Mining online data for public health surveillance

Vasileios Lampos *(a.k.a. Bill)*

Computer Science
University College London

@lampos

Structure

- ➡ Using online data for health applications
- ➡ From web searches to syndromic surveillance
  1. **Google Flu Trends**: original failure and correction
  2. **Better feature selection** using semantic concepts
  3. *Snapshot*: **Multi-task learning** for disease models

Online data

Online data for health (1/3)

When & why?

How?

Evaluation?

Online data for health (1/3)

When & why?

— coverage
— speed
— cost

How?

Evaluation?

Online data for health (1/3)

When & why?

— coverage
— speed
— cost

How?

— collaborate with experts
— access to user activity data
— machine learning
— natural language processing

Evaluation?

Online data for health (1/3)

When & why?

- coverage
- speed
- cost

How?

- collaborate with experts
- access to user activity data
- machine learning
- natural language processing

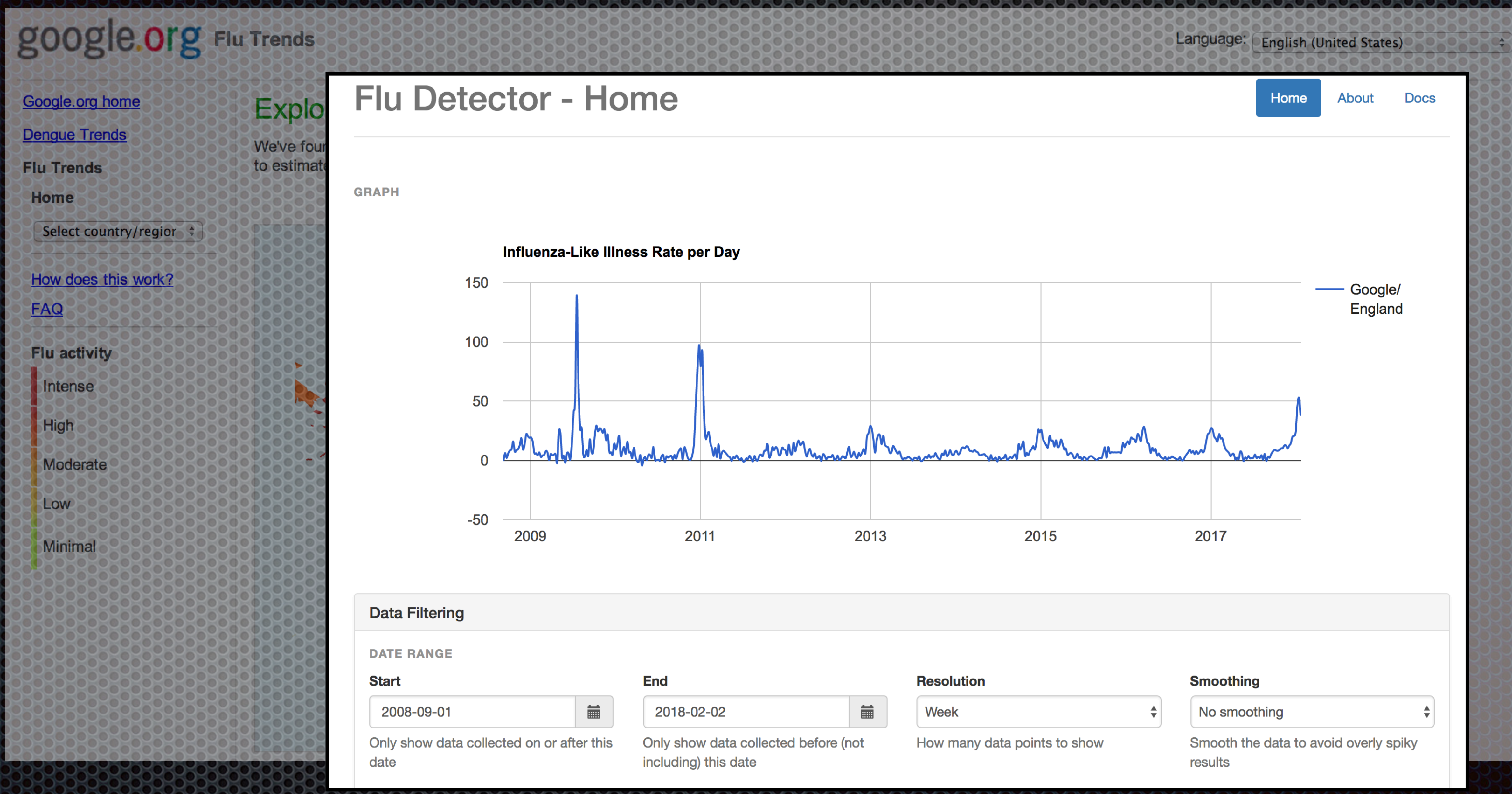
Evaluation?

- (partial) ground truth
- model interpretation
- real-time

Online data for health (2/3)

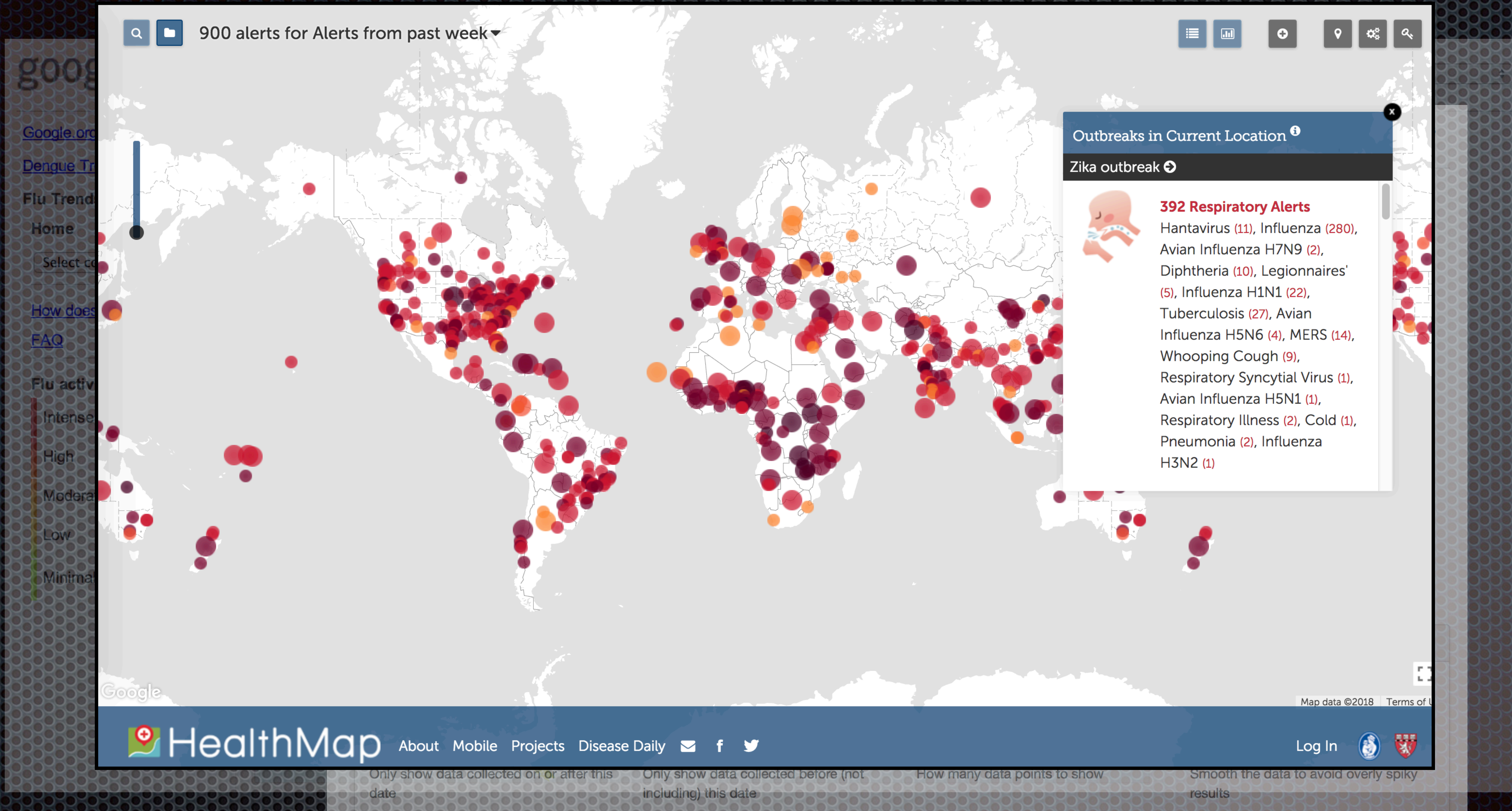
Google Flu Trends (*discontinued*)

Online data for health (2/3)



Flu Detector, fludetector.cs.ucl.ac.uk

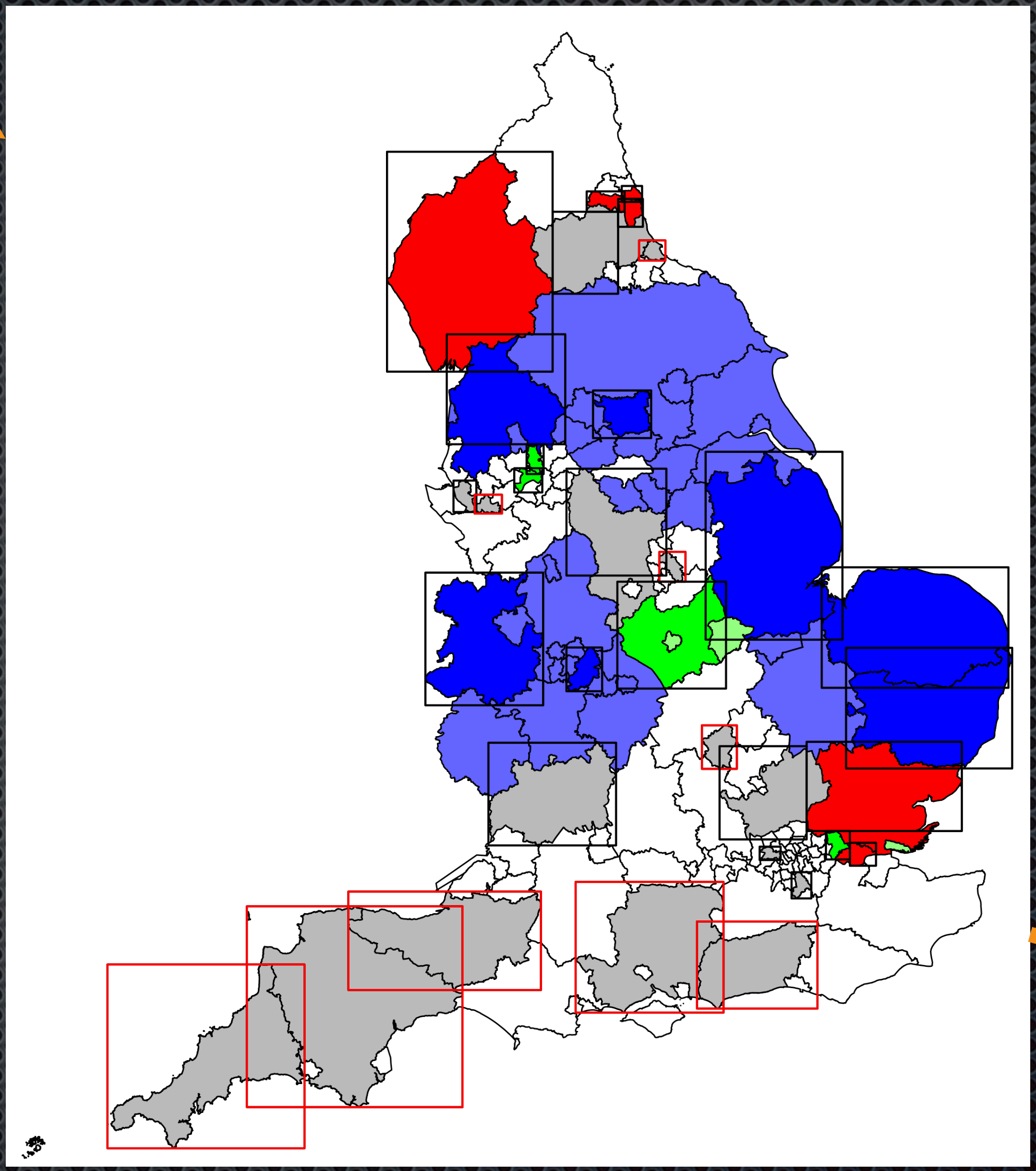
Online data for health (2/3)



Health Map, healthmap.org

Online data for health (3/3)

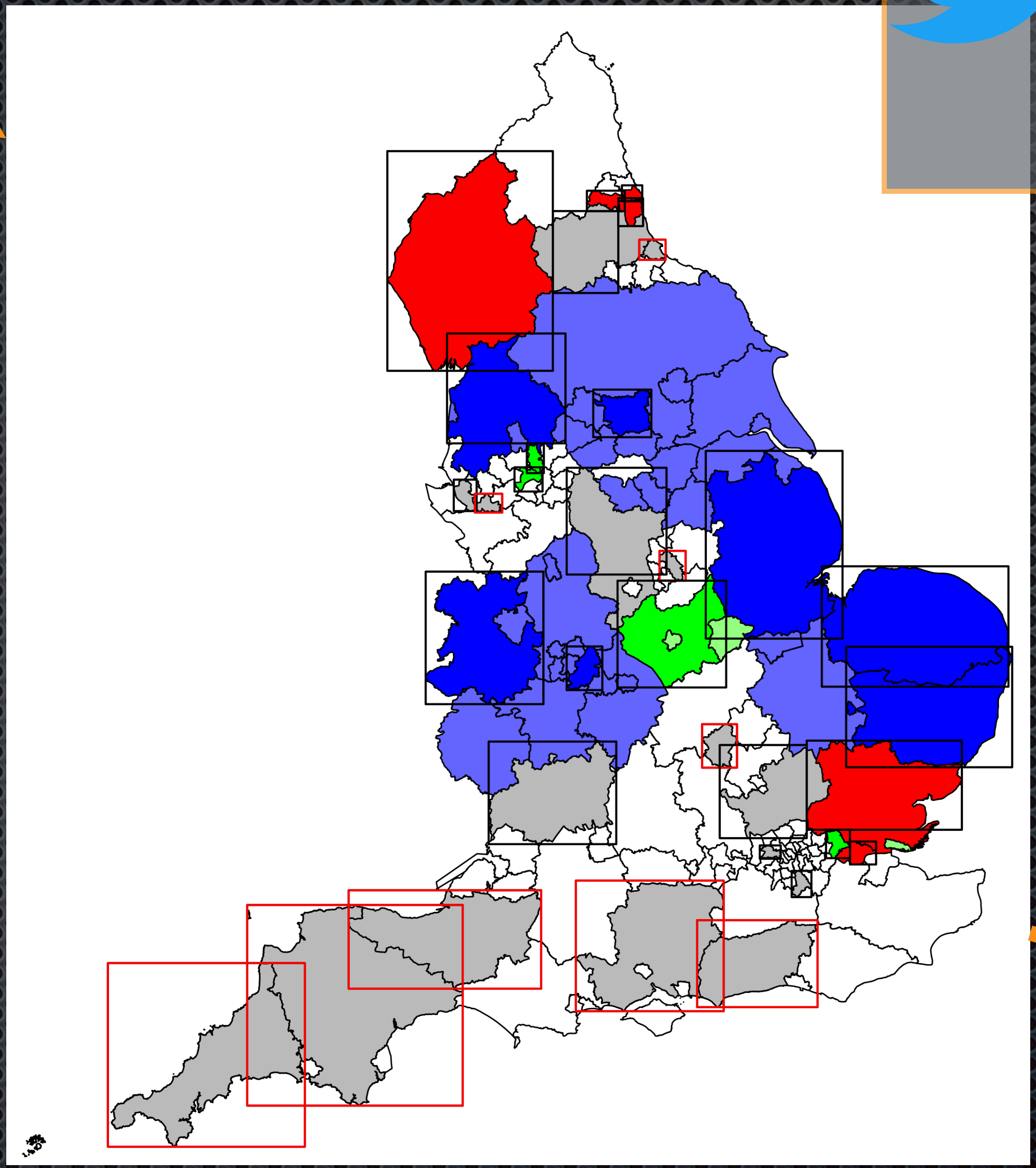
Health intervention



Impact?

Online data for health (3/3)

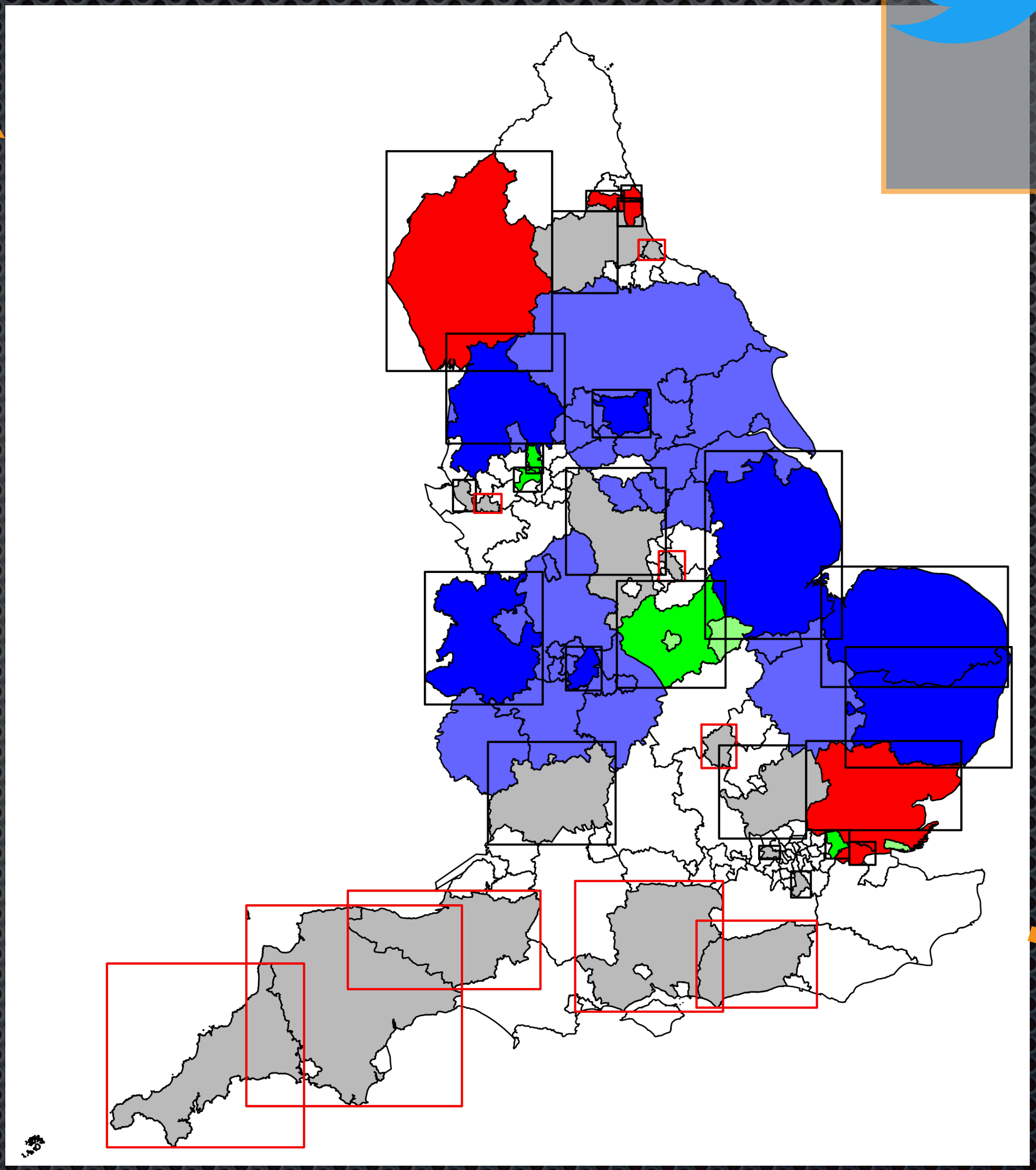
Health intervention



Impact?

Online data for health (3/3)

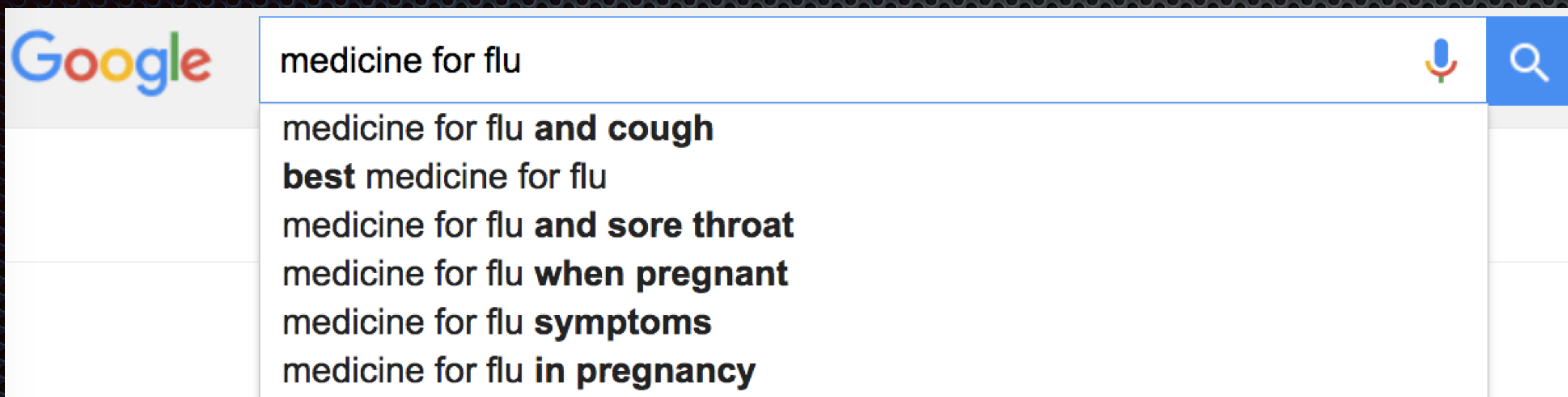
Vaccinations against flu

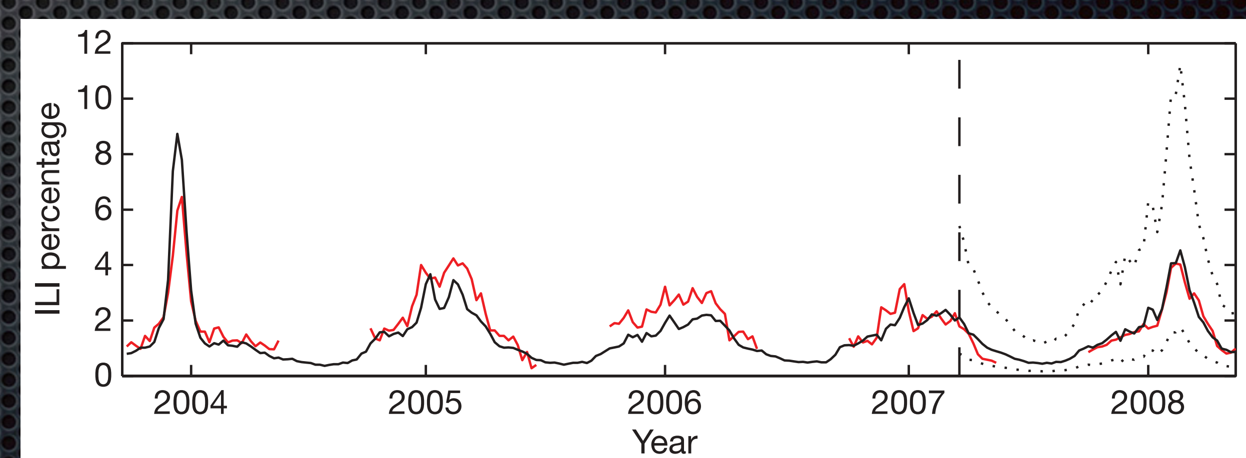


Impact?

