Gaussian Processes

Bruce Sharpe

My questions for any new technique

- what problems do they solve?
- disadvantages / what problems do they not solve?
- can I get some Python code working?
- what is the motivation for how they work?
- what is the "hello world" example?
- what are the definitions? assumptions? theorems?
- how do you know when to use them?
- are there famous examples where they were used?
- what is the motivation for how they work?
- has it won any Kaggle competitions?
- is there a YouTube video so I don't have to read the paper?
- can I even understand the Wikipedia page?

My questions for any new technique

- **what problems do they solve?**
- disadvantages / what problems do they not solve?
- **can I get some Python code working?**
- what is the motivation for how they work?
- **what is the "hello world" example?**
- what are the definitions? assumptions? theorems?
- how do you know when to use them?
- are there famous examples where they were used?
- what is the motivation for how they work?
- **has it won any Kaggle competitions?**
- is there a YouTube video so I don't have to read the paper?
- **can I even understand the Wikipedia page?**

Approaches to GPs

- State space <= most treatments start here
- Function space <= I prefer this

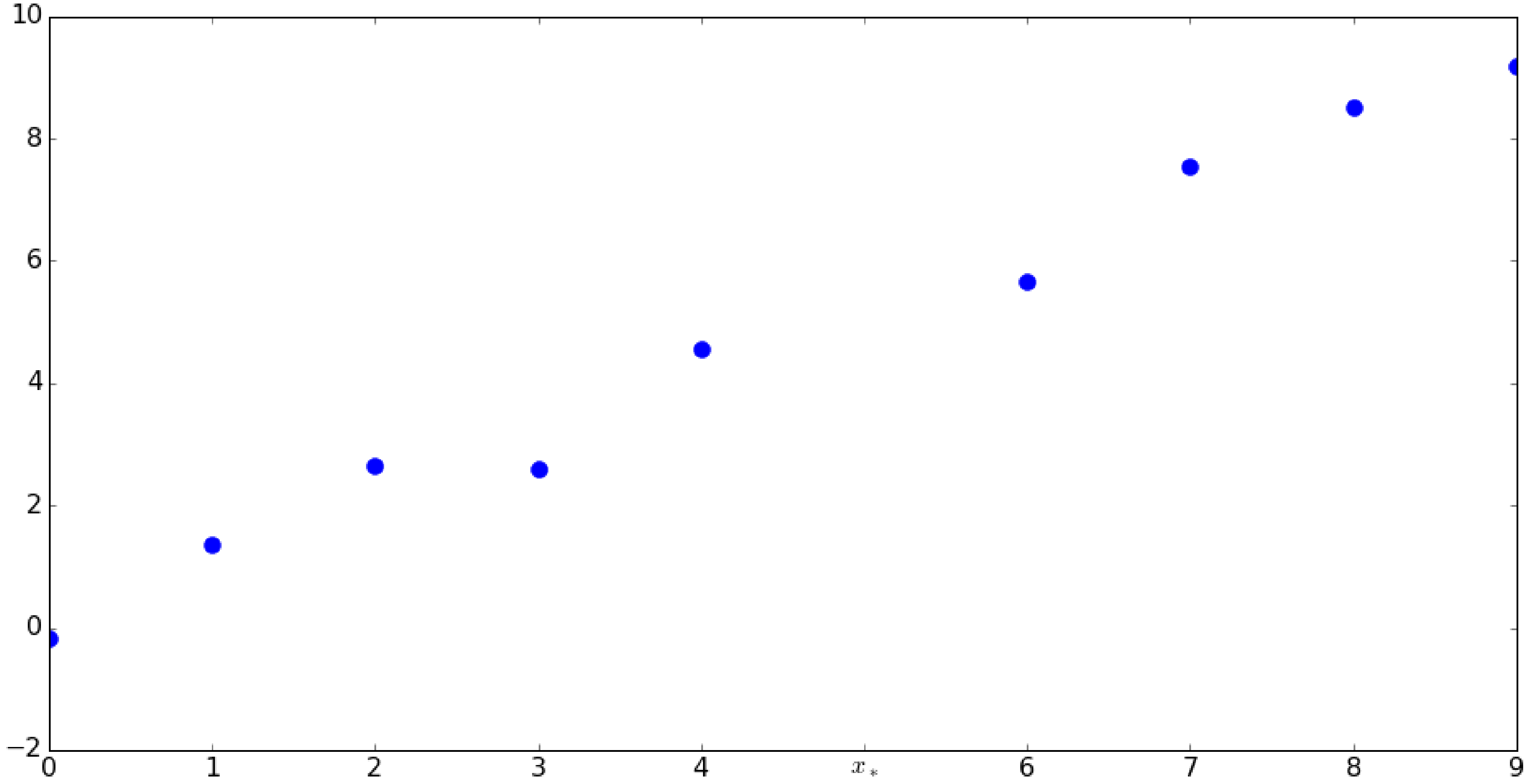
What problems do they solve?

