# Edge Classification

Thorsten Beier

May 29, 2017

# Vanilla Random Forest on Edges

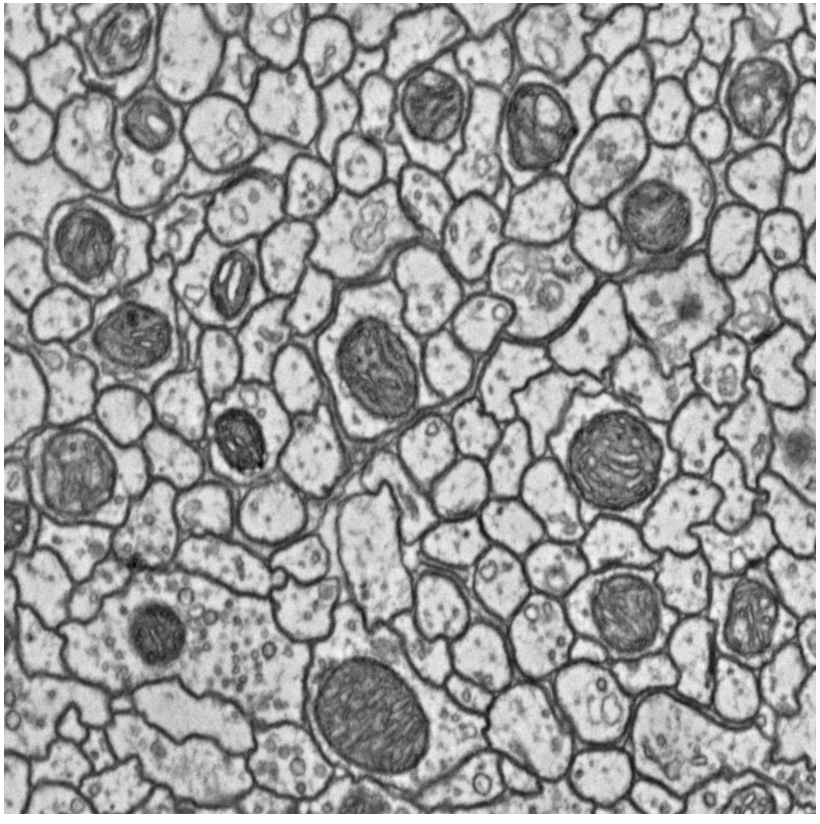


Figure: Raw Data

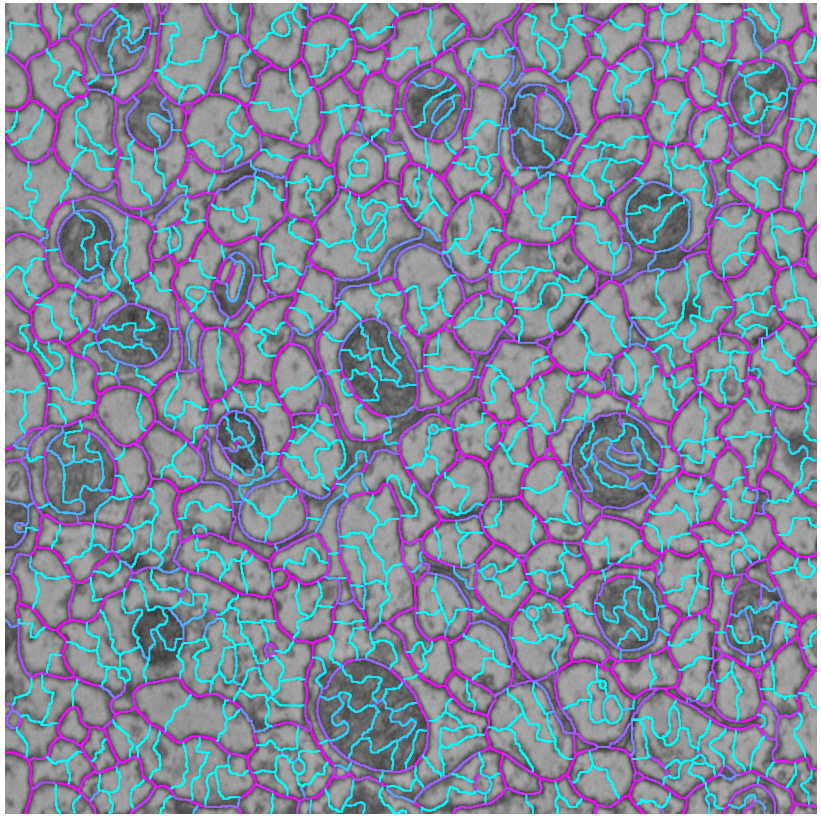


Figure: Vanilla RF on edges

# Vanilla Random Forest on Edges

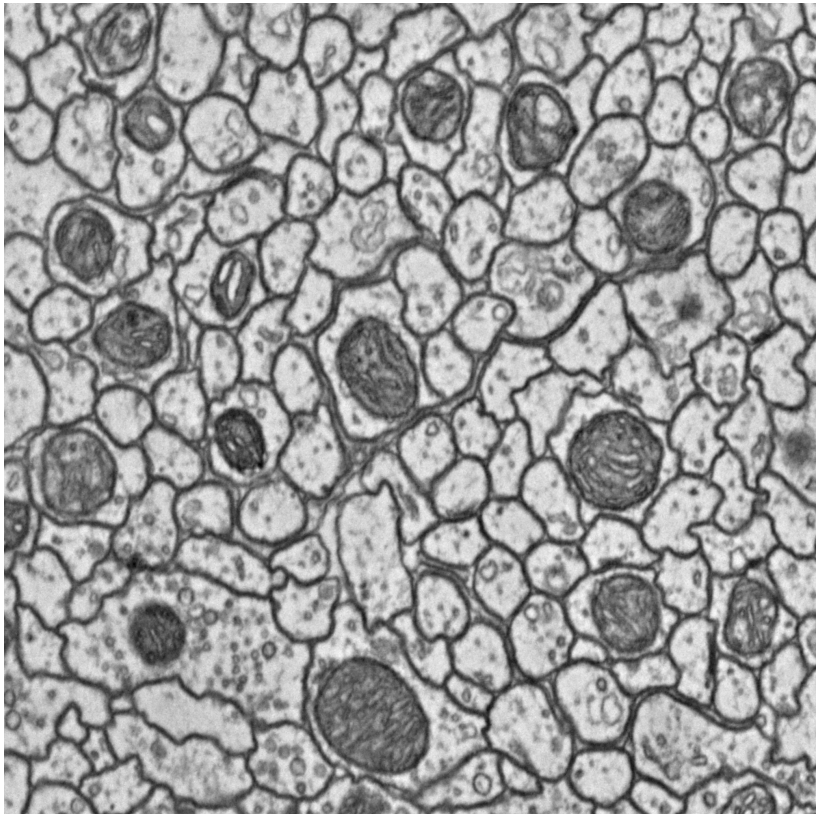


Figure: Raw Data

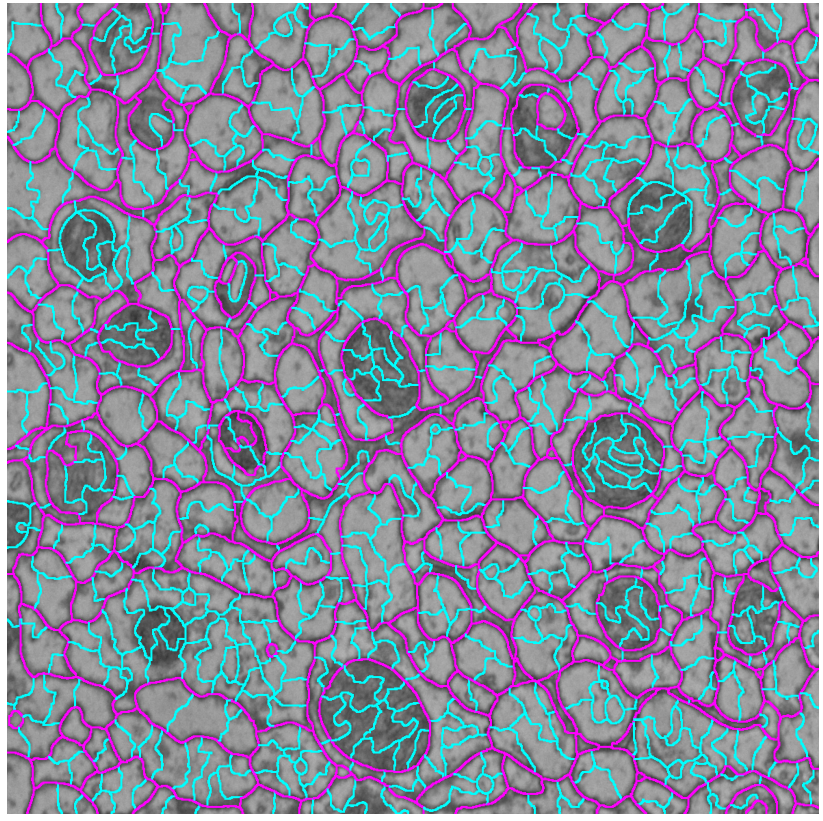
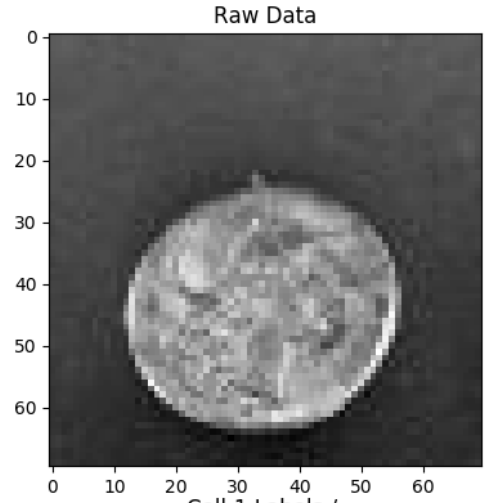
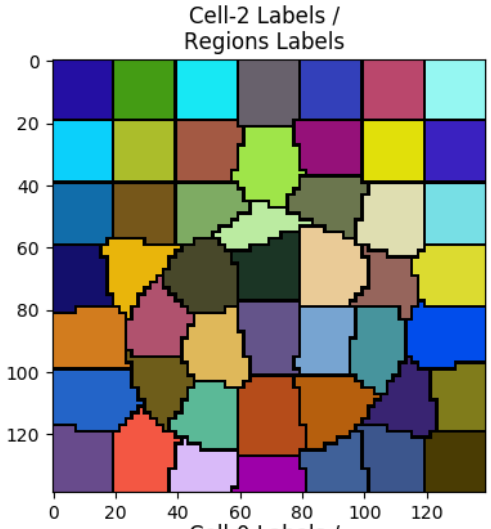
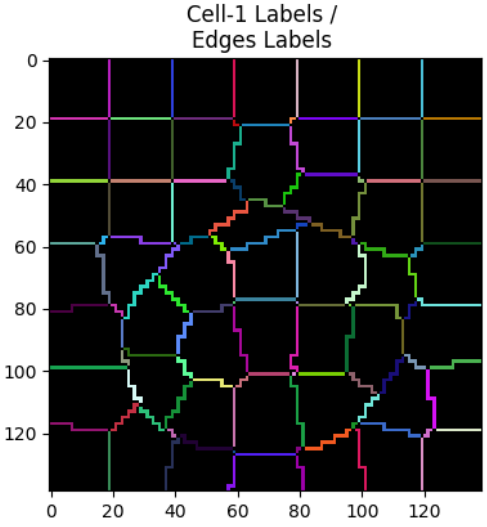