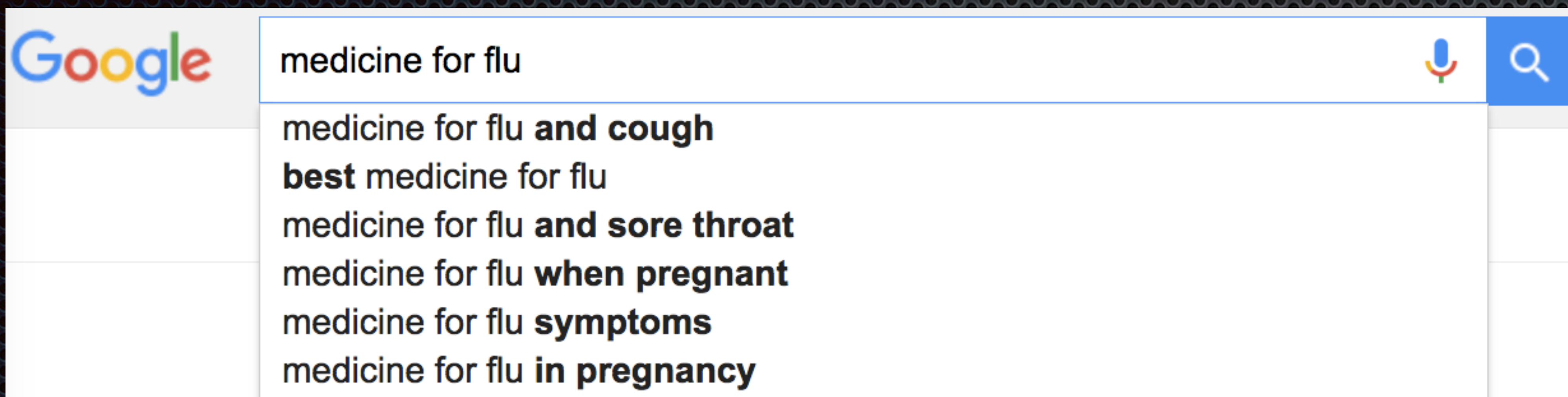
Lampos, Yom-Tov, Pebody, Cox (2015) doi:10.1007/s10618-015-0427-9
Wagner, Lampos, Yom-Tov, Pebody, Cox (2017) doi:10.2196/jmir.8184

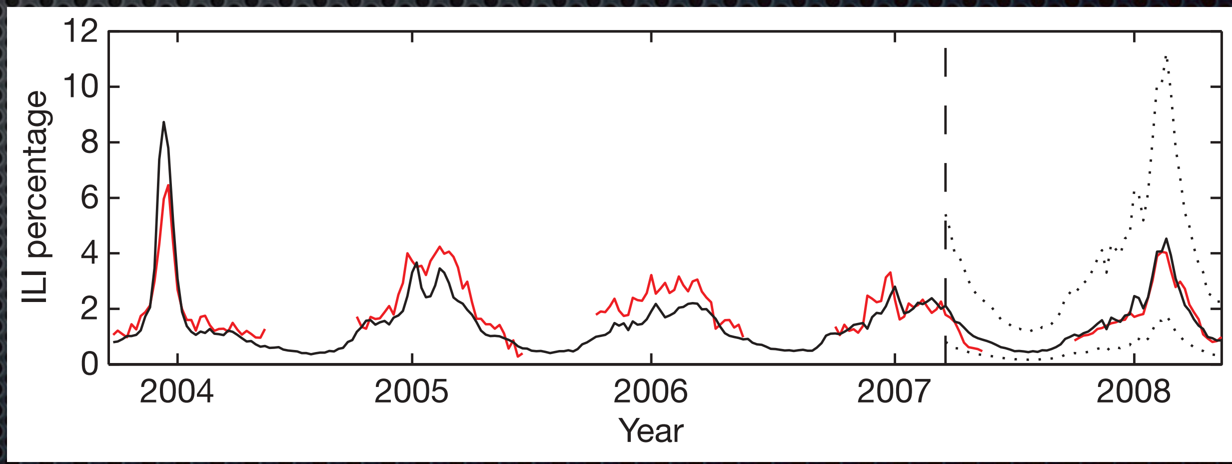
Google Flu Trends (GFT)





Google Flu Trends (GFT)





GFT — Supervised learning

- **Regression**
- Observations (**X**): **Frequencies** of *n* search queries for a location *L* and *m* contiguous time intervals of length τ
- Targets (**y**): **Rates of influenza-like illness (ILI)** for *L* and for the same *m* contiguous time intervals, obtained from a health agency
- Learn a function ***f*** such that ***f*: X** $\in \mathbb{R}^{n \times m} \longrightarrow$ **y** $\in \mathbb{R}^{n}$

GFT — Supervised learning

$$\text{frequency of } q_i = \frac{\text{count of } q_i}{\text{total count of all queries}}$$

- **Regression**
- Observations (**X**): **Frequencies** of *n* search queries for a location *L* and *m* contiguous time intervals of length τ
- Targets (**y**): **Rates of influenza-like illness (ILI)** for *L* and for the same *m* contiguous time intervals, obtained from a health agency
- Learn a function ***f*** such that ***f*: X** $\in \mathbb{R}^{n \times m} \longrightarrow$ **y** $\in \mathbb{R}^{n}$

GFT v.1 — Model

$$logit(P) = \beta_0 + \beta_1 \times logit(Q) + \varepsilon$$

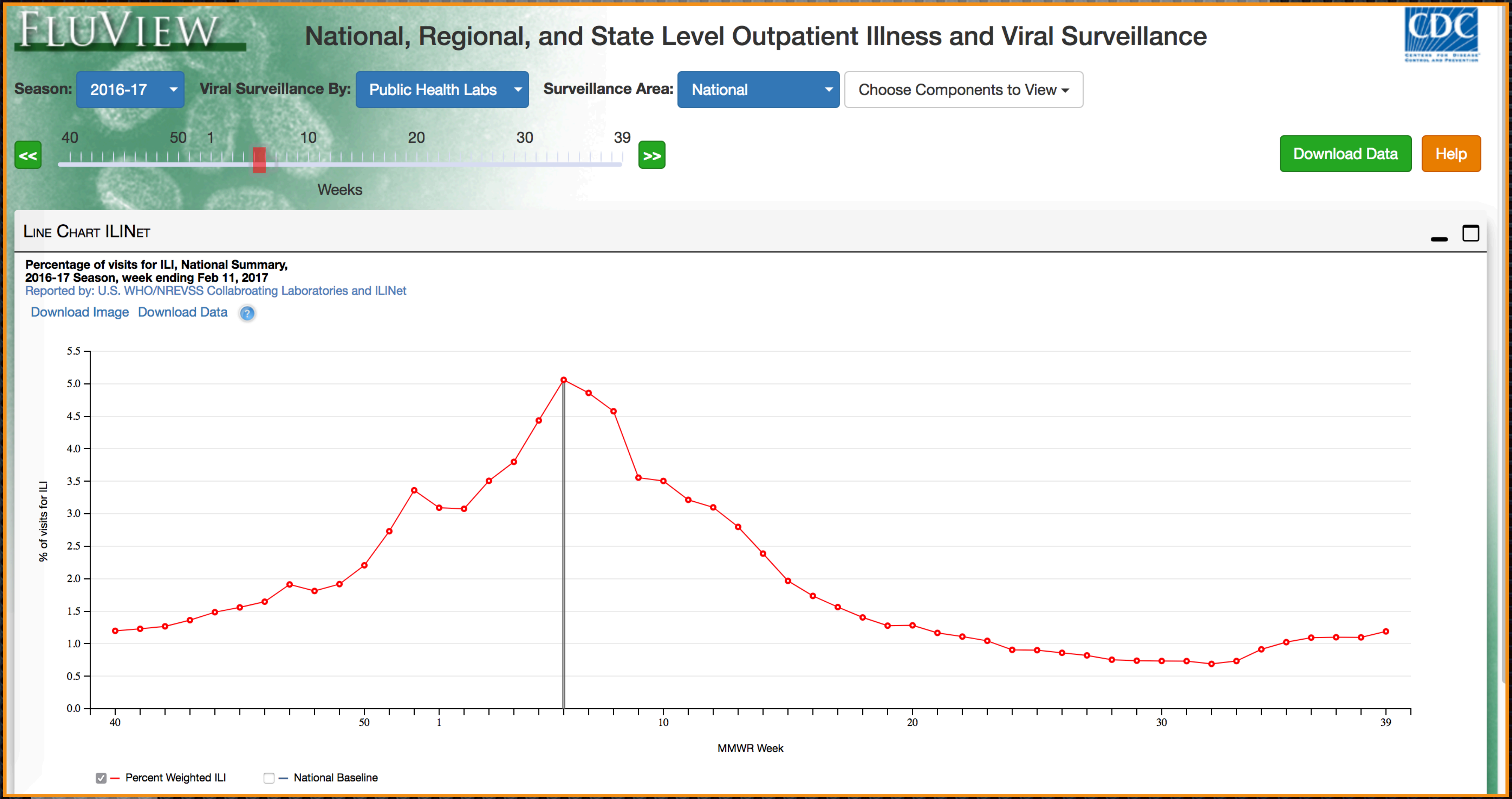
Q	Aggregate frequency of a set of search queries
P	Percentage (probability) of doctor visits
β_0	Regression bias term
β_1	Regression weight (one weight only)
ε	independent, zero-centered noise

GFT v.1 — Model

$$logit(P) = \beta_0 + \beta_1 \times logit(Q) + \varepsilon$$

Q	Aggregate frequency of a set of search queries
P	**Percentage (probability) of doctor visits**
β_0	Regression bias term
β_1	Regression weight (one weight only)
ε	independent, zero-centered noise

GFT v.1 — Model



ε independent, zero-centered noise

GFT v.1 — Model

$$logit(P) = \beta_0 + \beta_1 \times logit(Q) + \varepsilon$$

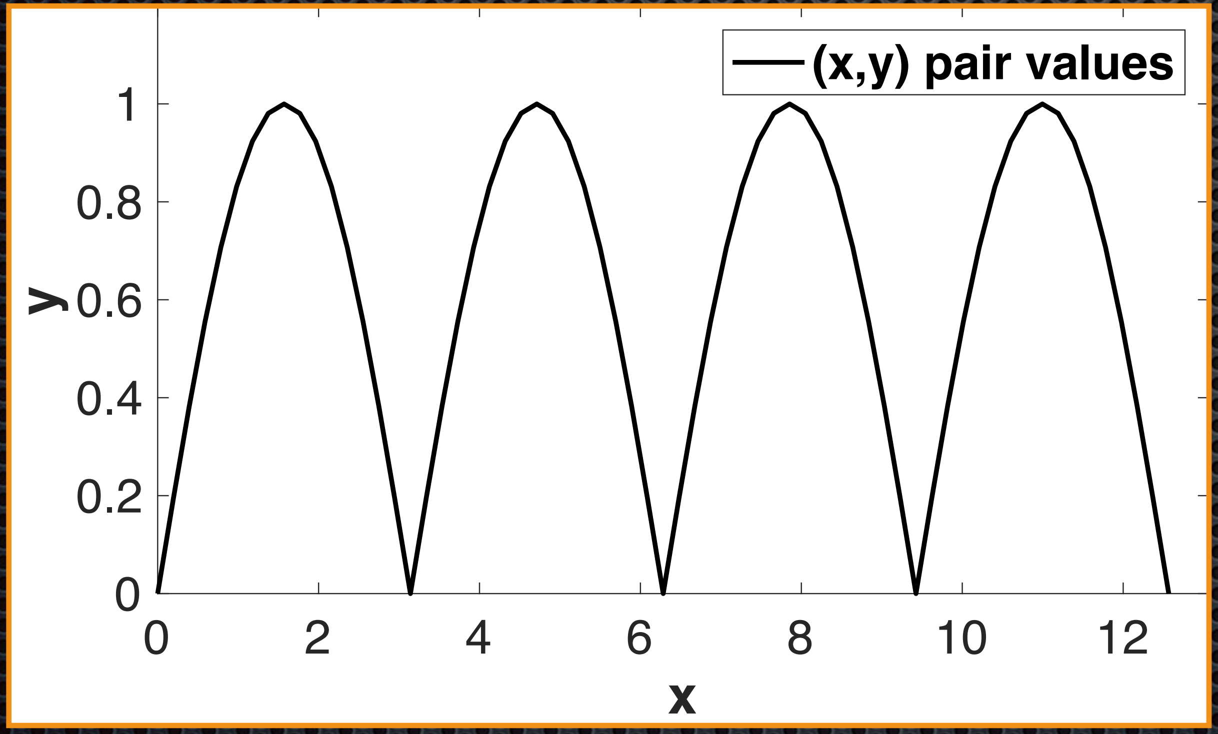
Q	Aggregate frequency of a set of search queries
P	Percentage (probability) of doctor visits
β_0	Regression bias term
β_1	**Regression weight (one weight only)**
ε	independent, zero-centered noise

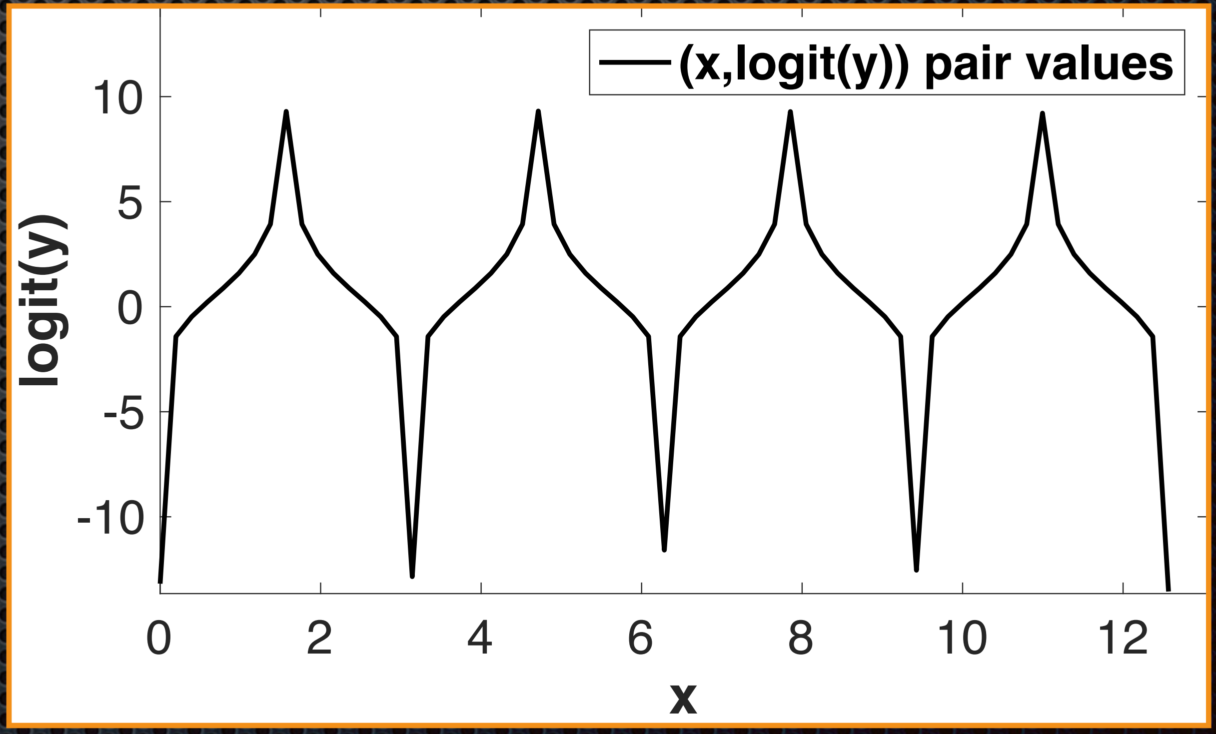
GFT v.1 — "Logit", why?

the ***logit*** function

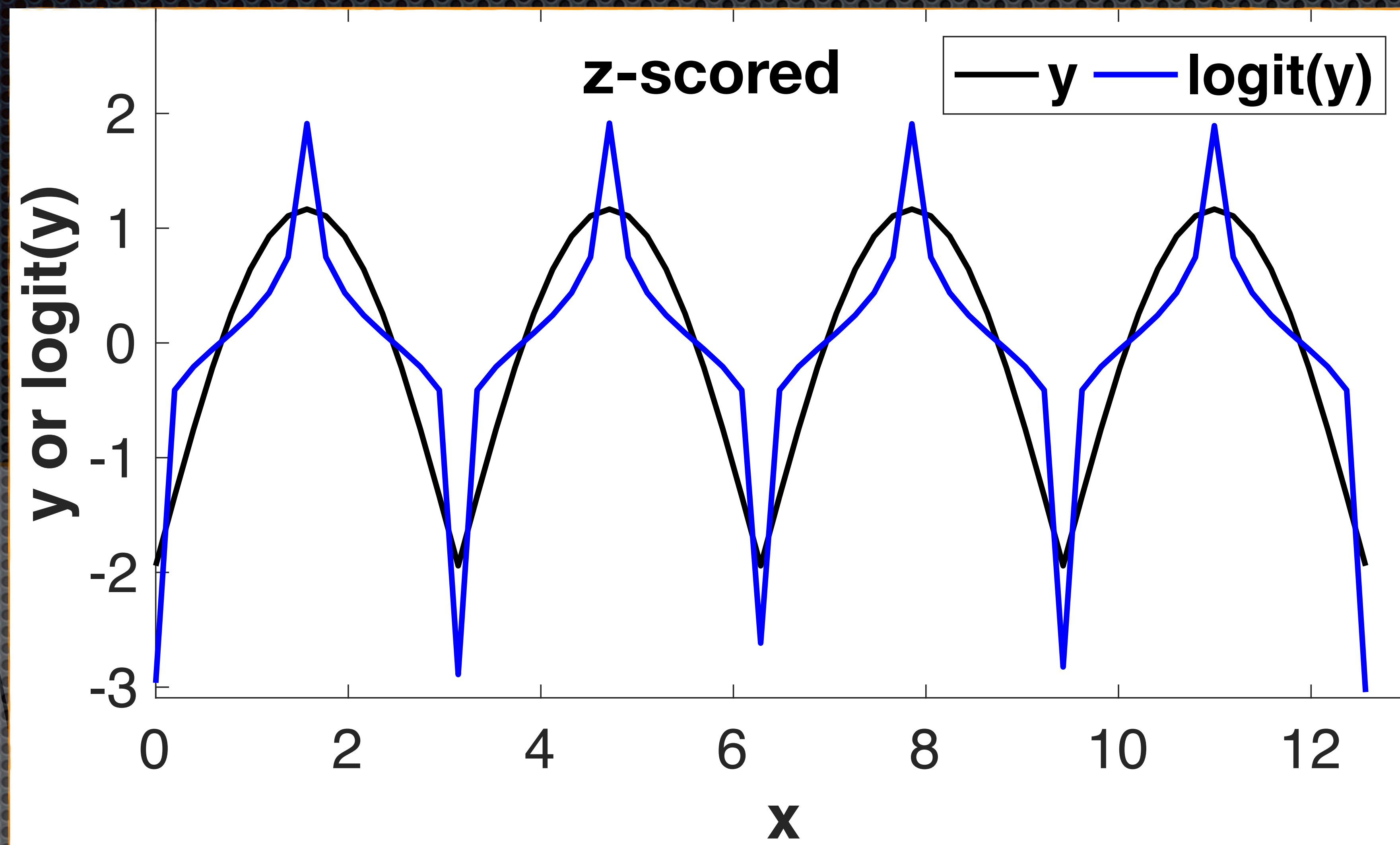
$$\mathrm{logit}(\alpha) = \log(\alpha/(1-\alpha))$$

$\alpha \in (0,1)$





GFT v.1 — “Logit”, why?



values close to 0.5 are “**squashed**”
border values (close to 0 or 1) are “**emphasised**”

GFT v.1 — Data

9 US regions considered

50 million search queries (*most frequent*) geolocated in these 9 US regions

Weekly **ILI rates from CDC**

170 weeks, 28/9/2003 to 11/5/2008 with ILI rate > 0

First 128 weeks: Training, 9 x 128 = **1,152 samples**

Last 42 weeks: Testing (per region)

GFT v.1 — Feature selection (1/2)

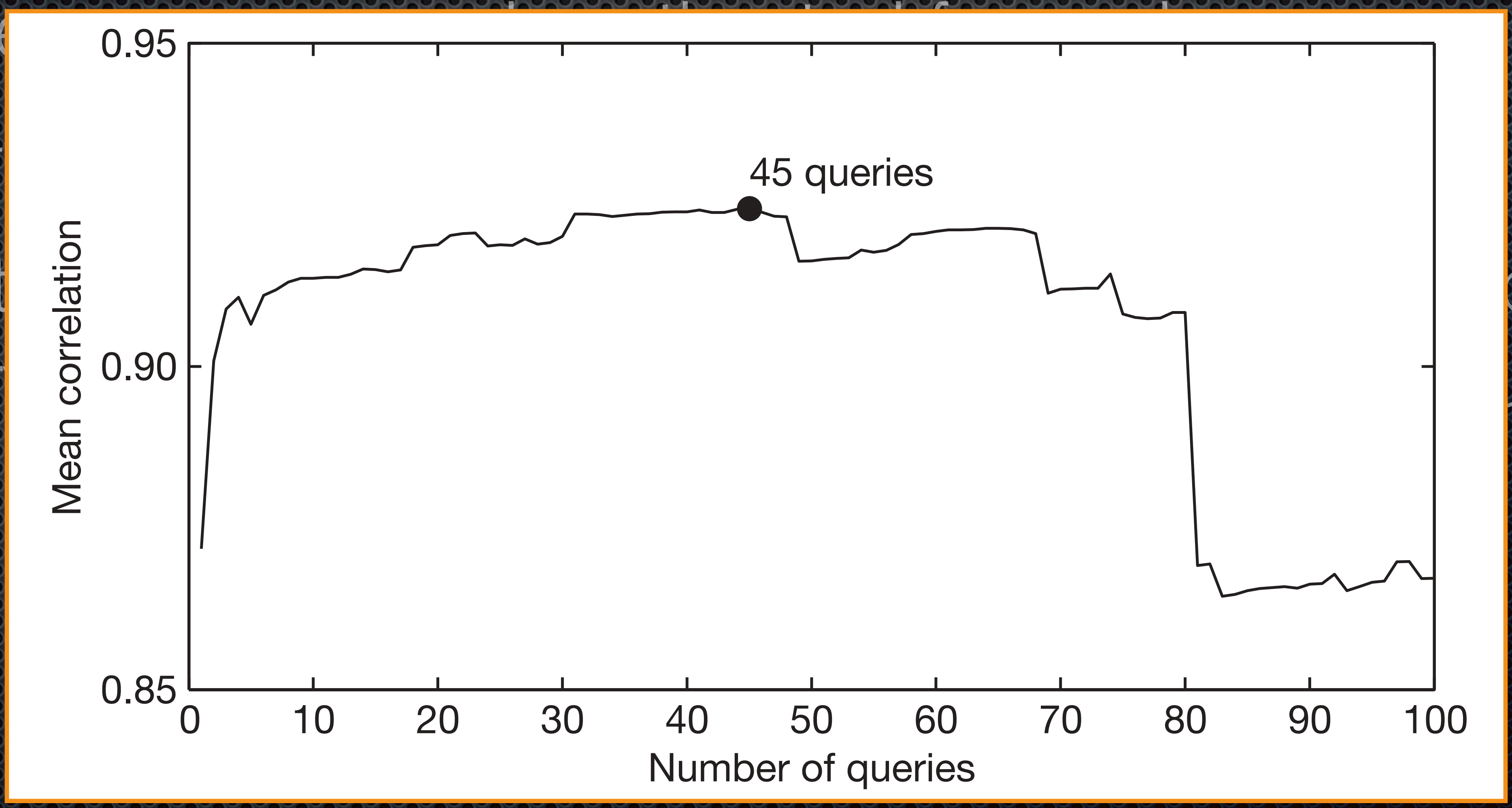
1. Single query flu models are trained for each US region
   50 million queries x 9 US regions = 450 million models
2. Inference accuracy is estimated for each query using linear correlation (Pearson) as the metric
3. Starting from the best performing query, adding up one query each time, a new model is trained and evaluated