- Regression
- Classification
- Optimization

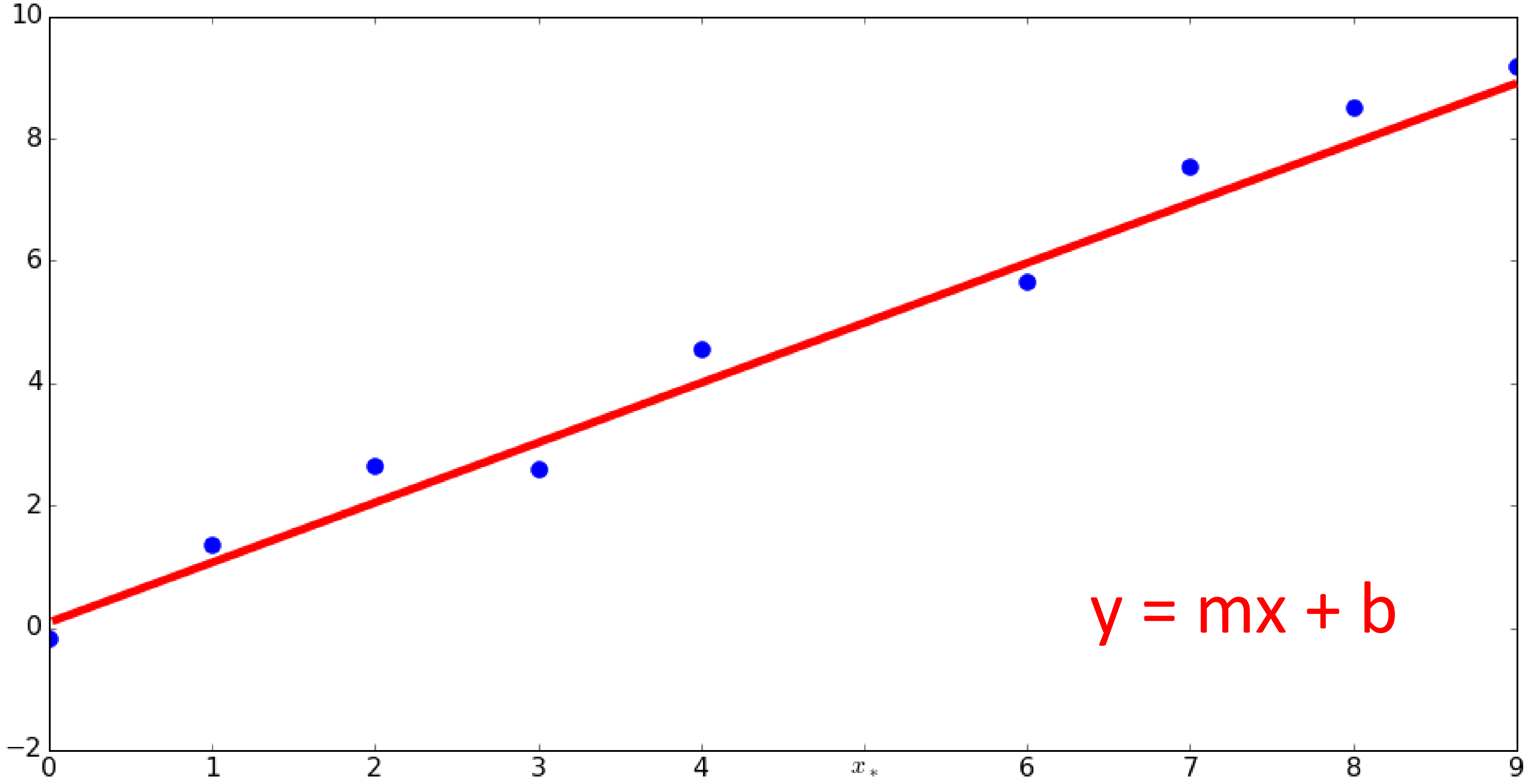
What problems do they solve?

- **Regression (especially things like time series)**
- Classification
- Optimization

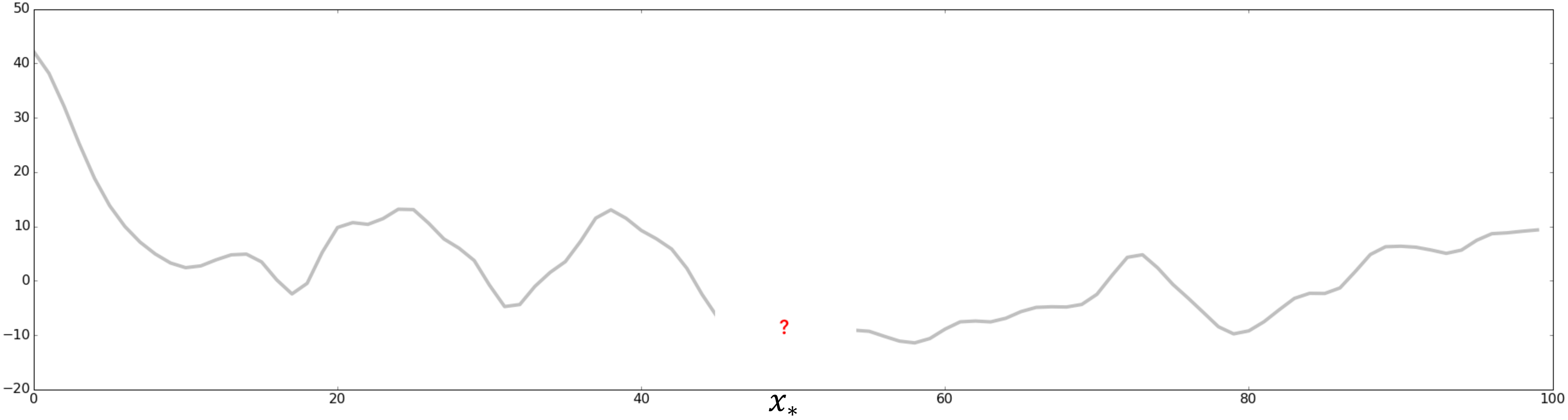
What regression problems do they solve?



