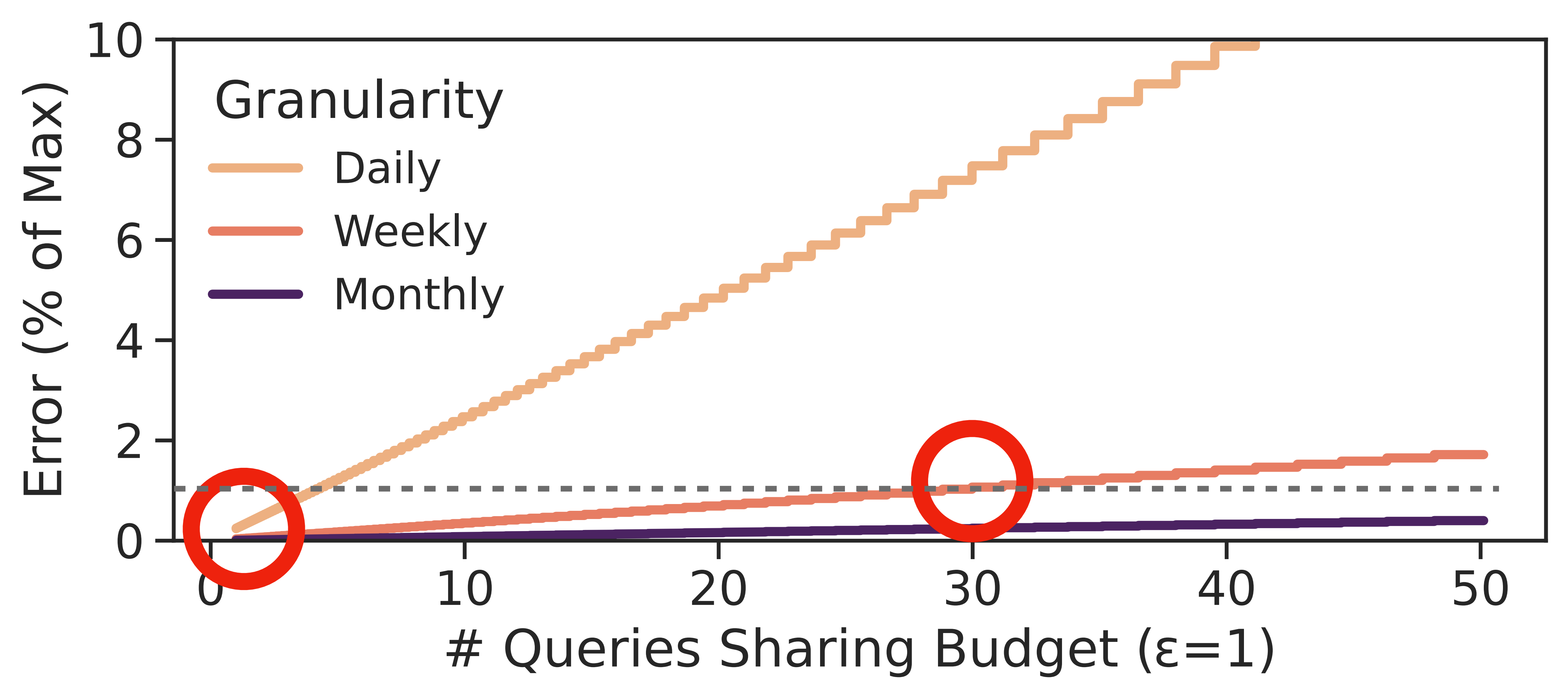
Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both



(Fixed 1% error goal)