GFT v.1 — Feature selection (1/2)

1. Single query flu models are trained for each US region
   50 million queries x 9 US regions = 450 million models

2. Infere … ing
   linea …

3. Start … o one
   quer … uated



GFT v.1 — Feature selection (2/2)

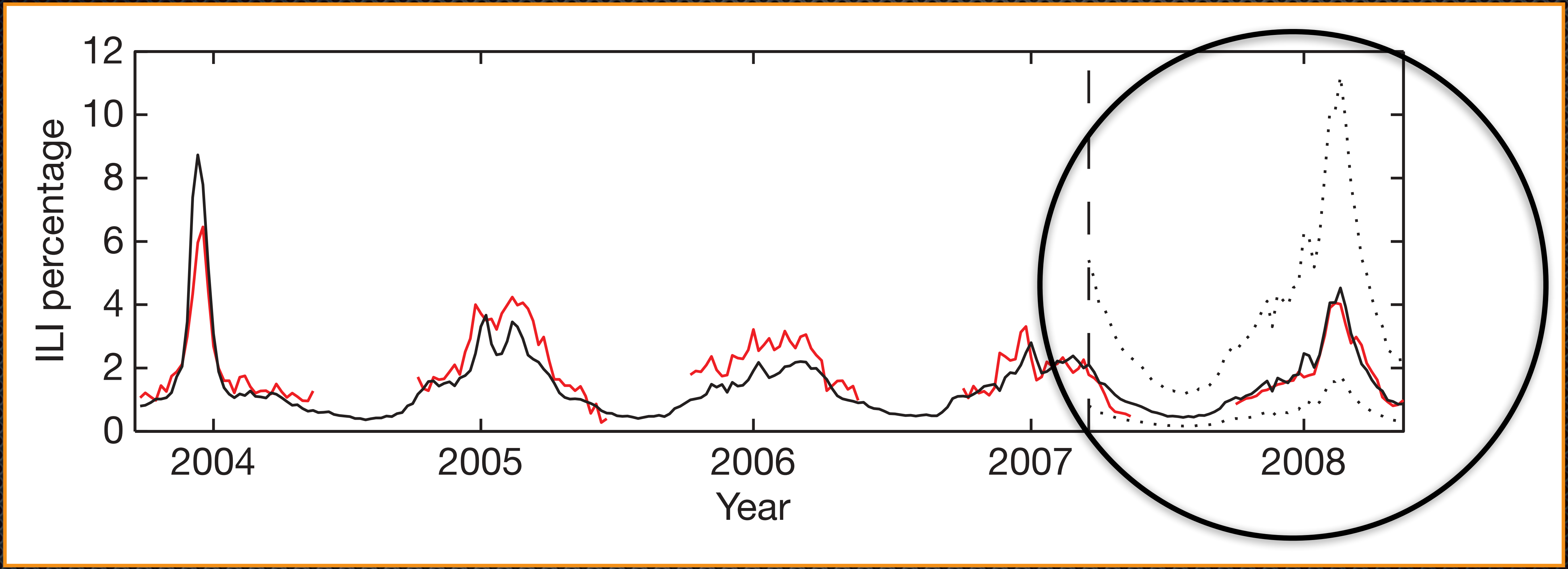
Search query topic	Top 45 queries	
	n	Weighted
Influenza complication	11	18.15
Cold/flu remedy	8	5.05
General influenza symptoms	5	2.60
Term for influenza	4	3.74
Specific influenza symptom	4	2.54
Symptoms of an influenza complication	4	2.21
Antibiotic medication	3	6.23
General influenza remedies	2	0.18
Symptoms of a related disease	2	1.66
Antiviral medication	1	0.39
Related disease	1	6.66
Unrelated to influenza	0	0.00
Total	45	49.40

GFT v.1 — Performance (1/2)

- Evaluated on 42 weeks (per region) from 2007-2008
- Evaluation metric: Pearson correlation
- **$\mu(r)$ = .97** with **min(r) = .92** and **max(r) = .99**
- Performance looked great at the time, ***but this is not a proper performance evaluation!***

  Why?
  Potentially ***misleading metric*** (not the loss function here) and rather ***small testing time span*** (< 1 flu season)

GFT v.1 — Performance (2/2)



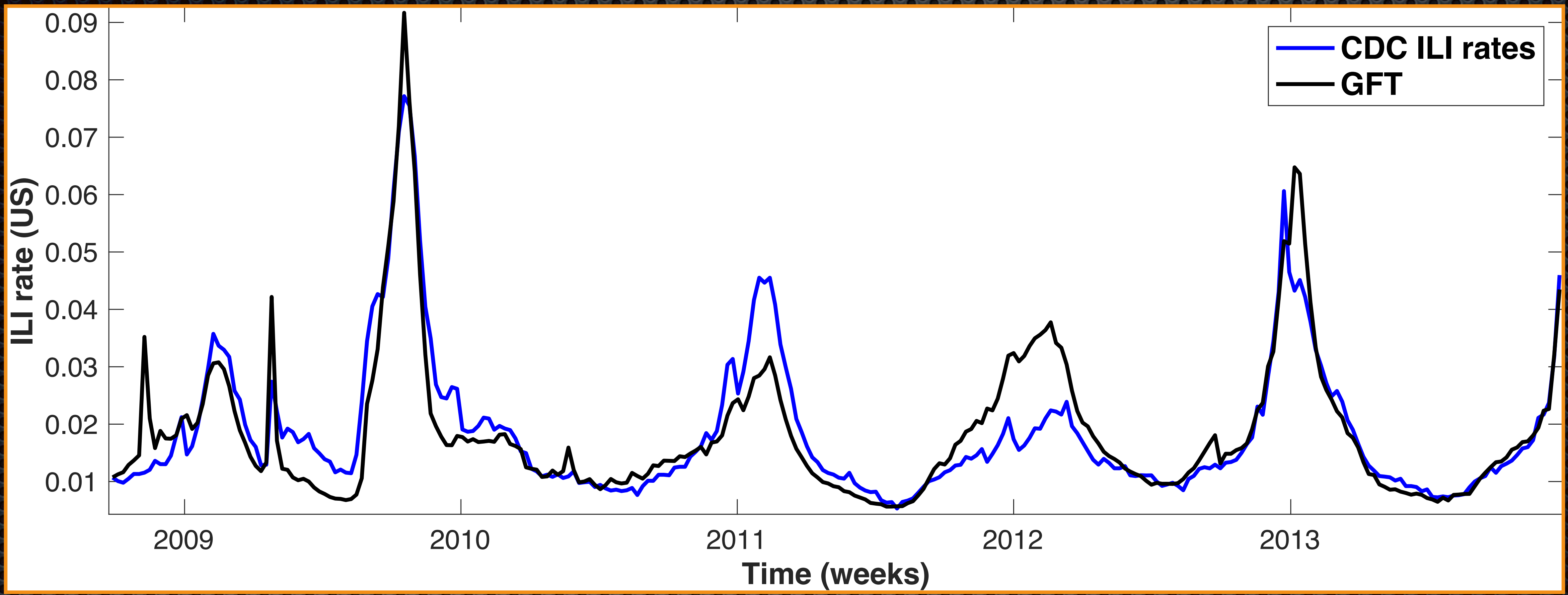
Mid-Atlantic US region

Pearson correlation, ***r* = .96**

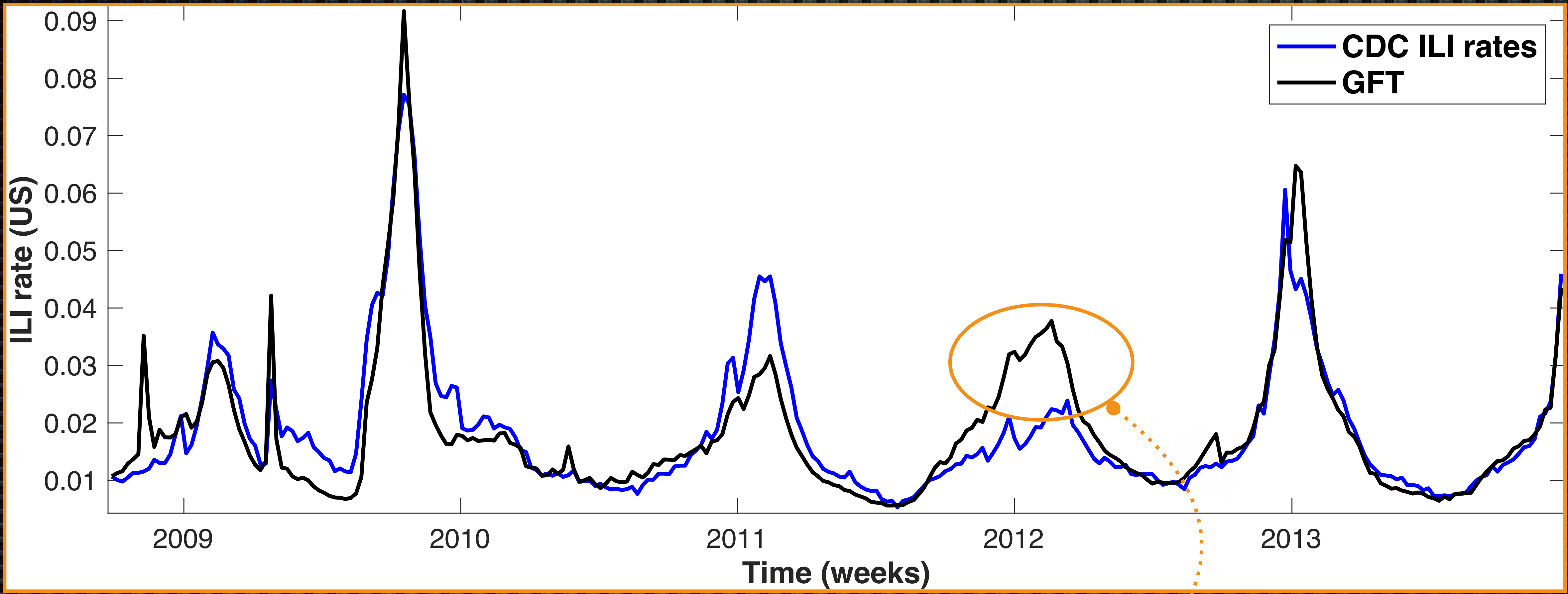
GFT v.2 — Data & evaluation

- weekly frequency of **49,708** search queries (US)
- filtered by a relaxed health topic classifier, intersection of frequent queries across all US regions
- from 4/1/**2004** to 28/12/**2013** (**521 weeks**)
- corresponding weekly US ILI rates from CDC
- **test on 5 flu seasons**, 5 year-long test sets (2008-13)
- **train on increasing data sets starting from 2004**, using all data prior to a test period

GFT v.1 was simple to a (*significant*) fault



GFT v.1 was simple to a (*significant*) fault



"rsv" — 25%
"flu symptoms" — 18%
"benzonatate" — 6%
"symptoms of pneumonia" — 6%
"upper respiratory infection" — 4%

GFT v.2 — Linear multivariate regression

$$\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 \right)$$

Least squares

GFT v.2 — Linear multivariate regression

$$\boxed{\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 \right)}$$

Least squares

$\mathbf{X} \in \mathbb{R}^{n \times m}$ frequency of *m* search queries for *n* weeks

$\mathbf{x}_i \in \mathbb{R}^m,\ i \in \{1, \ldots, n\}$ … for week *i*

GFT v.2 — Linear multivariate regression

$$\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 \right)$$

Least squares

$\mathbf{X} \in \mathbb{R}^{n \times m}$ frequency of *m* search queries for *n* weeks

$\mathbf{x}_i \in \mathbb{R}^m,\ i \in \{1, \ldots, n\}$ ... for week *i*

$\mathbf{y} \in \mathbb{R}^n$ ILI rates from CDC for *n* weeks

$y_i \in \mathbb{R}$ ... for week *i*

GFT v.2 — Linear multivariate regression

$$\boxed{\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 \right)}$$

Least squares

$\mathbf{X} \in \mathbb{R}^{n \times m}$ frequency of *m* search queries for *n* weeks

$\mathbf{x}_i \in \mathbb{R}^m,\ i \in \{1, \ldots, n\}$ ... for week *i*

$\mathbf{y} \in \mathbb{R}^n$ ILI rates from CDC for *n* weeks

$y_i \in \mathbb{R}$ ... for week *i*

$\mathbf{w} \in \mathbb{R}^m$ weights for the *m* search queries

$\beta \in \mathbb{R}$ intercept term

GFT v.2 — Linear multivariate regression

Least squares regression is **not** applicable here because we have **very few training samples** (*n*) but **many features** (search queries; *m*).

Models derived from least squares will tend to overfit the data, resulting to bad solutions.

$\mathbf{w} \in \mathbb{R}^m$ weights for the *m* search queries

$\beta \in \mathbb{R}$ intercept term

GFT v.2 — Regularisation with elastic net

$$\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 + \lambda_1 \sum_{j=1}^{m} |w_j| + \lambda_2 \sum_{j=1}^{m} w_j^2 \right)$$

GFT v.2 — Regularisation with elastic net

$$\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\underbrace{\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2}_{\text{least squares}} + \lambda_1 \sum_{j=1}^{m} |w_j| + \lambda_2 \sum_{j=1}^{m} w_j^2 \right)$$

GFT v.2 — Regularisation with elastic net

$$\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 + \lambda_1 \sum_{j=1}^{m} |w_j| + \lambda_2 \sum_{j=1}^{m} w_j^2 \right)$$

least squares

$$\boxed{\begin{array}{l} \lambda_1 \in \mathbb{R}^+ \\ \lambda_2 \in \mathbb{R}^+ \end{array}}$$

L1 & L2-norm **regularisers** for the weights

GFT v.2 — Regularisation with elastic net

$$\boxed{\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\underbrace{\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2}_{\text{least squares}} + \lambda_1 \sum_{j=1}^{m} |w_j| + \lambda_2 \sum_{j=1}^{m} w_j^2 \right)}$$

least squares

$$\boxed{\begin{array}{l} \lambda_1 \in \mathbb{R}^+ \\ \lambda_2 \in \mathbb{R}^+ \end{array}}$$

L1 & L2-norm **regularisers** for the weights

- Encourages sparse models (**feature selection**)
- Handles **collinear** features (search queries)
- Number of selected features is not limited to the number of samples (*n*)

GFT v.2 — Regularisation with elastic net

$$\underset{\mathbf{w},\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{w} + \beta - y_i)^2 + \lambda_1 \sum_{j=1}^{m} |w_j| + \lambda_2 \sum_{j=1}^{m} w_j^2 \right)$$

least squares

many weights will be set to zero!

$$\boxed{\begin{array}{l} \lambda_1 \in \mathbb{R}^+ \\ \lambda_2 \in \mathbb{R}^+ \end{array}}$$

L1 & L2-norm **regularisers** fo

- Encourages sparse models (**feature selection**)
- Handles **collinear** features (search queries)
- Number of selected features is not limited to the number of samples (*n*)

GFT v.2 — Feature selection

- 1st layer: Keep search queries that their frequency time series has a **≥ 0.5 Pearson correlation** with the CDC ILI rates (*in the training data*)
- 2nd layer: **Elastic net** will assign **weights equal to 0** to features (search queries) that are identified as statistically irrelevant to our task

μ (σ) # queries selected across all training data sets

# queries	r ≥ 0.5	GFT	Elastic net
49,708	937 (334)	46 (39)	278 (64)

GFT v.2 — Evaluation (1/2)

Target variable: $y = y_1, \ldots, y_N$

Estimates: $\hat{y} = \hat{y_1}, \ldots, \hat{y_N}$

Mean **S**quared **E**rror:

$$\mathrm{MSE}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \frac{1}{N} \sum_{t=1}^{N} \left(\hat{y}_t - y_t\right)^2$$

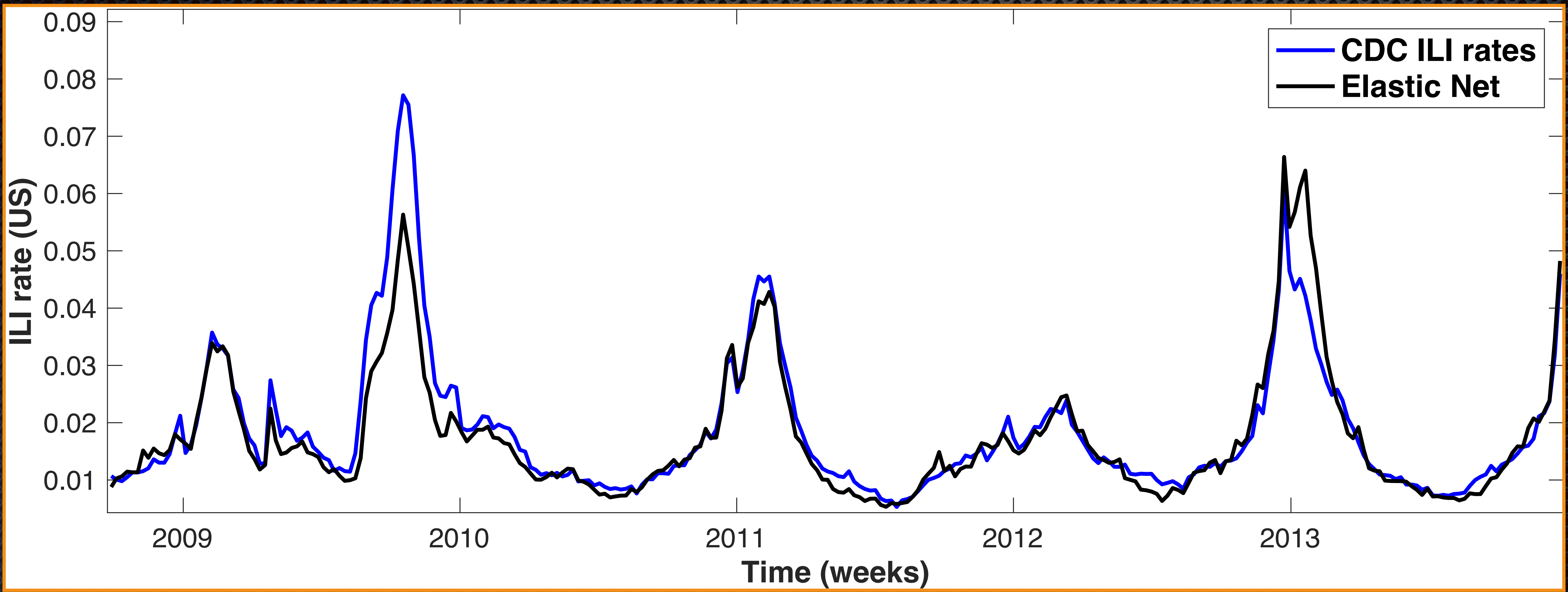
Mean **A**bsolute **E**rror:

$$\mathrm{MAE}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \frac{1}{N} \sum_{t=1}^{N} \left|\hat{y}_t - y_t\right|$$

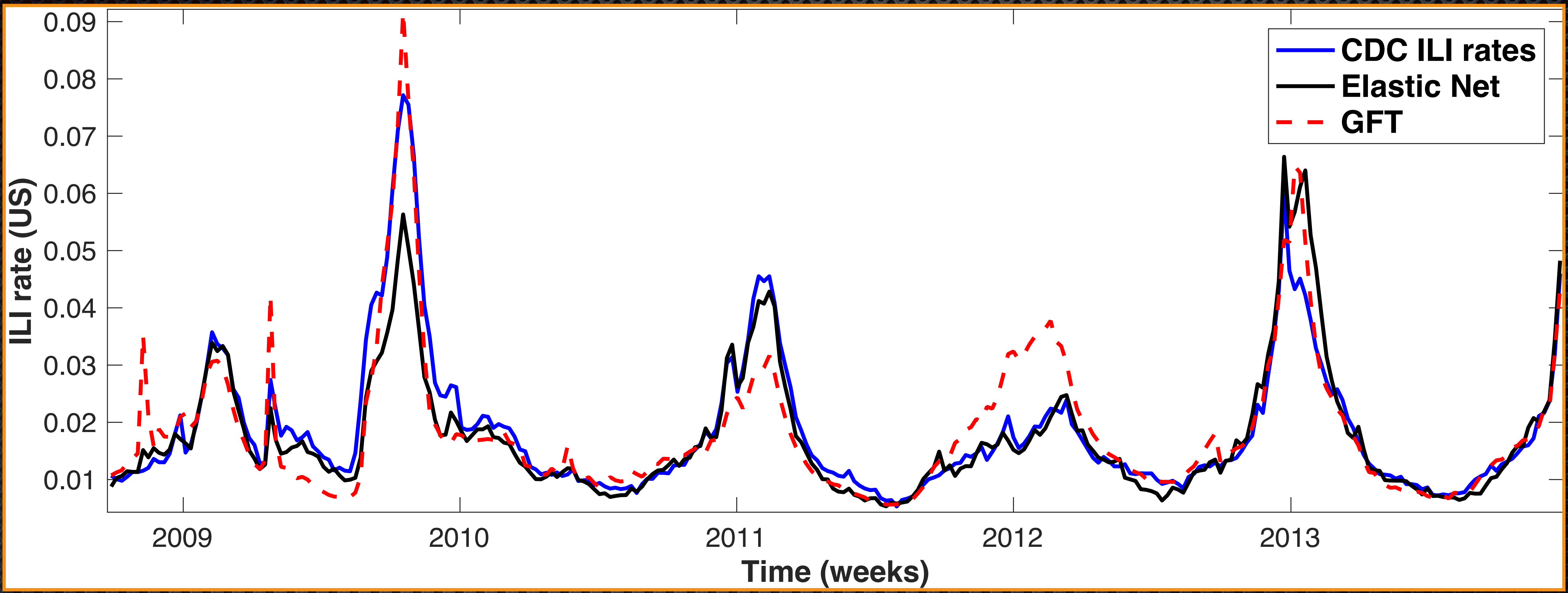
Mean **A**bsolute **P**ercentage of **E**rror:

$$\mathrm{MAPE}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \frac{1}{N} \sum_{t=1}^{N} \left|\frac{\hat{y}_t - y_t}{y_t}\right|$$

GFT v.2 — Evaluation (2/2)