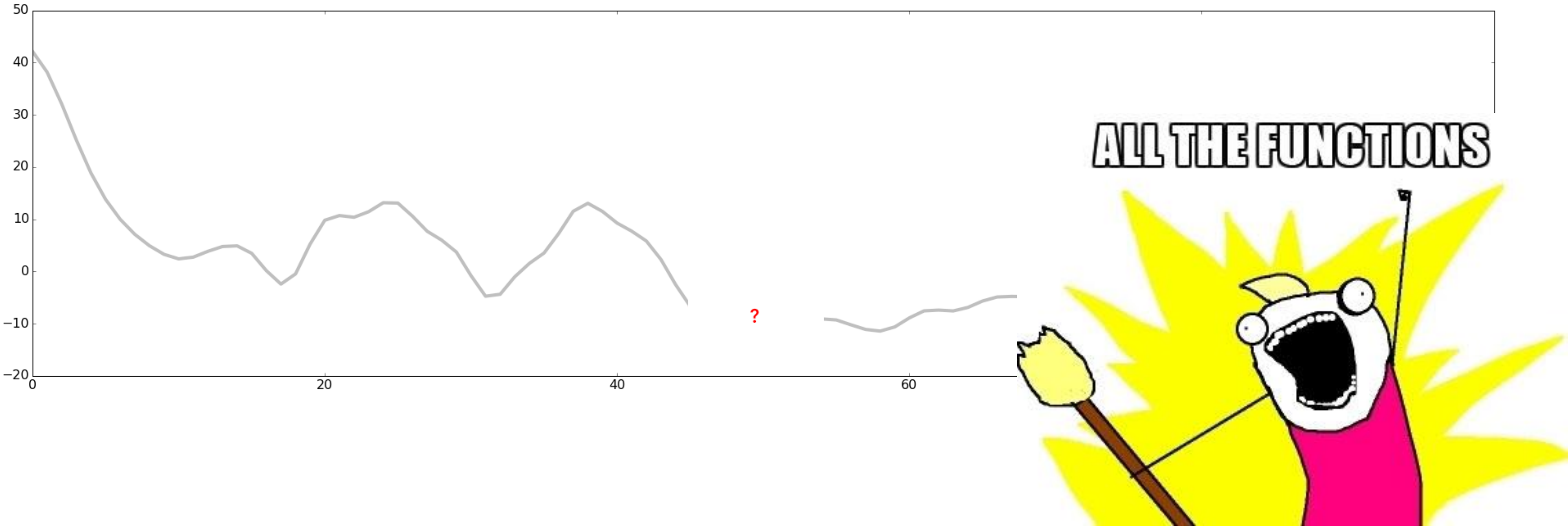
What regression problems do they solve?



What regression problems do they solve?



What regression problems do they solve?



Let’s narrow down the possibilities

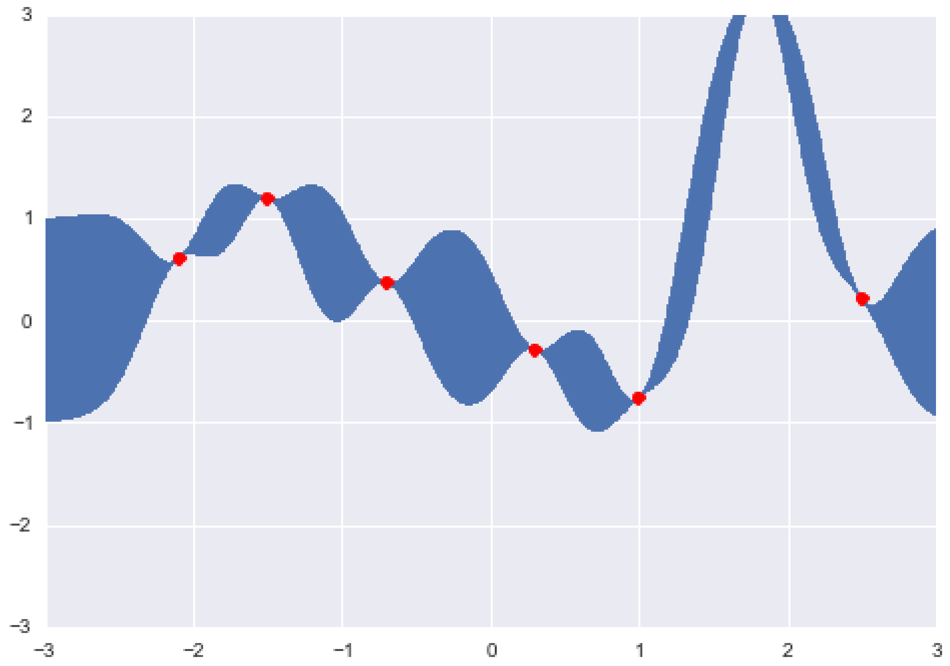
- #1: A class of functions
  - well, random field, really: for every x, $f(x)$ is a random variable
- #2: $f(x)$ is normally distributed
- #3: $\{f(x_i), f(x_j)\}$ is jointly Gaussian
- Every finite collection $\{f(x_i)\}_{i=1\ldots N}$ is multivariate normal
- A GP is completely characterized by its mean and covariance
  - $X \sim \mathrm{GP}(m, k)$
- Example: $k(x_i, x_j) = \sigma^2 e^{-|x_i - x_j|^2/2l^2}$ (squared exponential)