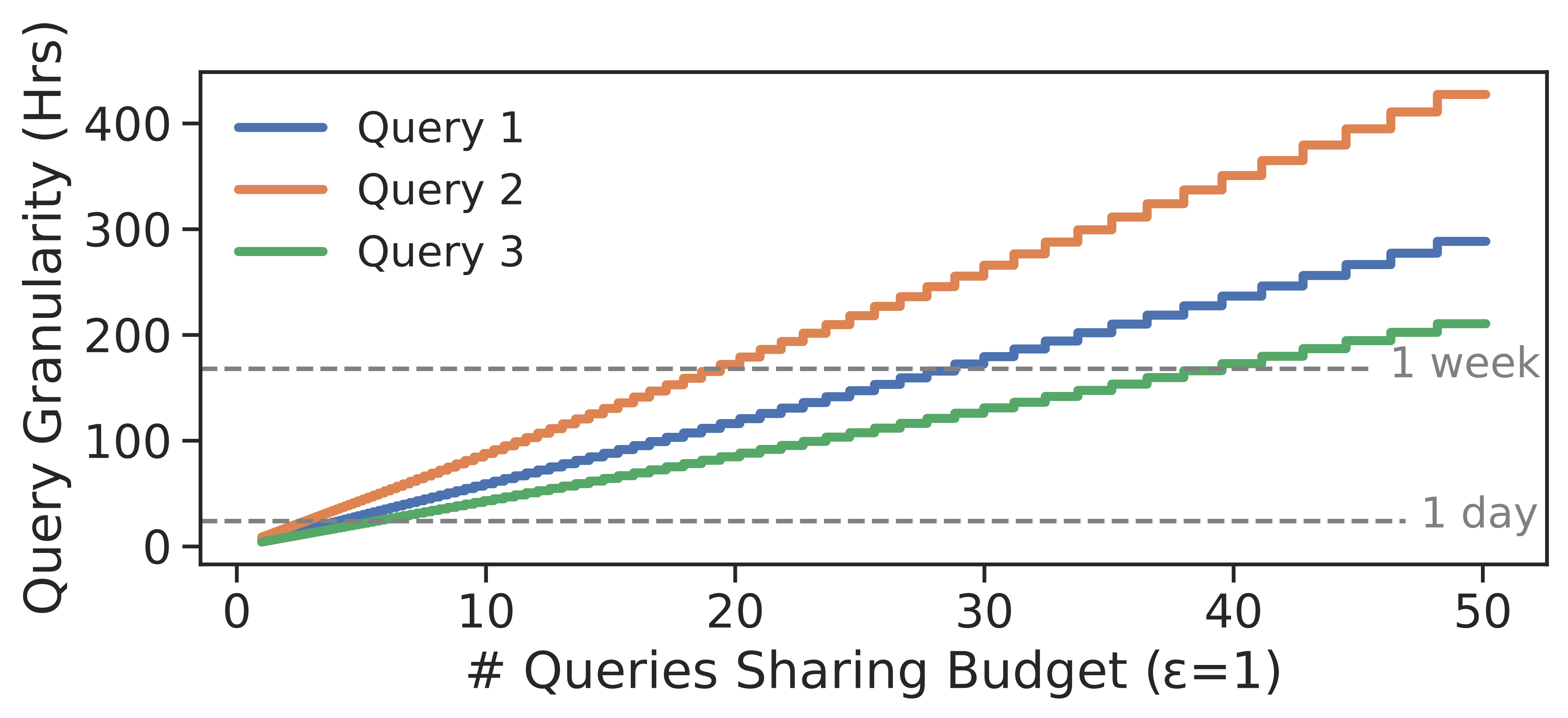


(Fixed Query 1)