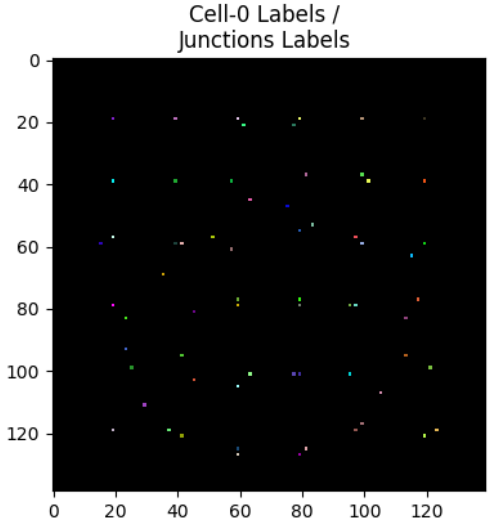
Figure: Vanilla RF on edges + MC

# Reminder CGP



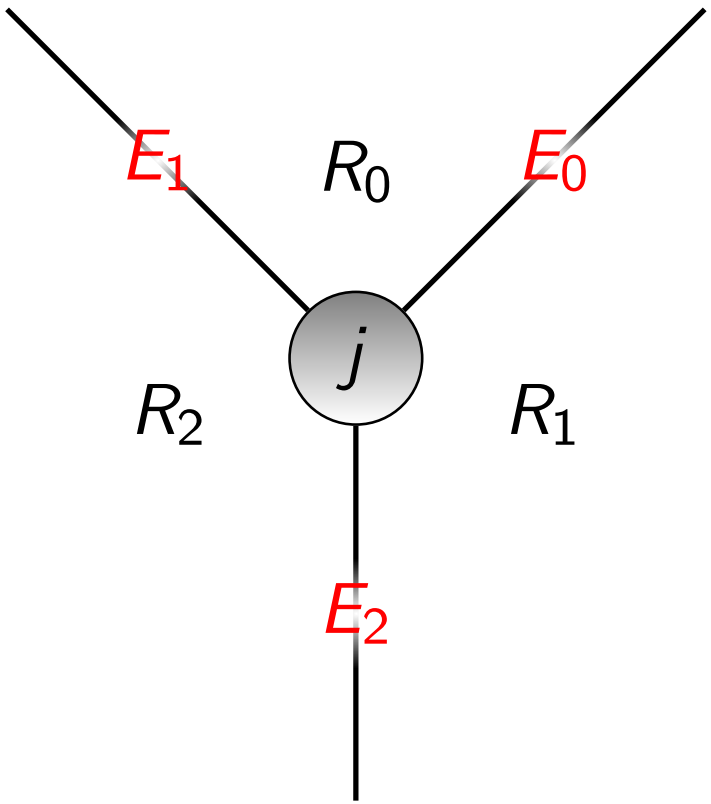



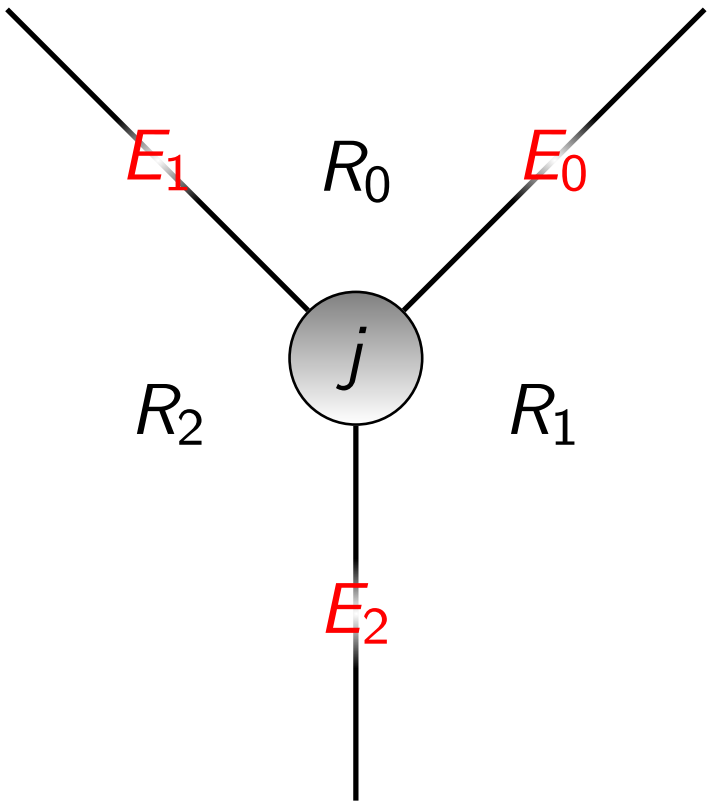

- Each Junction has 3 or 4 edges.
- Most edges have 2 Junctions
- 3-Junction has $2^3 - 3$ edge-configurations: $000, 011, 101, 110, 111$
- 4-Junction has $2^4 - 3$ edge-configurations: $0000, 0011, 0101, \ldots, 1111$

# Vanilla Random Forest on Junctions



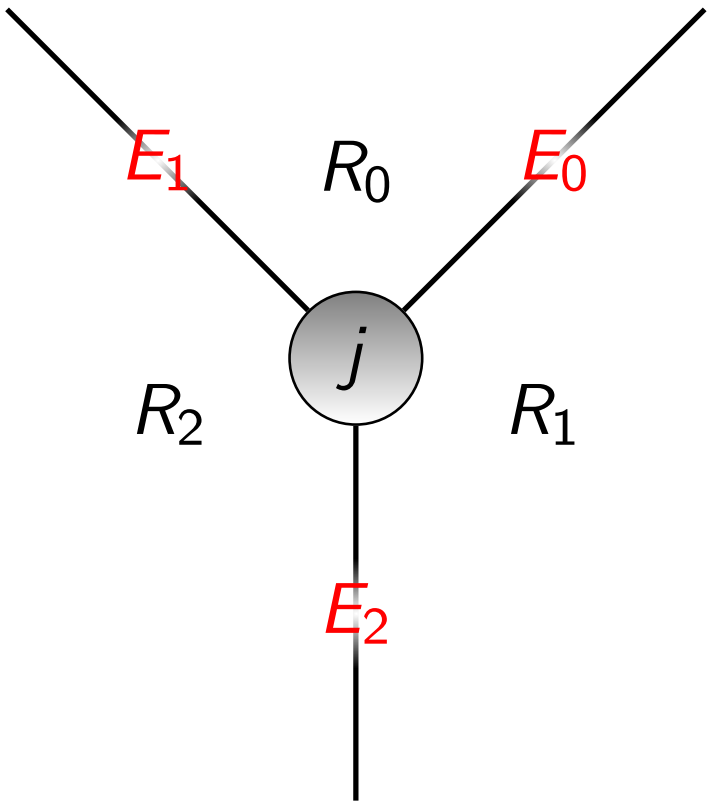

- 5 classes:
  $0 \equiv 000, \quad 1 \equiv 011, \quad 2 \equiv 101,$
  $3 \equiv 110, \quad 4 \equiv 111$

# Vanilla Random Forest on Junctions




- 5 classes:
  $0 \equiv 000, \quad 1 \equiv 011, \quad 2 \equiv 101,$
  $3 \equiv 110, \quad 4 \equiv 111$
- Features:
  - Naive Concatenation of edge Features: $F_j = F_{e0} + F_{e1} + F_{e2}$
  - Simple Statistics over the 3 edges:
  - argmin/argmax/min/max/sum across 3 edges for each edge feature

# Vanilla Random Forest on Junctions




- 5 classes:
  $0 \equiv 000, \quad 1 \equiv 011, \quad 2 \equiv 101,$
  $3 \equiv 110, \quad 4 \equiv 111$
- Features:
  - Naive Concatenation of edge Features: $F_j = F_{e0} + F_{e1} + F_{e2}$
  - Simple Statistics over the 3 edges:
  - argmin/argmax/min/max/sum across 3 edges for each edge feature
- Result: 3% improvement in hamming error