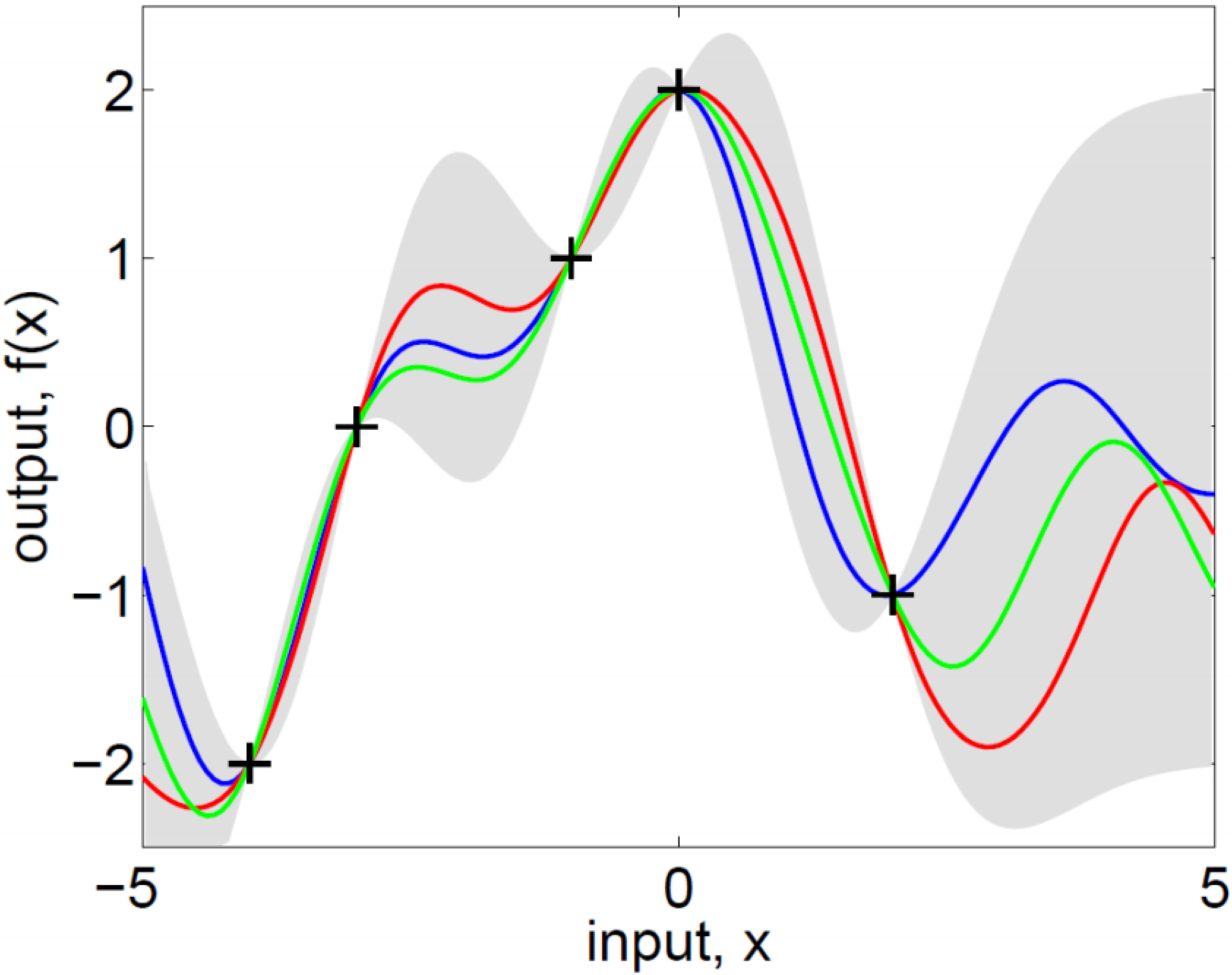
Slightly less mysterious (?)

- A GP is completely characterized by its mean and covariance
  - $f(x) \sim \mathrm{GP}(m, k)$
- The covariance k is called the *kernel*
  - main focus of interest
- Example: $k(x_i, x_j) = \sigma^2 e^{-\left|x_i - x_j\right|^2/2l^2}$ (squared exponential)

Back to regression (prediction)

- Compute $f(x_*)$ given some observations (x_i, y_i)
- We can calculate the posterior distribution (Bayes)
- Given the magic of normal distributions, can get analytical results





output, f(x)
2
1
0
−1
−2
−5
0
5
input, x

Parametric vs non-parametric

- Parametric: $y = mx + b$
- GPs are non-parametric
  - there are zillions of parameters, we just don't specify them
- GPs have hyperparameters (like σ and l)
  - $k(x_i, x_j) = \sigma^2 e^{-|x_i - x_j|^2/2l^2}$
  - you optimize the hyperparameters
- Usually you can't infer much from non-parametric models
  - but you can (sometimes) infer something from the hyperparameters