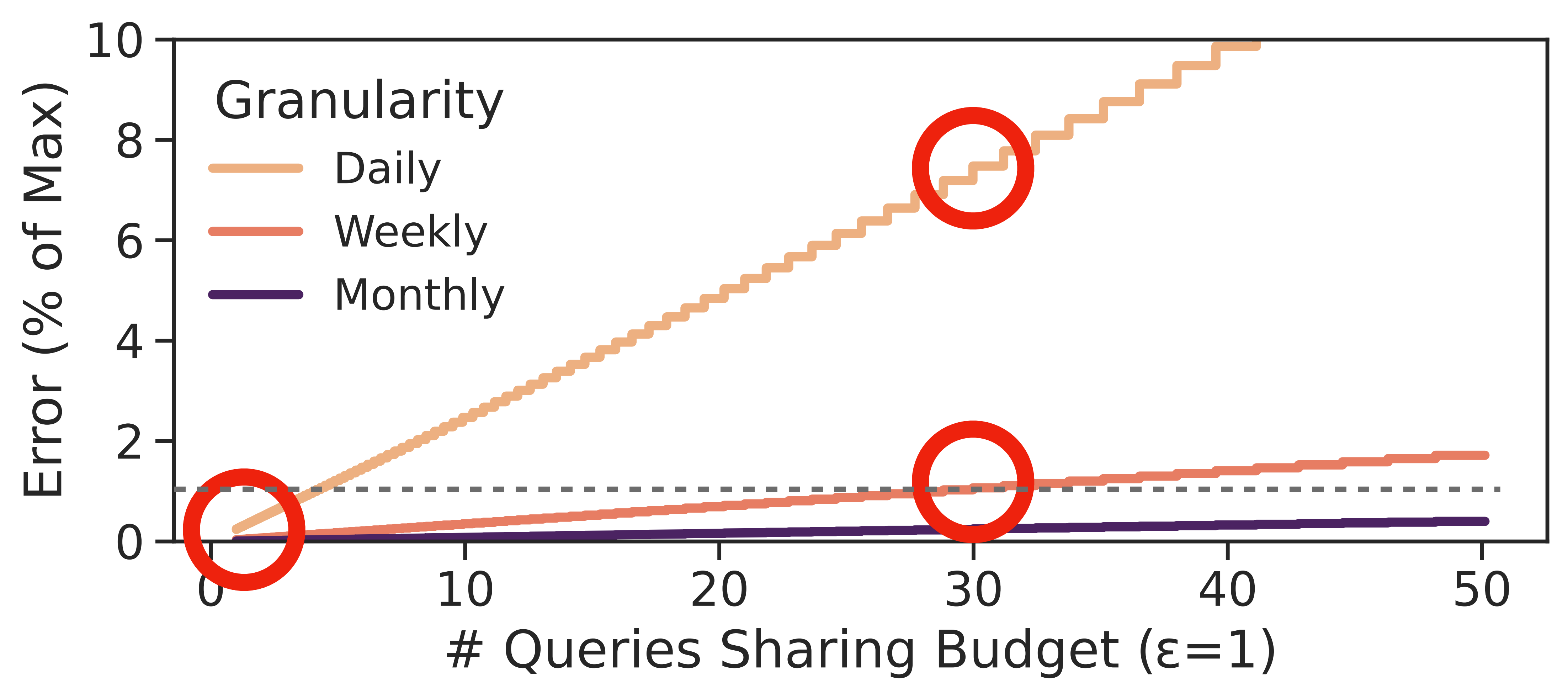
Analyst's Budget-Granularity Tradeoff

(# queries over same video) (count… per hour? per day?)

- Analyst has 2 utility knobs: query **budget** and temporal **granularity**
- Budget is *inversely proportional* to granularity and error bound
- To use less budget (i.e., execute more queries over same video): need…
  (a) coarser result OR (b) larger error bound OR (c) both



(Fixed 1% error goal)



(Fixed Query 1)

Takeaways

- Surveillance cameras could provide a lot of useful analytics, but are currently untapped due to privacy concerns
- Key technical challenge: satisfying DP over arbitrary analyst-provided NNs
- Our solution: **event-duration privacy** definition, implemented via **Privid**
- **Privid** prevents tracking, while still allowing many useful aggregation queries

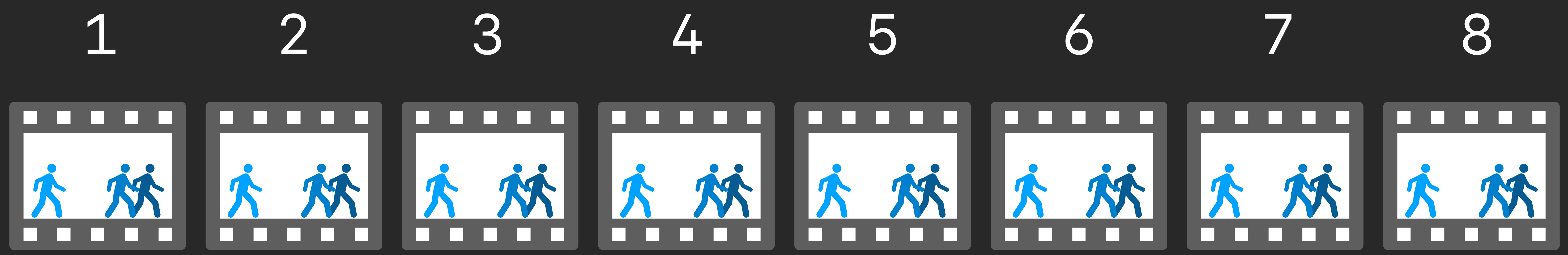
Guarantee Across Multiple Cameras

- Suppose I walk past 20 cameras on my way to work each day (250 days) and each of my appearances is bound by per-camera ρ
- Can learn:
  - was Frank **somewhere in Boston** today? (Coarse **location**, duration: 20xρ)
  - did Frank pass 5th street **this year**? (Coarse **time**, duration: 250xρ)
- **Can't learn**:
  - did Frank pass 5th street today? (Fine-grained time *and* location)