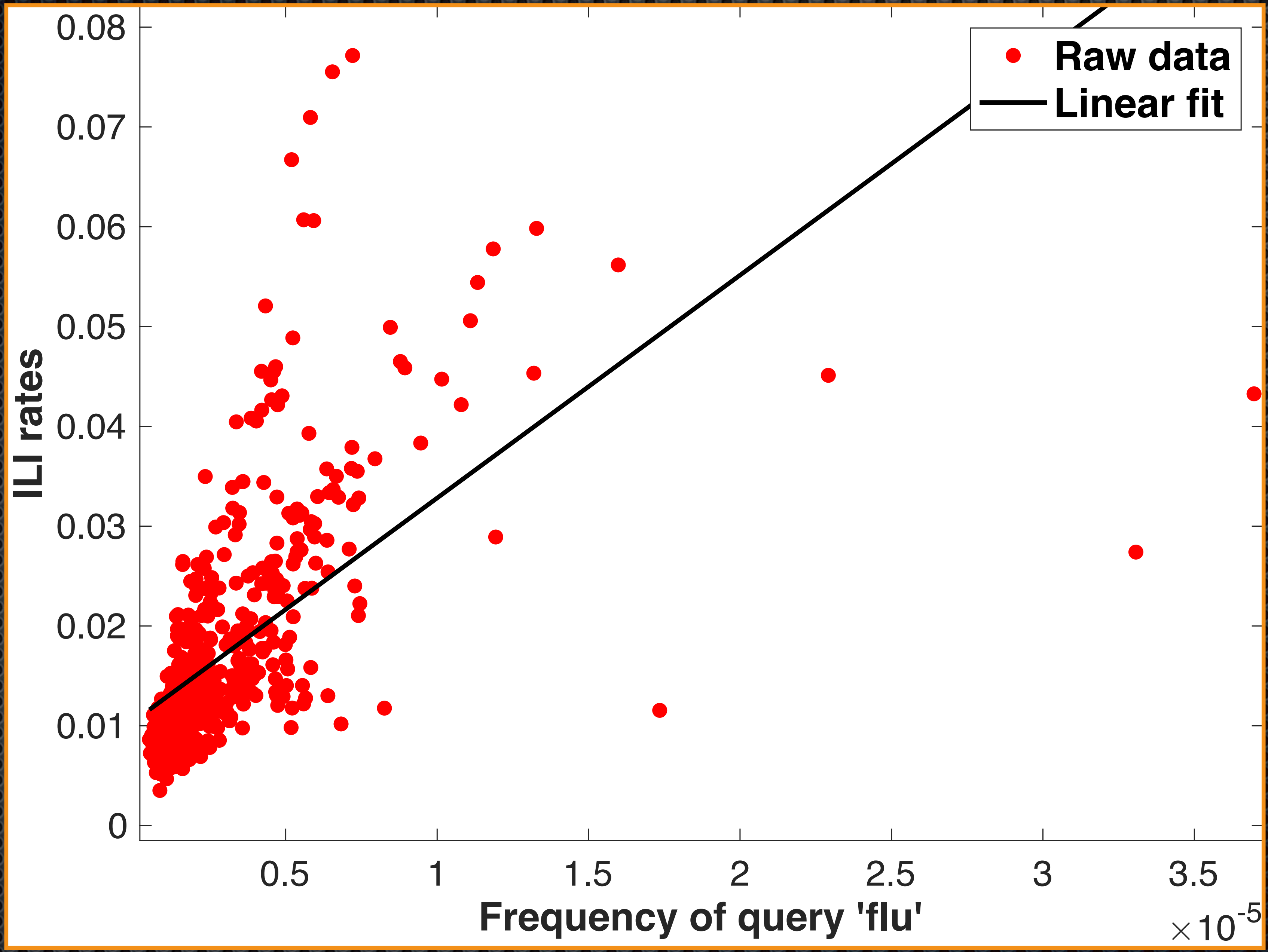
GFT v.2 — Evaluation (2/2)



GFT r = .89, MAE = $3.81 \cdot 10^{-3}$, MAPE = 20.4%

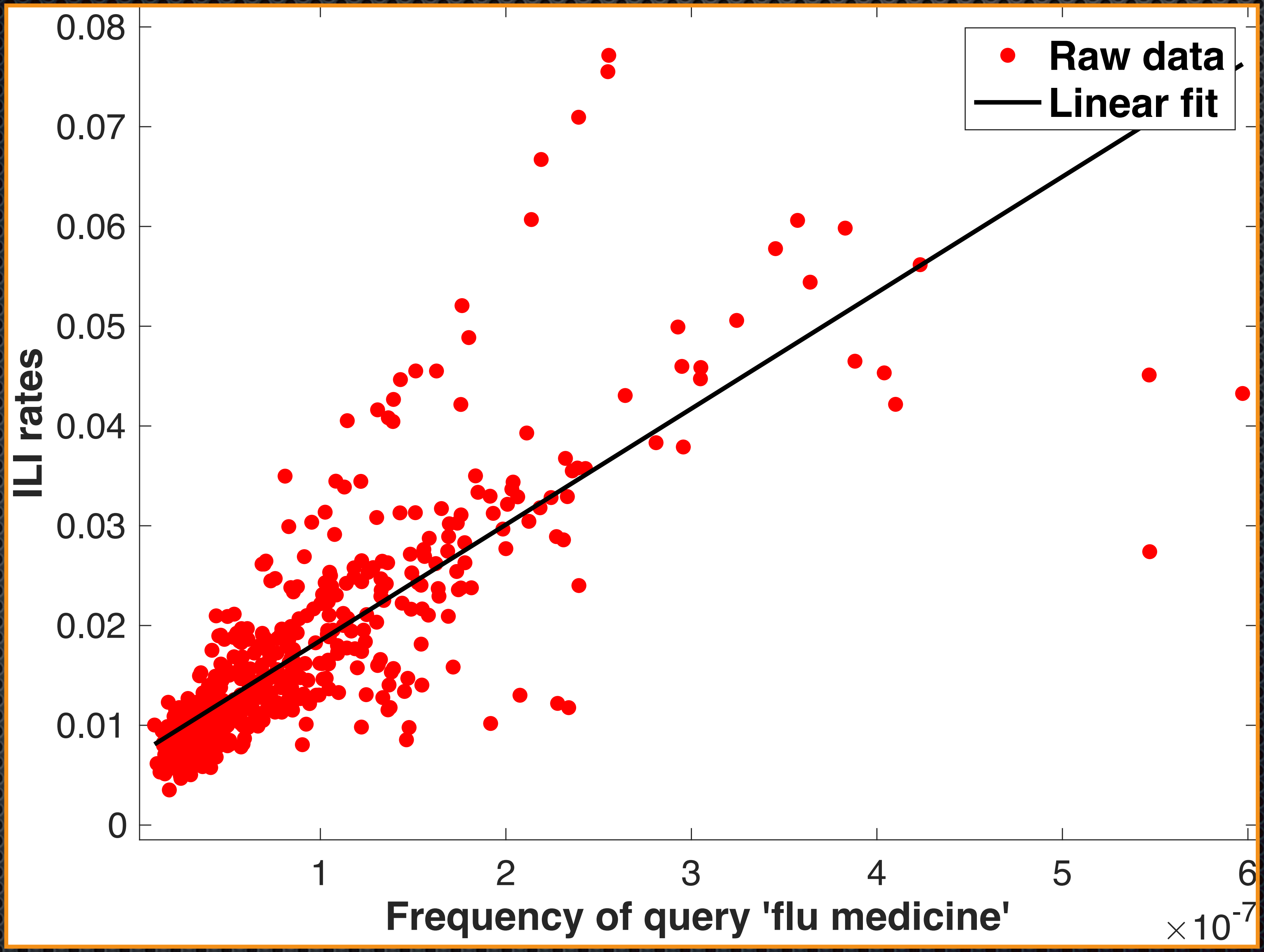
Elastic net r = .92, MAE = $2.60 \cdot 10^{-3}$, MAPE = 11.9%

GFT v.2 — Nonlinearities in the data



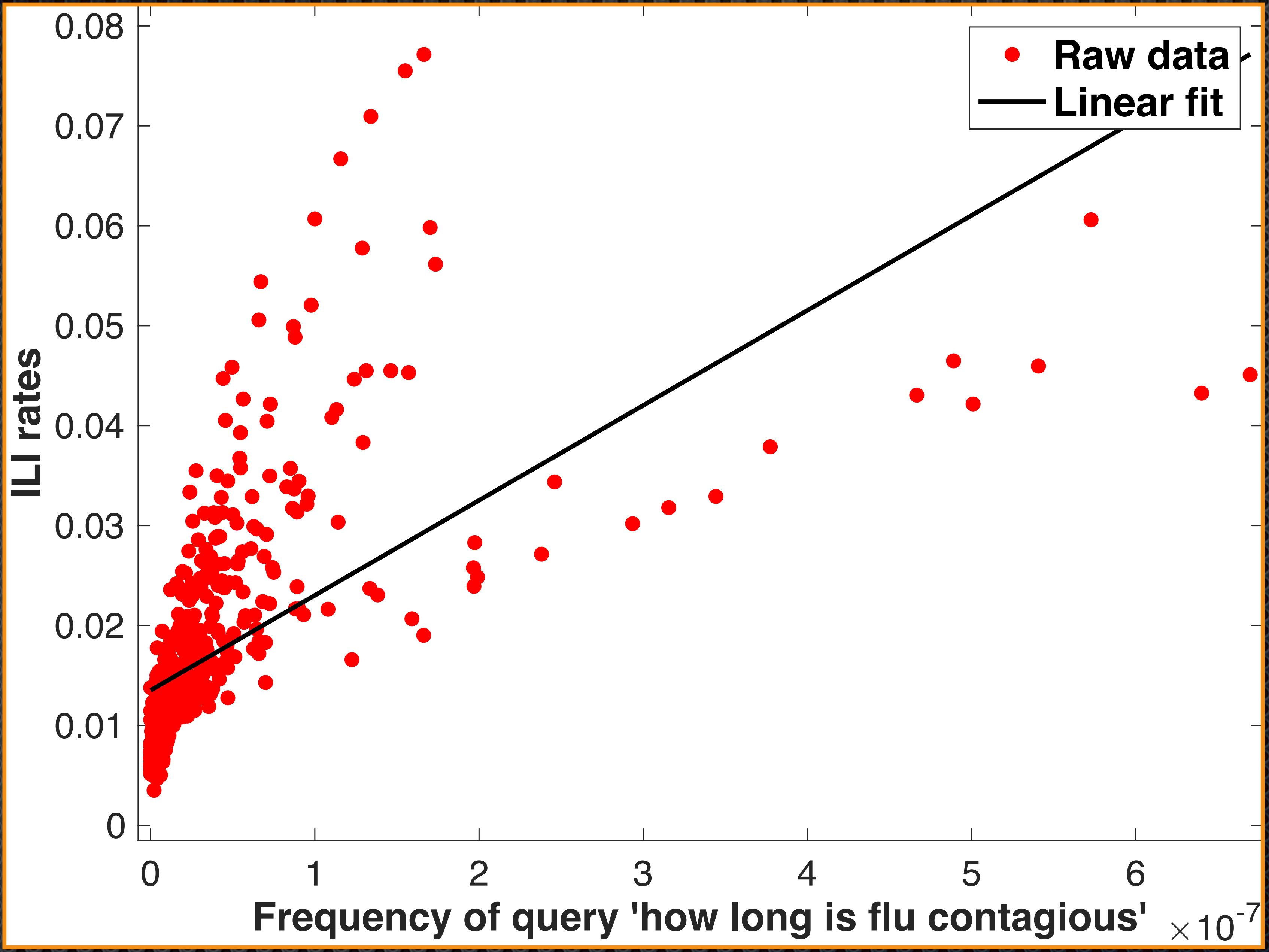
US ILI rates (CDC) ~ freq. of query ‘*flu*’

GFT v.2 — Nonlinearities in the data



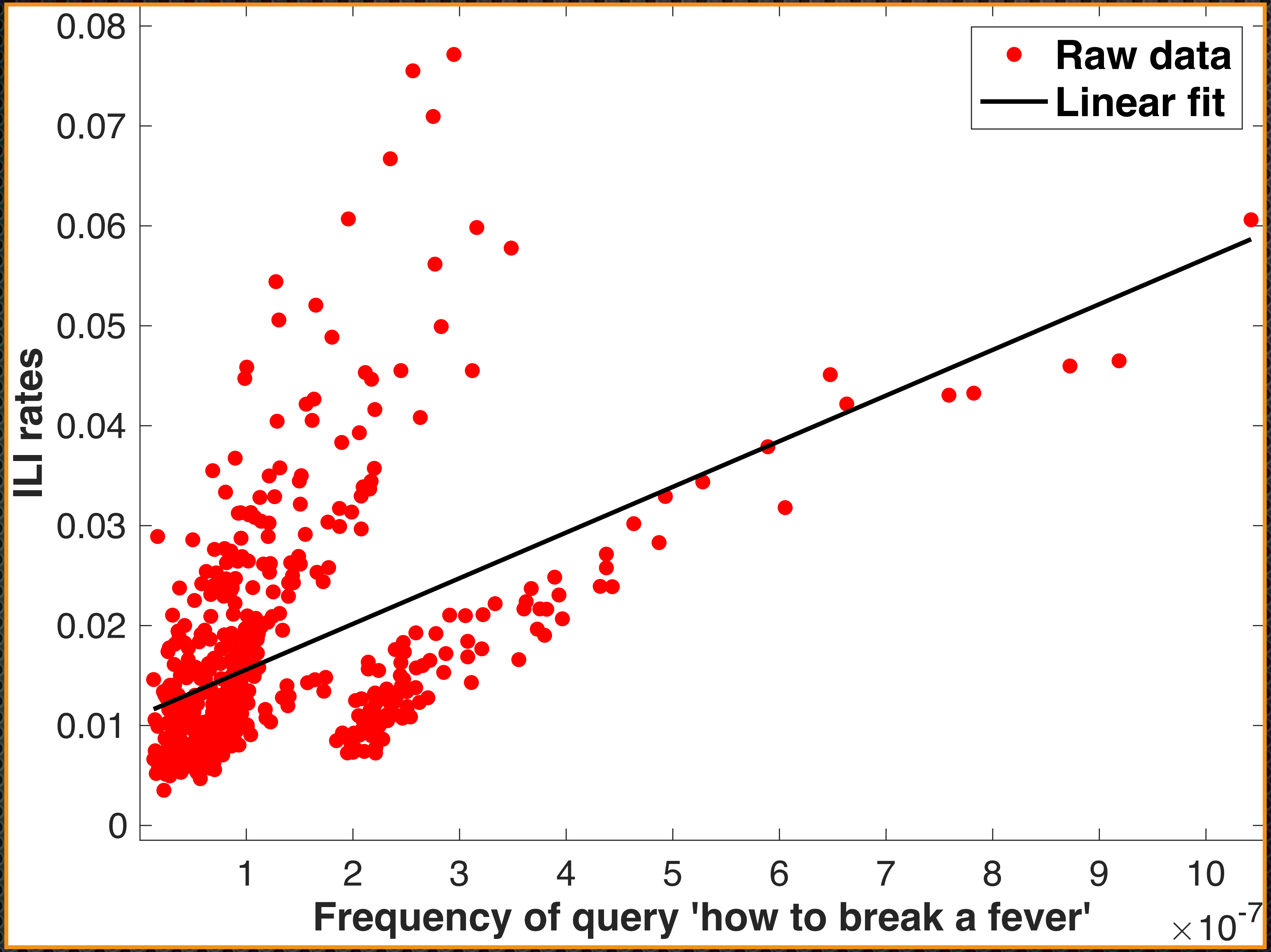
US ILI rates (CDC) ~ freq. of query ‘*flu medicine*’

GFT v.2 — Nonlinearities in the data



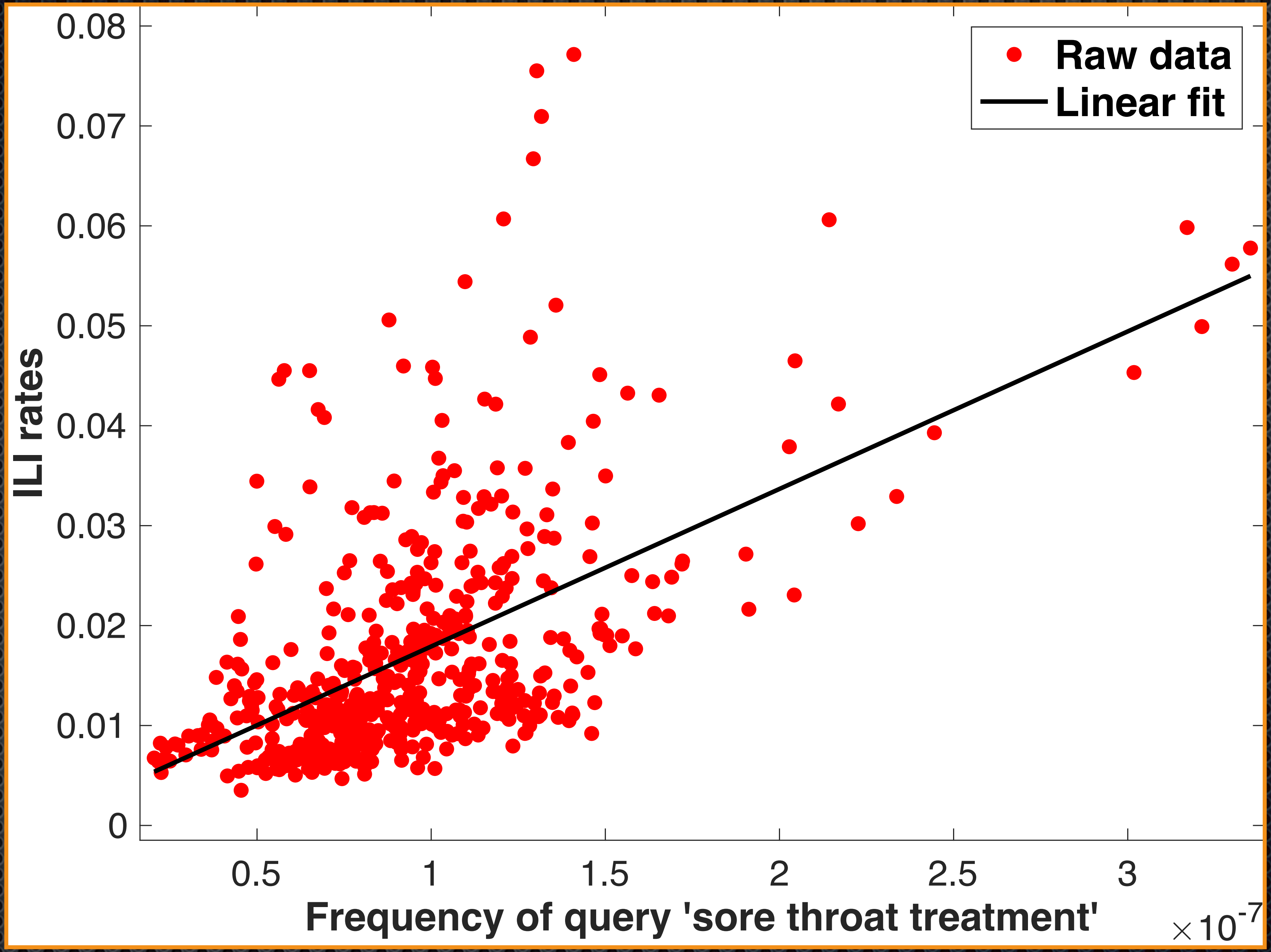
US ILI rates (CDC) ~ freq. of query ‘*how long is flu contagious*’

GFT v.2 — Nonlinearities in the data



US ILI rates (CDC) ~ freq. of query ‘*how to break a fever*’

GFT v.2 — Nonlinearities in the data



US ILI rates (CDC) ~ freq. of query ‘*sore throat treatment*’

GFT v.2 — Gaussian Processes (1/4)

$$\boxed{f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right), \; \mathbf{x}, \mathbf{x}' \in \mathbb{R}^m, \; f : \mathbb{R}^m \to \mathbb{R}}$$

- A Gaussian Process (GP) learns a **distribution over functions** that can explain the data
- Fully specified by a **mean** (*m*) and a **covariance** (**kernel**) function (*k*); we set $m(\mathbf{x}) = 0$ in our experiments
- Collection of random variables any finite number of which have a **multivariate Gaussian distribution**

GFT v.2 — Gaussian Processes (1/4)

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right), \ \ \mathbf{x}, \mathbf{x}' \in \mathbb{R}^m, \ \ f : \mathbb{R}^m \to \mathbb{R}$$

- A Gaussian Process (GP) learns a **distribution over functions** that can explain the data
- Fully specified by a **mean** (*m*) and a **covariance** (**kernel**) function (*k*); we set $m(\mathbf{x}) = 0$ in our experiments
- Collection of random variables any finite number of which have a $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$

GFT v.2 — Gaussian Processes (1/4)

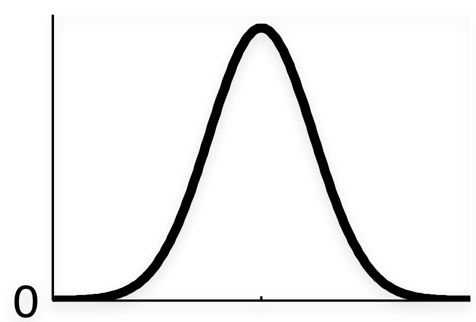
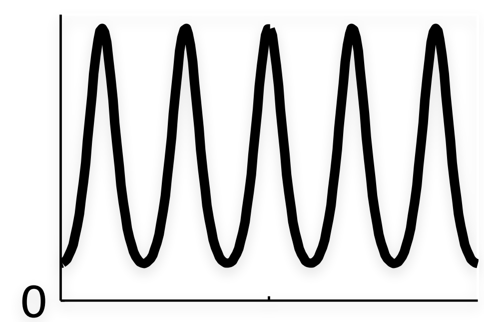
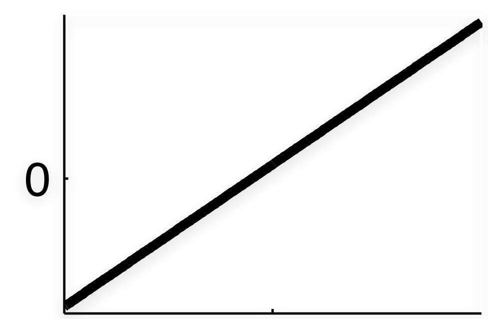
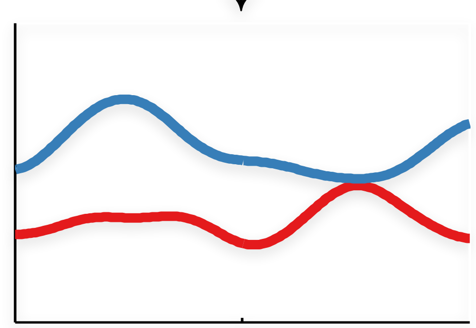
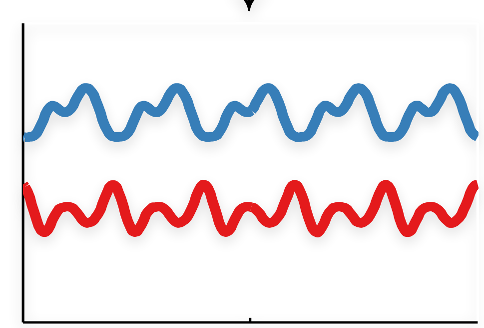
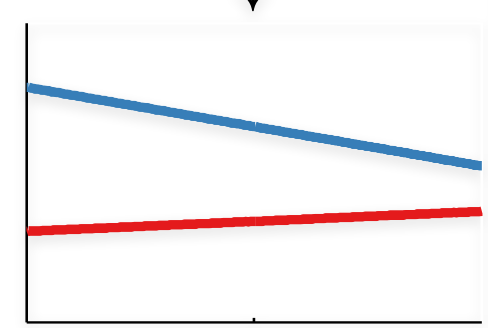
$$\boxed{f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right),\ \mathbf{x}, \mathbf{x}' \in \mathbb{R}^m,\ f : \mathbb{R}^m \to \mathbb{R}}$$

- A Gaussian Process (GP) learns a **distribution over functions** that can explain the data
- Fully specified by a **mean** (*m*) and a **covariance** (**kernel**) function (*k*); we set *m*(**x**) = 0 in our experiments
- Collection of random variables any finite number of which have a $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\}$

inference $f^* \sim \mathcal{N}\left(0, \mathbf{K}\right),\ (\mathbf{K})_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$