# Reminder: Autocontext RF

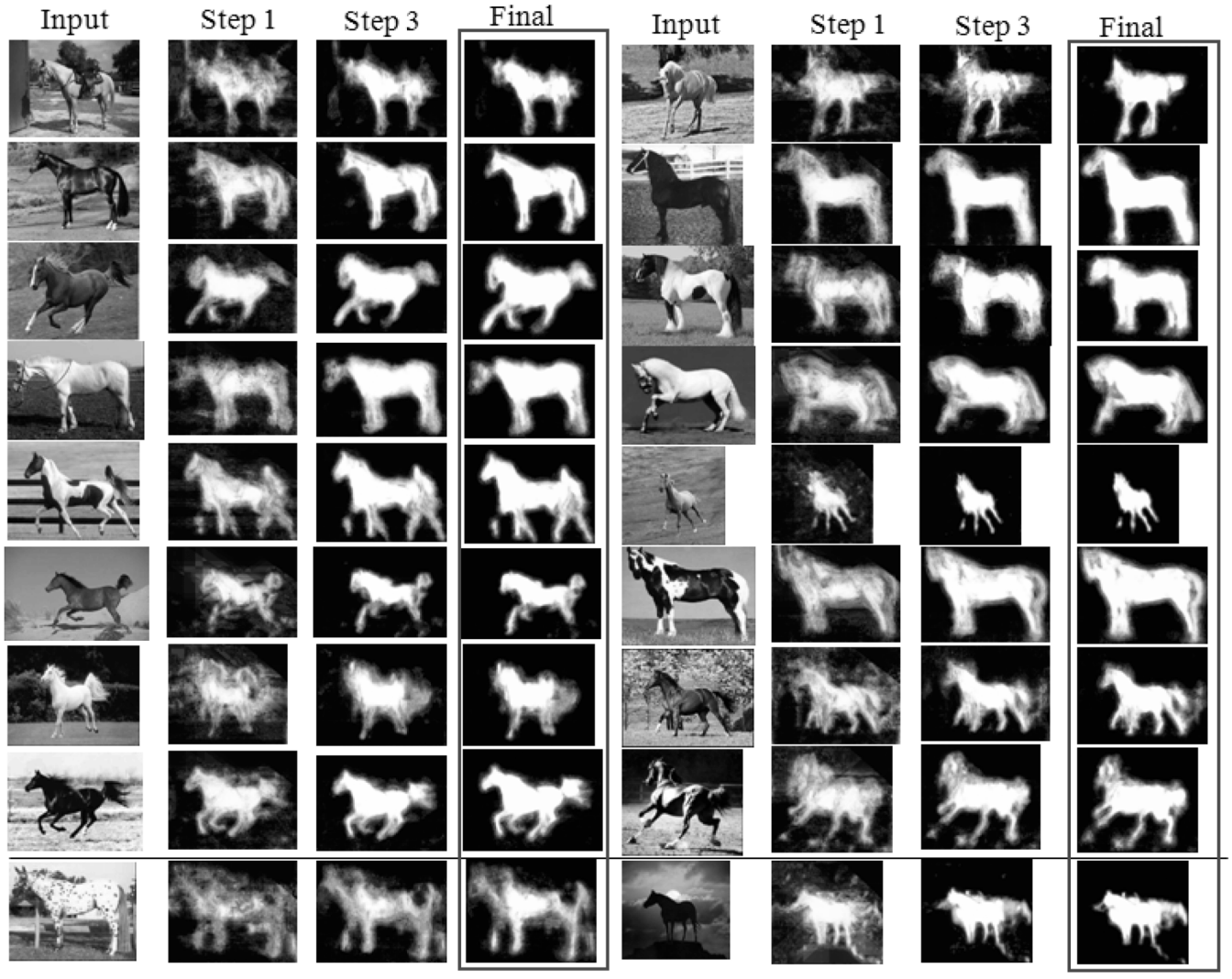



Figure: Auto-Context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation

# Graph Autocontext

- Compute initial features for each edge

# Graph Autocontext

- Compute initial features for each edge
  - Statistics of raw/pmap accumulated along edges

# Graph Autocontext

- Compute initial features for each edge
  - Statistics of raw/pmap accumulated along edges
  - ...

# Graph Autocontext

- Compute initial features for each edge
  - Statistics of raw/pmap accumulated along edges
  - ...
- Train initial random forest & Predict probabilities all edges

# Graph Autocontext

- Compute initial features for each edge
  - Statistics of raw/pmap accumulated along edges
  - ...
- Train initial random forest & Predict probabilities all edges
- Compute new features on probabilities

# Graph Autocontext

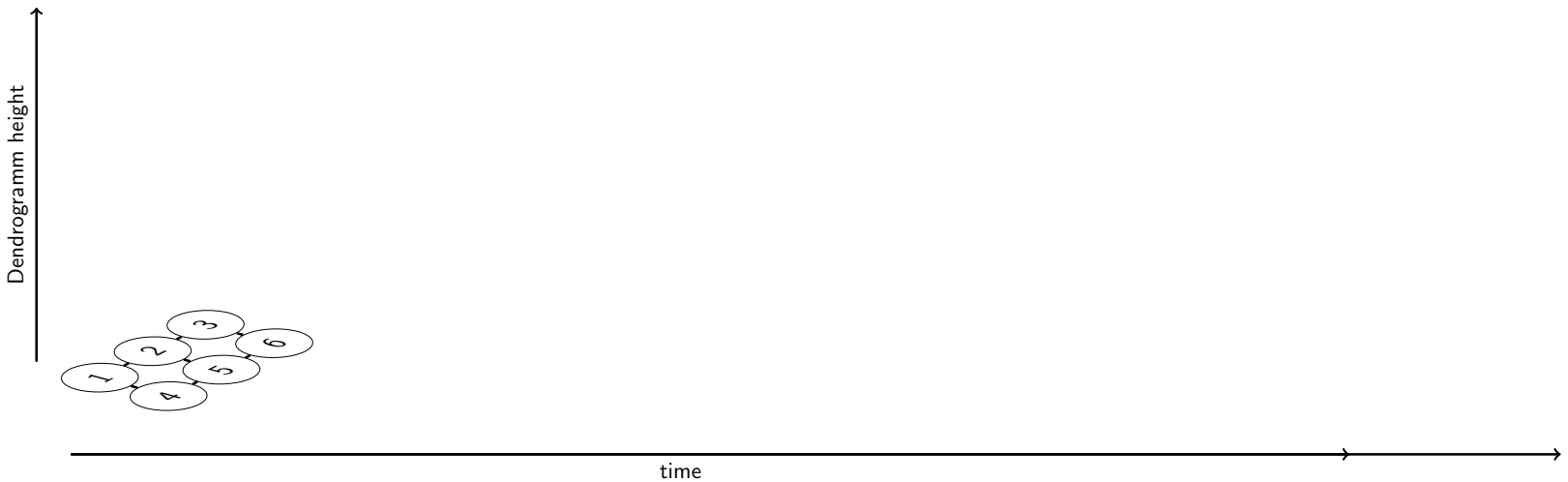
- Compute initial features for each edge
  - Statistics of raw/pmap accumulated along edges
  - ...
- Train initial random forest & Predict probabilities all edges
- Compute new features on probabilities

