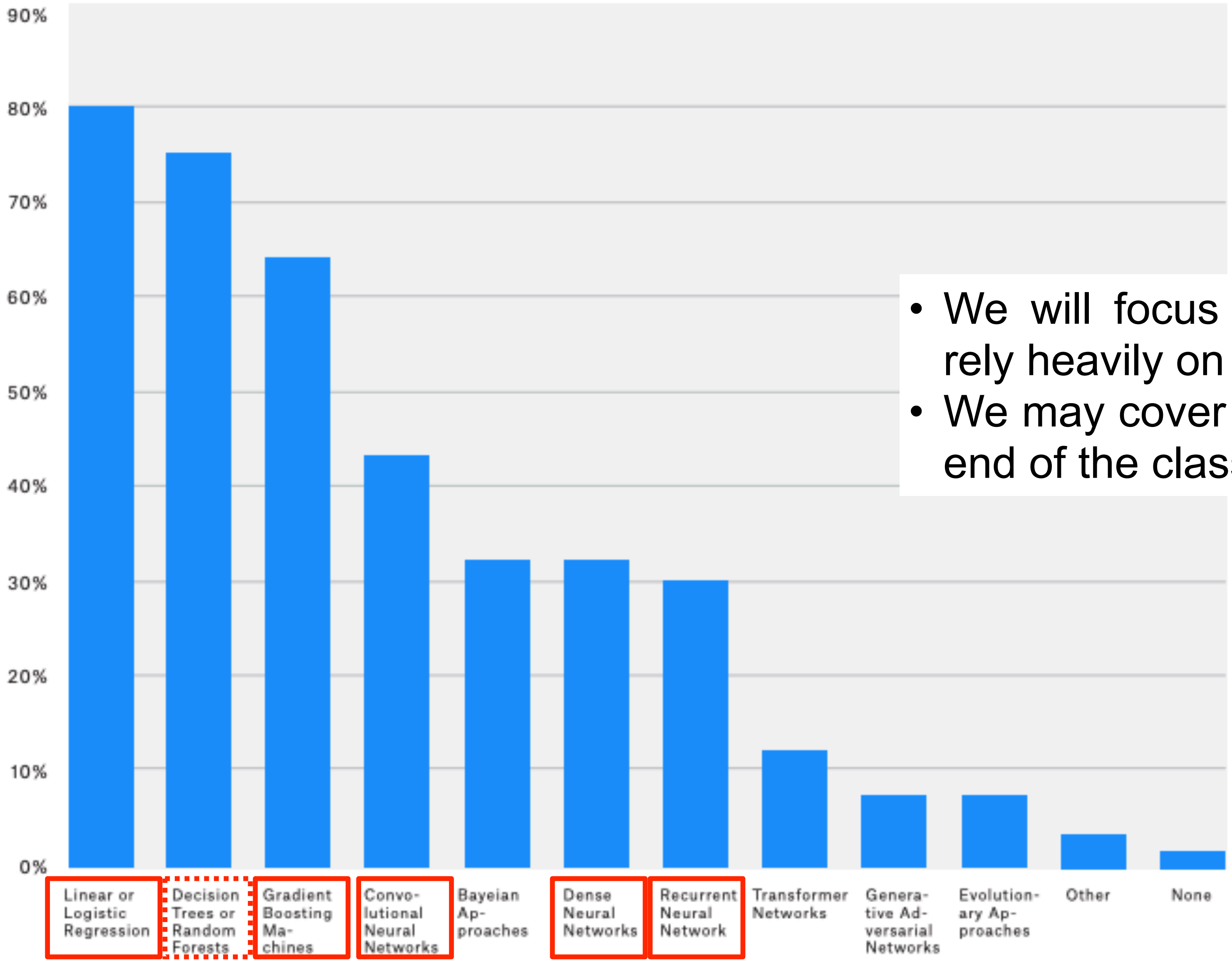
Decision Trees and Forests

Pascal Fua
IC-CVLab

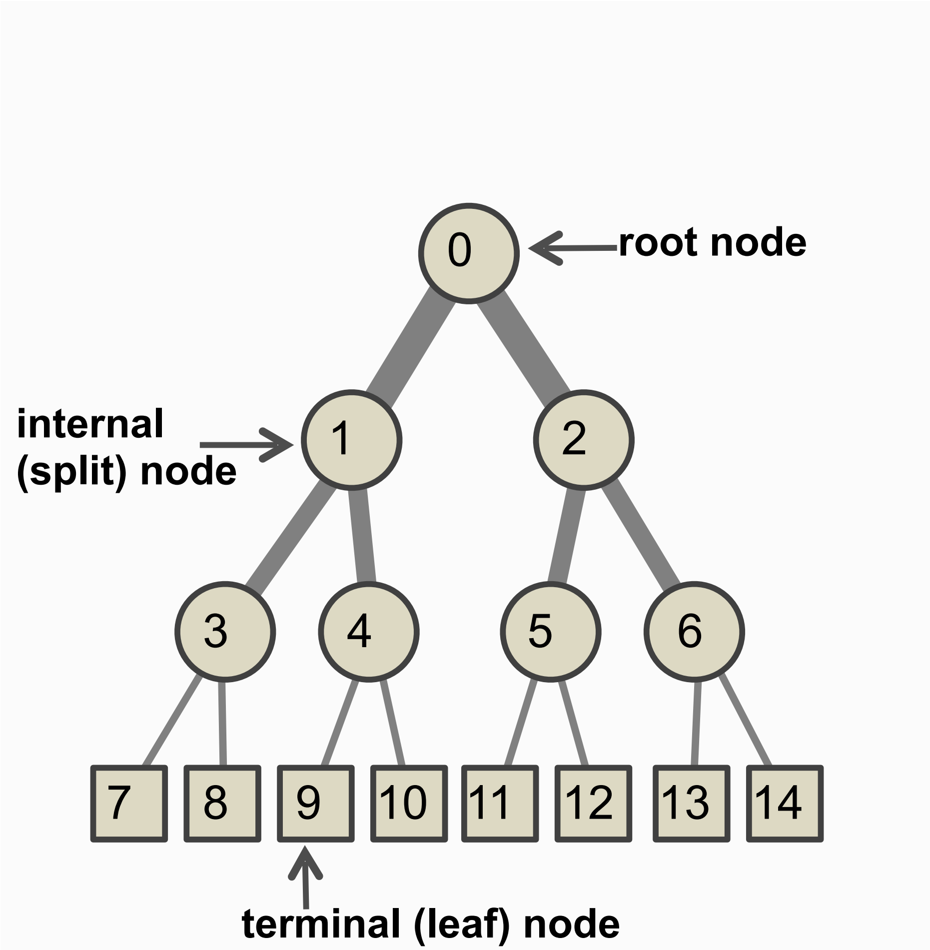
Kaggle Survey (2019)



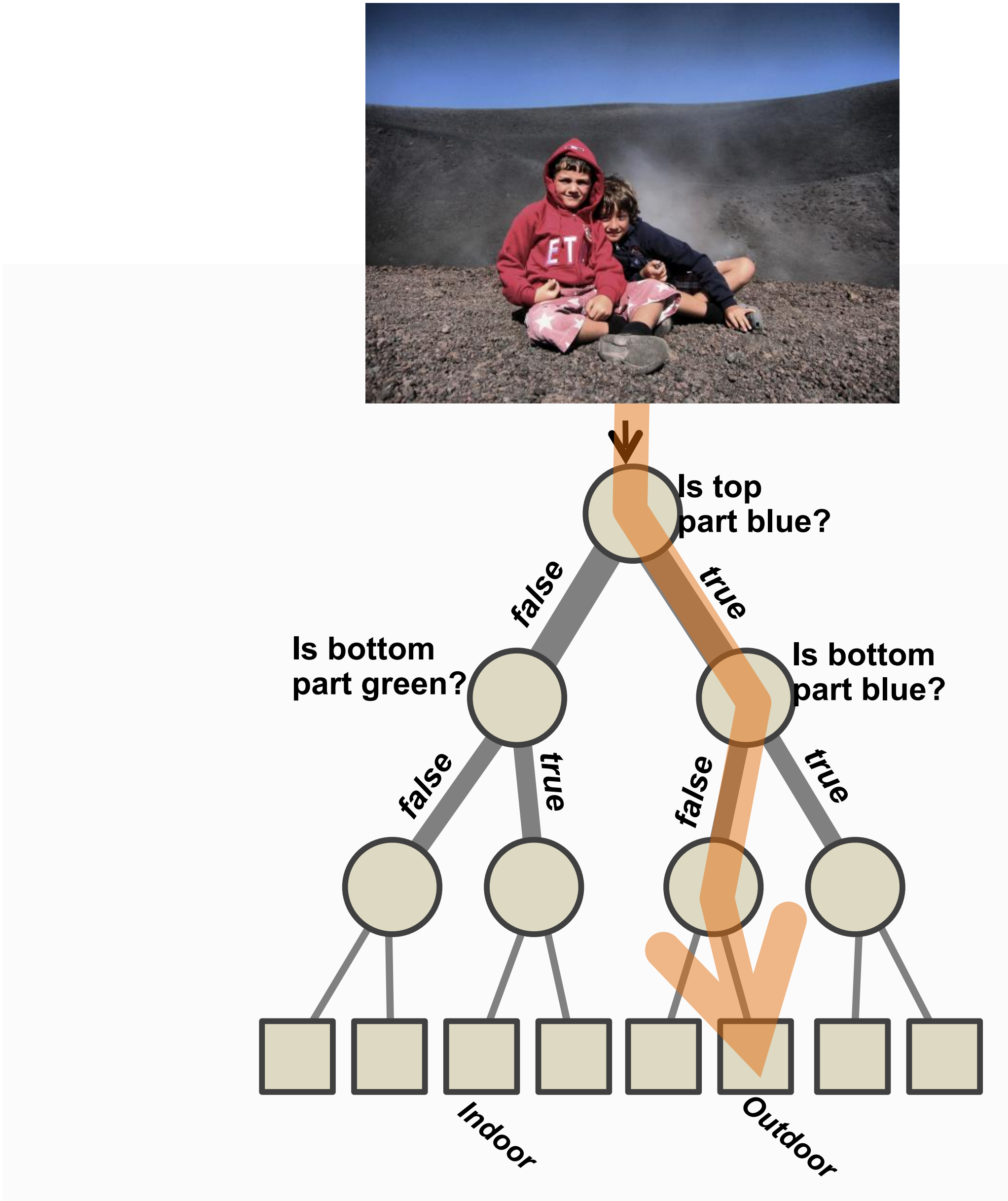
- We will focus on methods that do not rely heavily on probabilities.
- We may cover some of the others at the end of the class time permitting.

What data science methods do you use at work?