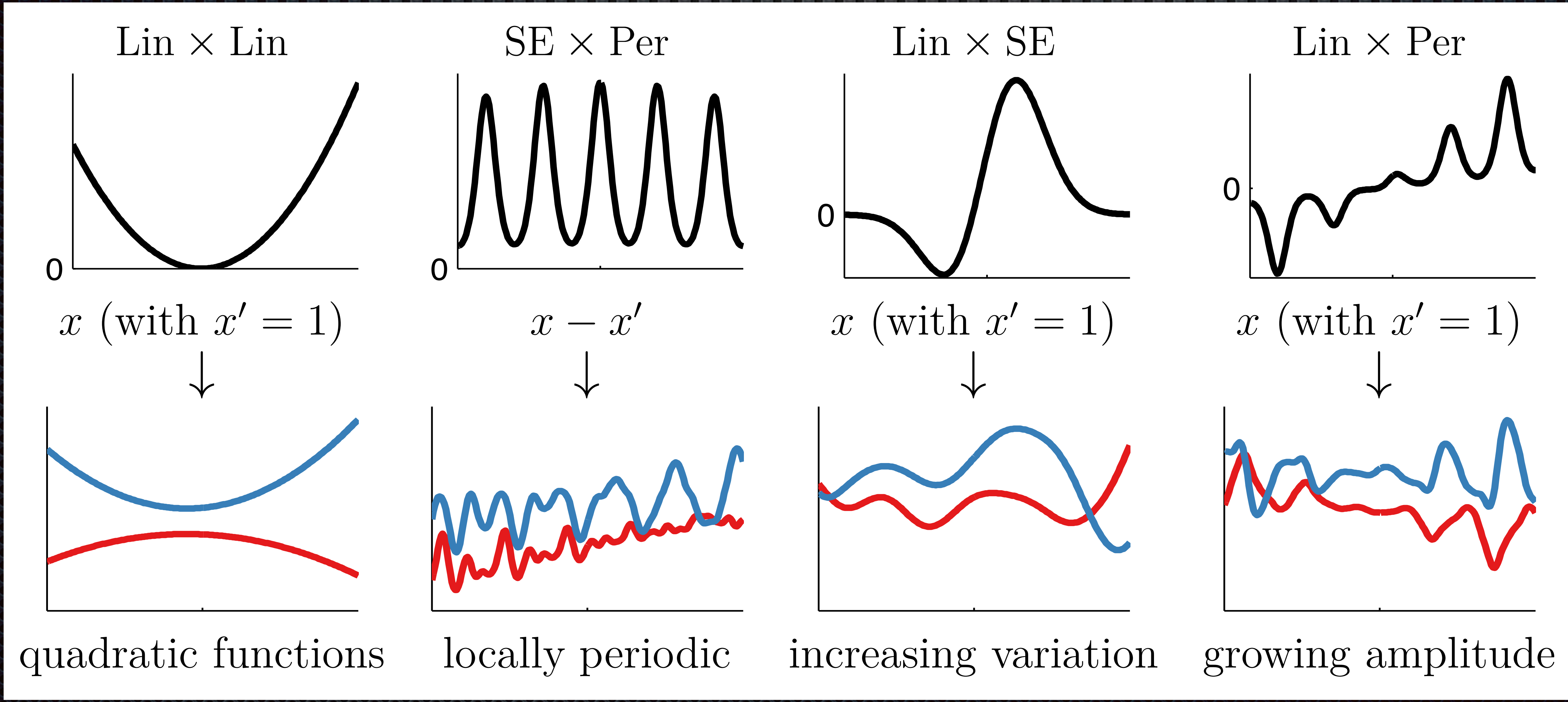
GFT v.2 — Gaussian Processes (2/4)

Common **GP kernels** (covariance functions)

Kernel name:	Squared-exp (SE)	Periodic (Per)	Linear (Lin)
$k(x, x') =$	$\sigma_f^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$	$\sigma_f^2 \exp\left(-\frac{2}{\ell^2} \sin^2\left(\pi \frac{x-x'}{p}\right)\right)$	$\sigma_f^2 (x-c)(x'-c)$
Plot of $k(x, x')$:			
	$x - x'$	$x - x'$	x (with $x' = 1$)
	$\downarrow$	$\downarrow$	$\downarrow$
Functions $f(x)$ sampled from GP prior:			
	x	x	x
Type of structure:	local variation	repeating structure	linear functions

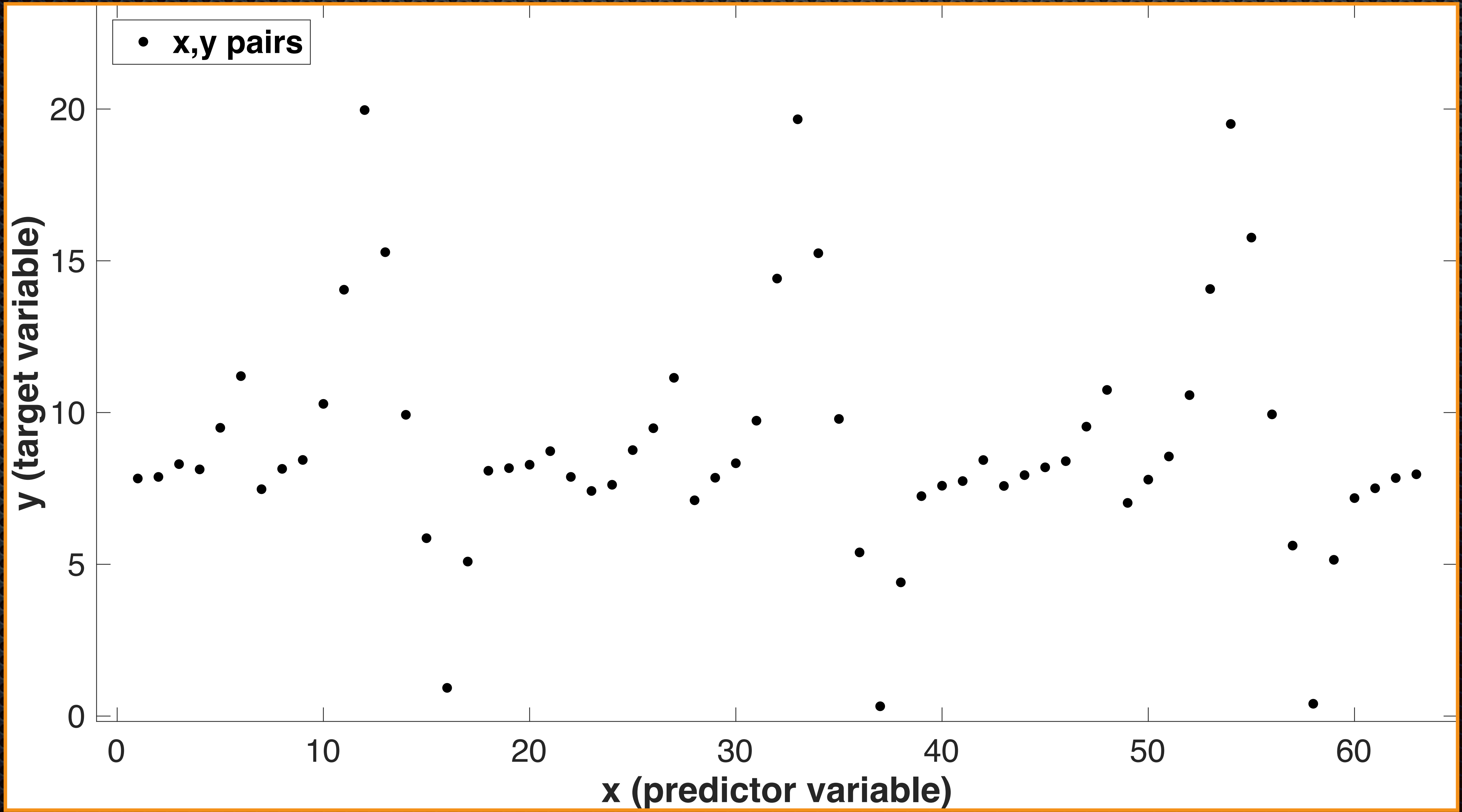
GFT v.2 — Gaussian Processes (3/4)

Adding or **multiplying** GP kernels
produces a new valid GP kernel



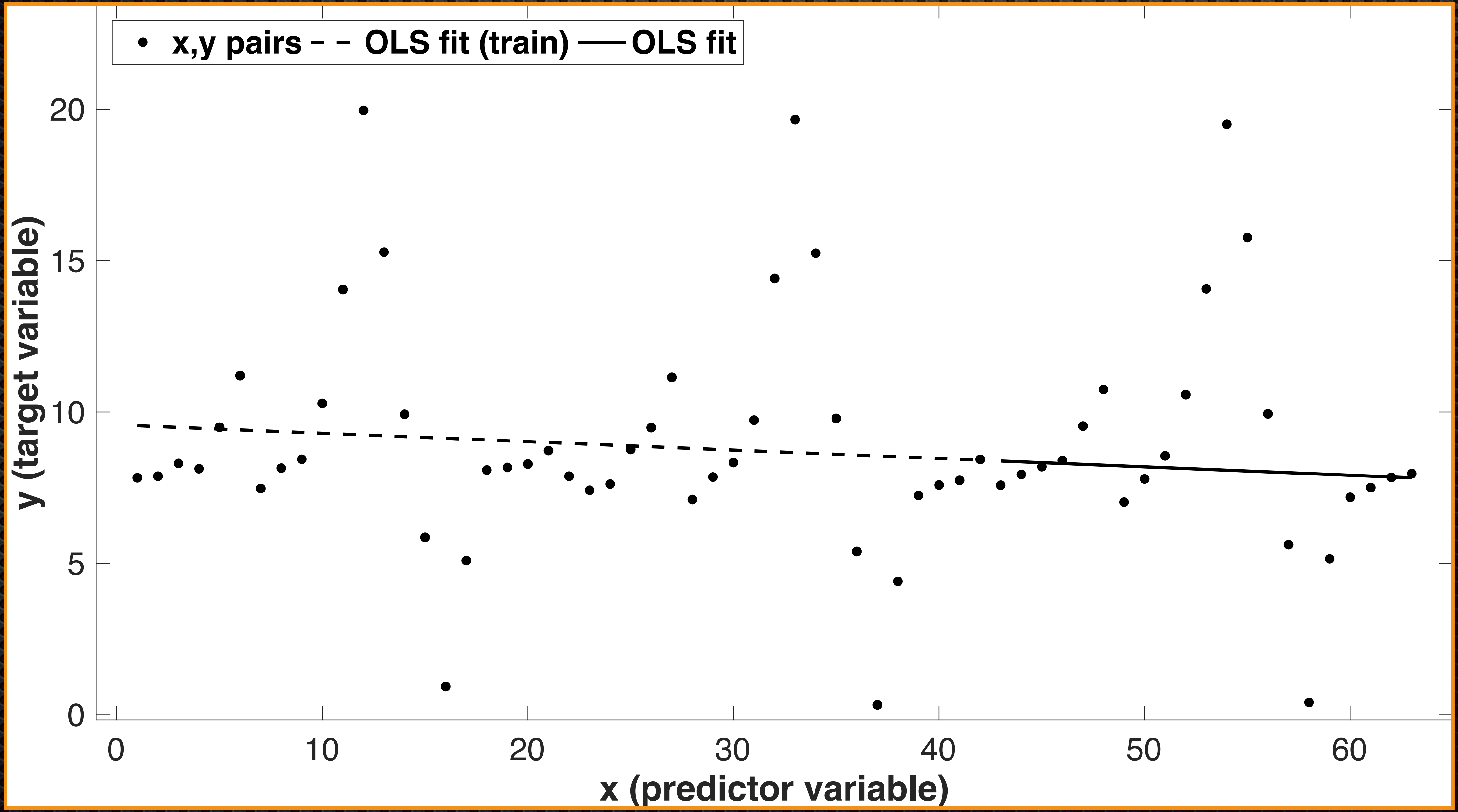
GFT v.2 — Gaussian Processes (4/4)

(x,y) pairs with obvious nonlinear relationship



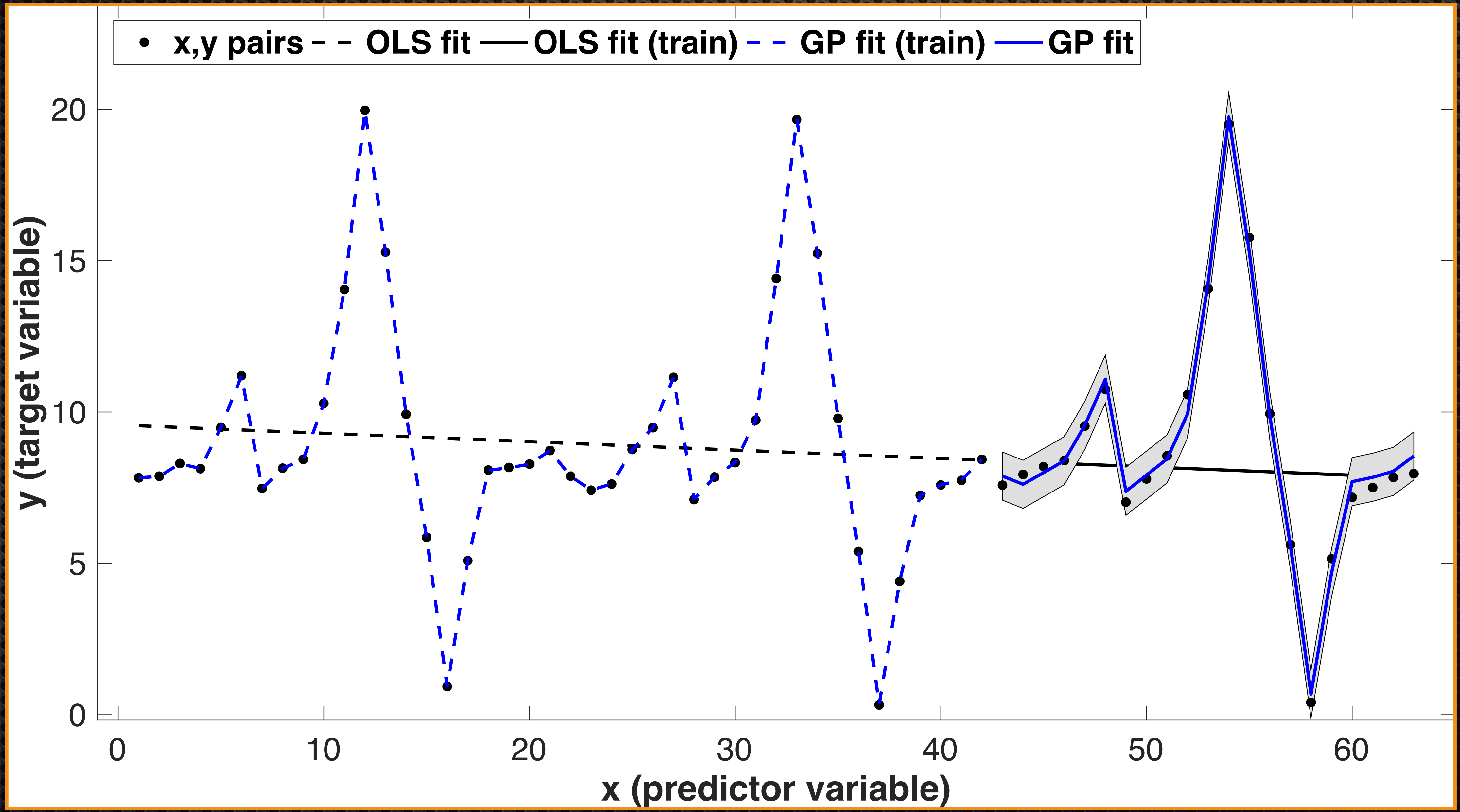
GFT v.2 — Gaussian Processes (4/4)

least squares regression (*poor solution*)



GFT v.2 — Gaussian Processes (4/4)

sum of 2 GP kernels (*periodic* + *squared exponential*)



GFT v.2 — *k*-means and GP regression

- **Clustering** queries selected by elastic net into C clusters with *k*-means
- Clusters are determined by using **cosine similarity** as the distance metric (on query frequency time series)
- Groups queries with similar **topicality** & **usage patterns**

$$k(\mathbf{x}, \mathbf{x}') = \left(\sum_{i=1}^{C} k_{\mathrm{SE}}(\mathbf{x}_{c_i}, \mathbf{x}'_{c_i}) \right) + \sigma^2 \cdot \delta(\mathbf{x}, \mathbf{x}')$$