→ **Can learn some common trends, but can't track me**

Privacy Budget ϵ

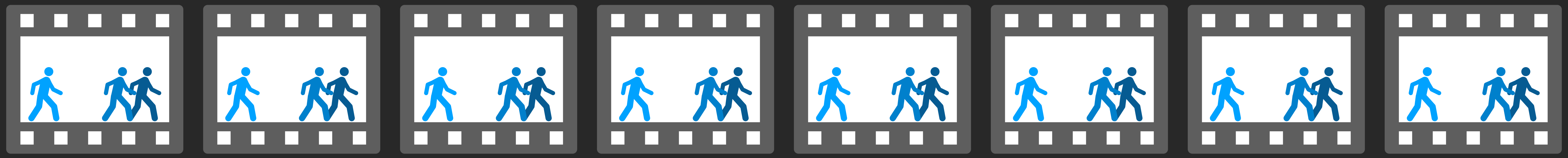
- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried



Privacy Budget ϵ

- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried

	1	2	3	4	5	6	7	8

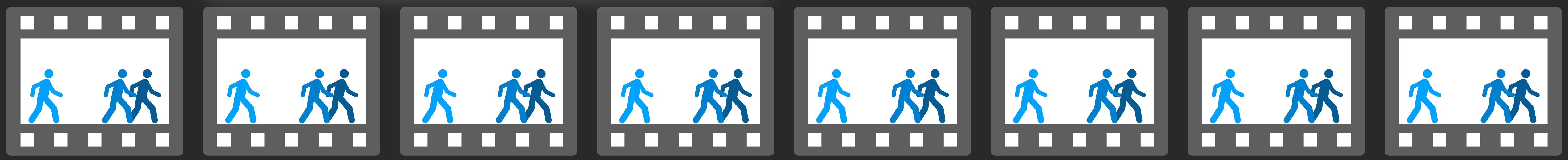


Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ

Privacy Budget ϵ

- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried

	1	2	3	4	5	6	7	8

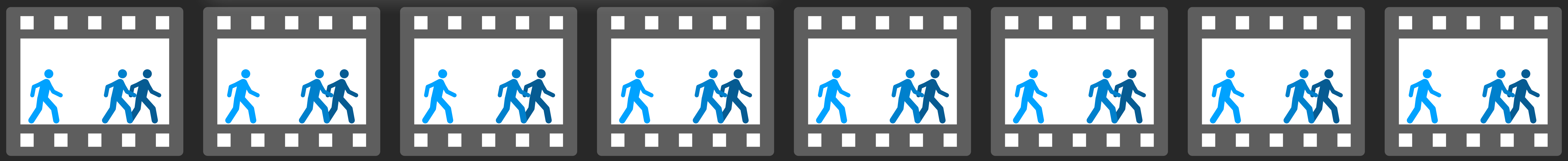


Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ

Query t=[2,4] using e/2

Privacy Budget ϵ

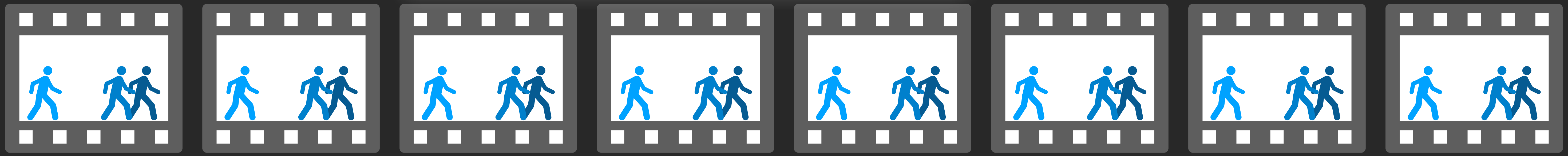
- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried



	1	2	3	4	5	6	7	8
Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
Query t=[2,4] using e/2	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	ϵ	ϵ	ϵ

Privacy Budget ϵ

- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried

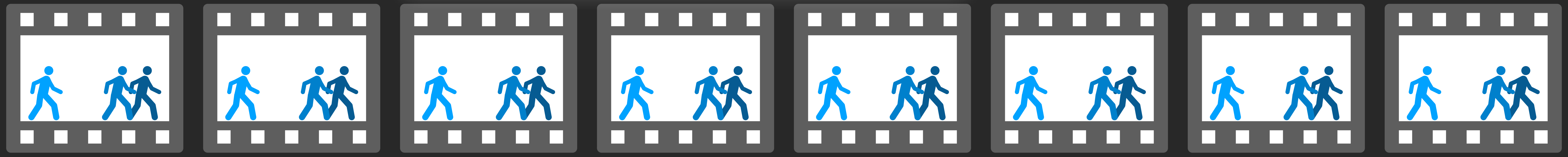


	1	2	3	4	5	6	7	8
Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
Query t=[2,4] using e/2	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	ϵ	ϵ	ϵ
Query t=[3,5] using e								