Decision Tree

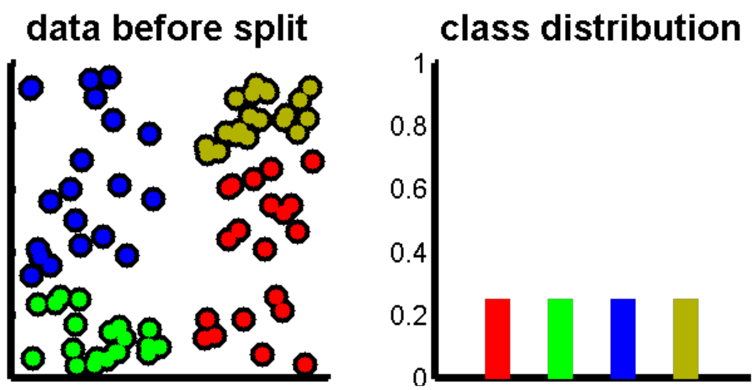


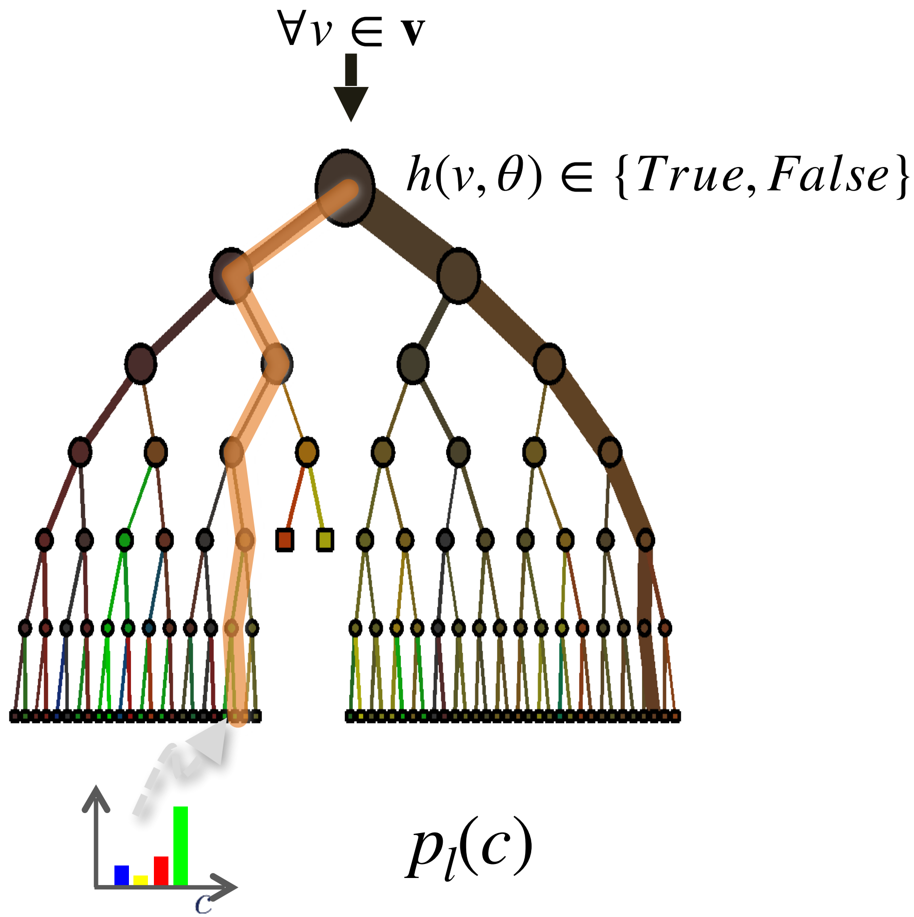
Tree structure



A decision tree

Training

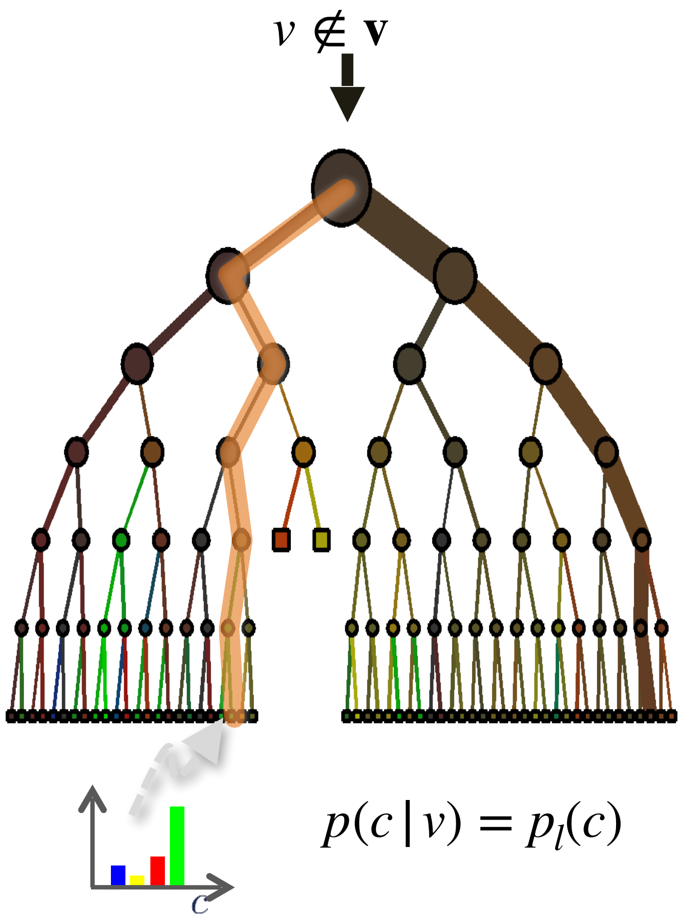




- Training set **v**
- To each sample v is assigned a class c.

- Compute $p_l(c)$, the proportion of samples in each class that lands in leaf l.

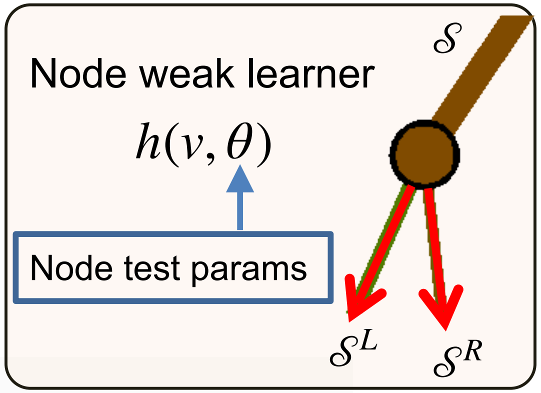
Testing



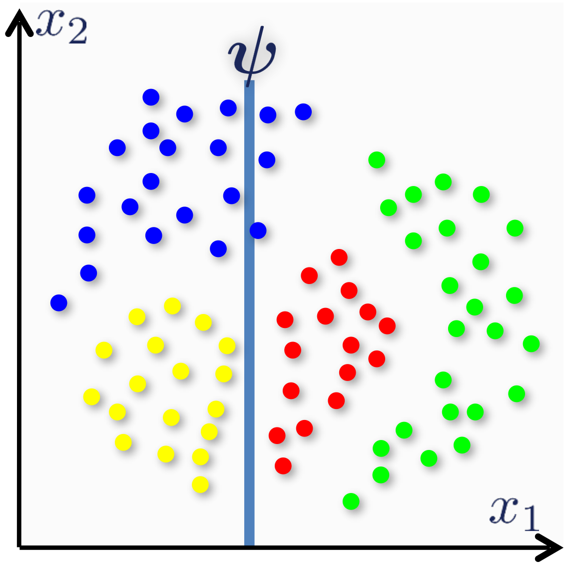
$$p(c \mid v) = p_l(c)$$

- Let us assume that v falls into leaf l.
- We take the probability of belonging to class c, $p(c \mid v)$, to be $p_l(c)$ if it lands in leaf l.

Weak Learners



Weak learner examples

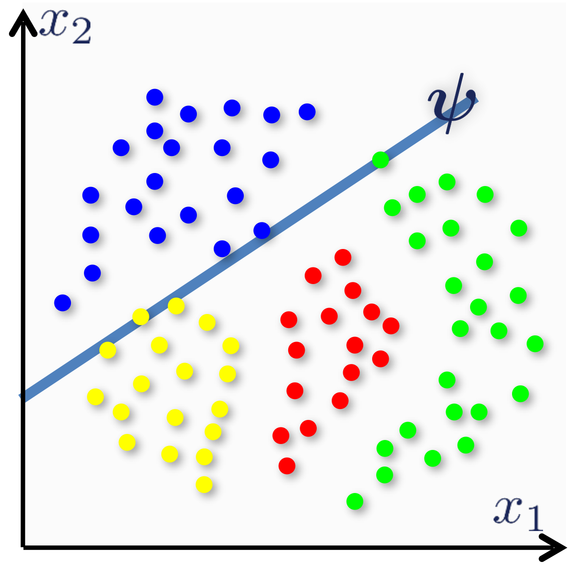


Weak learner: axis aligned.

$$h(\mathbf{v}, \boldsymbol{\theta}) = [\tau_1 > \boldsymbol{\phi}(\mathbf{v}) \cdot \boldsymbol{\psi} > \tau_2]$$

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$

With $\boldsymbol{\psi} = (1 \ 0 \ \psi_3)$ or $\boldsymbol{\psi} = (0 \ 1 \ \psi_3)$

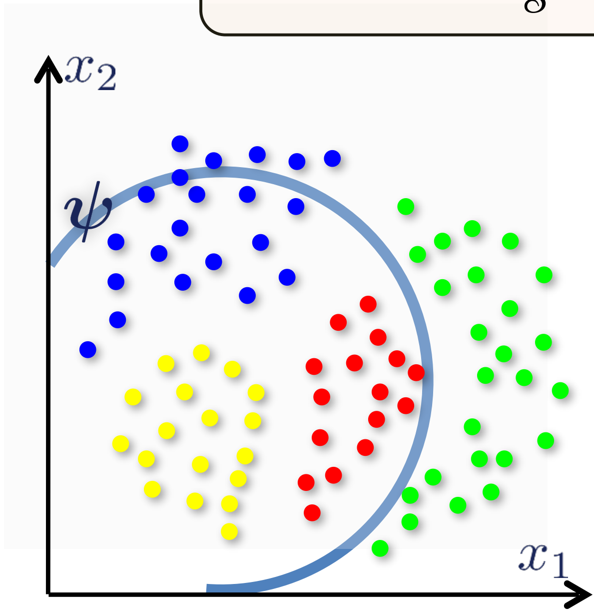


Weak learner: oriented line.

$$h(\mathbf{v}, \boldsymbol{\theta}) = [\tau_1 > \boldsymbol{\phi}(\mathbf{v}) \cdot \boldsymbol{\psi} > \tau_2]$$

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^3$ a generic line in homog. coordinates.



Weak learner: conic section.

$$h(\mathbf{v}, \boldsymbol{\theta}) = \left[\tau_1 > \boldsymbol{\phi}^\top(\mathbf{v}) \ \boldsymbol{\psi} \ \boldsymbol{\phi}(\mathbf{v}) > \tau_2\right]$$

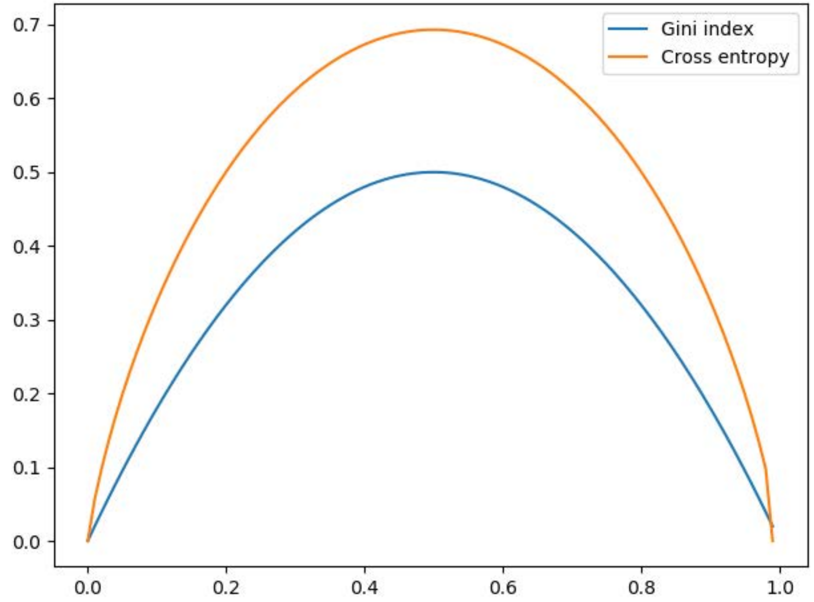
Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^{3\times 3}$ a matrix representing a conic.

Entropy and Gini Index

Let p^k be the proportion of data points in $\mathcal{S}$ that are assigned to class k. We can define

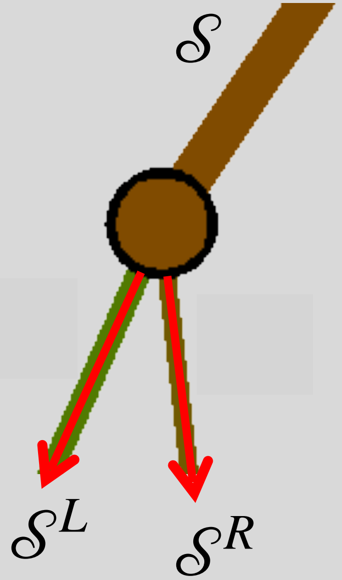
- the Gini index $Q(\mathcal{S}) = \sum_{k=1}^{K} p^k(1-p^k)$,
- the entropy $Q(\mathcal{S}) = -\sum_{k=1}^{K} p^k \ln p^k$.
- They both vanish when $\exists k \ , p^k = 1$.
- They are maximized when all p^k are equal.



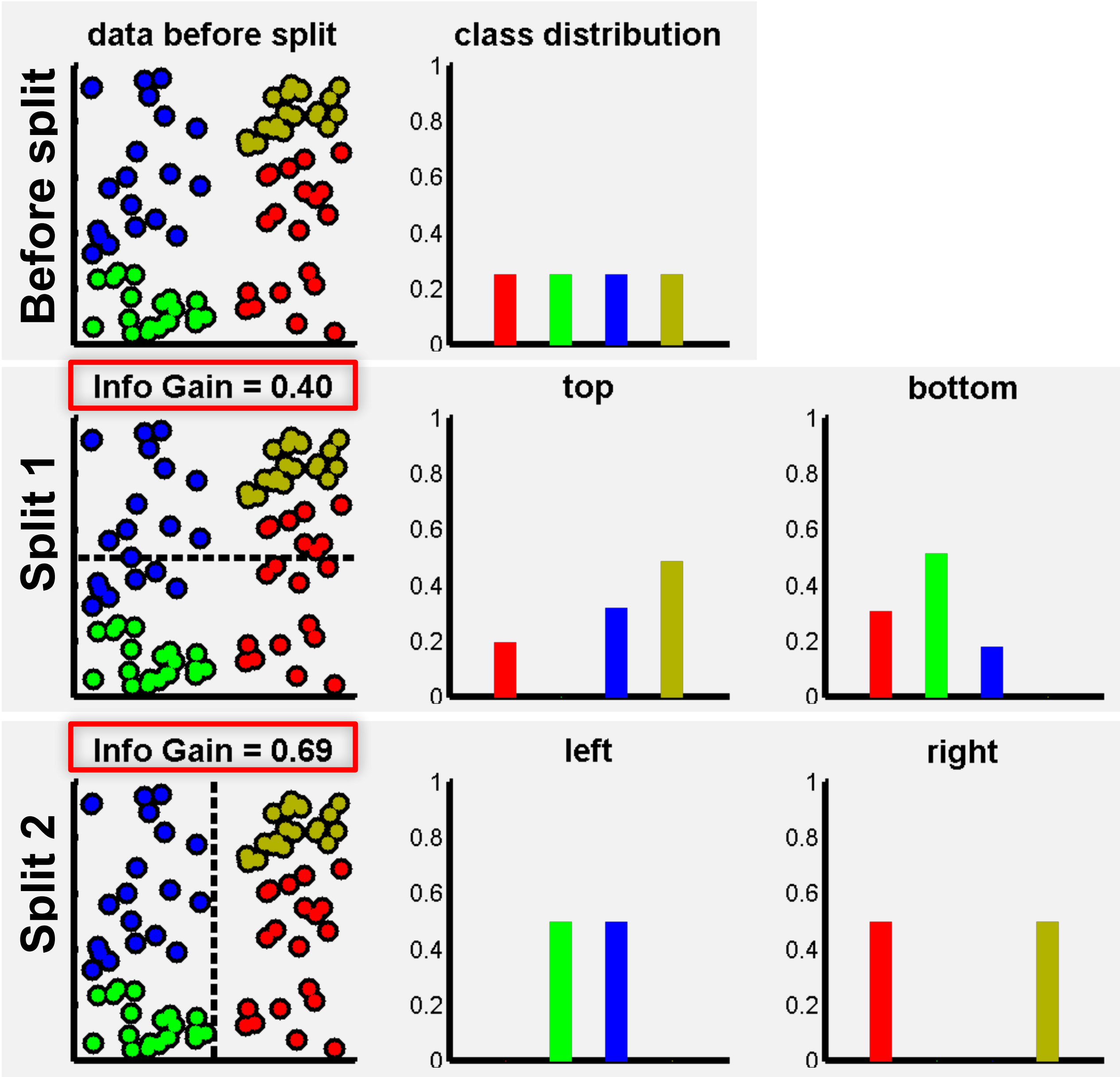
Two classes case.