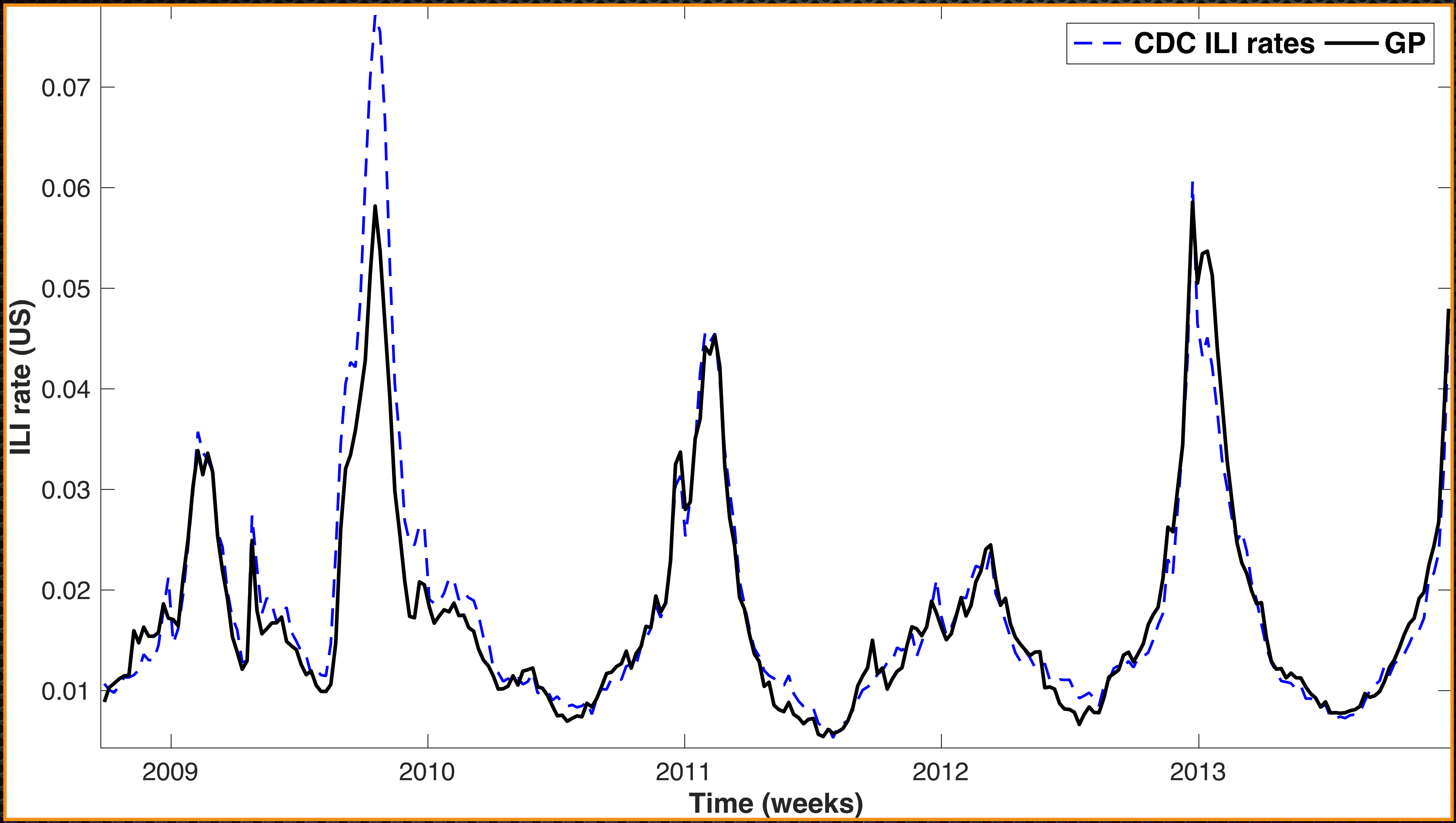
noise

$$\mathbf{x} = \{\mathbf{x}_{c_1}, \ldots, \mathbf{x}_{c_{10}}\}$$

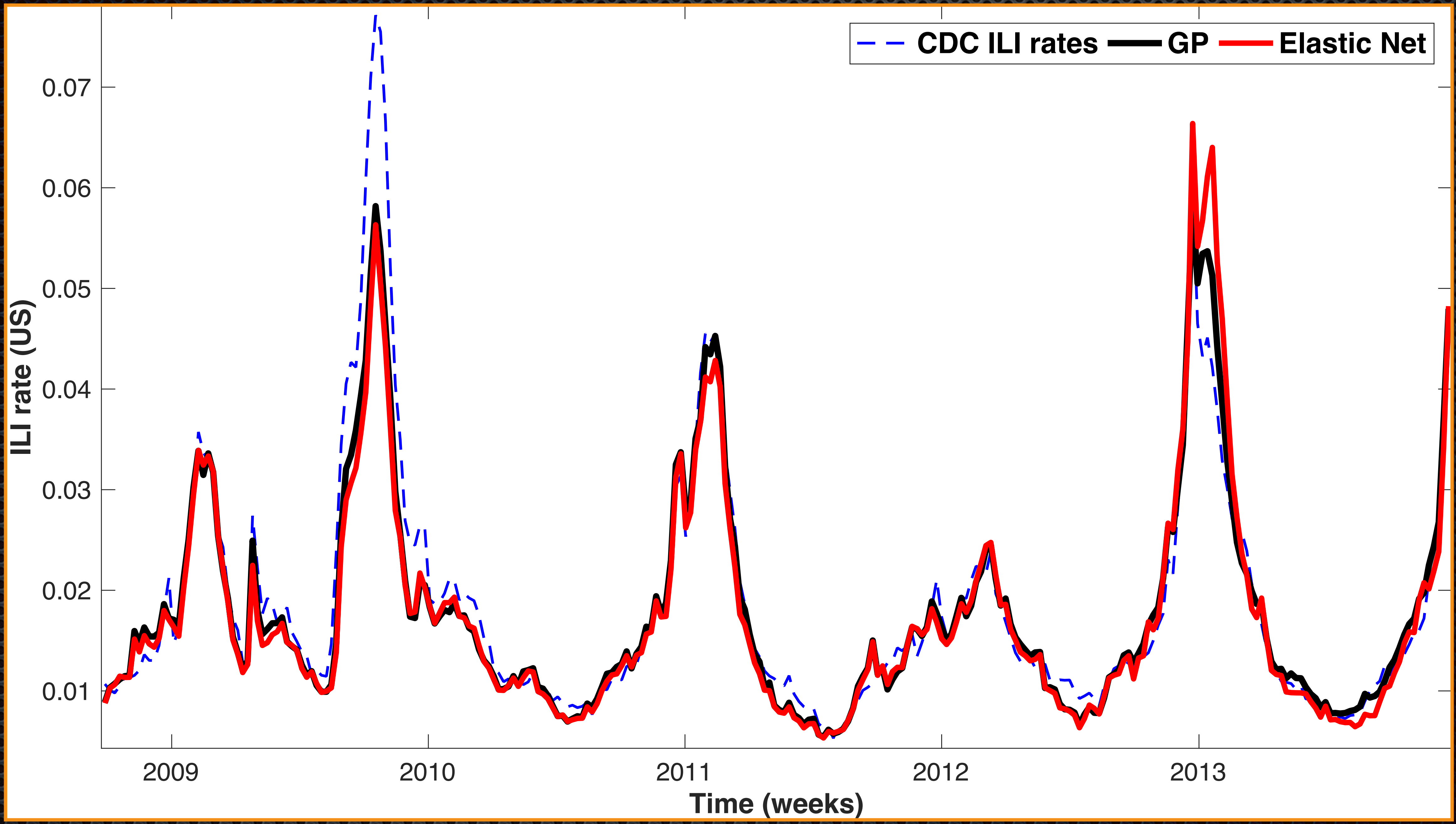
clusters

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2} \right)$$

GFT v.2 — Performance



GFT v.2 — Performance



Elastic net r = .92, MAE = $2.60 \cdot 10^{-3}$, MAPE = 11.9%

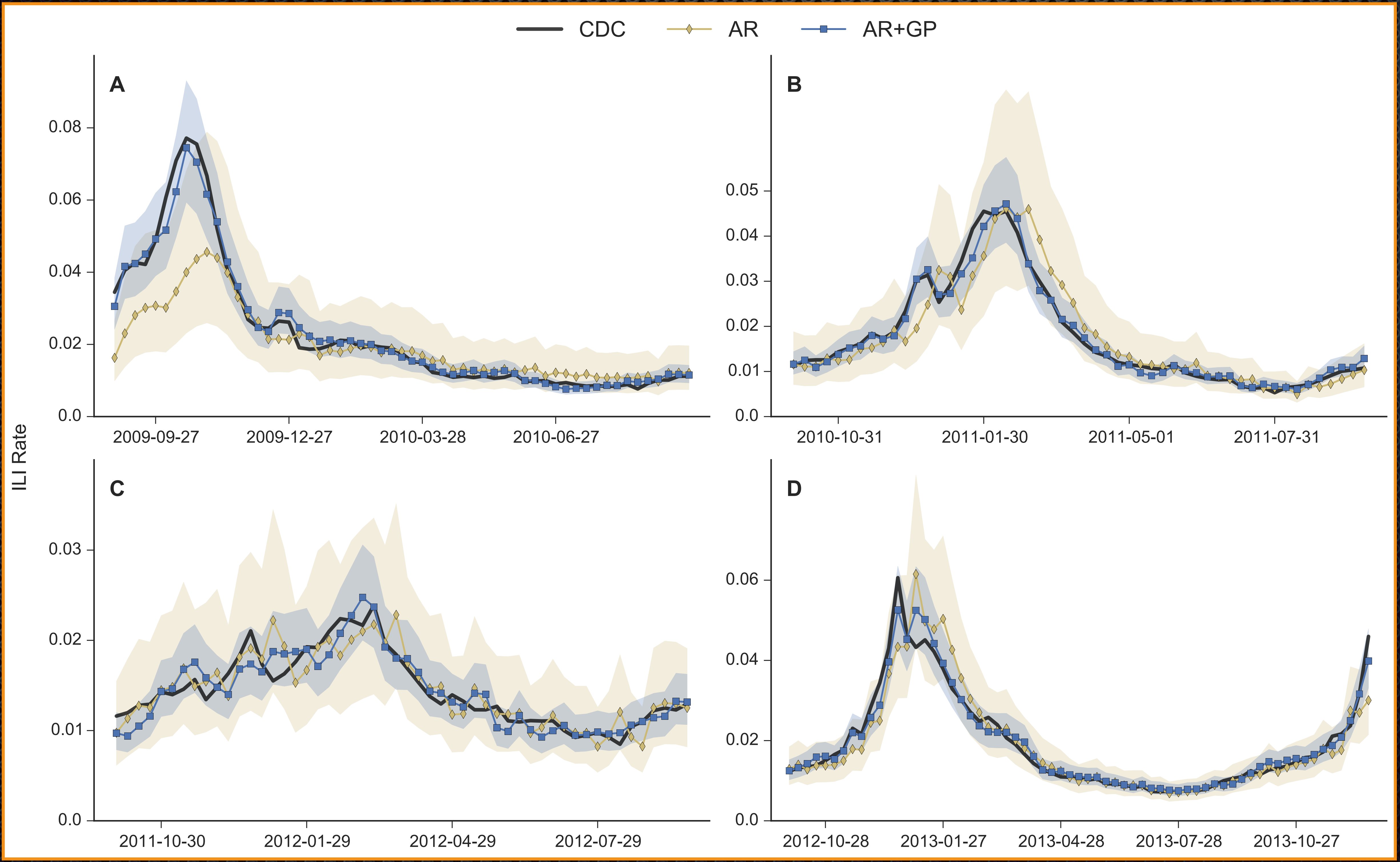
GP r = .95, MAE = $2.21 \cdot 10^{-3}$, MAPE = 10.8%

GFT v.2 — Queries' added value

$$y_t = \underbrace{\sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{J} \omega_i y_{t-52-i}}_{\text{AR and seasonal AR}} + \underbrace{\sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{K} \nu_i \epsilon_{t-52-i}}_{\text{MA and seasonal MA}} + \underbrace{\sum_{i=1}^{D} w_i h_{t,i}}_{\text{regression}} + \epsilon_t$$

- **Autoregression**: Combine CDC ILI rates from the previous week(s) with the ILI rate estimate from search queries for the current week
- Various week lags explored (1, 2,…, 6 weeks)

GFT v.2 — Performance



GFT v.2 — Performance

1-week lag for the CDC data

AR(CDC) r = .97, MAE = $1.87 \cdot 10^{-3}$, MAPE = 8.2%

AR(CDC,GP) r = .99, MAE = $1.05 \cdot 10^{-3}$, MAPE = 5.7%

—

2-week lag for the CDC data

AR(CDC) r = .87, MAE = $3.36 \cdot 10^{-3}$, MAPE = 14.3%

AR(CDC,GP) r = .99, MAE = $1.35 \cdot 10^{-3}$, MAPE = 7.3%

—

GP r = .95, MAE = $2.21 \cdot 10^{-3}$, MAPE = 10.8%

GFT v.2 — Non-optimal feature selection

- Queries **irrelevant to flu** are still maintained, e.g. "*nba injury report*" or "*muscle building supplements*"
- Feature selection is primarily based on **correlation**, then on a **linear relationship**
- Introduce a **semantic feature selection**
  — enhance causal connections (*implicitly*)
  — circumvent the painful training of a classifier

GFT v.3 — Word embeddings

- Word embeddings are **vectors** of a certain dimensionality (usually from 50 to 1024) that represent words in a corpus
- Derive these vectors by **predicting contextual word occurrence** in large corpora (**word2vec**) using a shallow neural network approach:
  — Continuous Bag-Of-Words (**CBOW**): Predict centre word from surrounding ones
  — **skip-gram**: Predict surrounding words from centre one
- Other methods available: **GloVe**, **fastText**

GFT v.3 — Word embedding data sets

Use **tweets** geolocated in the UK to learn word embeddings that may capture

— **informal language** used in searches
— **British English** language / expressions
— **cultural biases**

(a) 215 million tweets (February 2014 to March 2016), CBOW, 512 dimensions, 137,421 words covered

https://doi.org/10.6084/m9.figshare.4052331.v1

(b) 1.1 billion tweets (2012 to 2016), skip-gram, 512 dimensions, 470,194 words covered

https://doi.org/10.6084/m9.figshare.5791650.v1

GFT v.3 — Cosine similarity

word embeddings

$$\cos(\mathbf{v}, \mathbf{u}) = \frac{\mathbf{v} \cdot \mathbf{u}}{\|\mathbf{v}\| \|\mathbf{u}\|} = \frac{\sum_{i=1}^{n} v_i u_i}{\sqrt{\sum_{i=1}^{n} v_i^2} \sqrt{\sum_{i=1}^{n} u_i^2}}$$

GFT v.3 — Cosine similarity

word embeddings

$$\cos(\mathbf{v},\mathbf{u}) = \frac{\mathbf{v}\cdot\mathbf{u}}{\|\mathbf{v}\|\|\mathbf{u}\|} = \frac{\sum_{i=1}^{n} v_i u_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} u_i^2}}$$

$$\max_{\mathbf{v}}\left(\cos(\mathbf{v},\text{‘king’}) + \cos(\mathbf{v},\text{‘woman’}) - \cos(\mathbf{v},\text{‘man’})\right) \Rightarrow \mathbf{v} = \text{‘queen’}$$

GFT v.3 — Cosine similarity

word embeddings

$$\cos(\mathbf{v},\mathbf{u}) = \frac{\mathbf{v}\cdot\mathbf{u}}{\|\mathbf{v}\|\|\mathbf{u}\|} = \frac{\sum_{i=1}^{n} v_i u_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} u_i^2}}$$

$$\max_{\mathbf{v}} \left(\cos(\mathbf{v}, \text{‘king’}) + \cos(\mathbf{v}, \text{‘woman’}) - \cos(\mathbf{v}, \text{‘man’})\right) \Rightarrow \mathbf{v} = \text{‘queen’}$$

or

$$\max_{\mathbf{v}} \left(\frac{\cos(\mathbf{v}, \text{‘king’}) \times \cos(\mathbf{v}, \text{‘woman’})}{\cos(\mathbf{v}, \text{‘man’})}\right) \Rightarrow \mathbf{v} = \text{‘queen’}$$

where $\cos(\cdot,\cdot) = \left(\cos(\cdot,\cdot)+1\right)/2$

GFT v.3 — Cosine similarity

word embeddings

$$\cos(\mathbf{v},\mathbf{u}) = \frac{\mathbf{v}\cdot\mathbf{u}}{\|\mathbf{v}\|\|\mathbf{u}\|} = \frac{\sum_{i=1}^{n} v_i u_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} u_i^2}}$$

$$\max_{\mathbf{v}}\left(\cos(\mathbf{v},\text{‘king’}) + \cos(\mathbf{v},\text{‘woman’}) - \cos(\mathbf{v},\text{‘man’})\right) \Rightarrow \mathbf{v} = \text{‘queen’}$$

Positive context

or

$$\max_{\mathbf{v}}\left(\frac{\cos(\mathbf{v},\text{‘king’}) \times \cos(\mathbf{v},\text{‘woman’})}{\cos(\mathbf{v},\text{‘man’})}\right) \Rightarrow \mathbf{v} = \text{‘queen’}$$

Negative context

where $= \left(\cos(\cdot,\cdot)+1\right)/2$

GFT v.3 — Analogies in Twitter embd.

The …	for …	not the …	is … ?
woman	king	man	?
him	she	he	?
better	bad	good	?
England	Rome	London	?
Messi	basketball	football	?
Guardian	Conservatives	Labour	?
Trump	Europe	USA	?
rsv	fever	skin	?

GFT v.3 — Analogies in Twitter embd.

The …	for …	not the …	is … ?
woman	king	man	*queen*
him	she	he	?
better	bad	good	?
England	Rome	London	?
Messi	basketball	football	?
Guardian	Conservatives	Labour	?
Trump	Europe	USA	?
rsv	fever	skin	?

GFT v.3 — Analogies in Twitter embd.

The …	for …	not the …	is … ?
woman	king	man	*queen*
him	she	he	*her*
better	bad	good	*?*
England	Rome	London	*?*
Messi	basketball	football	*?*
Guardian	Conservatives	Labour	*?*
Trump	Europe	USA	*?*
rsv	fever	skin	*?*

GFT v.3 — Analogies in Twitter embd.

The …	for …	not the …	is … ?
woman	king	man	*queen*
him	she	he	*her*
better	bad	good	*worse*
England	Rome	London	*Italy*
Messi	basketball	football	*Lebron*
Guardian	Conservatives	Labour	*Telegraph*
Trump	Europe	USA	*Farage*
rsv	fever	skin	*flu*

GFT v.3 — Better query selection (1/3)

1. **Query embedding** = Average token embedding
2. Derive a **concept** by specifying a **positive** (*P*) and a **negative** (*N*) **context** (sets of n-grams)
3. **Rank** all queries using their **similarity score** with this concept

GFT v.3 — Better query selection (1/3)

1. **Query embedding** = Average token embedding
2. Derive a **concept** by specifying a **positive** (*P*) and a **negative** (*N*) **context** (sets of n-grams)
3. **Rank** all queries using their **similarity score** with this concept

$$S\left(\mathcal{Q},\mathcal{C}\right)=\frac{\sum_{i=1}^{k}\cos\left(\mathbf{e}_{\mathcal{Q}},\mathbf{e}_{P_i}\right)}{\sum_{j=1}^{z}\cos\left(\mathbf{e}_{\mathcal{Q}},\mathbf{e}_{N_j}\right)+\gamma}$$

GFT v.3 — Better query selection (1/3)

1. **Query embedding** = Average token embedding
2. Derive a **concept** by specifying a **positive** (*P*) and a **negative** (*N*) **context** (sets of n-grams)
3. **Rank** all queries using their **similarity score** with this concept

query embedding

$$S(\mathcal{Q}, \mathcal{C}) = \frac{\sum_{i=1}^{k} \cos\left(\mathbf{e}_{\mathcal{Q}}, \mathbf{e}_{P_i}\right)}{\sum_{j=1}^{z} \cos\left(\mathbf{e}_{\mathcal{Q}}, \mathbf{e}_{N_j}\right) + \gamma}$$

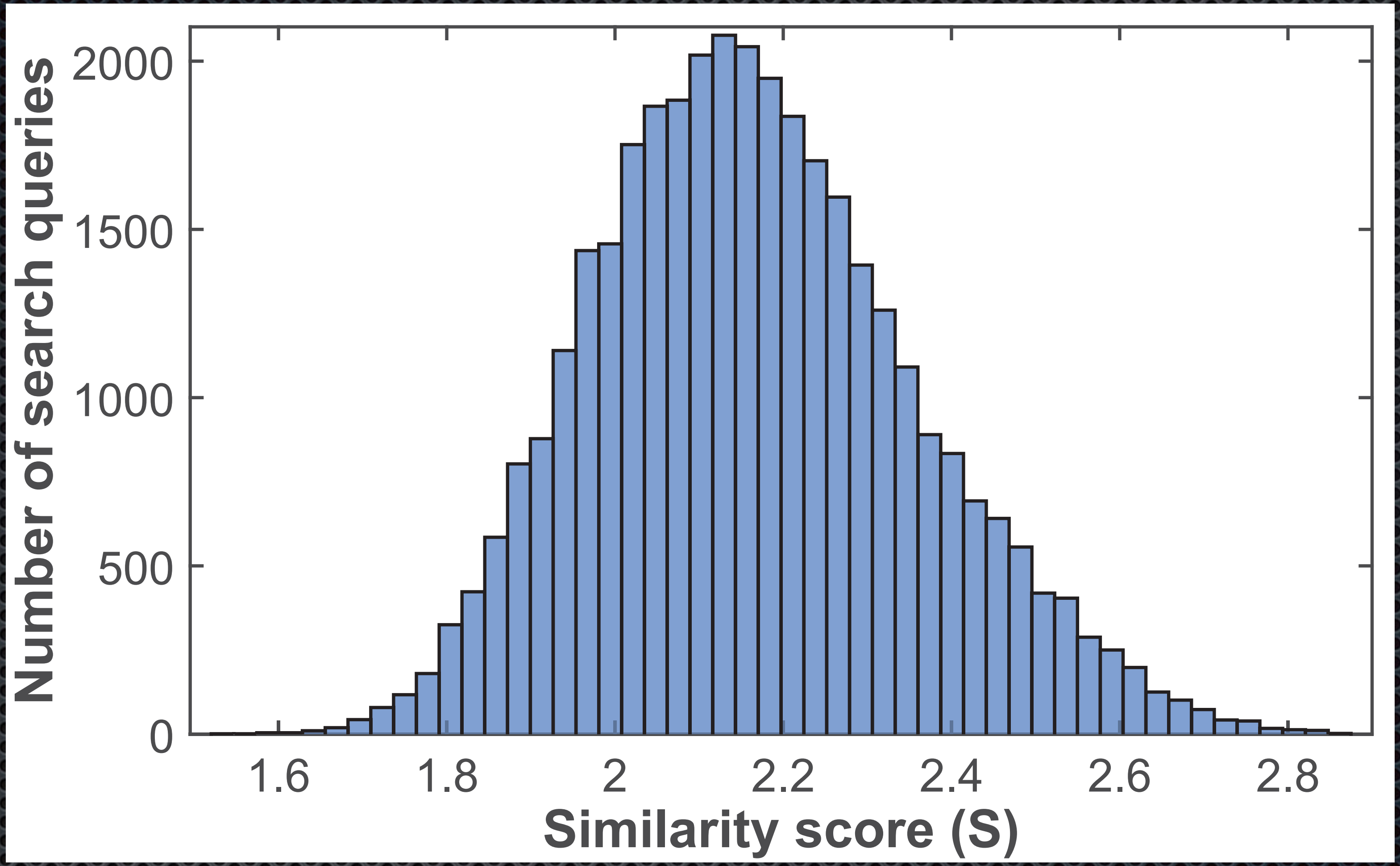
embedding of a negative concept n-gram

constant to avoid division by 0

GFT v.3 — Better query selection (2/3)

Positive context	Negative context	Most similar queries
#flu fever flu flu medicine gp hospital	bieber ebola wikipedia	cold flu medicine flu aches cold and flu cold flu symptoms colds and flu
flu flu gp flu hospital flu medicine	ebola wikipedia	flu aches flu colds and flu cold and flu cold flu medicine

GFT v.3 — Better query selection (3/3)



Given that the **distribution** of concept similarity scores appears to be **unimodal**, we standard deviations from the mean ($\mu_S + \theta\sigma_S$) to determine the selected queries

GFT v.3 — Hybrid feature selection

- Embedding based feature selection is an **unsupervised technique**, thus non optimal
- If we combine it with the previous ways for selecting features, will we obtain **better inference accuracy**?

  We test **7 feature selection approaches**:
    - similarity → elastic net (**1**)
    - correlation → elactic net (**2**) → GP (**3**)
    - similarity → correlation → elastic net (**4**) → GP (**5**)
    - similarity → correlation → GP (**6**)
    - correlation → GP (**7**)

GFT v.3 — GP model details

Skipped in the interest of time!

If you're interested, check Section 3.1 of

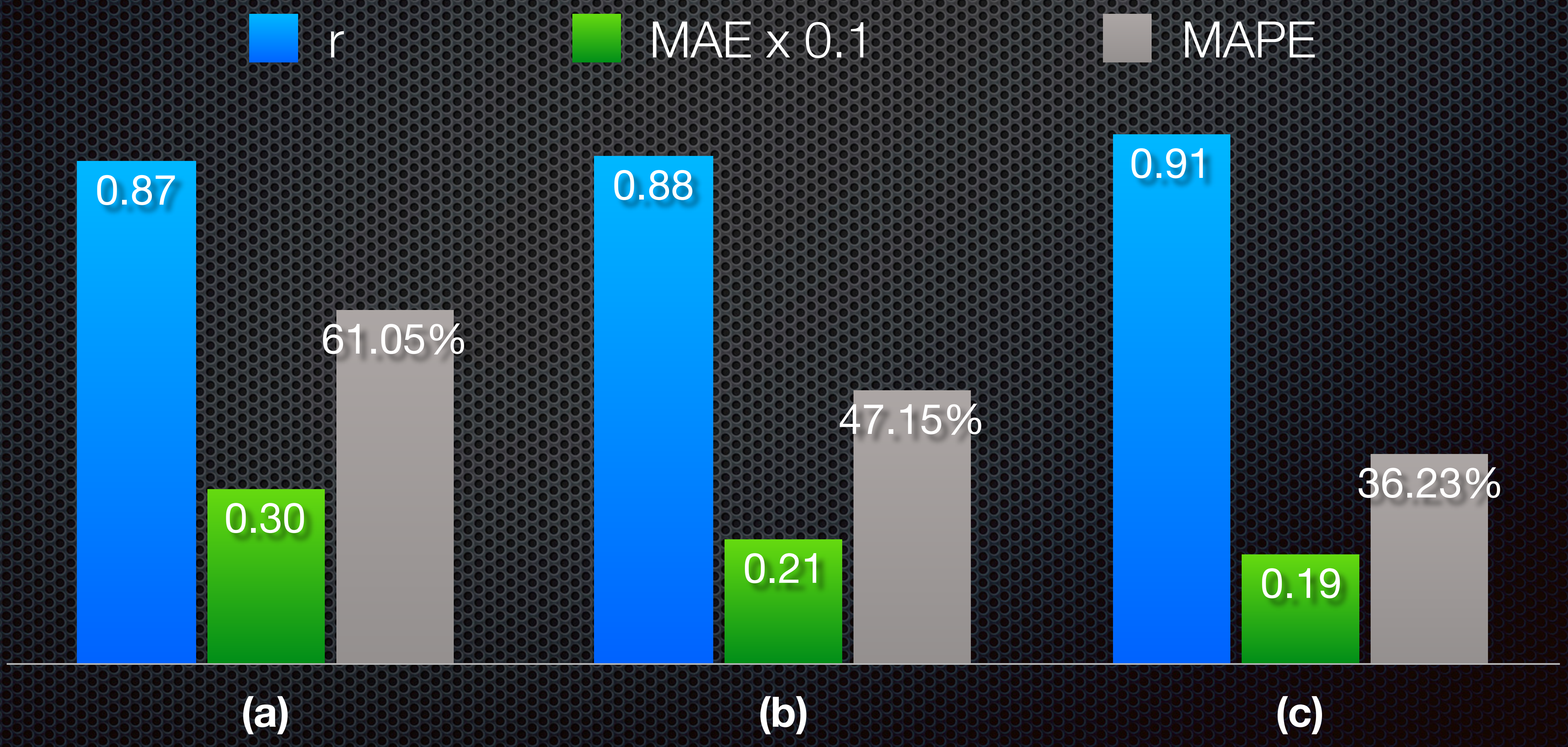
https://doi.org/10.1145/3038912.3052622

GFT v.3 — Data & evaluation

- weekly frequency of **35,572** search queries (UK)
- from 1/1/**2007** to 9/08/**2015** (**449 weeks**)
- access to a private Google Health Trends API for health-oriented research
- corresponding ILI rates for England (Royal College of General Practitioners and Public Health England)
- **test on the last 3 flu seasons** in the data (2012-2015)
- **train on increasing data sets starting from 2007**, using all data prior to a test period

GFT v.3 — Performance (1/3)

(a) similarity → elastic net
(b) correlation → elactic net
(c) similarity → correlation → elastic net

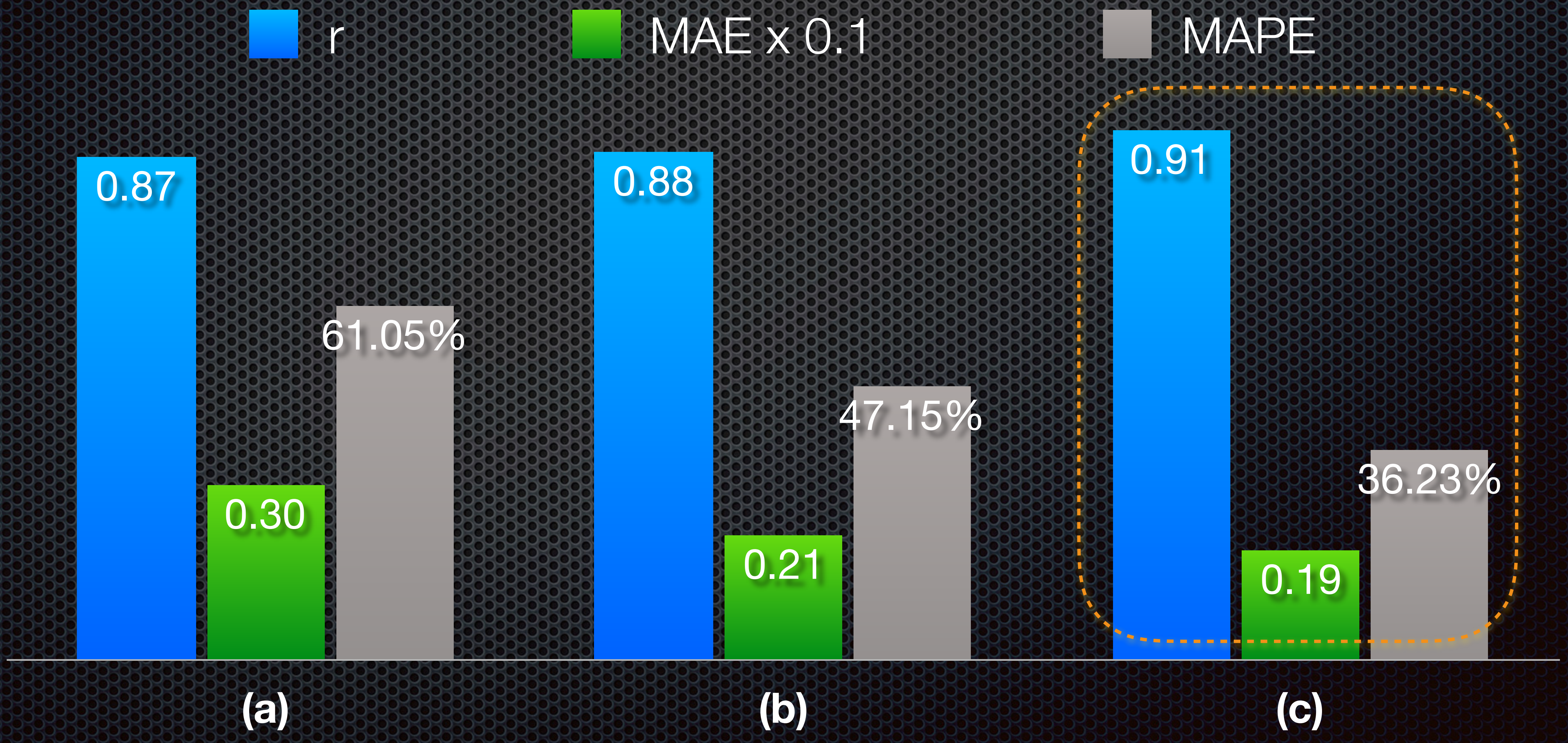


GFT v.3 — Performance (1/3)