- Stack new features with old features & train improved random forest

# Graph Autocontext

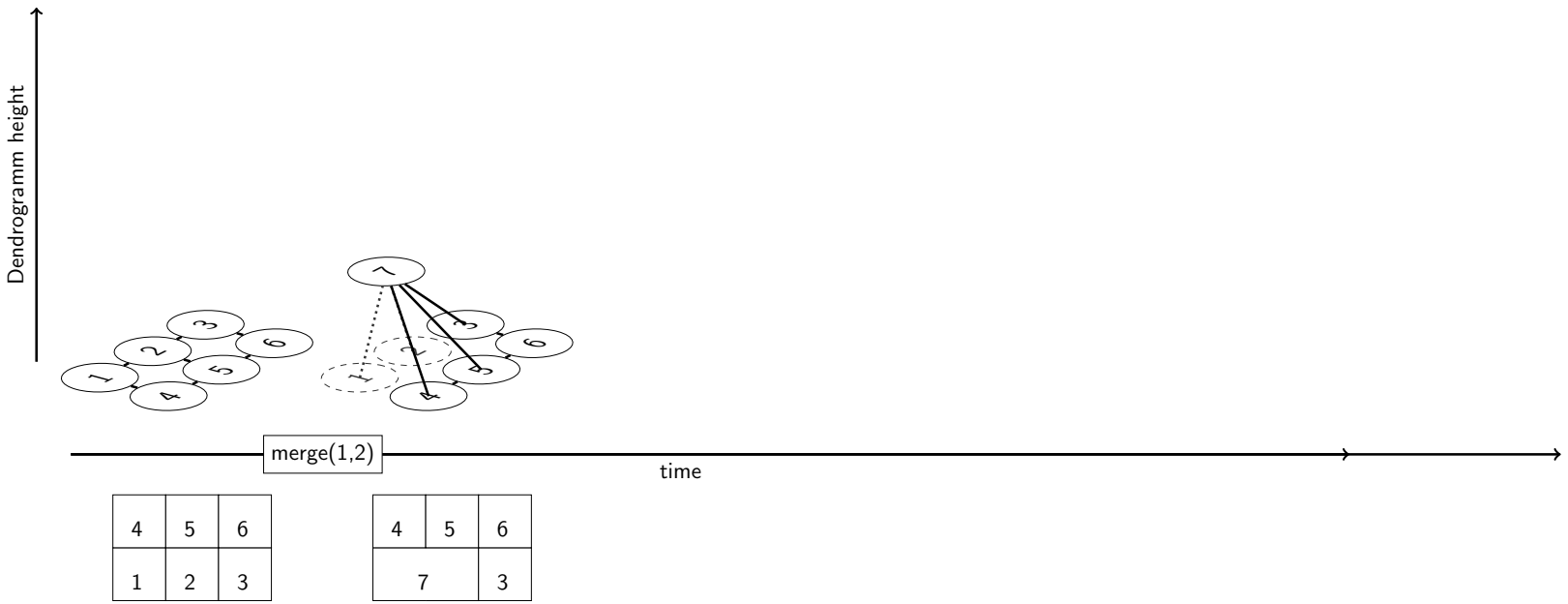
- Compute initial features for each edge
  - Statistics of raw/pmap accumulated along edges
  - ...
- Train initial random forest & Predict probabilities all edges
- Compute new features on probabilities
  - ???
- Stack new features with old features & train improved random forest