The paper

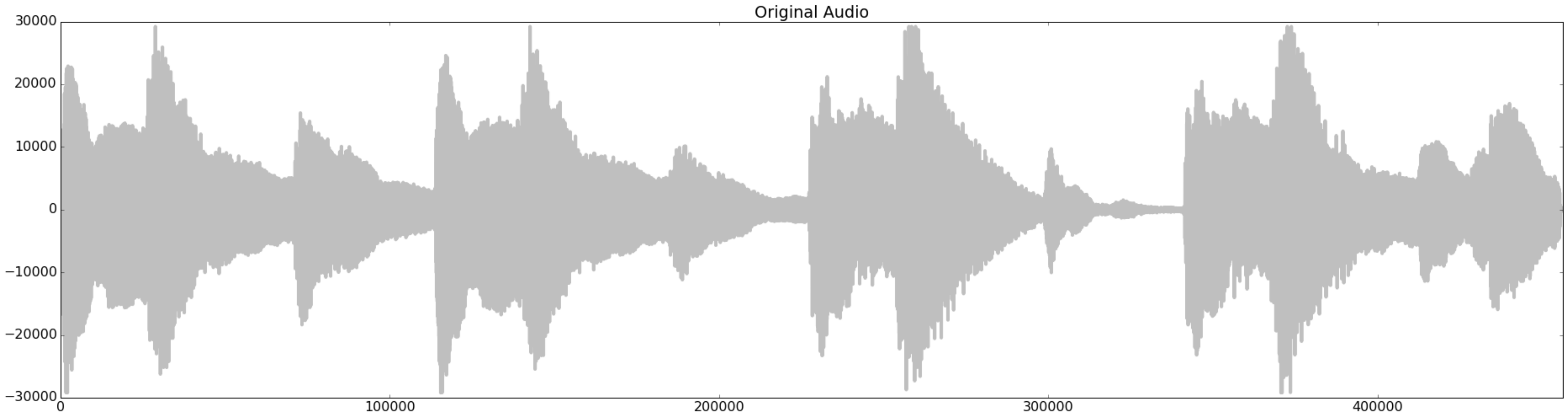
- “Principled”? Hmm…
- Changepoints: cool!
- Examples: pretty good!

Questions

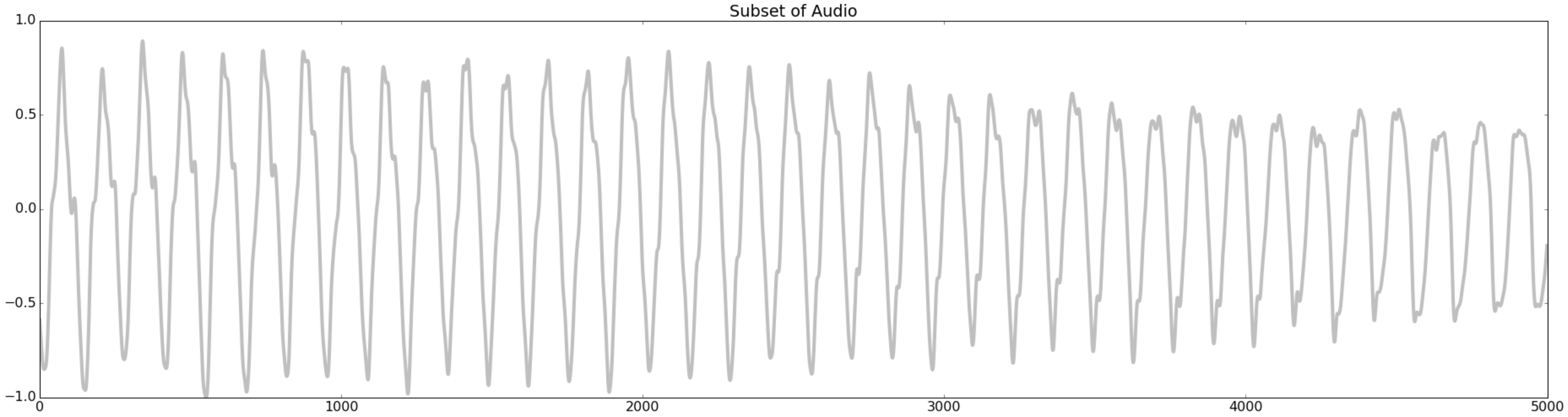
- Can you use an experimentally determined covariance rather than an analytical form?
- GP for big data? (sparse approximations?)