—> Minimizing these measures favors leaves in which a large fraction of samples belong to the same class.

Maximizing Information Gain

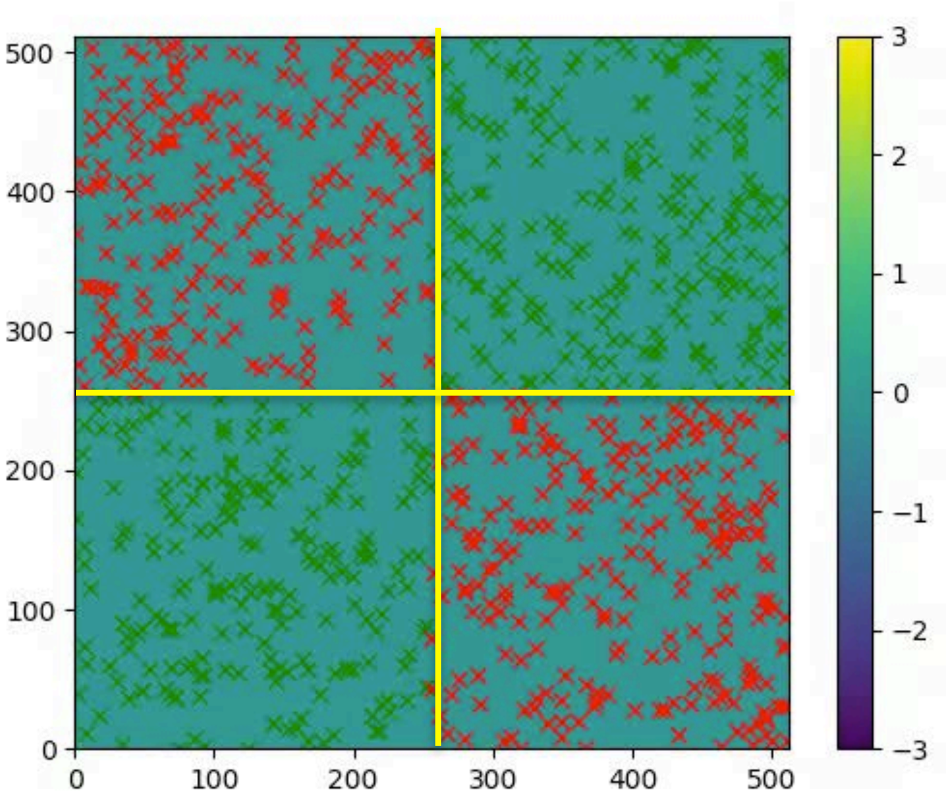


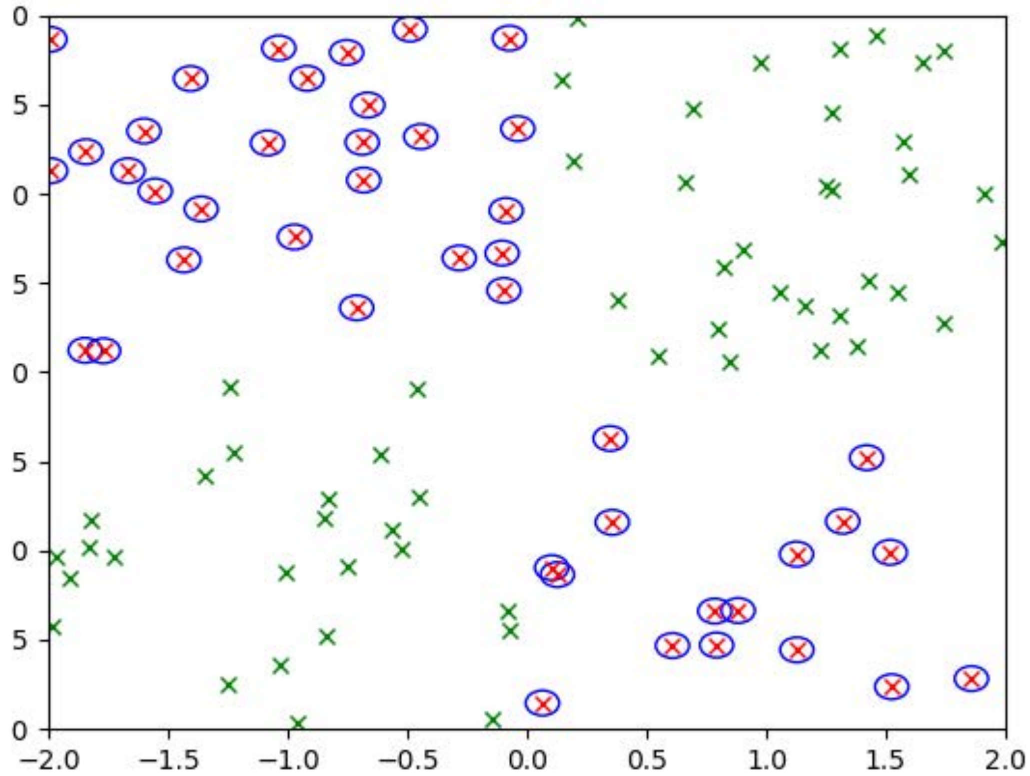
$$Q(\mathcal{S}) - \sum_{\tau\in L,R} \frac{|\mathcal{S}^{\tau}|}{|\mathcal{S}|} Q(S^{\tau})$$



At each node, pick the weak learner that delivers the highest information gain.

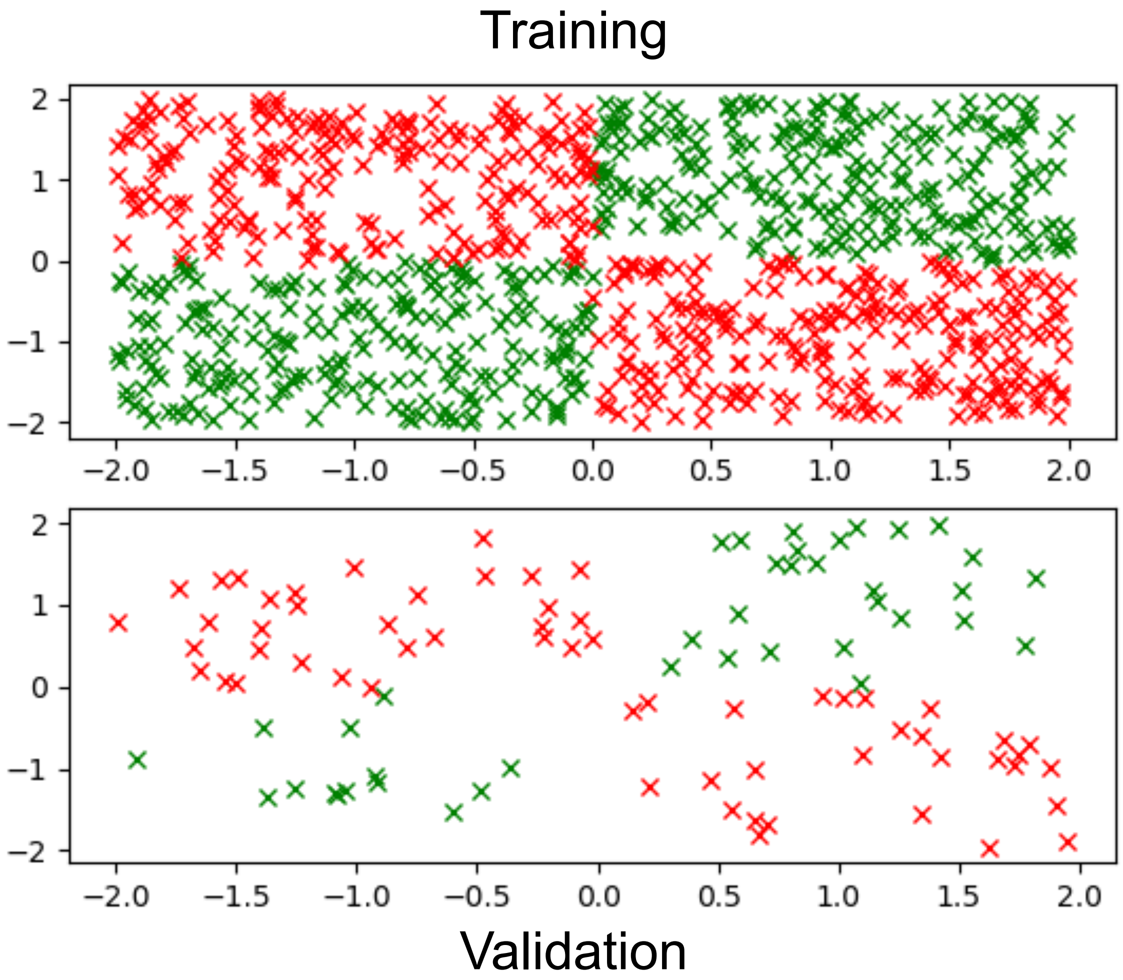
Problematic for AdaBoost ….

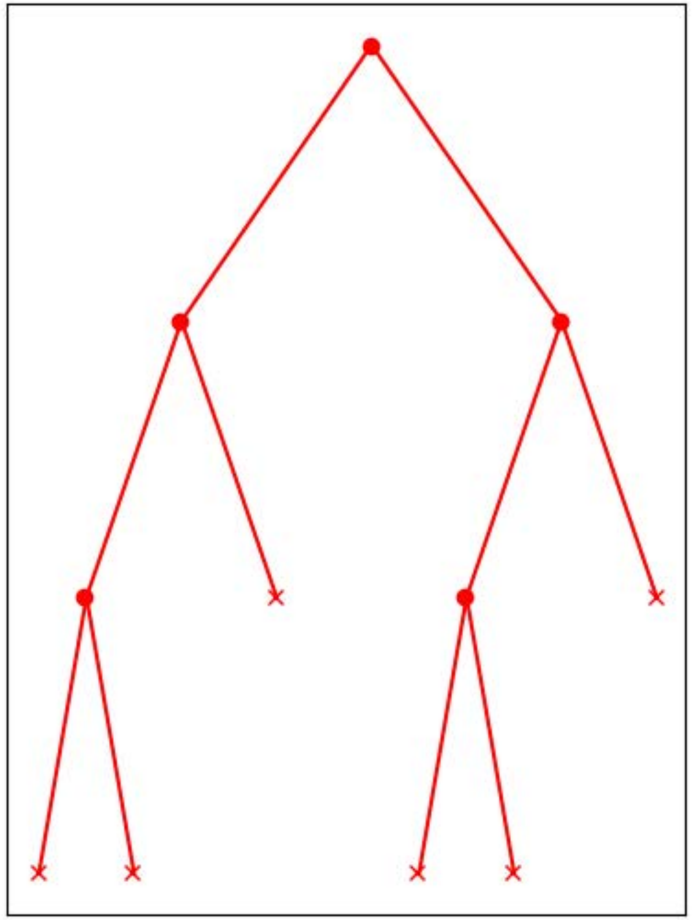




When using linear classifiers as weak learners.

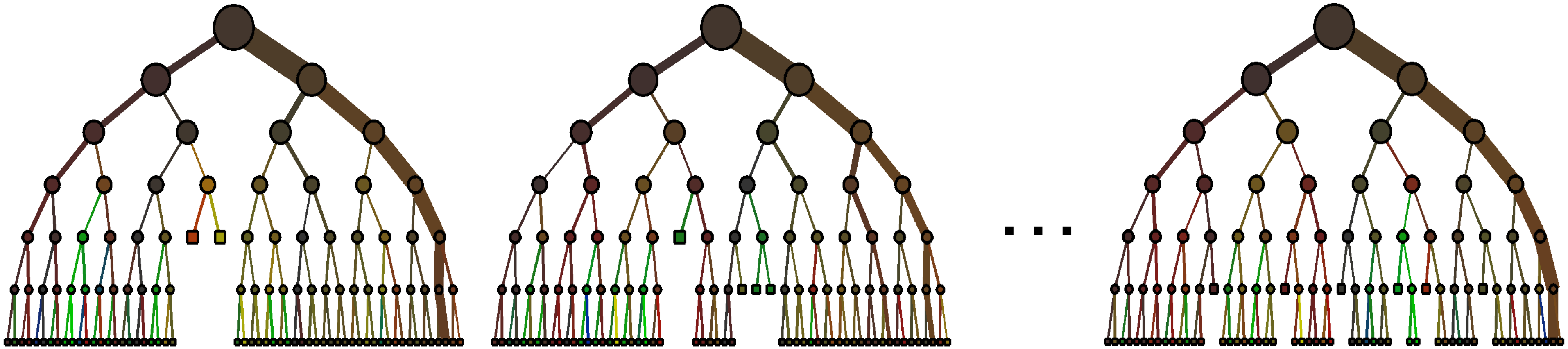
… but not for Trees





From Trees to Forests

Use multiple trees to increase robustness:



$p_1(c\,|\,\mathbf{v})$ $p_2(c\,|\,\mathbf{v})$ ... $p_n(c\,|\,\mathbf{v})$

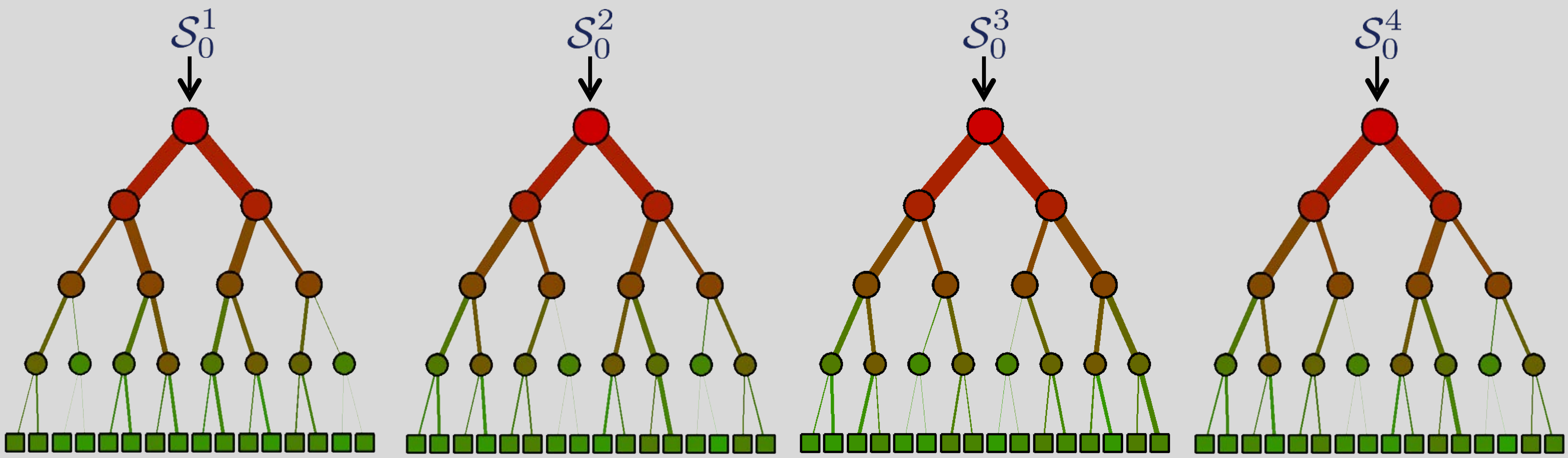
$$p(c|\mathbf{v}) = f(p_1(c|\mathbf{v}), \ldots, p_T(c|\mathbf{v}))$$

- How many trees?
- How different should they be?
- How do we fuse their outputs?

Creating Multiple Trees

$\mathcal{S}_0$	Full training set
$\mathcal{S}_0^t \subset \mathcal{S}_0$	Randomly sampled subsets made available to train the tree t

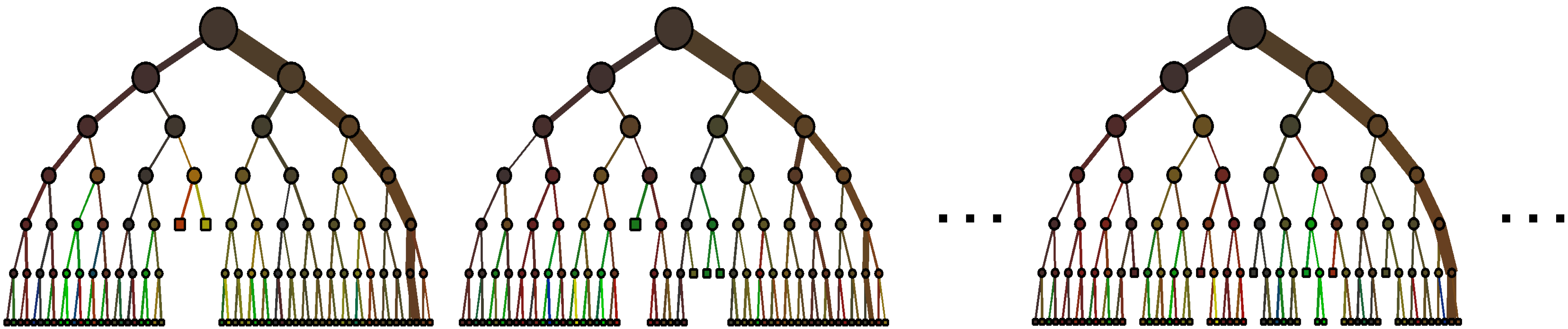
Forest training



- The subsets are typically chosen randomly with replacement.
- This is known as bagging.

EPFL

Fusing the Output



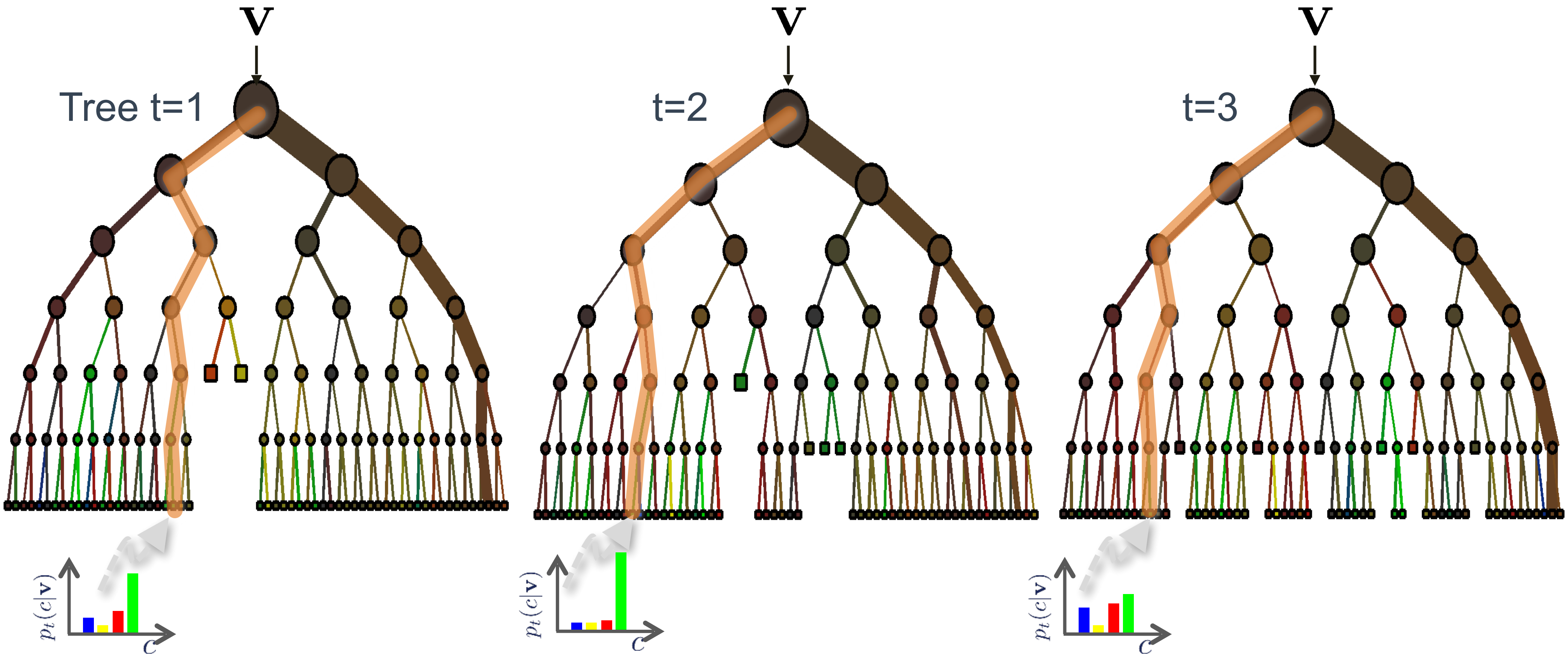
Naive Bayesian: $$p(c\,|\,\mathbf{v}) \propto \prod_t p_t(c\,|\,\mathbf{v})$$

$$L(c,\mathbf{v}) = \frac{1}{T}\sum_t -\log(p_t(c\,|\,\mathbf{v}))$$

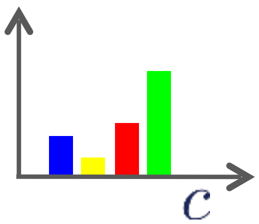
- Assumes the output of each tree is independent from each other.
- Valid assumption if the training subsets are disjoint.
- Justifiable assumption if the training database is large enough.