(a) similarity → elastic net

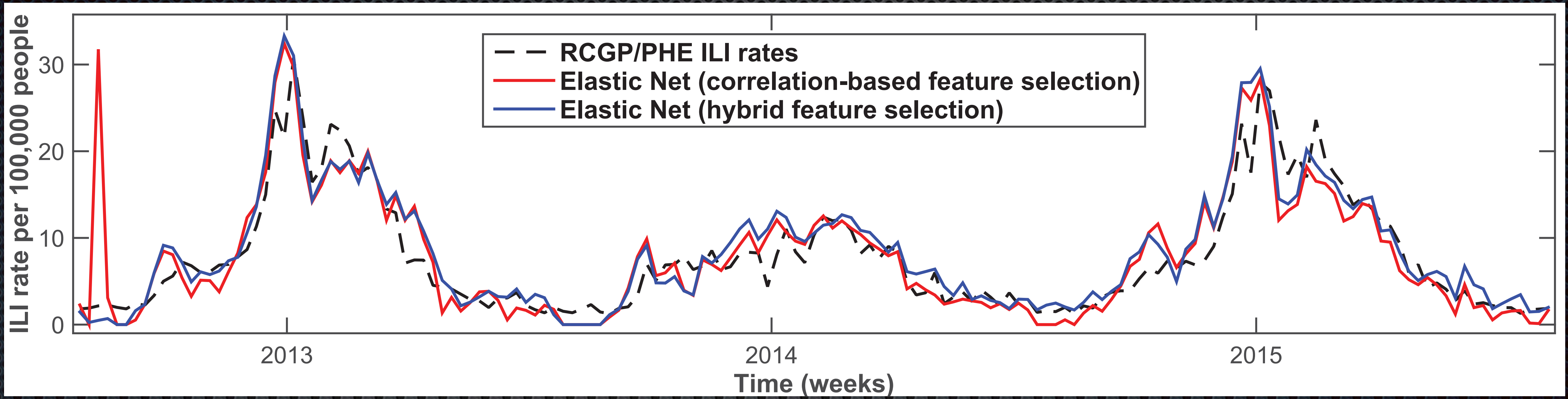
(b) correlation → elactic net

(c) similarity → correlation → elastic net



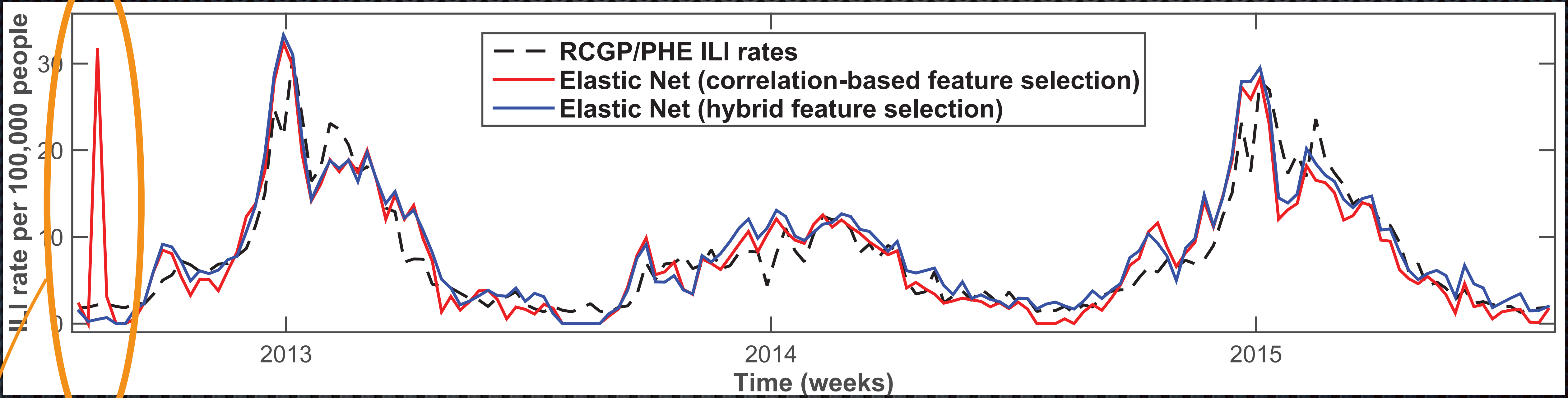
GFT v.3 — Performance (2/3)

Elastic net with and without word embeddings filtering



GFT v.3 — Performance (2/3)

Elastic net with and without word embeddings filtering

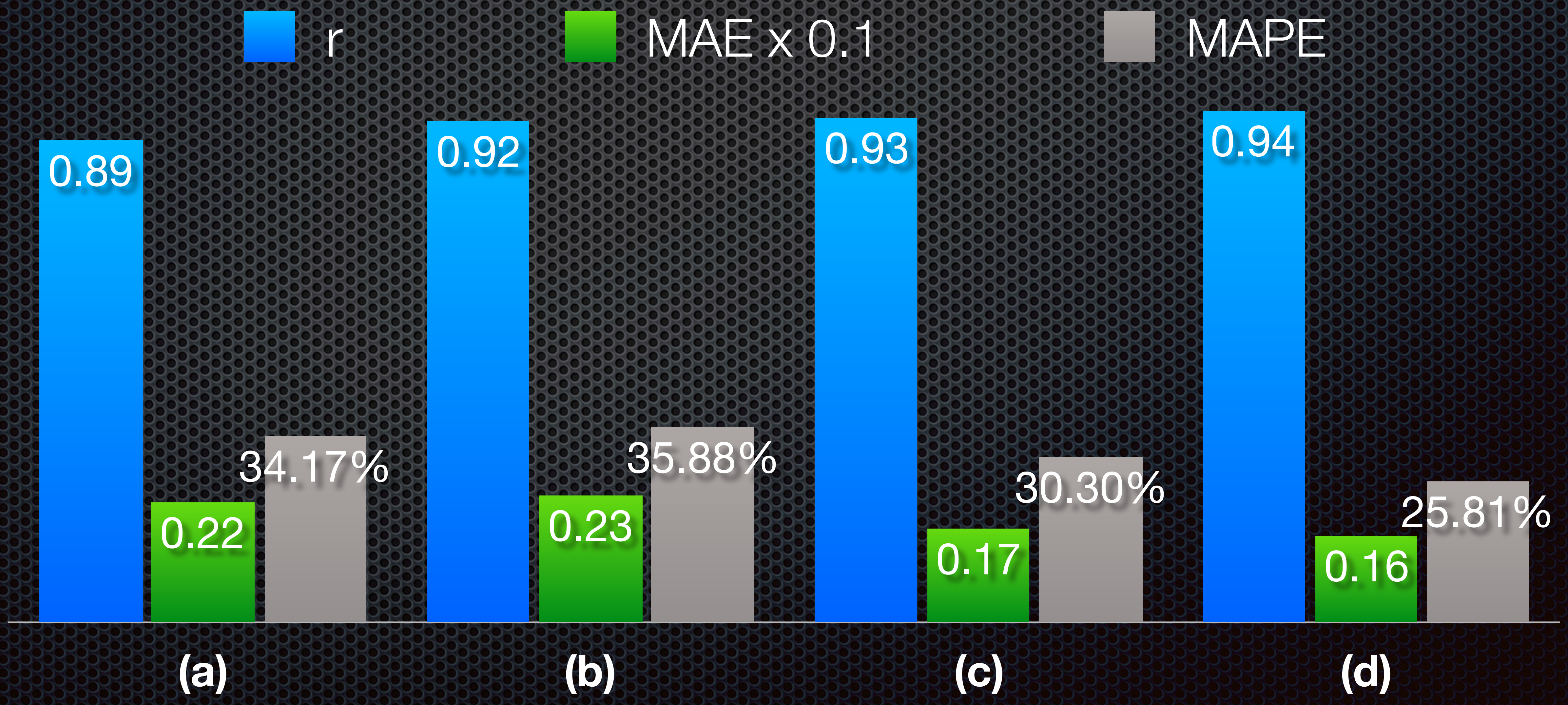


ratio over highest weight

prof. *surname* (70.3%), *name surname* (27.2%), heal the world (21.9%), heating oil (21.2%), *name surname* recipes (21%), tlc diet (13.3%), blood game (12.3%), swine flu vaccine side effects (7.2%)

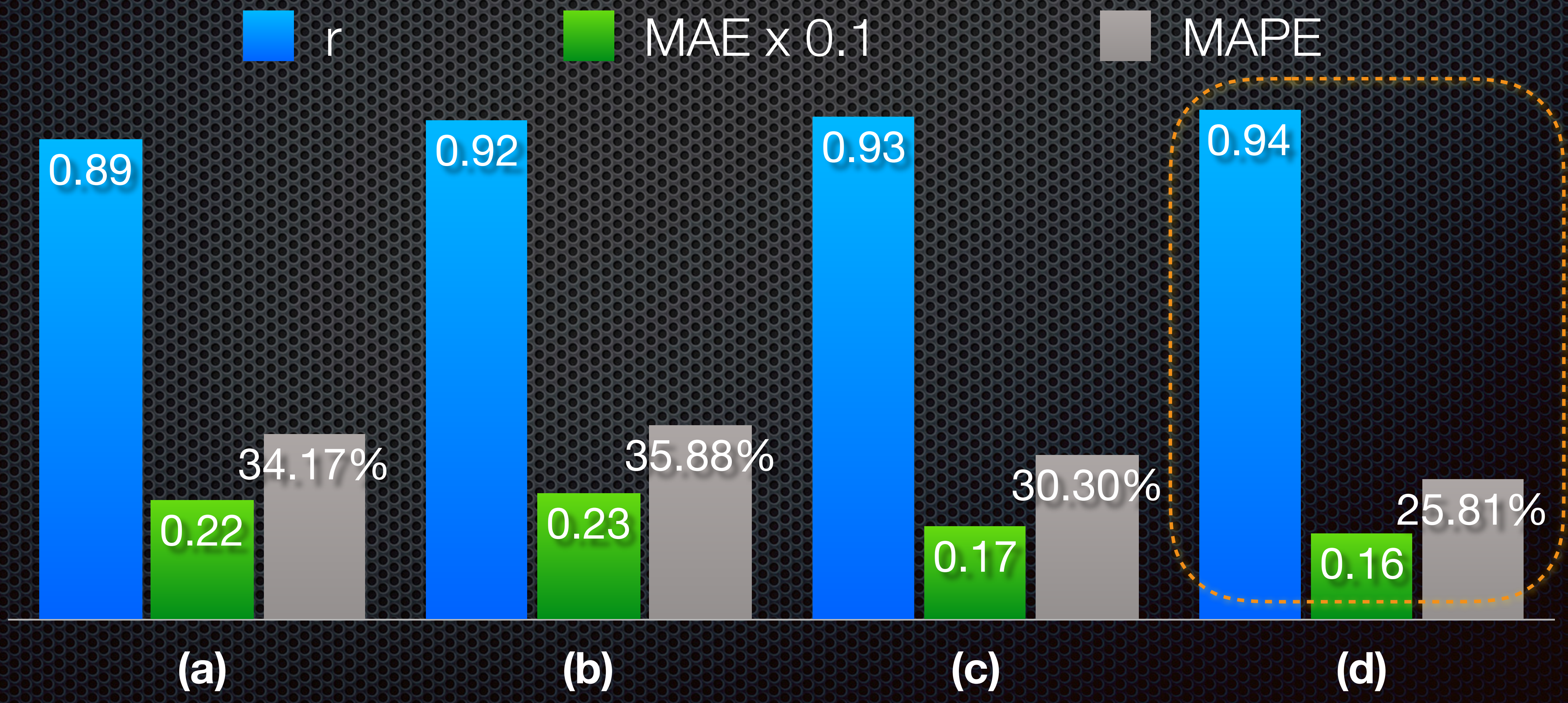
GFT v.3 — Performance (3/3)

(a) correlation → GP
(b) correlation → elastic net → GP
(c) similarity → correlation → elactic net → GP
(d) similarity → correlation → GP



GFT v.3 — Performance (3/3)

(a) correlation → GP
(b) correlation → elastic net → GP
(c) similarity → correlation → elactic net → GP
(d) similarity → correlation → GP



Multi-task learning

- m tasks (problems) $t_1,\ldots,t_m$
- observations $\mathbf{X}_{t_1},\mathbf{y}_{t_1},\ldots,\mathbf{X}_{t_m},\mathbf{y}_{t_m}$
- learn models f_{t_i}**:** $\mathbf{X}_{t_i} \rightarrow \mathbf{y}_{t_i}$ **jointly** (*and not independently*)

Why?

- When tasks are related, multi-task learning is expected to **perform better** than learning each task independently
- Model learning possible even with a **few training samples**

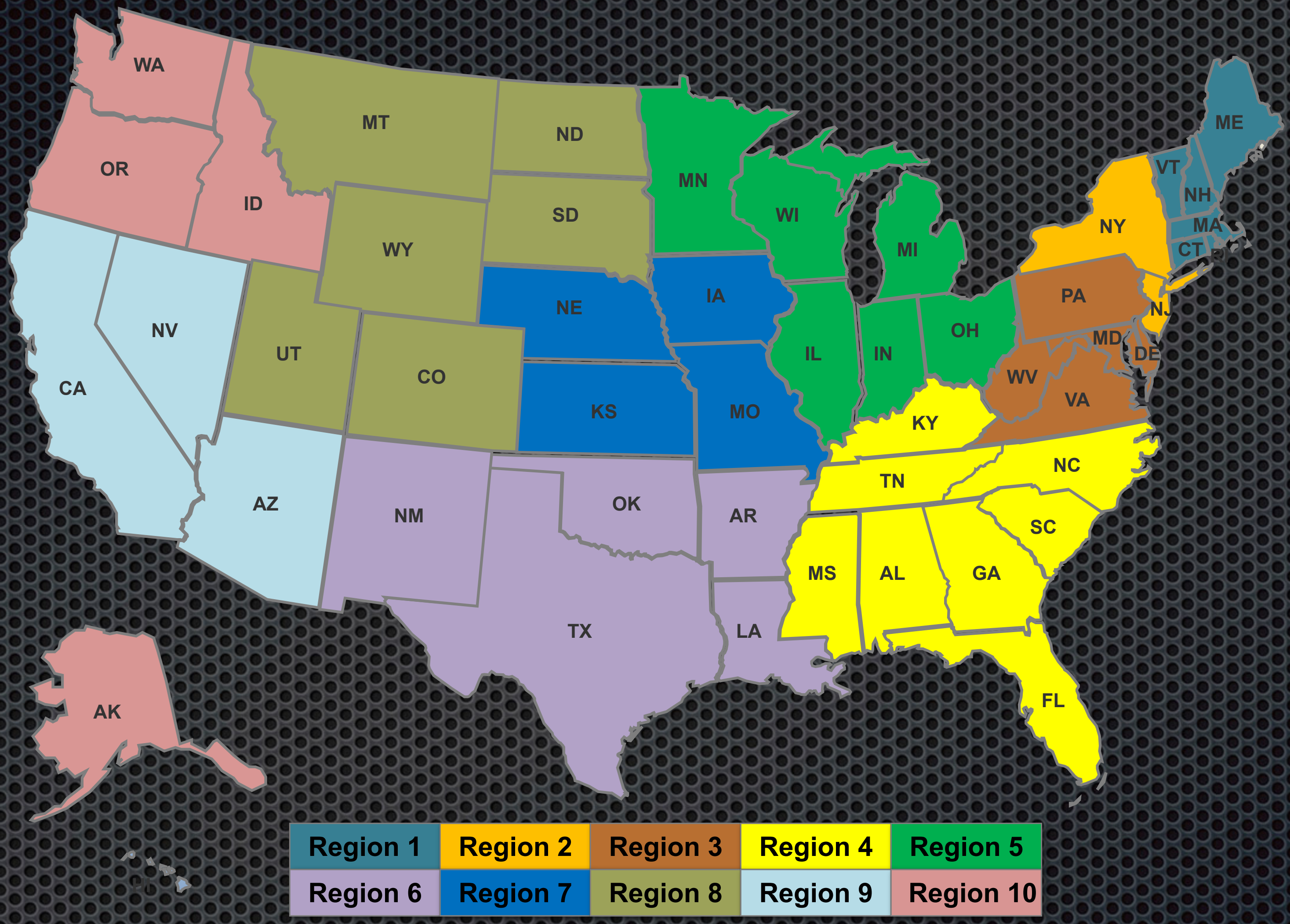
Multi-task learning for disease modelling

- m tasks (problems) $t_1,\ldots,t_m$
- observations $\mathbf{X}_{t_1},\mathbf{y}_{t_1},\ldots,\mathbf{X}_{t_m},\mathbf{y}_{t_m}$
- learn models f_{t_i}: $\mathbf{X}_{t_i} \rightarrow \mathbf{y}_{t_i}$ **jointly** (*and not independently*)

Can we **improve disease models** (*flu*) from online search:

- when sporadic training data are available?
- across the geographical regions of a country?
- across two different countries?

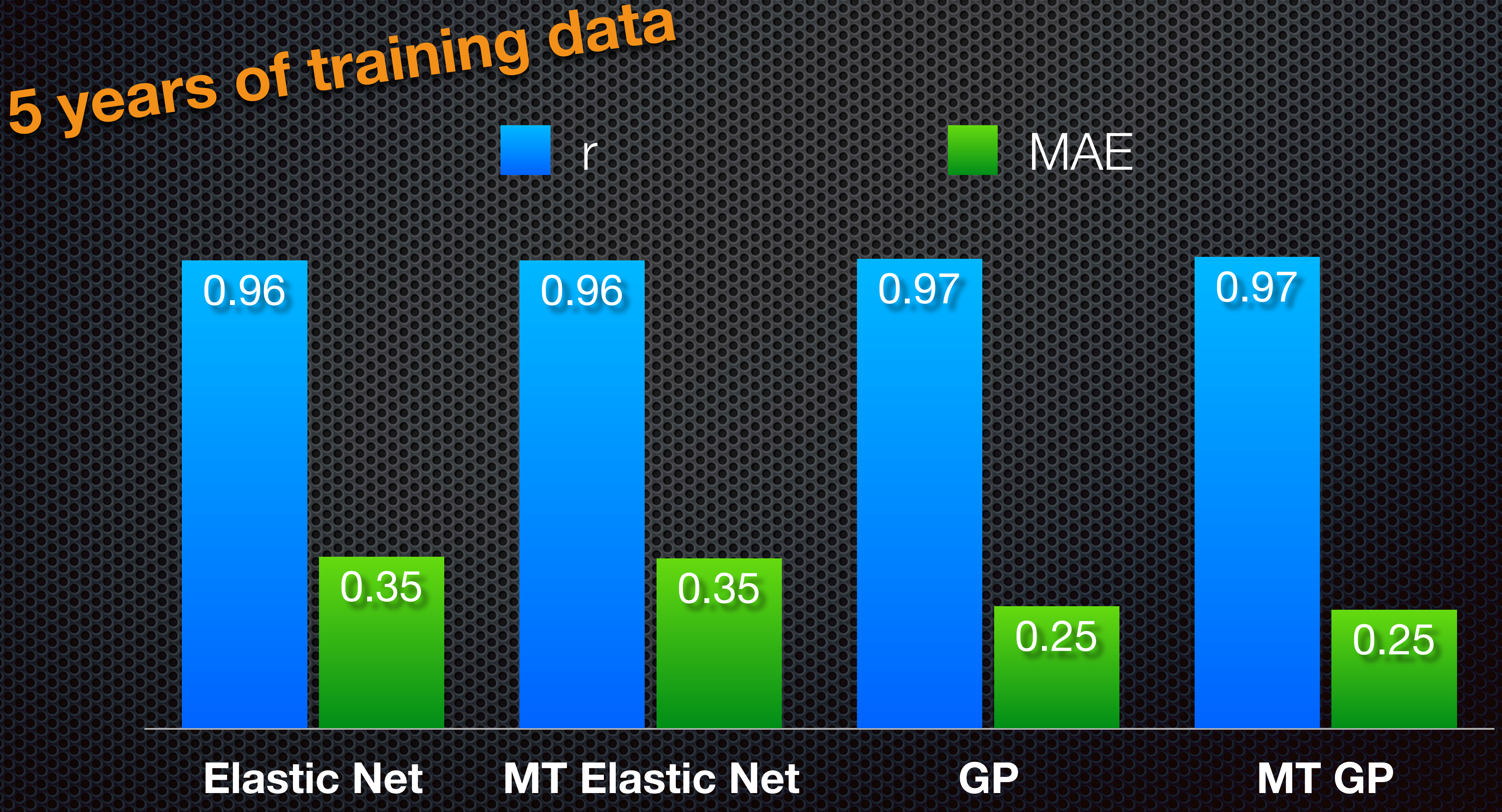
Multi-task learning GFT (1/5)



- Can multi-task learning across the 10 US regions help us improve the **national** ILI model?