# Reminder: Agglomerative Clustering

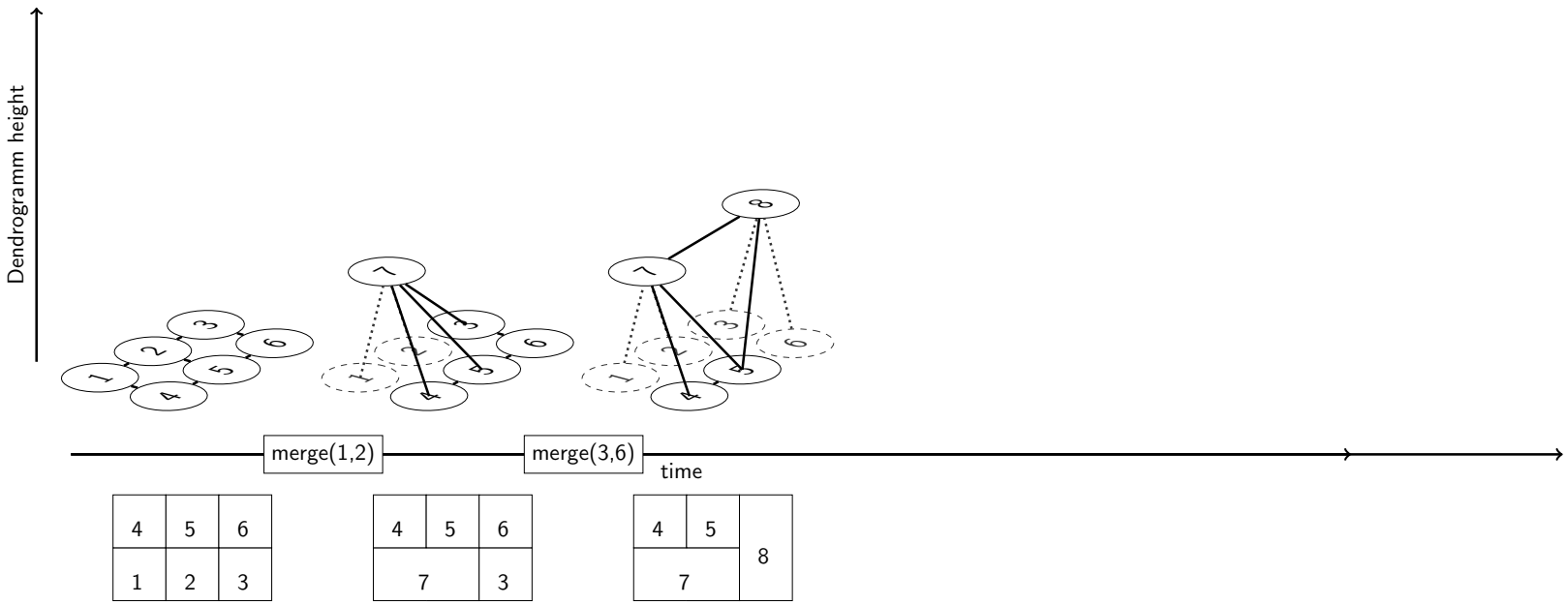



| | | |
|---|---|---|
| 4 | 5 | 6 |
| 1 | 2 | 3 |

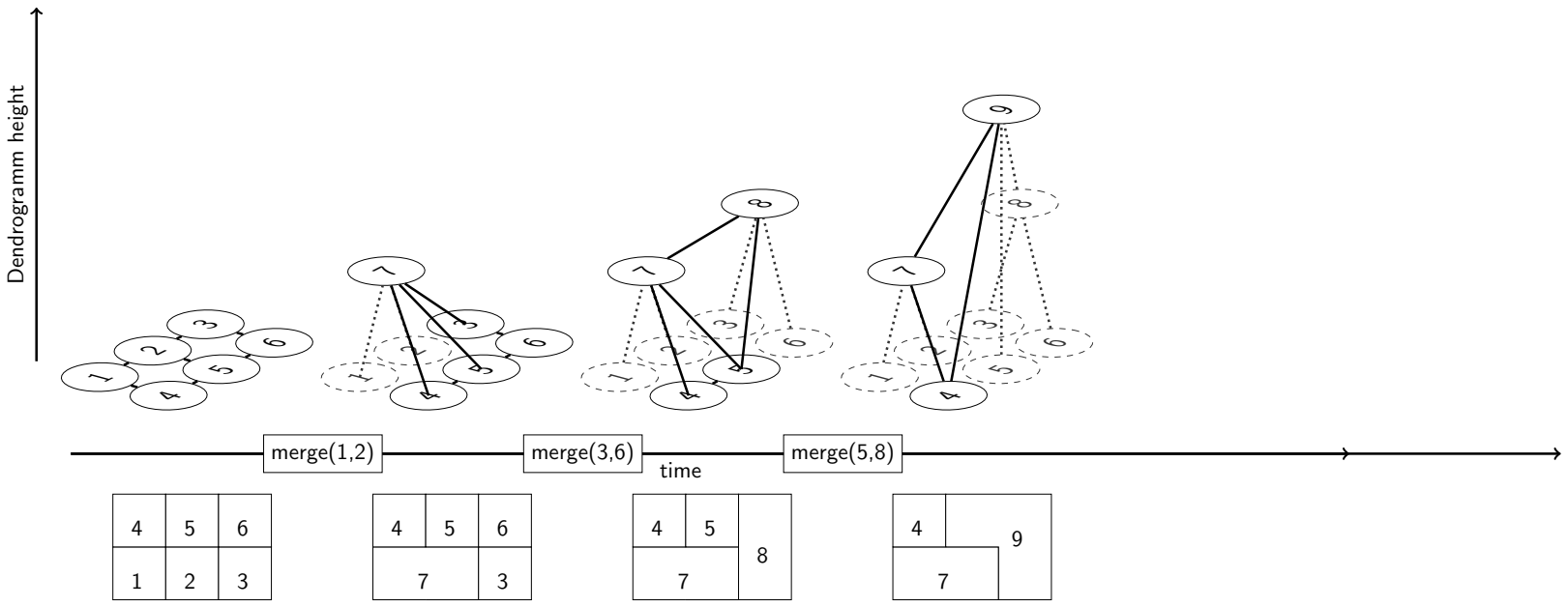
# Reminder: Agglomerative Clustering



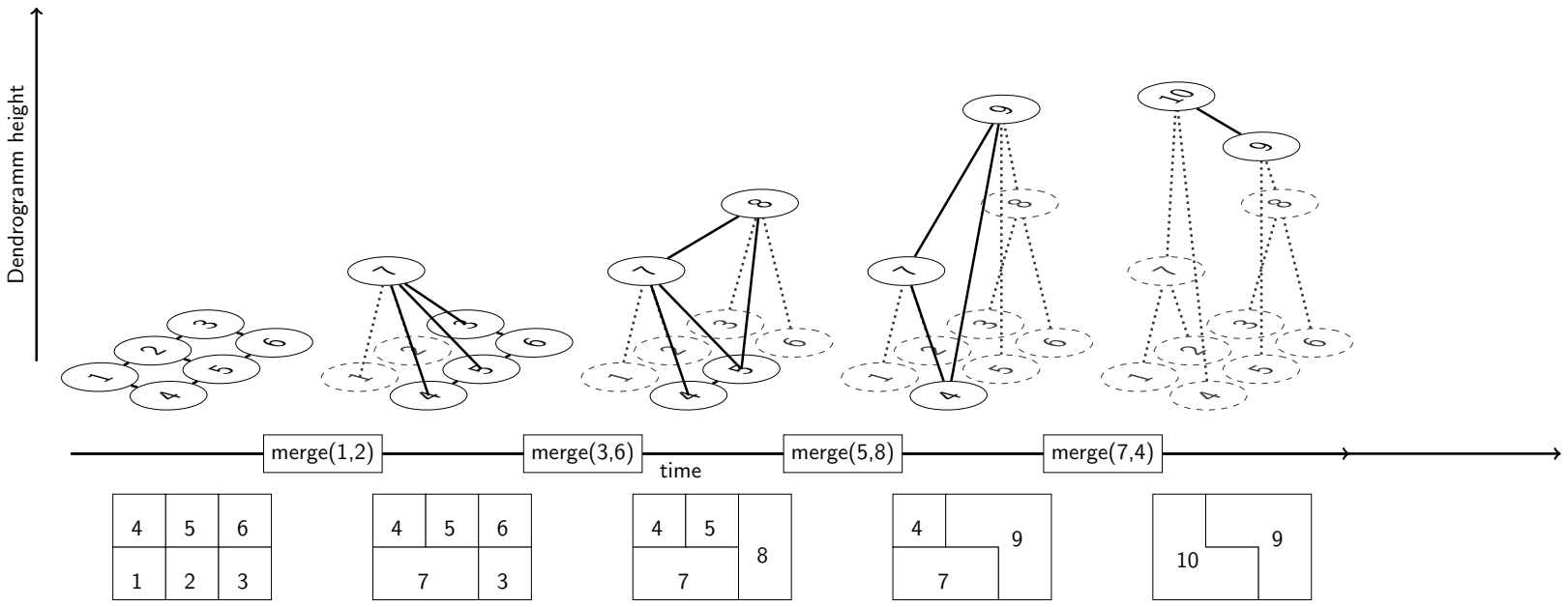
# Reminder: Agglomerative Clustering



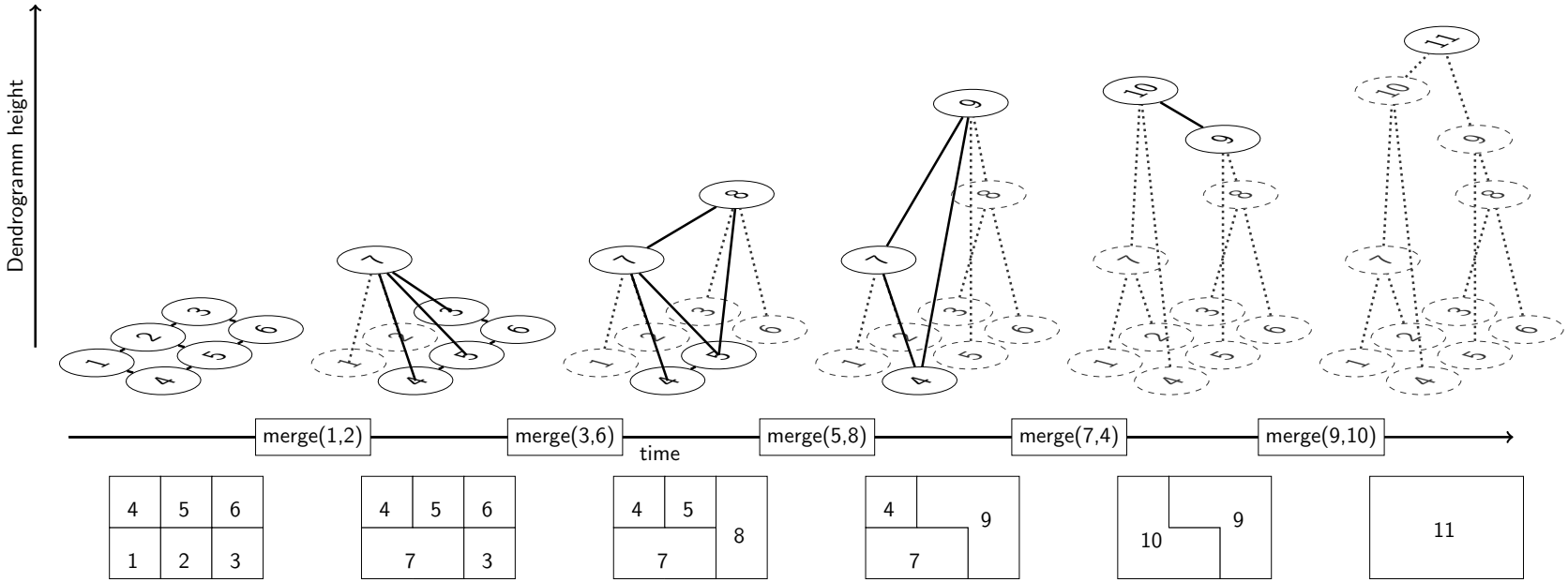
# Reminder: Agglomerative Clustering



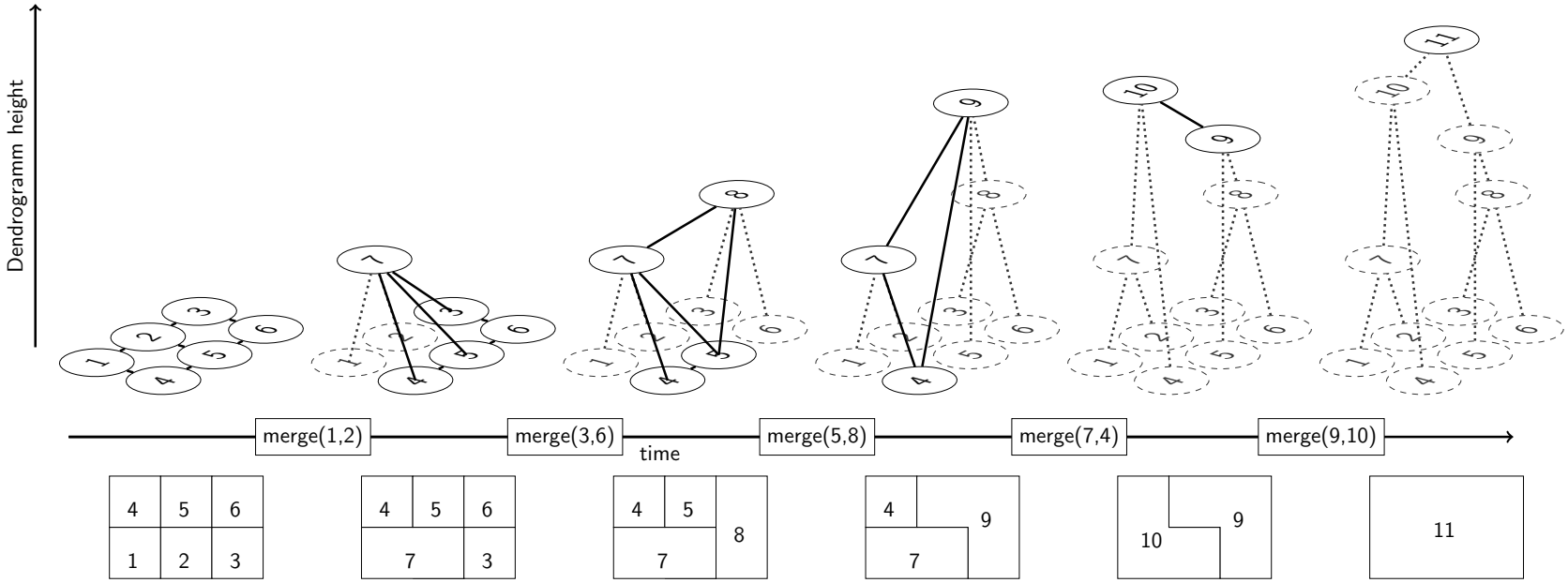