EPFL

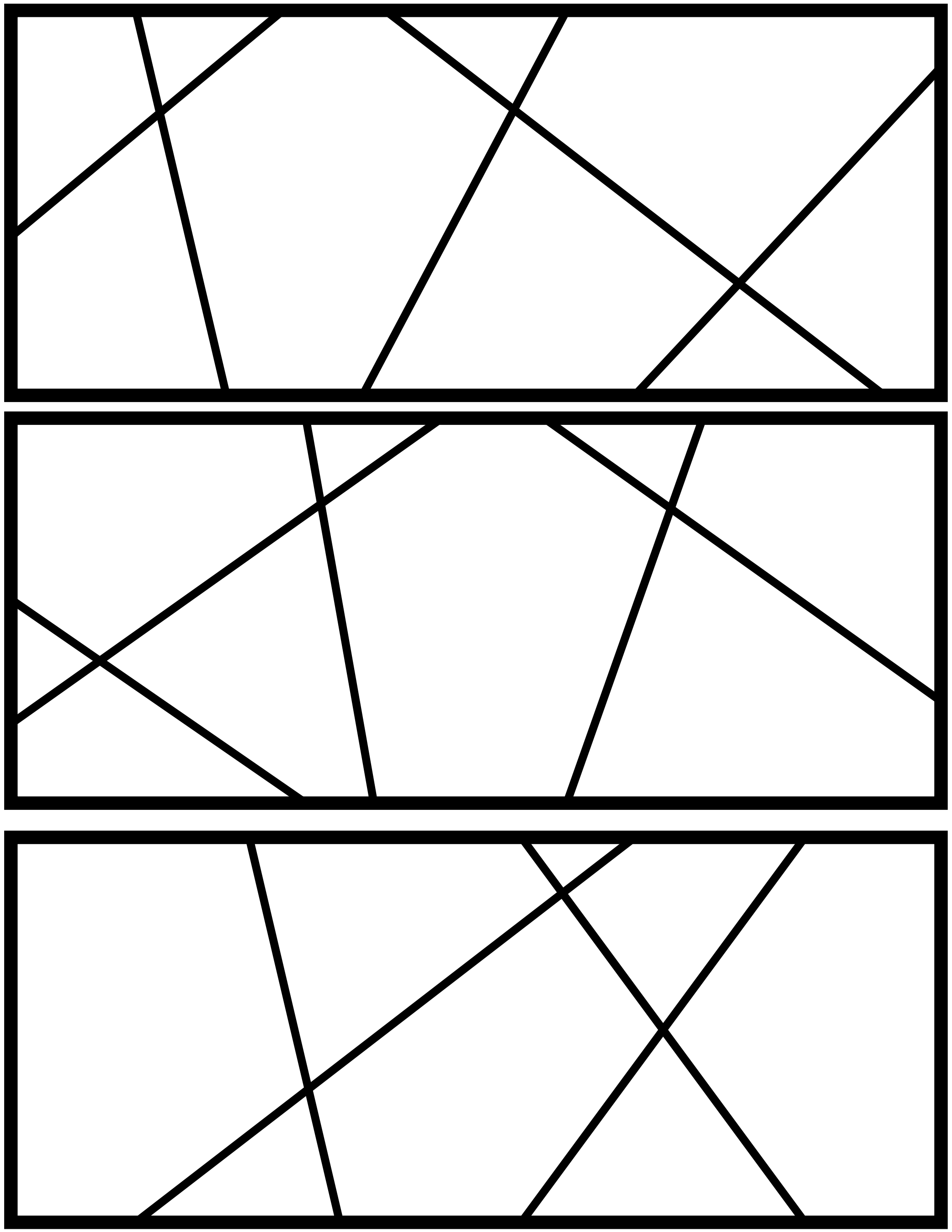
Ensemble Model



$$L(c, \mathbf{v}) = \frac{1}{T}\sum_t -\log(p_t(c|\mathbf{v}))$$

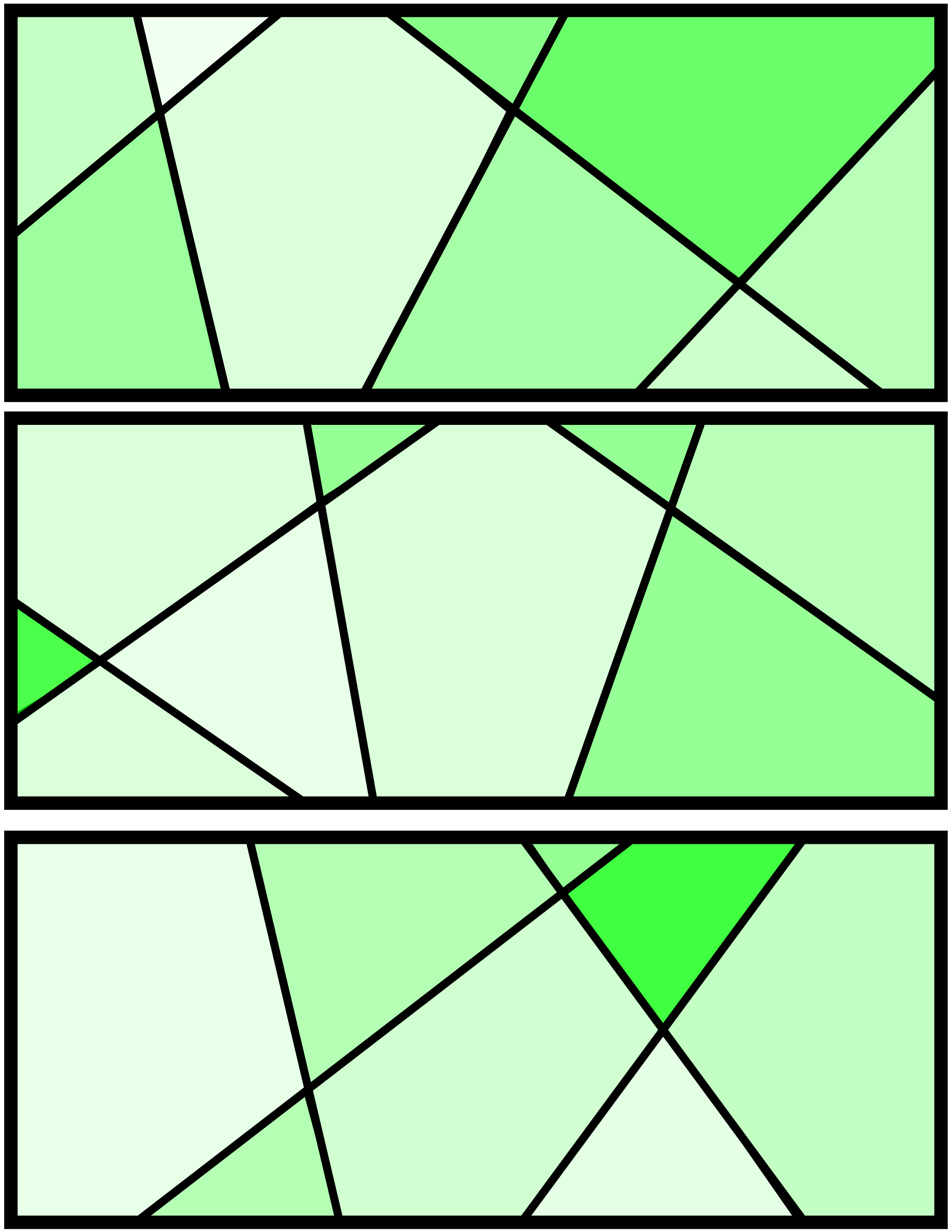


Graphical Interpretation



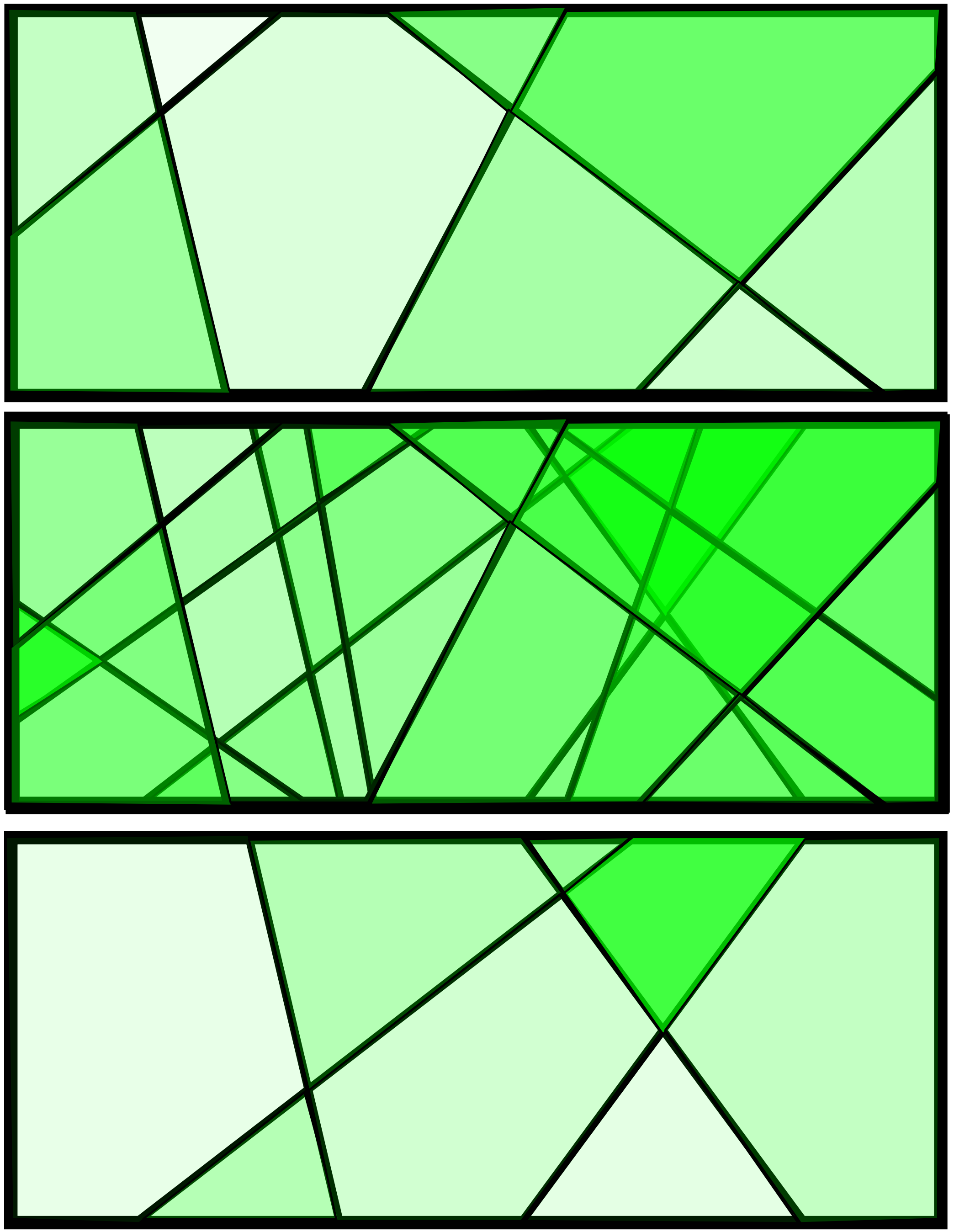
Weak classifiers at every level of the tree split the space.

Graphical Interpretation



Weak classifiers at every level of the tree split the space.

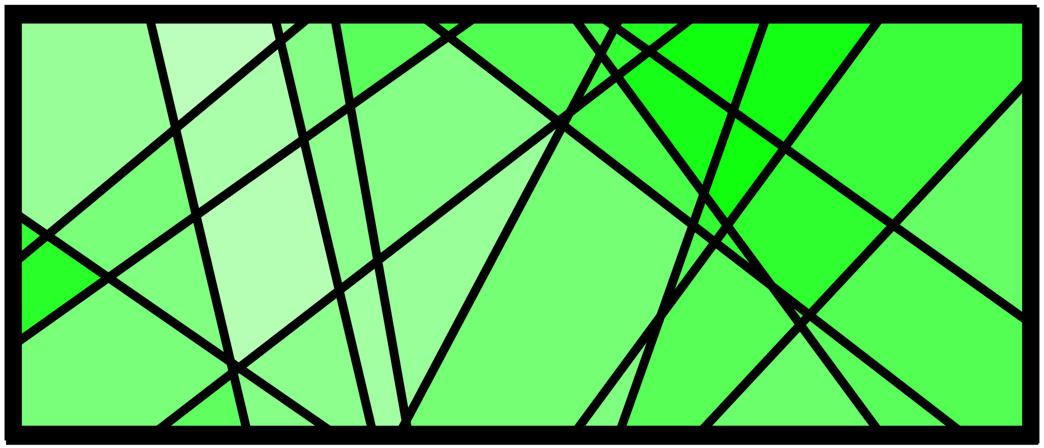
Graphical Interpretation



The splits are combined by the hierarchical nature of the tree.

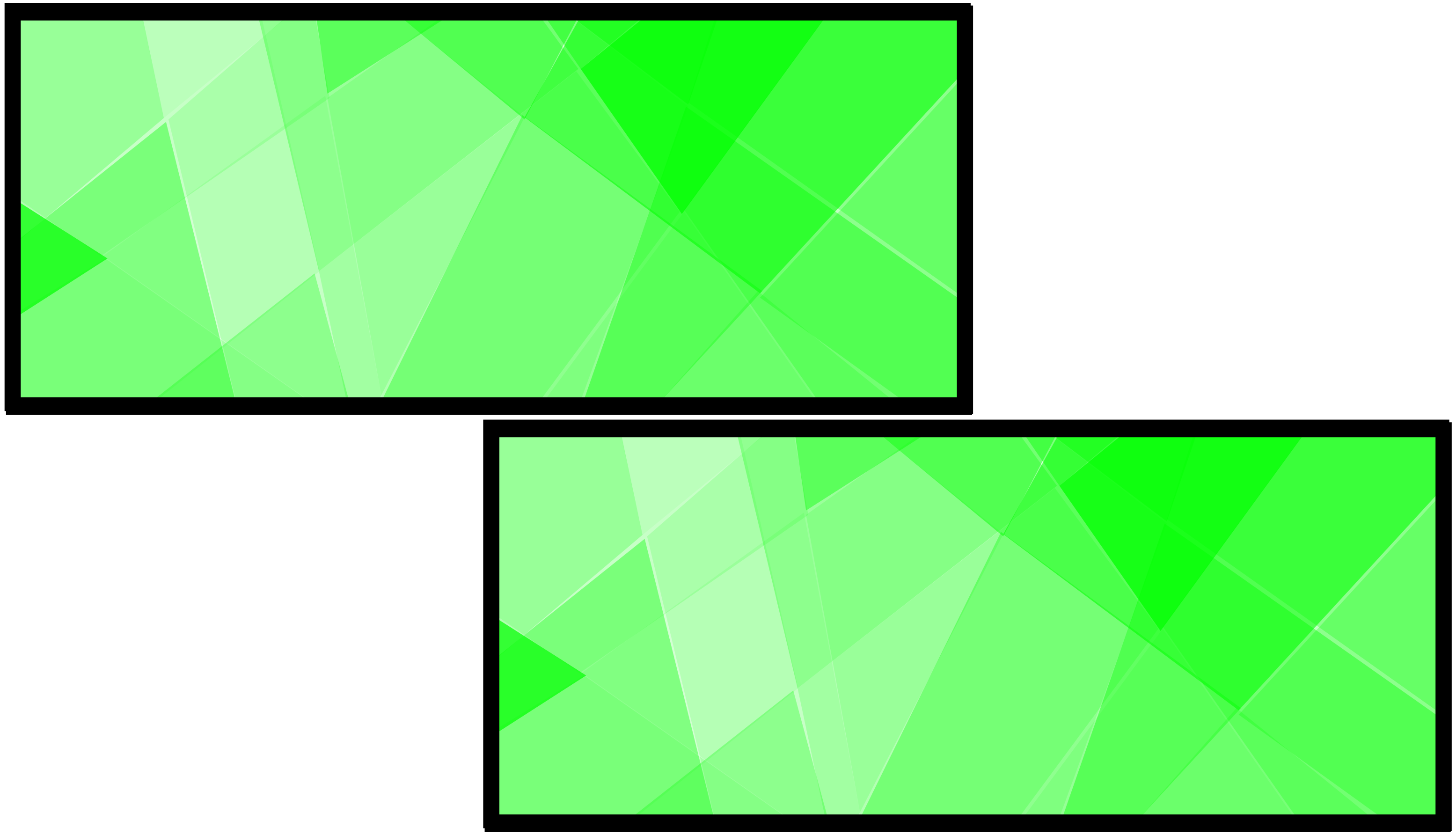
EPFL

Graphical Interpretation

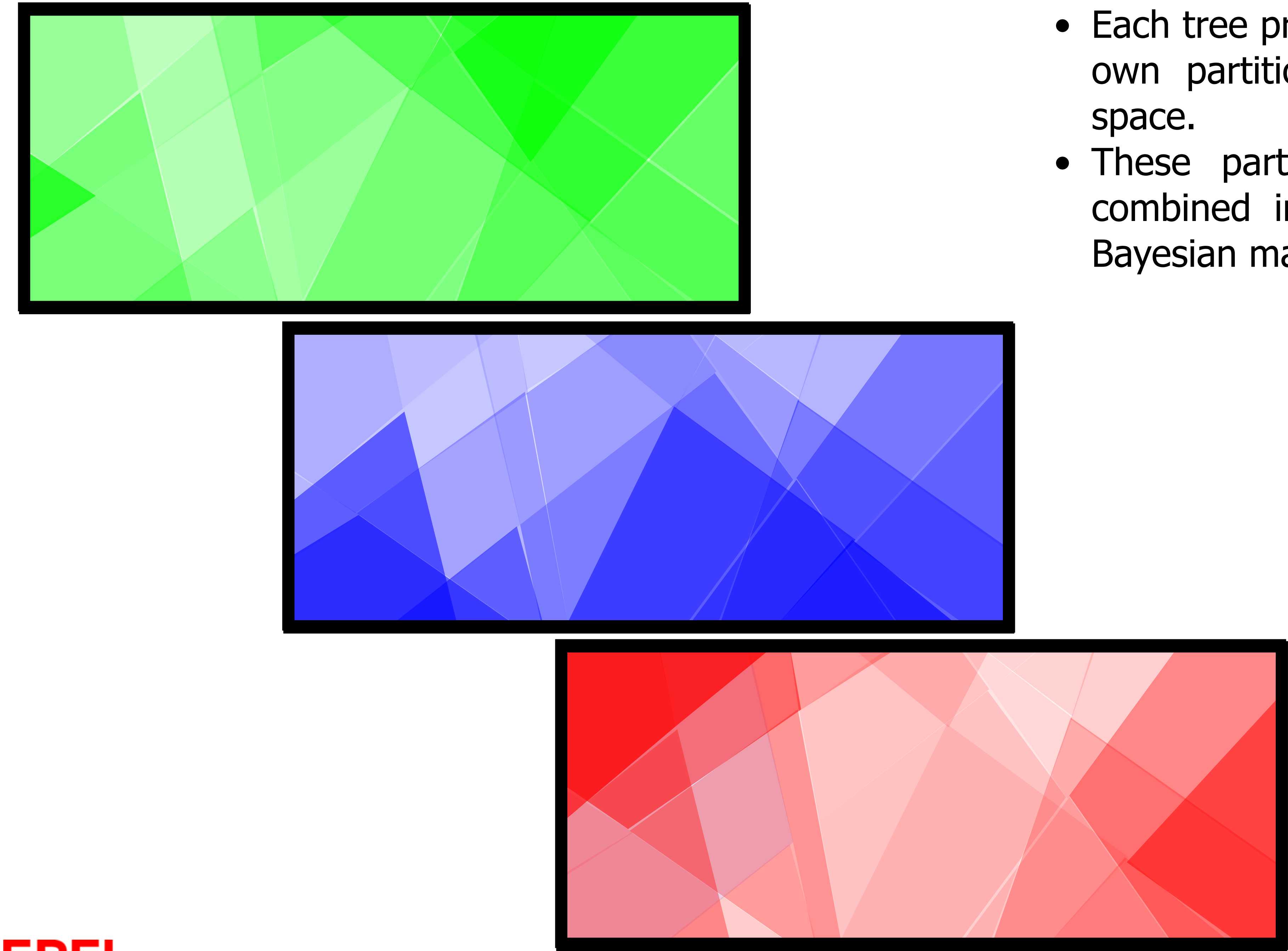


The splits are combined by the hierarchical nature of the tree.

