**Motivation**
- **Pointillism**: point observations, group patterns.
- **Active search**: filter as many positives as possible for later processing.
- **Flexibility**: allow for arbitrary definitions of the desired pattern, in the form of a functional classifier.

**Applications**
- Environmental monitoring: autonomous boats searching a pond for polluted areas
- Astronomy: choosing where to point a telescope to find interesting objects
- Polling: carefully surveying to find electoral races that need attention

**Problem Setup**
1. **Select locations to observe function values**
   - \( f \sim \mathcal{GP}(\mu(.), \sigma^2) \)
   - \( f = \{ f(x_1), f(x_2), \ldots, f(x_n) \} \)

2. **Collect reward for region matches**
   - \( r(D) = \sum_g \mathbbm{1}\{ \mathbb{E}[h_g(f) | D] > \theta \} \)

**Finding Vortices with a Black-Box Classifier**
We want to find vortices in a 2d map of fluid flow.

The velocity dataset; each arrow represents the mean of a 2x2 square. This run was initialized with the points at the green circles and selected the ones at the red circles.

Trained a 2-layer neural network on a small training set:

Train set: Mean and standard error of recall for matching regions, over 15 runs. True labels are determined by using the classifier on the full velocity dataset.

**Algorithm**
Choose point to greedily maximize expected reward
- \( \max_{x_*} \mathbb{E} \sum_{g \in G} [r(g, D_\ast) | x_\ast, D_\ast] \)

Estimate expected reward with Monte Carlo:
- Sample observation: \( z_* \sim \mathcal{D}_t \sim \mathbb{N}(\mu_{f|\mathcal{D}_t}(x_*), \kappa_{f|\mathcal{D}_t}(x_*, x_*) + \sigma^2) \)

Sample enough \( f \) to get \( h_g(f) \)
- \( r(D_t \cup \{x_\ast, z_\ast\}) = \sum_{g \in G} \mathbb{E}[h_g(f) | D_t, x_\ast, z_\ast] > \theta \)

Analytical form:
- If \( h_g(f) = \Phi(L_g f + b) \) where \( \Phi \) is normal cdf and \( L_g \) is linear, e.g., \( L_g f = \int_{x \in x_g} w(x)^T f(x) dx \)
  - then \( \mathbb{E}[r_g(D_t) | x_\ast, D_t] \) has a closed form.

**Intuition**
If linear classifiers and regions are independent:
- If two regions have the same marginal probability, picks the one whose variance can be reduced more with one point.
- If two regions’ variances can be equally well reduced, picks the one with higher marginal probability.
- In a given region, picks point most correlated to region’s label.

**Related Work**
Most Bayesian optimization:
- Models functions with GPs
- Usually maximization of observable point values

Active search (e.g., Garnett et al., IJCAI 2012)
- Usually assumes that labels are directly observable and correspond to single points.

Active Area Search (Ma et al., AISTATS 2014)
- Similar setup and algorithm, but can only detect thresholds on mean of a region.
- APPS generalizes to any pattern.

Level set estimation (Gotsman et al., IJCAI ‘13; Low et al., AMAS ‘12)
- Actively finds a particular level set in a function.
- Related to AAS; can’t model arbitrary patterns.