# Reminder: Agglomerative Clustering



# Reminder: Agglomerative Clustering



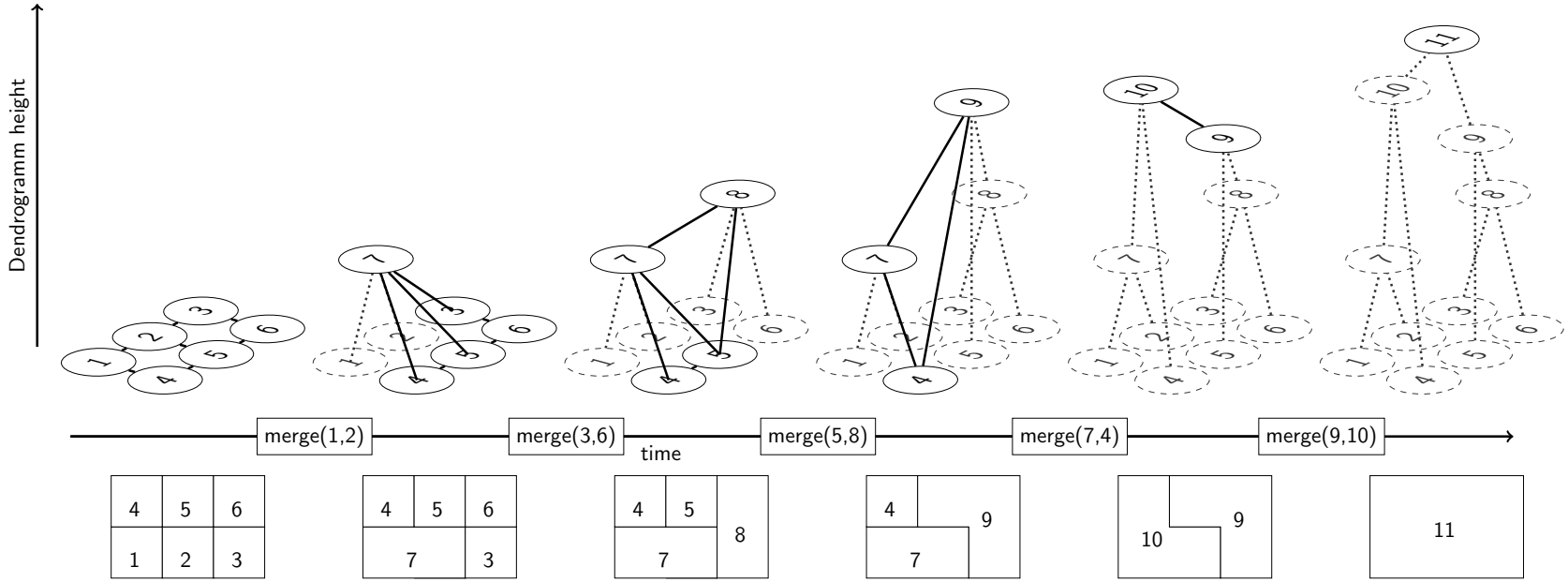
# Reminder: Agglomerative Clustering




Contract edge with lowest dissimilarity:

- Size Regularized Cluster Criterion:
  $\text{dissimilarity}_e = p_e \cdot \frac{2}{\left(\frac{1}{|u_e|}\right)^\gamma + \left(\frac{1}{|v_e|}\right)^\gamma}$

# Reminder: Agglomerative Clustering




Contract edge with lowest dissimilarity:

- Size Regularized Cluster Criterion:
  $\text{dissimilarity}_e = p_e \cdot \frac{2}{\left(\frac{1}{|u_e|}\right)^\gamma + \left(\frac{1}{|v_e|}\right)^\gamma}$

UCM transform / Use dendrogramm height as feature

- Remember the height in the dendrogramm when the endpoints are merged.
- "Edgelet"-Averaging