Graphical Interpretation

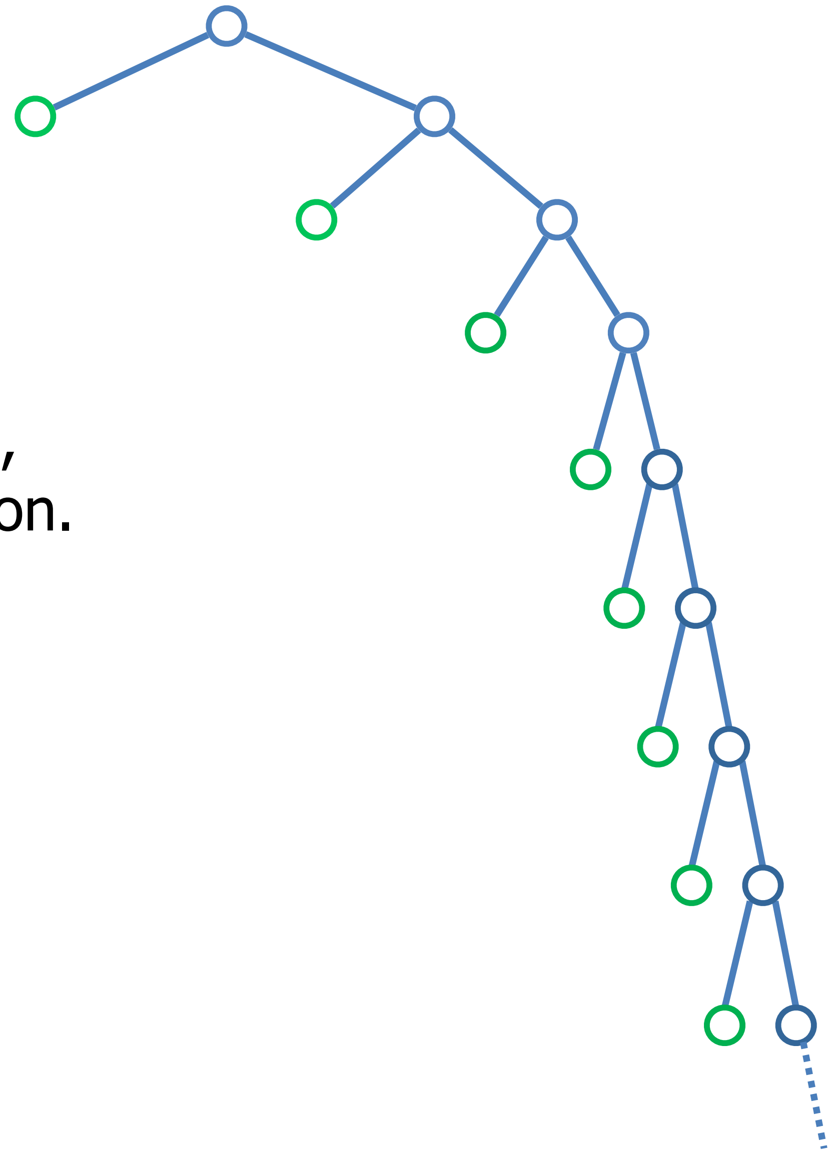


Graphical Interpretation



- Each tree produces its own partition of the space.
- These partitions are combined in a Naive Bayesian manner.

Relationship to Boosting



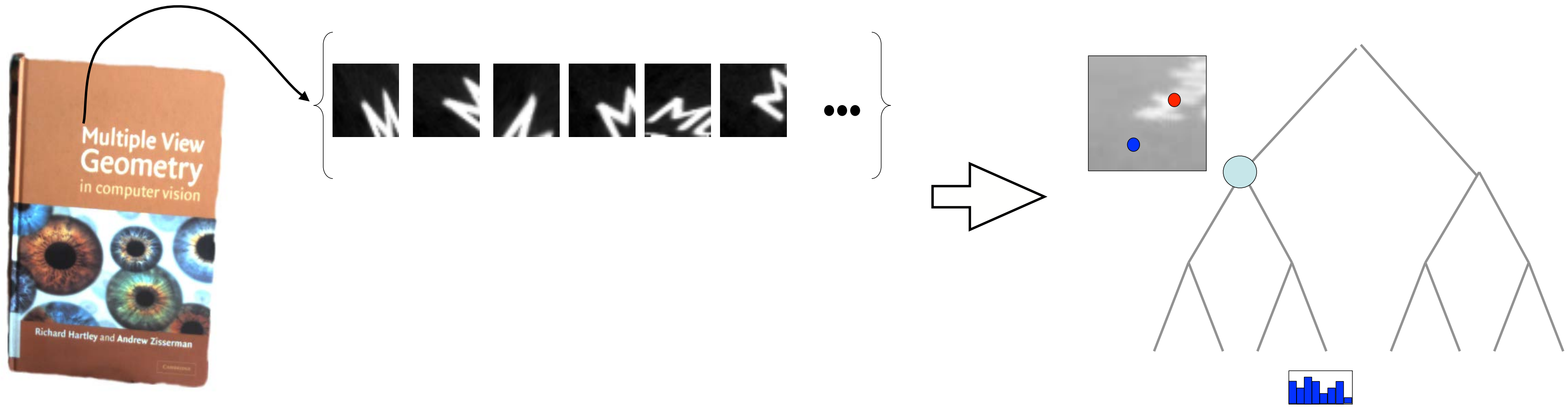
- Boosted Cascades:
  - Very unbalanced tree.
  - Good for unbalanced binary problems, such as sliding window object detection.

- Randomized forests:
  - Less deep, more balanced.
  - Ensemble of trees gives robustness.
  - Good for multi-class problems.

3D Pose Estimation

To track the car:
1. We track interest points in the image.
2. We infer their 3D position from the tracks.

Classification-Based Approach to Matching

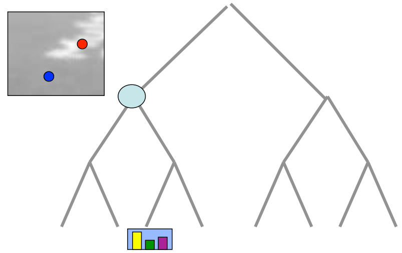


- One class per keypoint.
- Train a decision forest to recognize them.

Lepetit and Fua, PAMI'06

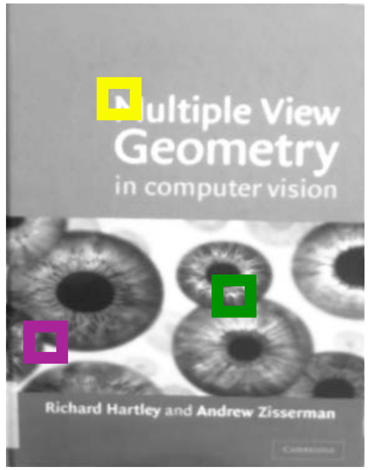
Simple Weak Learners

The nodes contain simple tests of the form "Is $I(\mathbf{m}_1) > I(\mathbf{m}_2)$?"



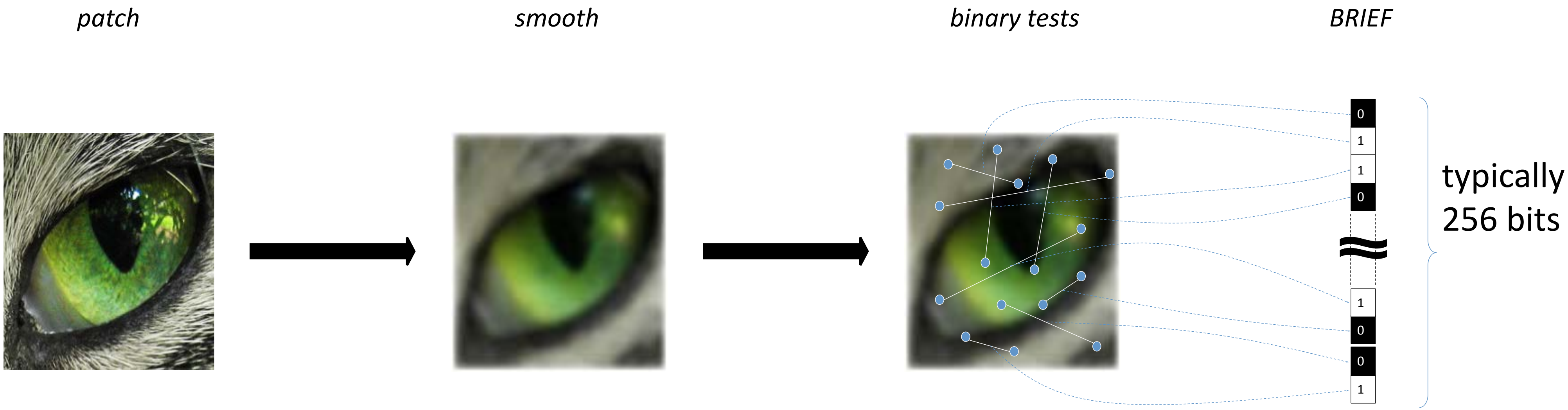
Posteriors can be learned from:

- Warped images



- Video sequences

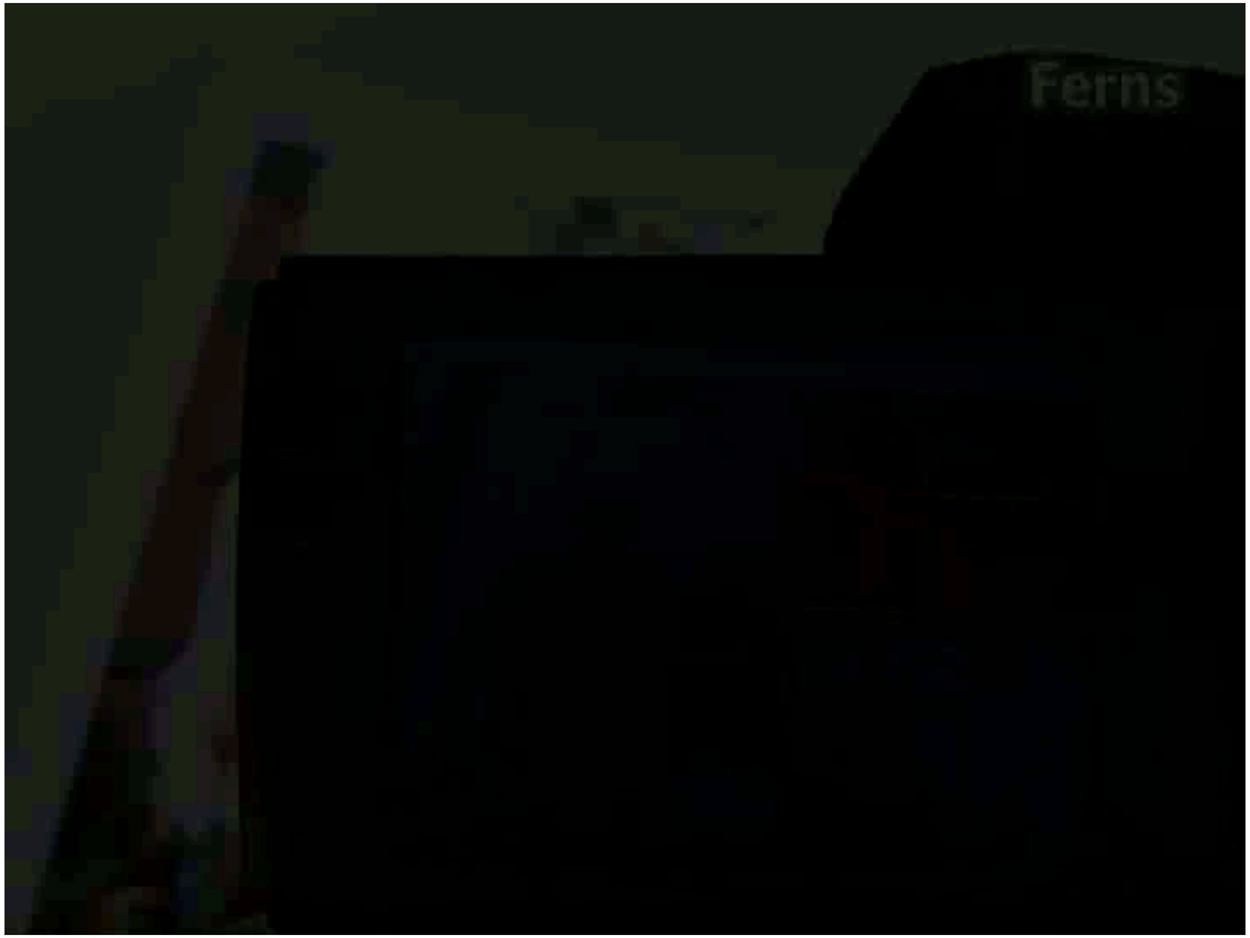
BRIEF



- Most smooth kernels work, even simple box filters.
- 128, 256, or 512 binary tests usually suffice.
- Random arrangement of tests effective as long as they are evenly sampled.