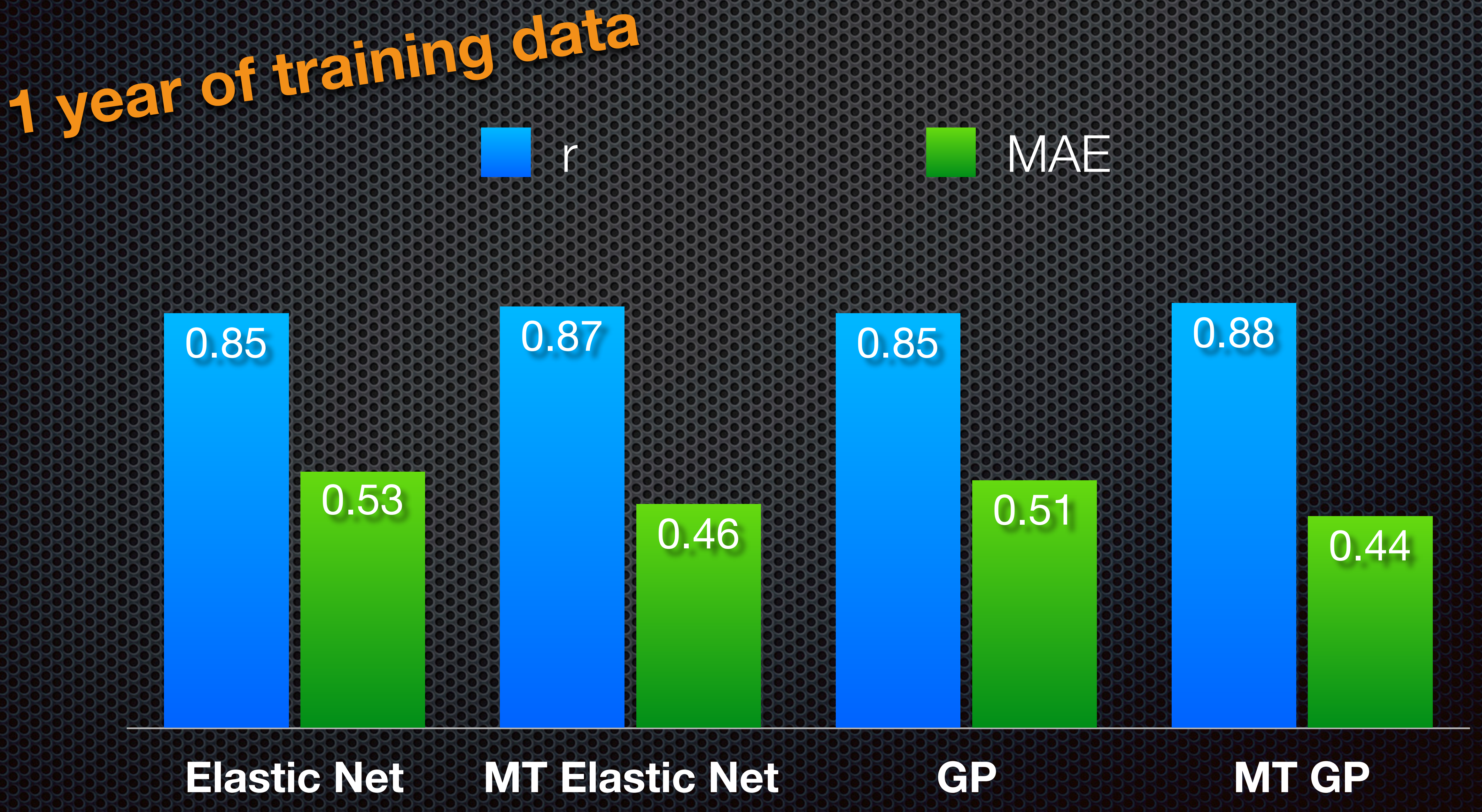
Multi-task learning GFT (1/5)

- Can multi-task learning across the 10 US regions help us improve the **national** ILI model?



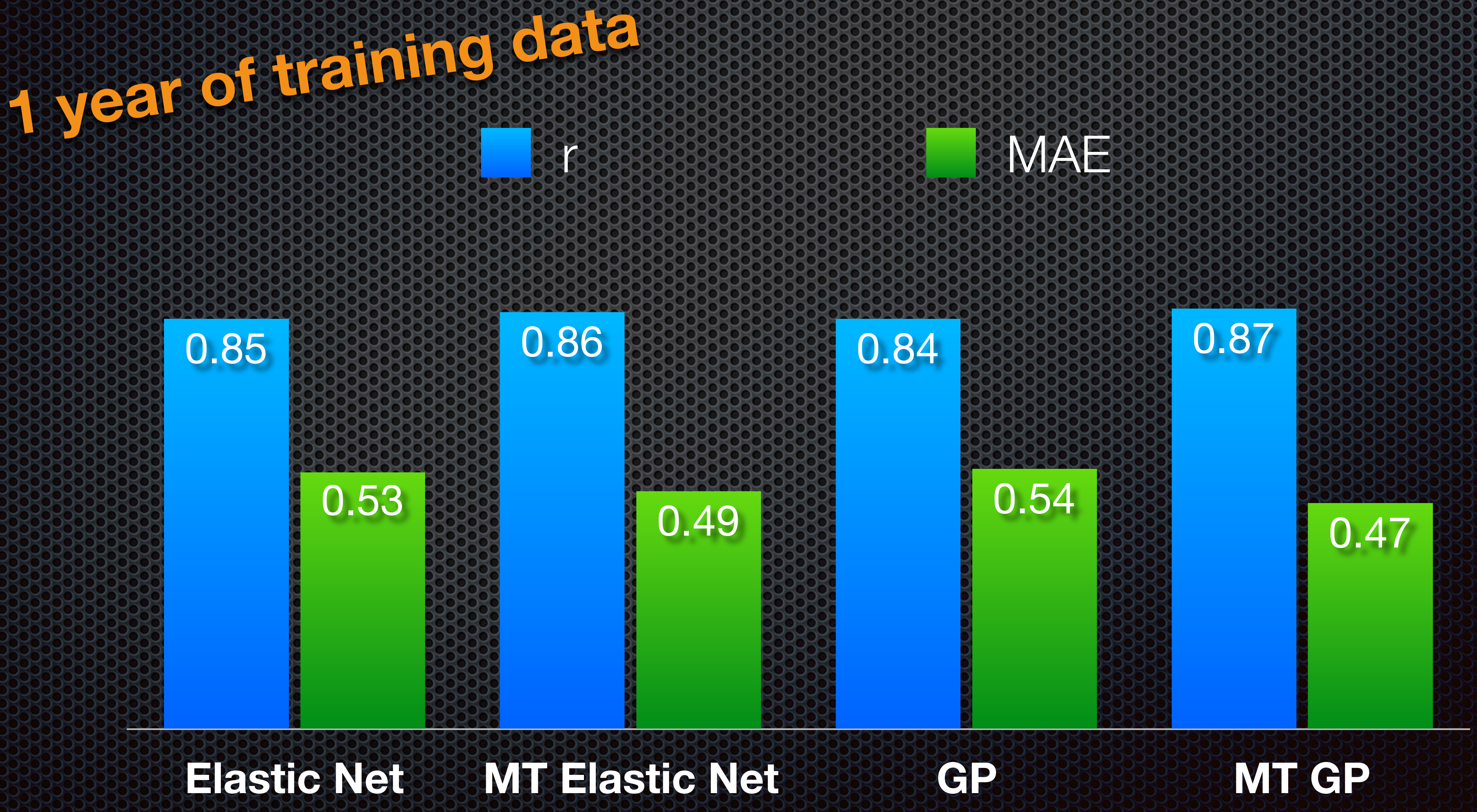
Multi-task learning GFT (1/5)

- Can multi-task learning across the 10 US regions help us improve the **national** ILI model?



Multi-task learning GFT (2/5)

- Can multi-task learning across the 10 US regions help us improve the **regional** ILI models?

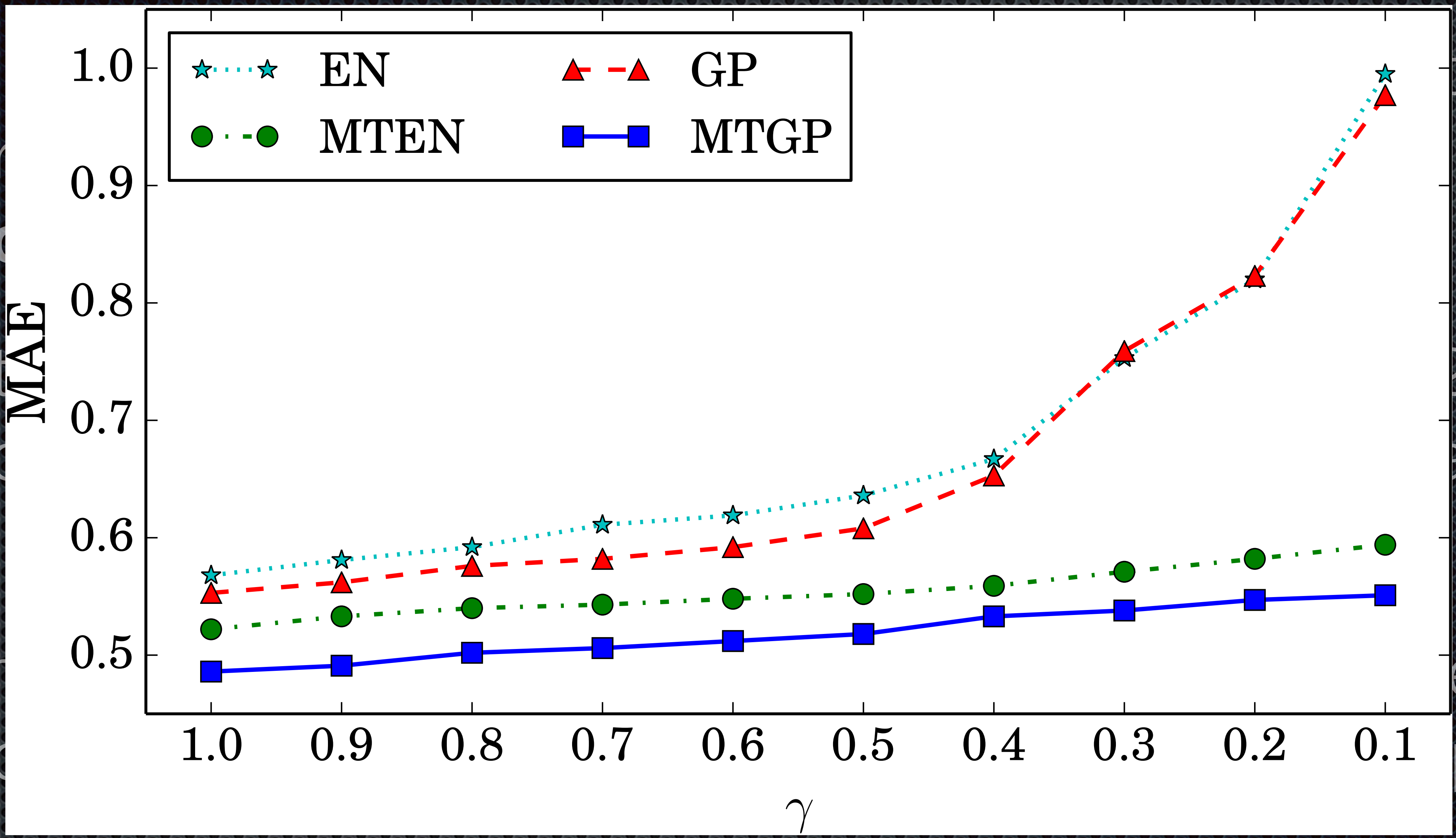


Multi-task learning GFT (3/5)

- Can multi-task learning across the 10 US regions help us improve **regional** models under **sporadic health reporting**?
- Split US regions into **two groups**, one including the 2 regions with the highest population (4 and 9 in the map), and the other having the remaining 8 regions
- Train and evaluate models for the **8 regions** under the hypothesis that there might exist **sporadic health reports**
- Start **downsampling** the data from the 8 regions using **burst error sampling** (random data blocks removed) with rate γ (1 no sampling, 0.1 10% sample)

Multi-task learning GFT (3/5)

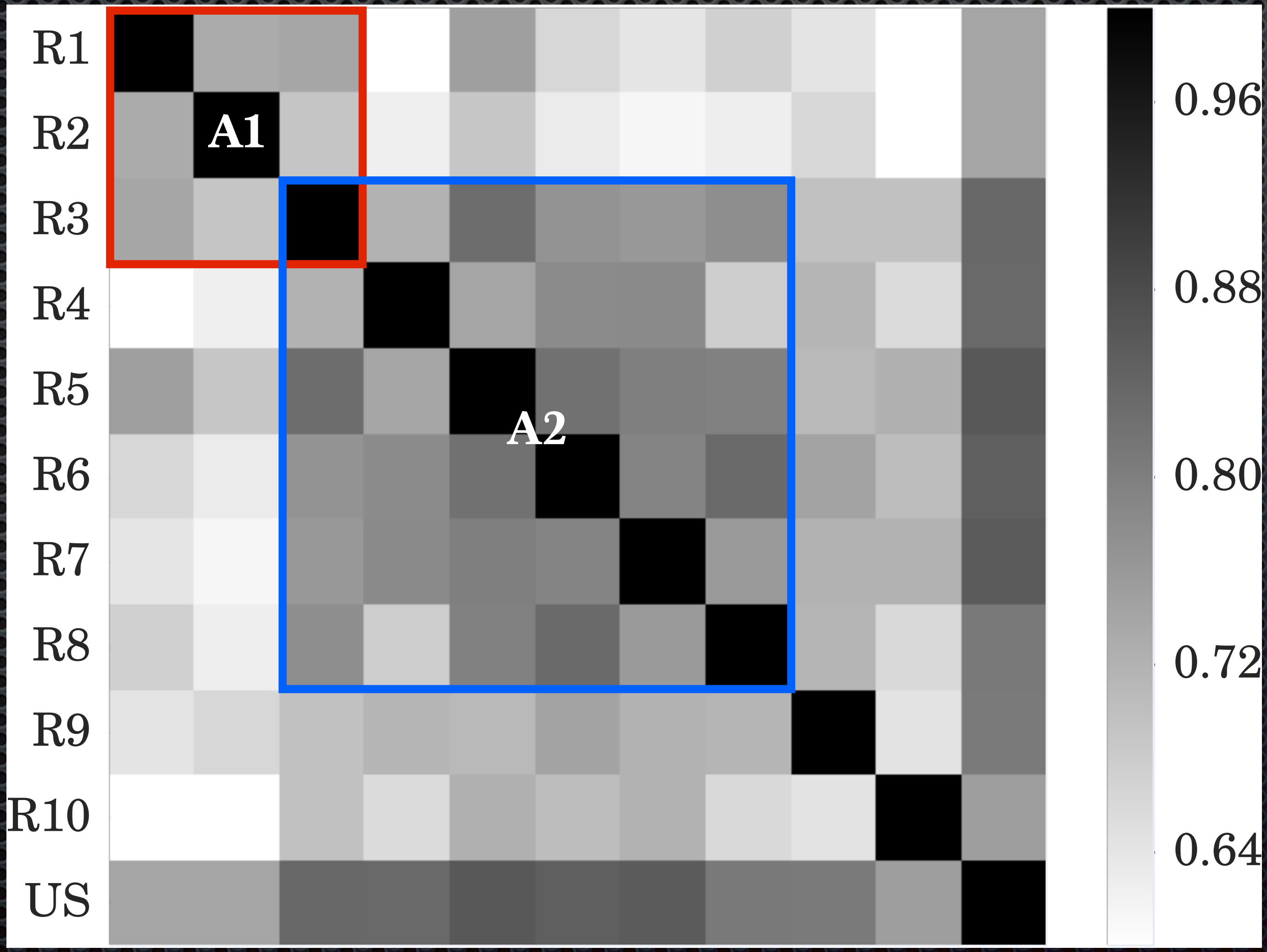
- Can ... us impr... repo...
- Split ... 2 regio... ap), and ...
- Trai... e hypo... ports



- Start **downsampling** the data from the 8 regions using **burst error sampling** (random data blocks removed) with rate γ (1 no sampling, 0.1 10% sample)

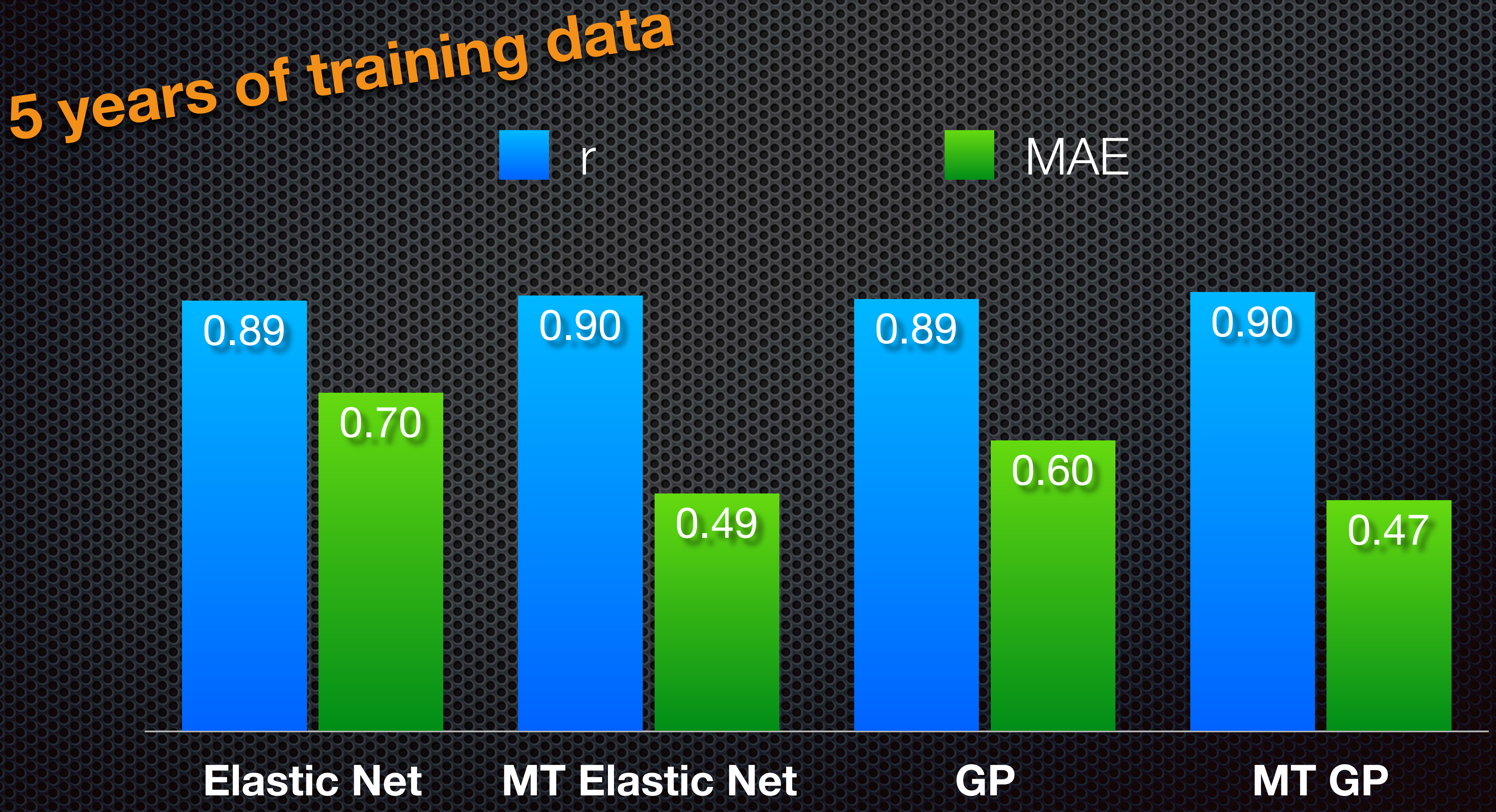
Multi-task learning GFT (4/5)

- Correlations between US regions induced by the covariance matrix of the MT GP model
- Multi-task learning model seems to be capturing existing geographical relations



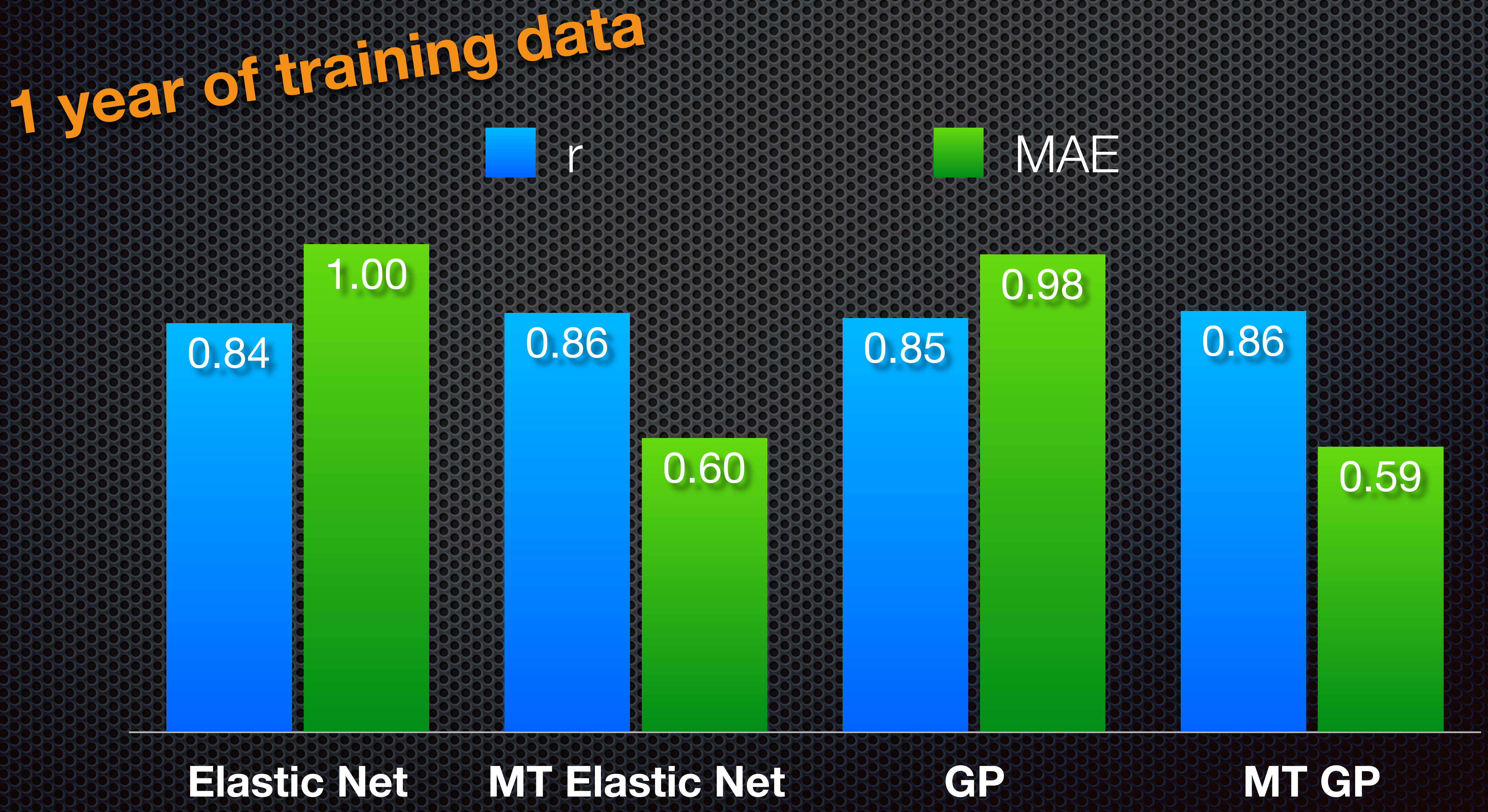
Multi-task learning GFT (5/5)

- Can multi-task learning across countries (US, England) help us improve the ILI model for England?



Multi-task learning GFT (5/5)

- Can multi-task learning across countries (US, England) help us improve the ILI model for England?



Conclusions

- Online (***user-generated***) data can help us **improve** our current understanding about **public health** matters
- The original **Google Flu Trends** was based on a good idea, but on very **limited modelling effort**, resulting to **major errors**
- Subsequent models improved the **statistical modelling** as well as the **semantic disambiguation** between possible features and delivered **better** / **more robust performance**
- **Multi-task learning** improves disease models further
- *Future direction*: **Models without strong supervision**

Acknowledgements

Collaborators: Andrew Miller, Bin Zou, Ingemar J. Cox

Industrial partners

— Microsoft Research (Elad Yom-Tov)
— Google

Public health organisations

— Public Health England
— Royal College of General Practitioners

Funding: EPSRC (“i-sense”)

Thank you.

Vasileios Lampos *(a.k.a. Bill)*

Computer Science
University College London

@lampos

References

GFT v.1 — Ginsberg et al. Detecting influenza epidemics using search engine query data. Nature 457, pp. 1012—1014 (2009).

GFT v.2 — Lampos, Miller, Crossan and Stefansen. Advances in nowcasting influenza-like illness rates using search query logs. Scientific Reports 5, 12760 (2015).

GFT v.3 — Lampos, Zou and Cox. Enhancing feature selection using word embeddings: The case of flu surveillance. WWW '17, pp. 695–704 (2017).

MTL — Zou, Lampos and Cox. Multi-task learning improves disease models from Web search. WWW '18, In Press (2018).