Privacy Budget ϵ

- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried

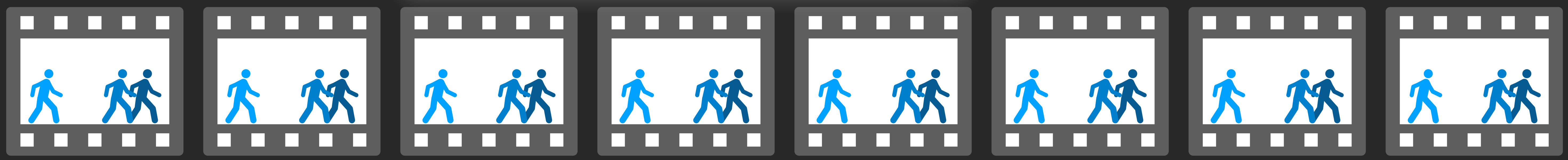
	1	2	3	4	5	6	7	8



Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
Query t=[2,4] using e/2	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	ϵ	ϵ	ϵ
Query t=[3,5] using e								

Privacy Budget ϵ

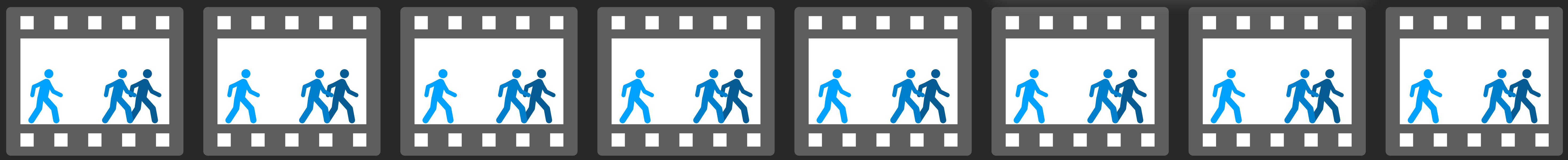
- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried



	1	2	3	4	5	6	7	8
Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
Query t=[2,4] using e/2	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	ϵ	ϵ	ϵ
Query t=[3,5] using e	**REJECTED**		insufficient budget in [3,4]					

Privacy Budget ϵ

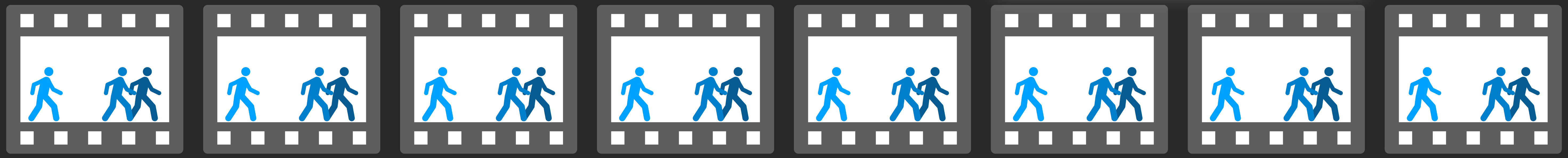
- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried



	1	2	3	4	5	6	7	8
Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
Query t=[2,4] using e/2	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	ϵ	ϵ	ϵ
Query t=[3,5] using e	REJECTED			insufficient budget in [3,4]				
Query t=[6,7] using e								

Privacy Budget ϵ

- Analyst picks privacy budget for each camera (standard DP parameter ϵ), budget is distributed *per-frame*, and is public knowledge
- Each Privid query consumes budget from all frames in query window
- Once budget is depleted for a frame, it can no longer be queried



	1	2	3	4	5	6	7	8
Initial Budget	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
Query t=[2,4] using e/2	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	ϵ	ϵ	ϵ
Query t=[3,5] using e	**REJECTED**			insufficient budget in [3,4]				
Query t=[6,7] using e	ϵ	$\epsilon/2$	$\epsilon/2$	$\epsilon/2$	ϵ	0	0	ϵ