Calonder et al. ECCV'10

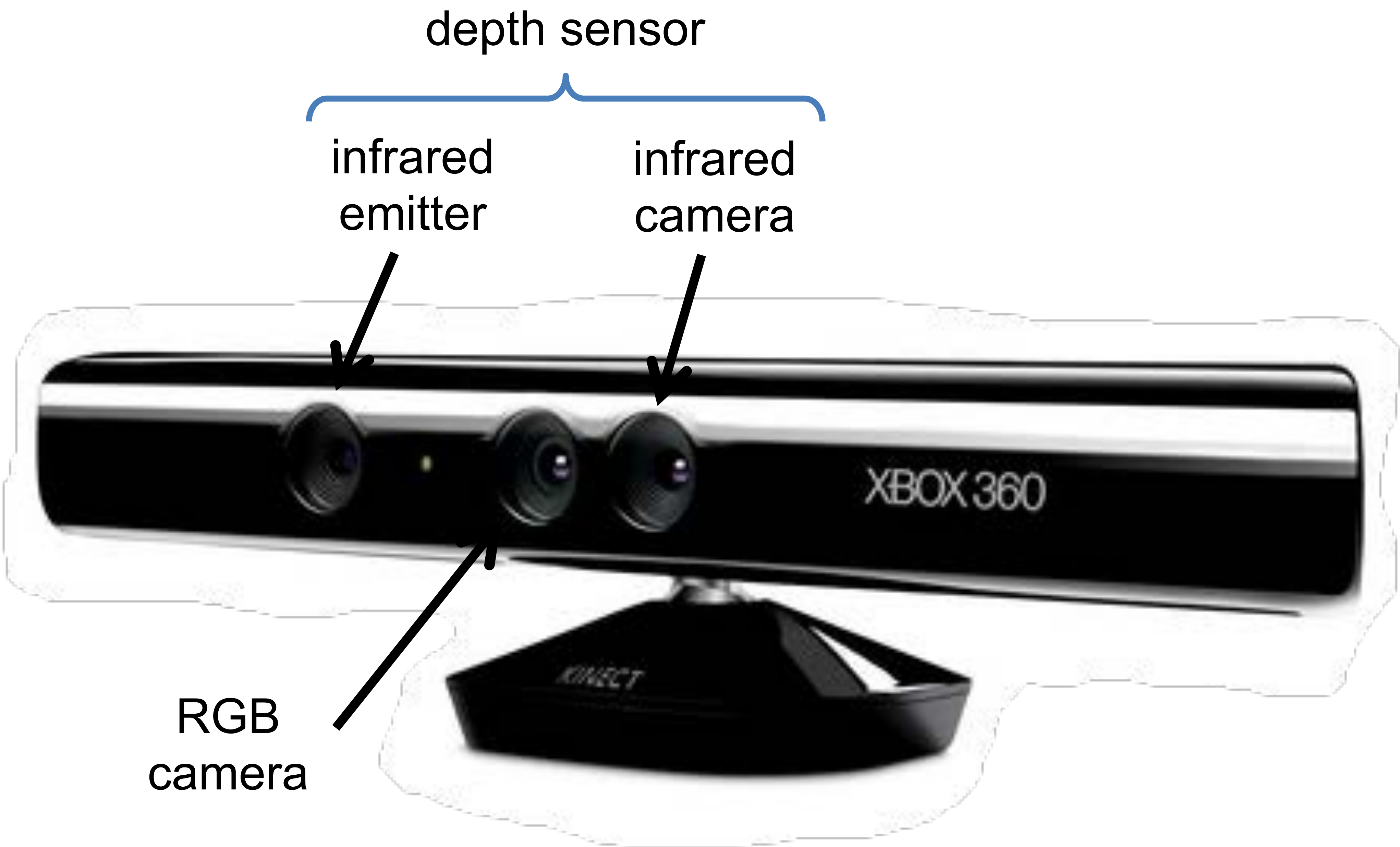
Point Correspondences



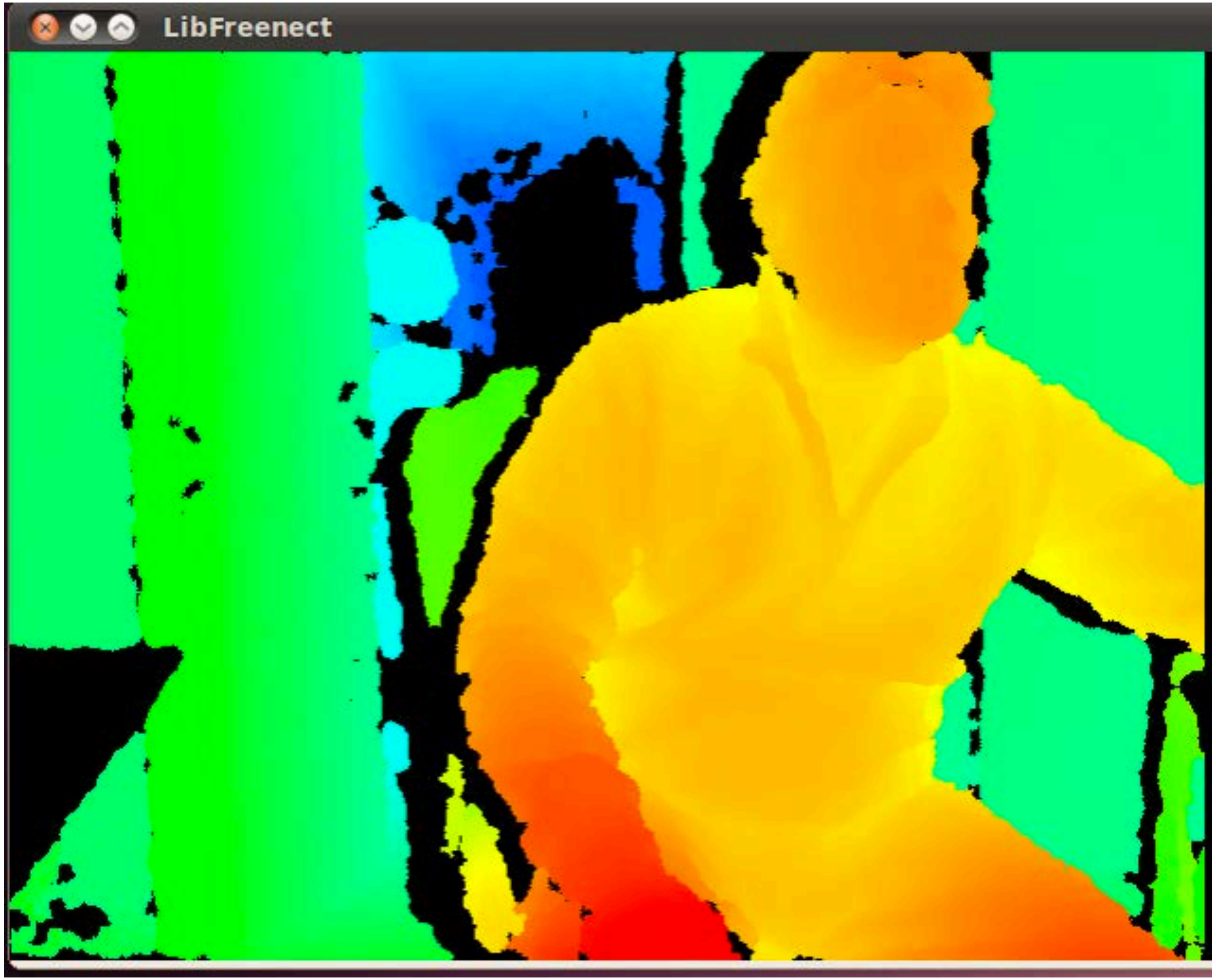
--> Real-time on a 2008 cell phone.

Wagner et al. ISMAR'08

Body Part Estimation



Depth Image



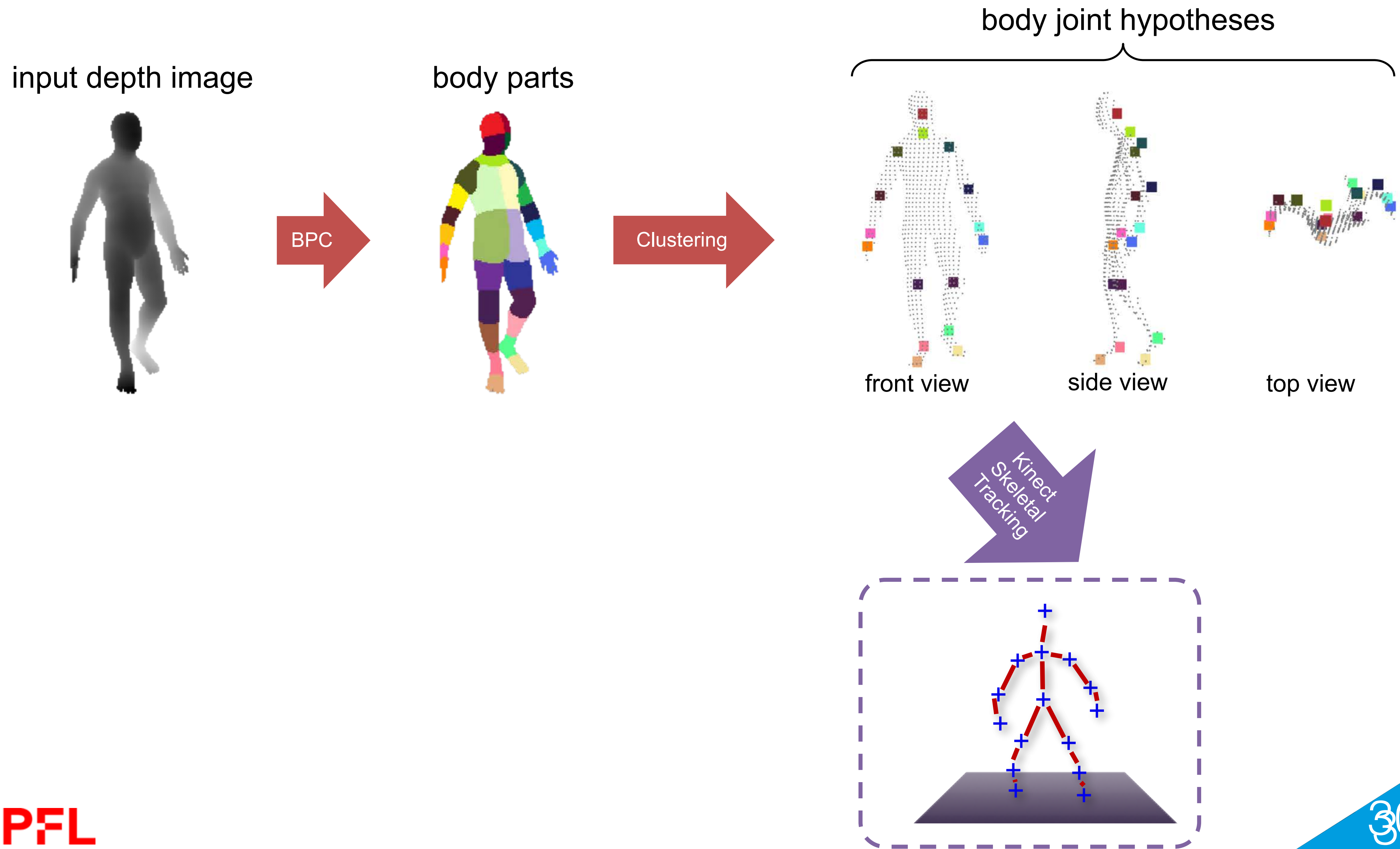
Depth Sequence

Depth image.

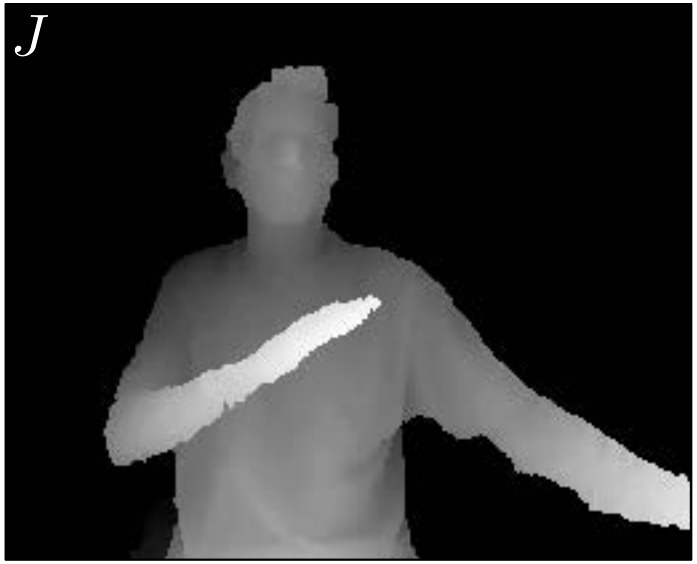
Side view

Top view

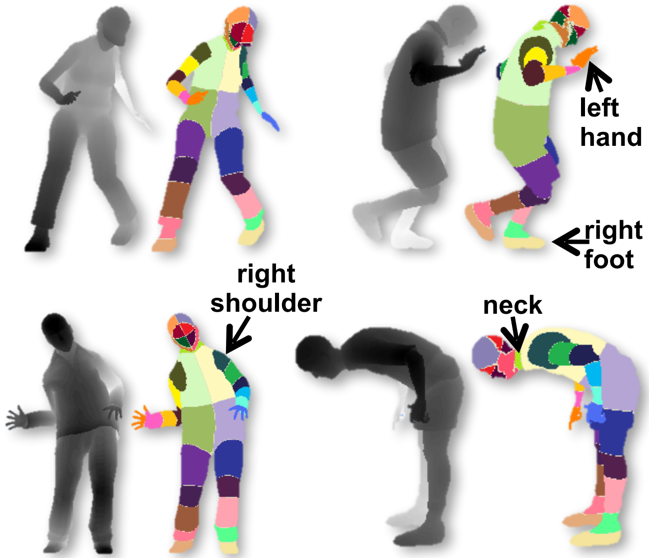
Processing Pipeline



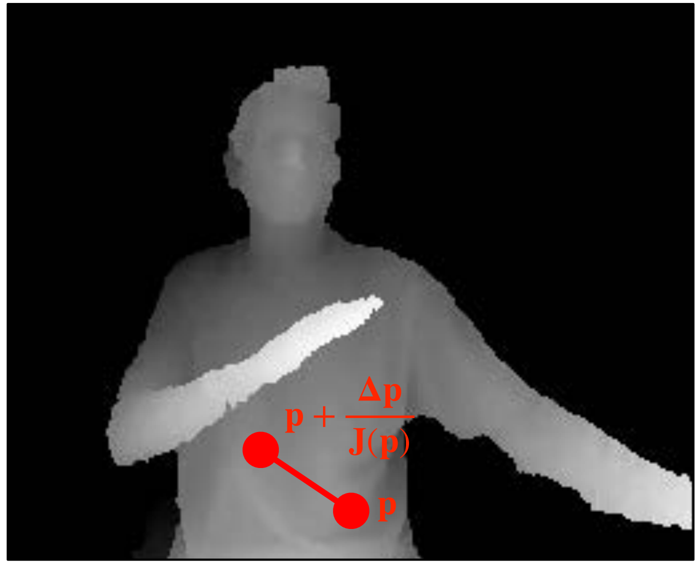
Body Part Recognition



Input depth image



Training labelled data



Visual features

Visual feature: $$x(\mathbf{p}, \Delta\mathbf{p}) = J(\mathbf{p}) - \mathbf{J}(\mathbf{p} + \frac{\Delta\mathbf{p}}{\mathbf{J}(\mathbf{p})})$$