My big fat dumb GP

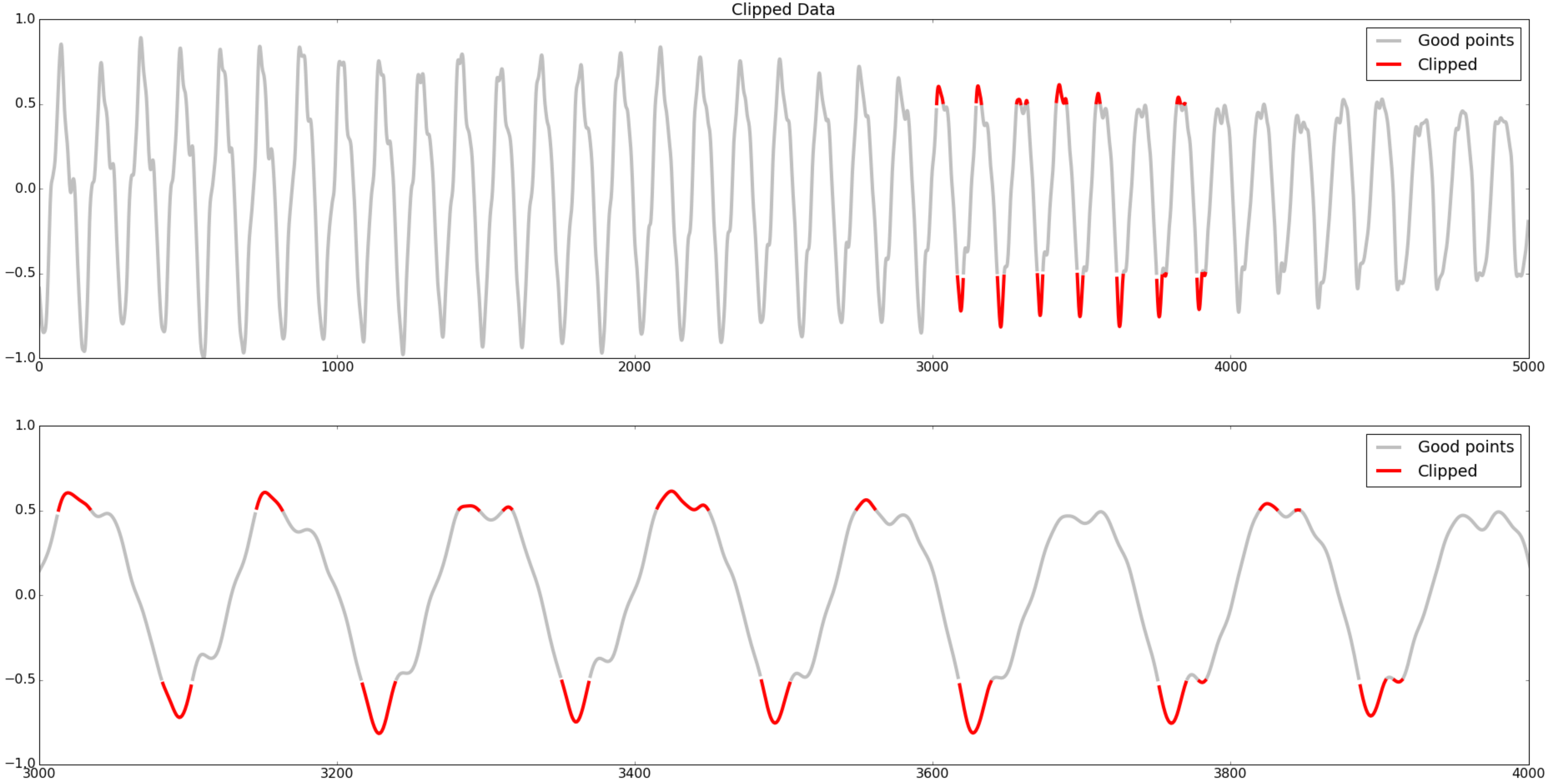
Audio


Original Audio
30000
20000
10000
0
−10000
−20000
−30000
0
100000
200000
300000
400000