Weak classifier: $$h(\mathbf{p}, \Delta\mathbf{p}, \tau) = x(\mathbf{p}, \Delta\mathbf{p}) - \tau$$

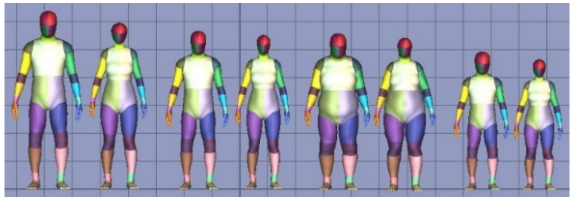
- Very fast to compute.
- Real-time performance

Synthetic Training Data

[Vicon]

Record mocap
100,000s of poses

Retarget to varied body shapes



Render (depth, body parts) pairs

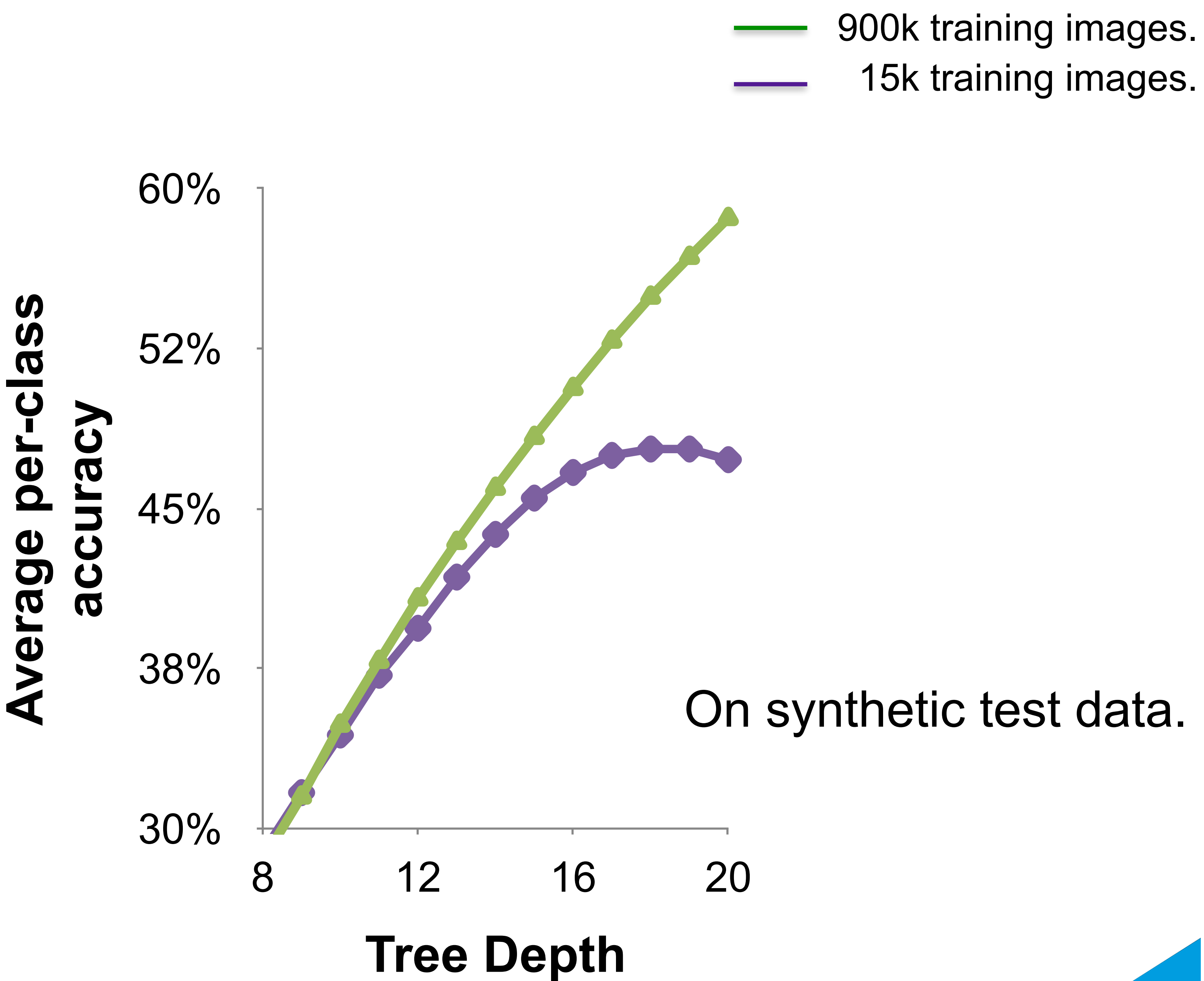
Train invariance to:

Influence of Tree Depth

Input depth | Ground truth parts | Inferred parts

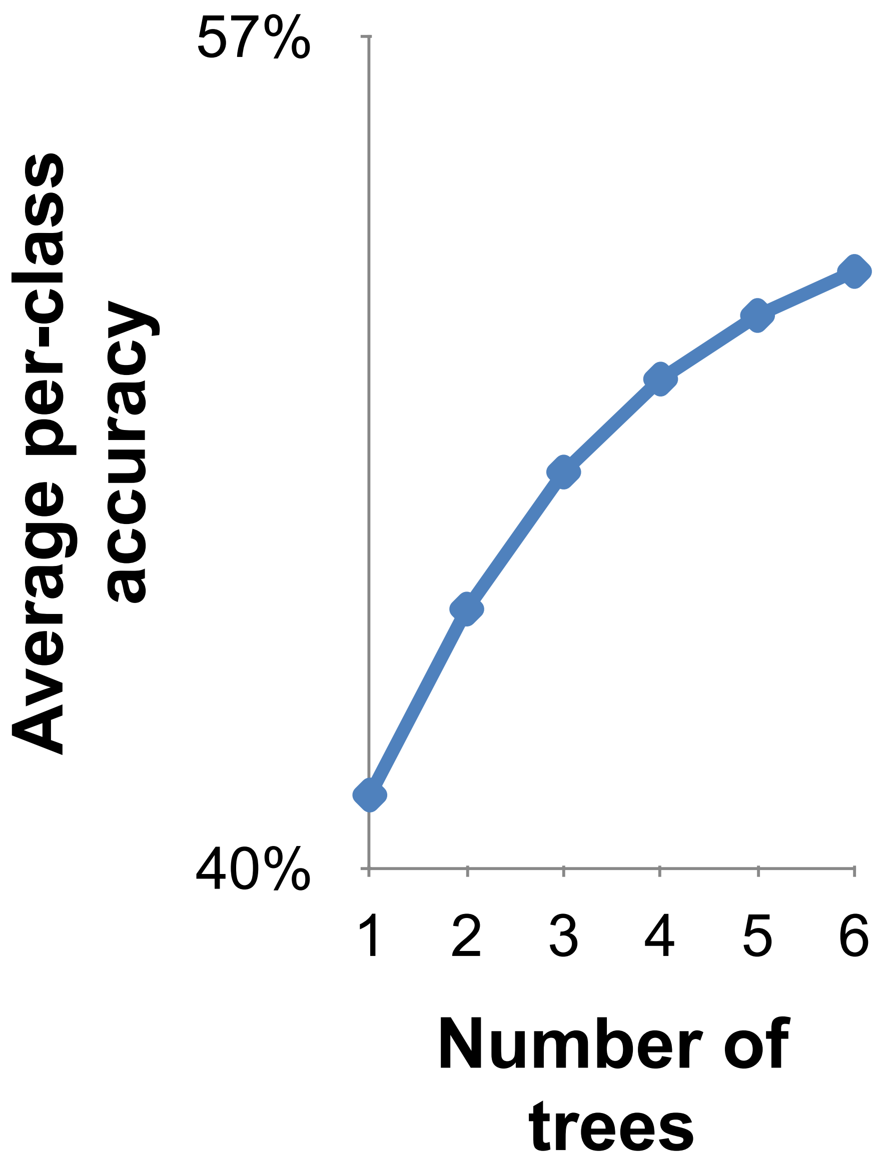
depth 18

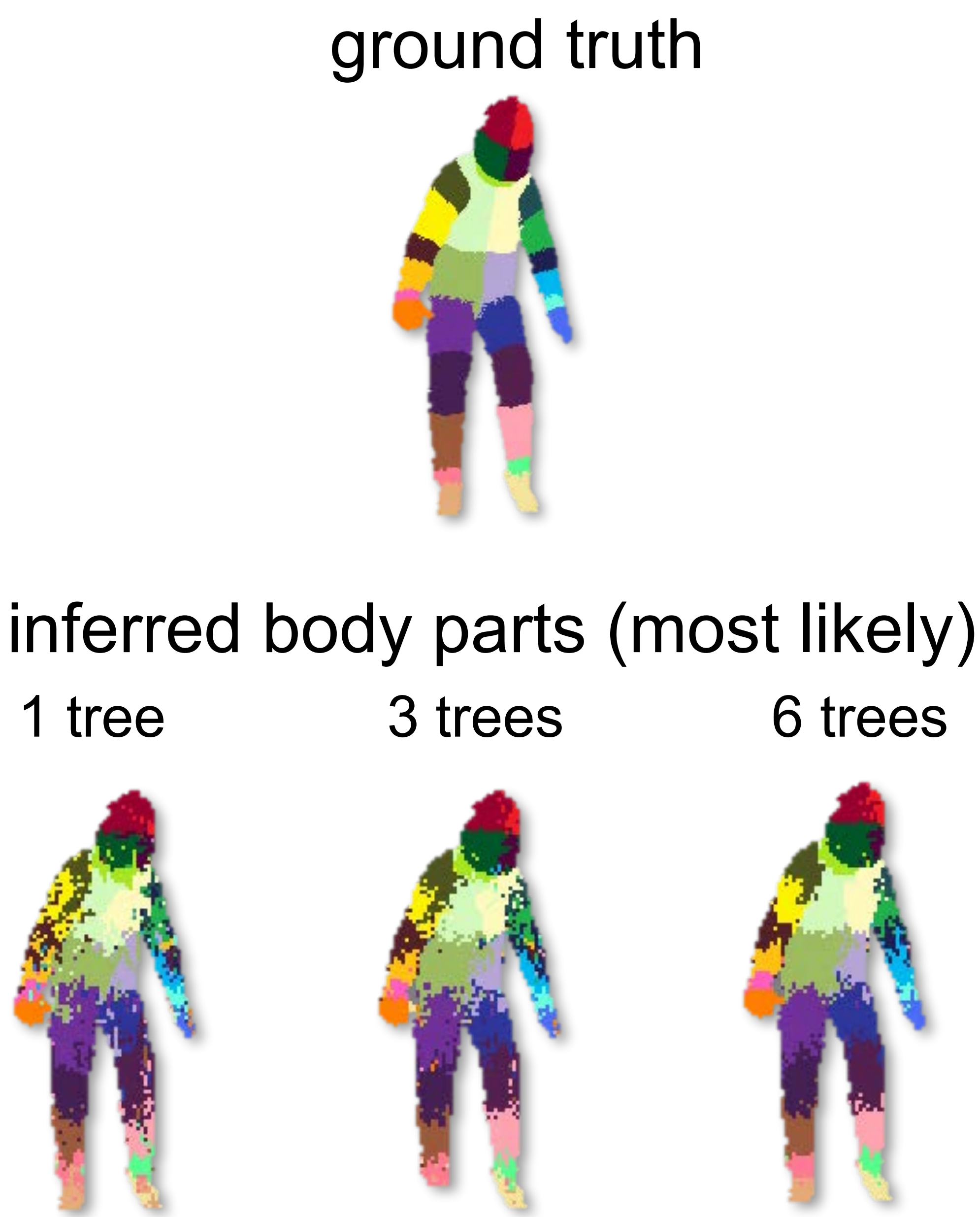
Choosing the Tree Depth



On synthetic test data.

Choosing the Number of Trees





Result

Input depth image with background removed.

Inferred body parts posterior $p(c|\mathbf{v})$

(2 videos here)

Decision Forests in Short

- They make it comparatively easy to interpret what is happening.
- Their behavior is easy to modify.
- They can be trained using moderate amounts of data.

—> Very useful in many practical applications.