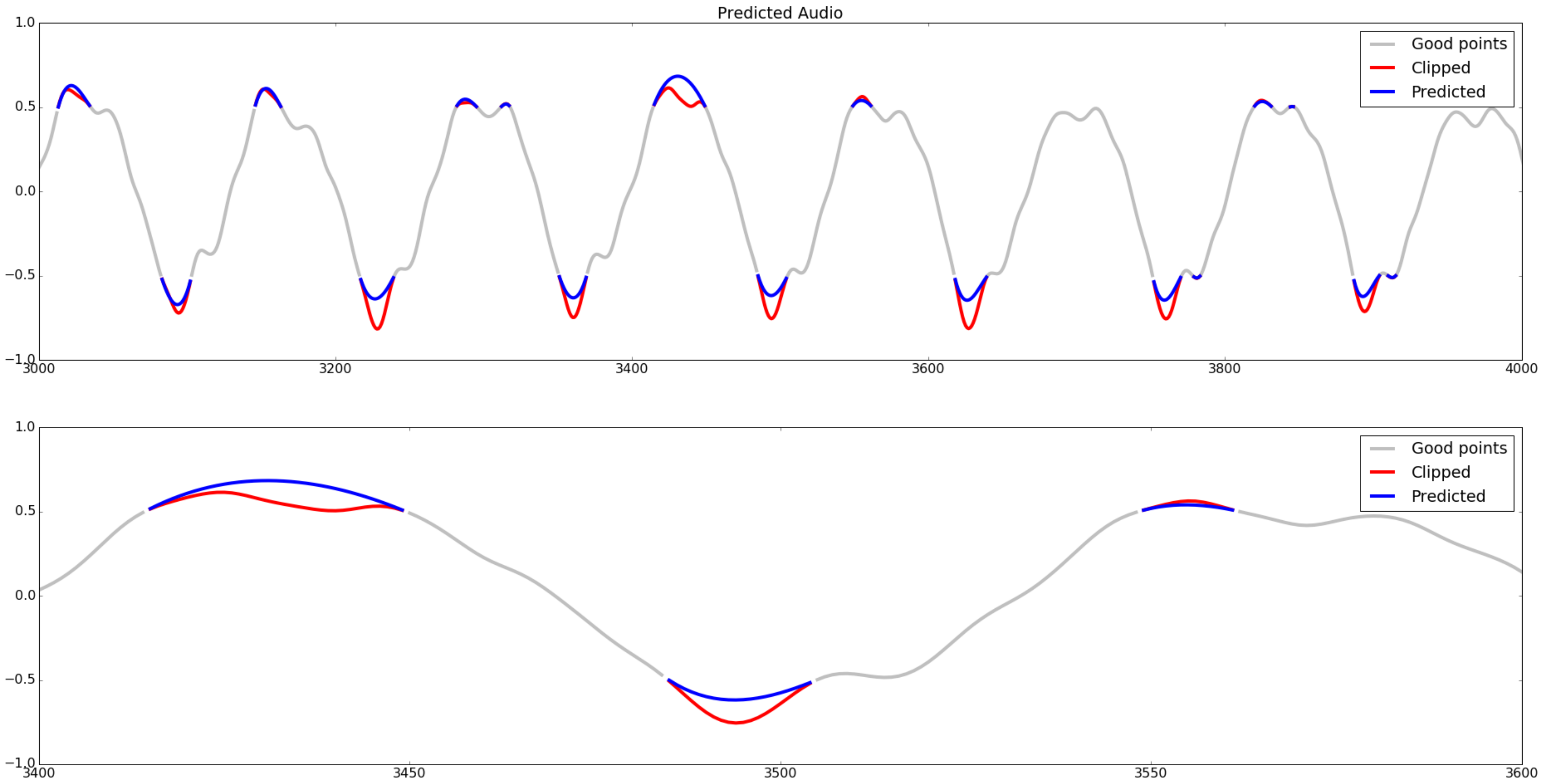
Ahem, let’s not blow up my memory



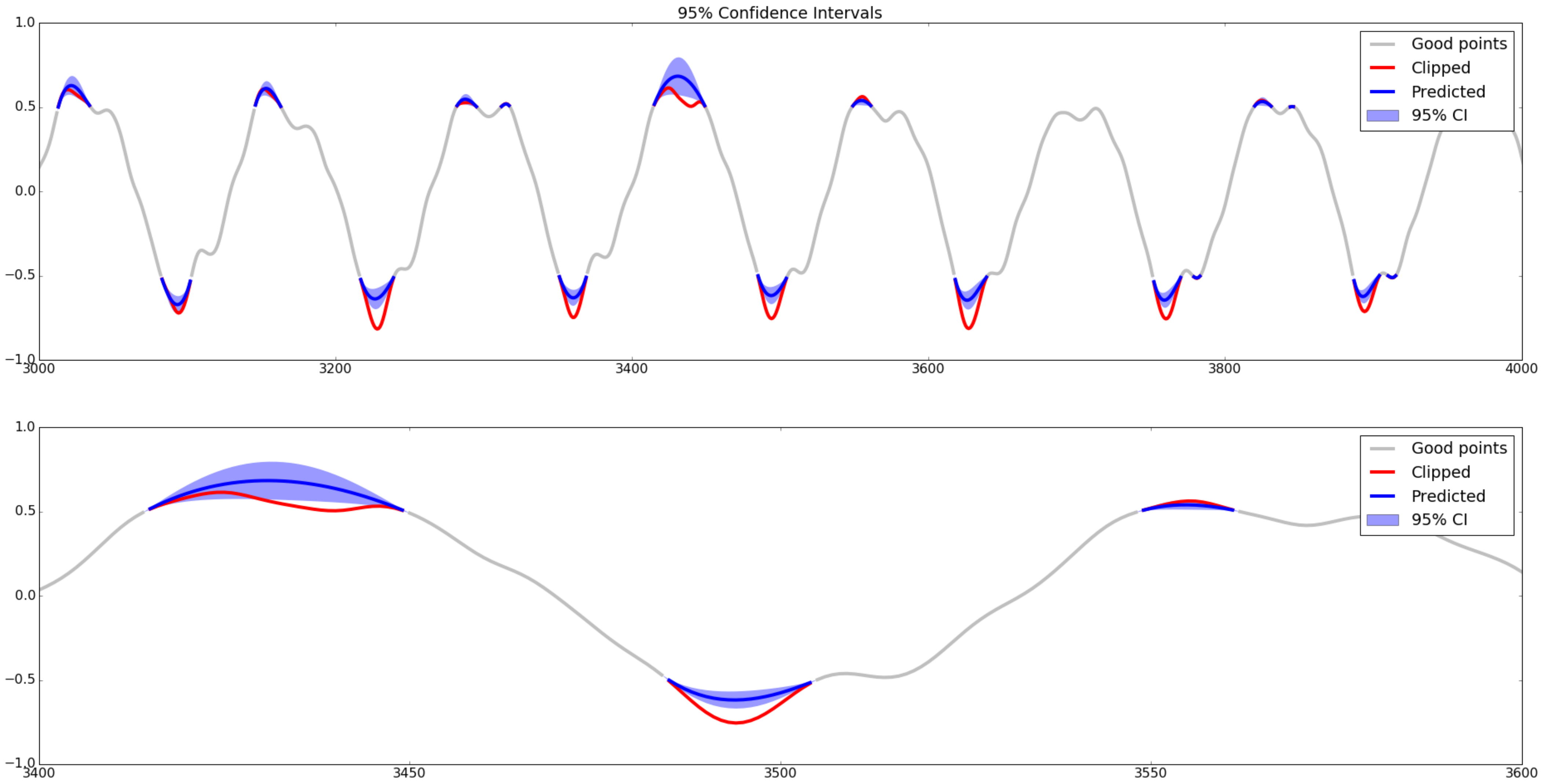
Clipping distortion



Fill in the gaps

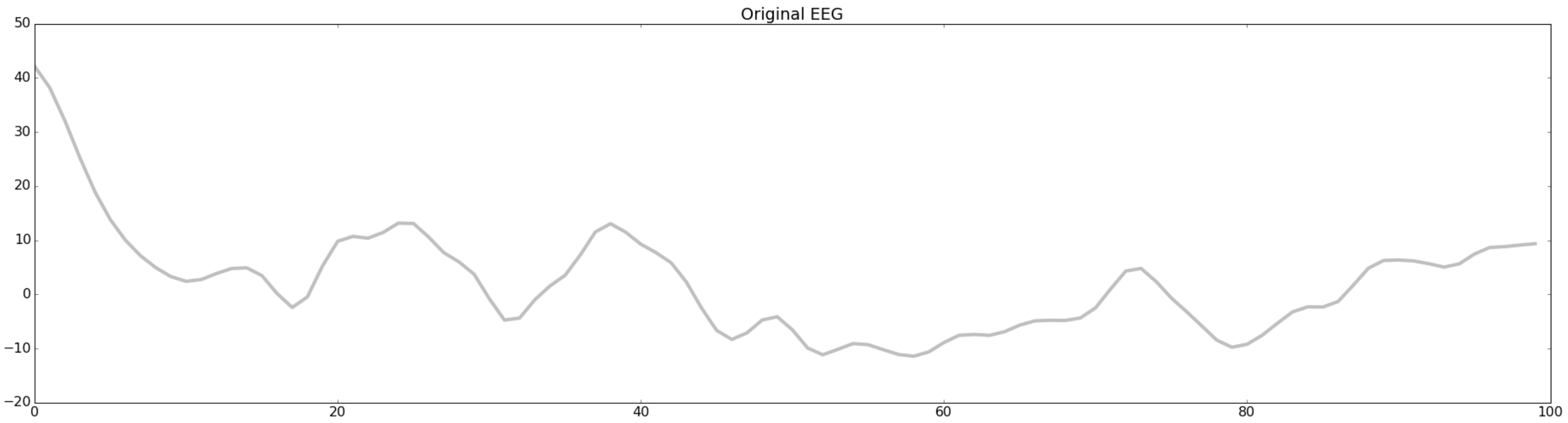


And now, with confidence

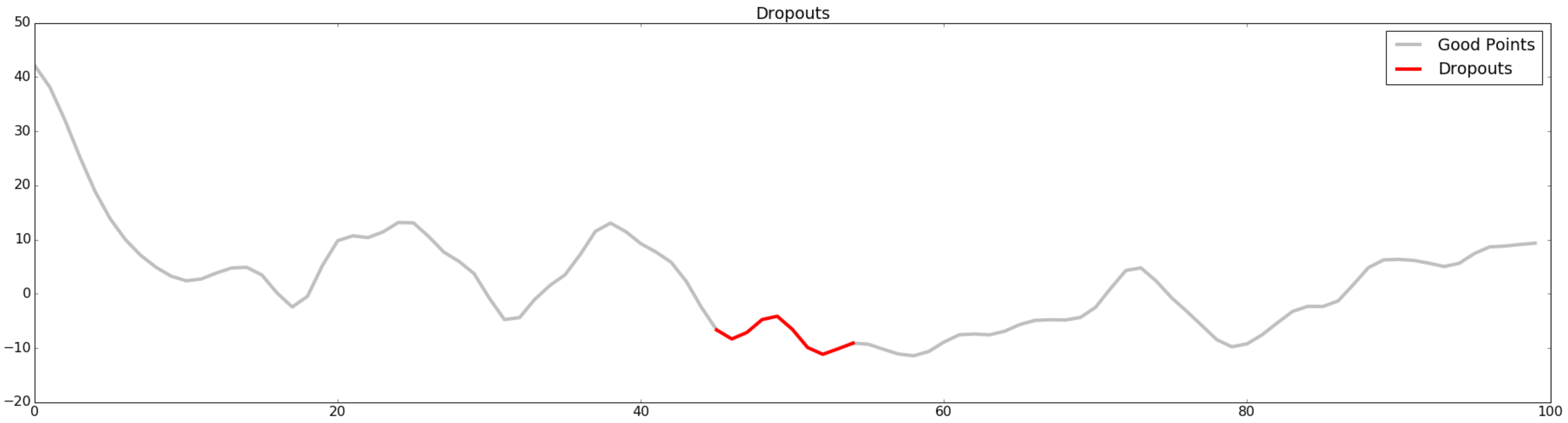


My big fat dumb GP #2

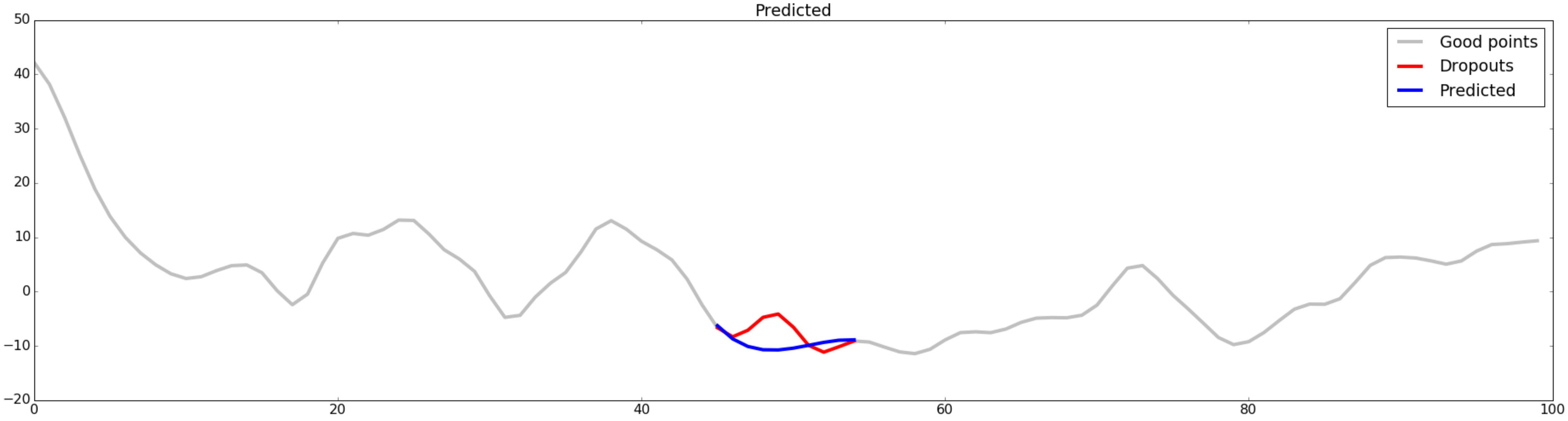
EEG data



EEG data with dropouts



EEG data with predictions



EEG data with confidence