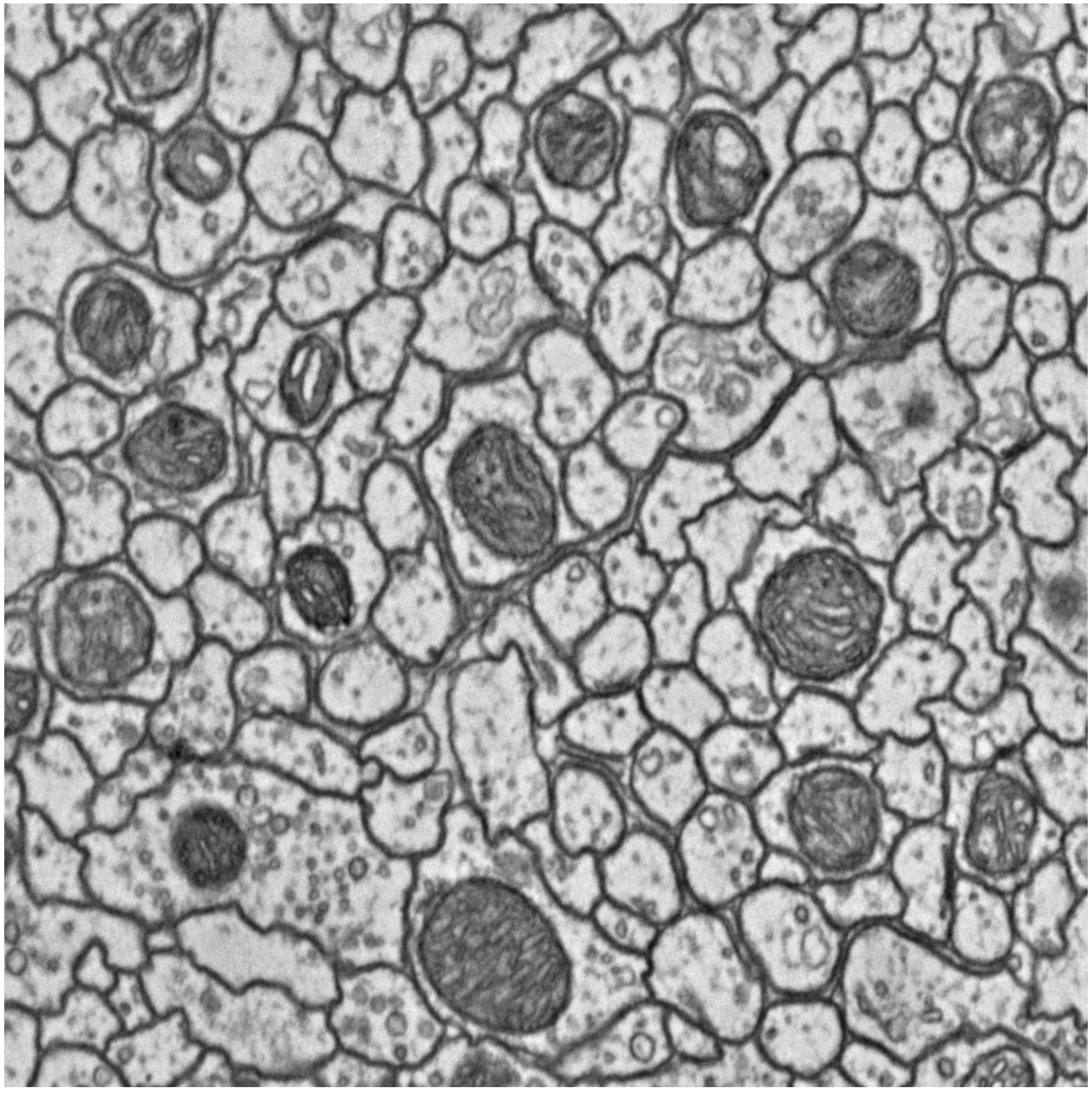


Figure: Raw Data

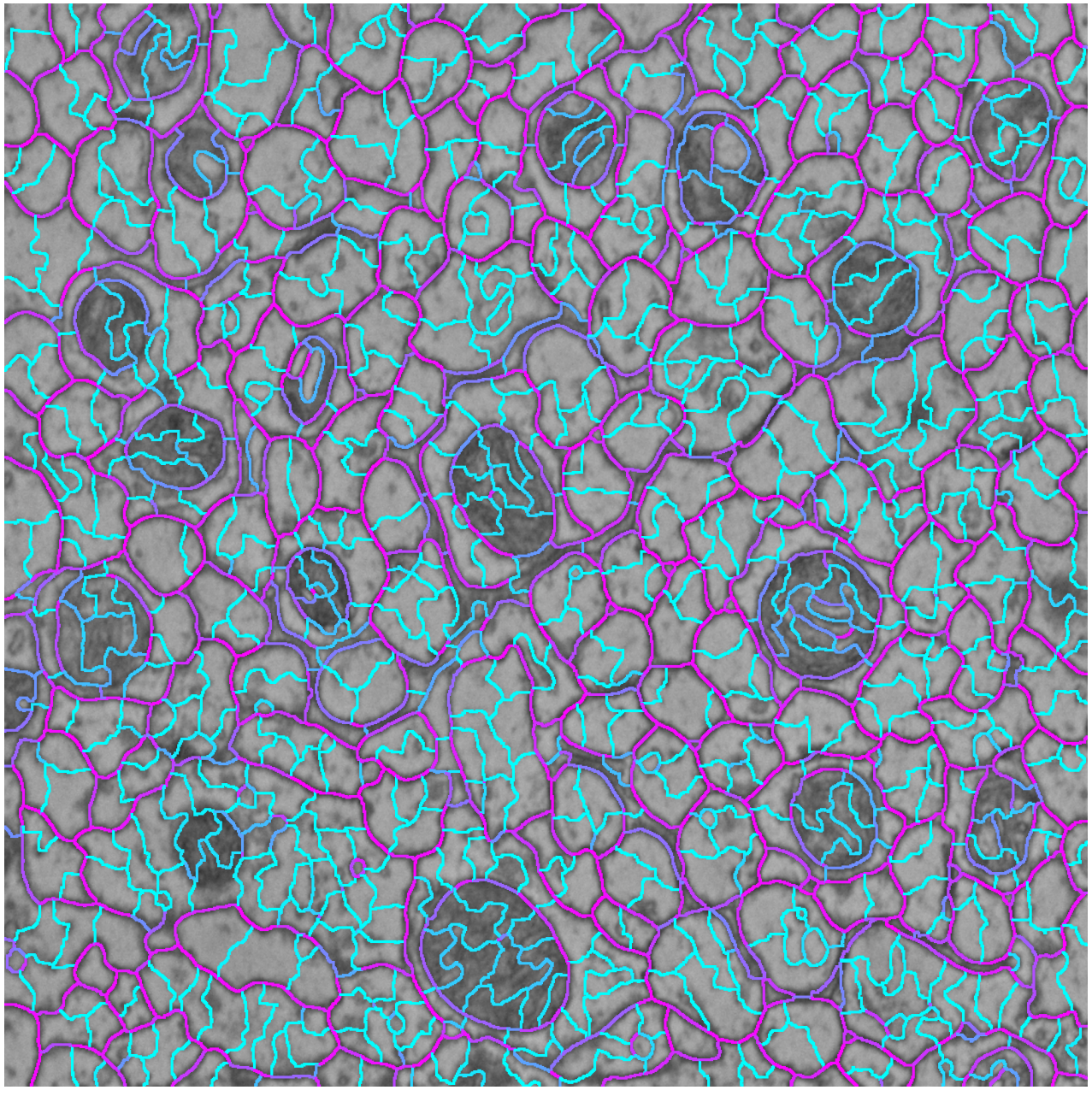


Figure: Vanilla RF on edges

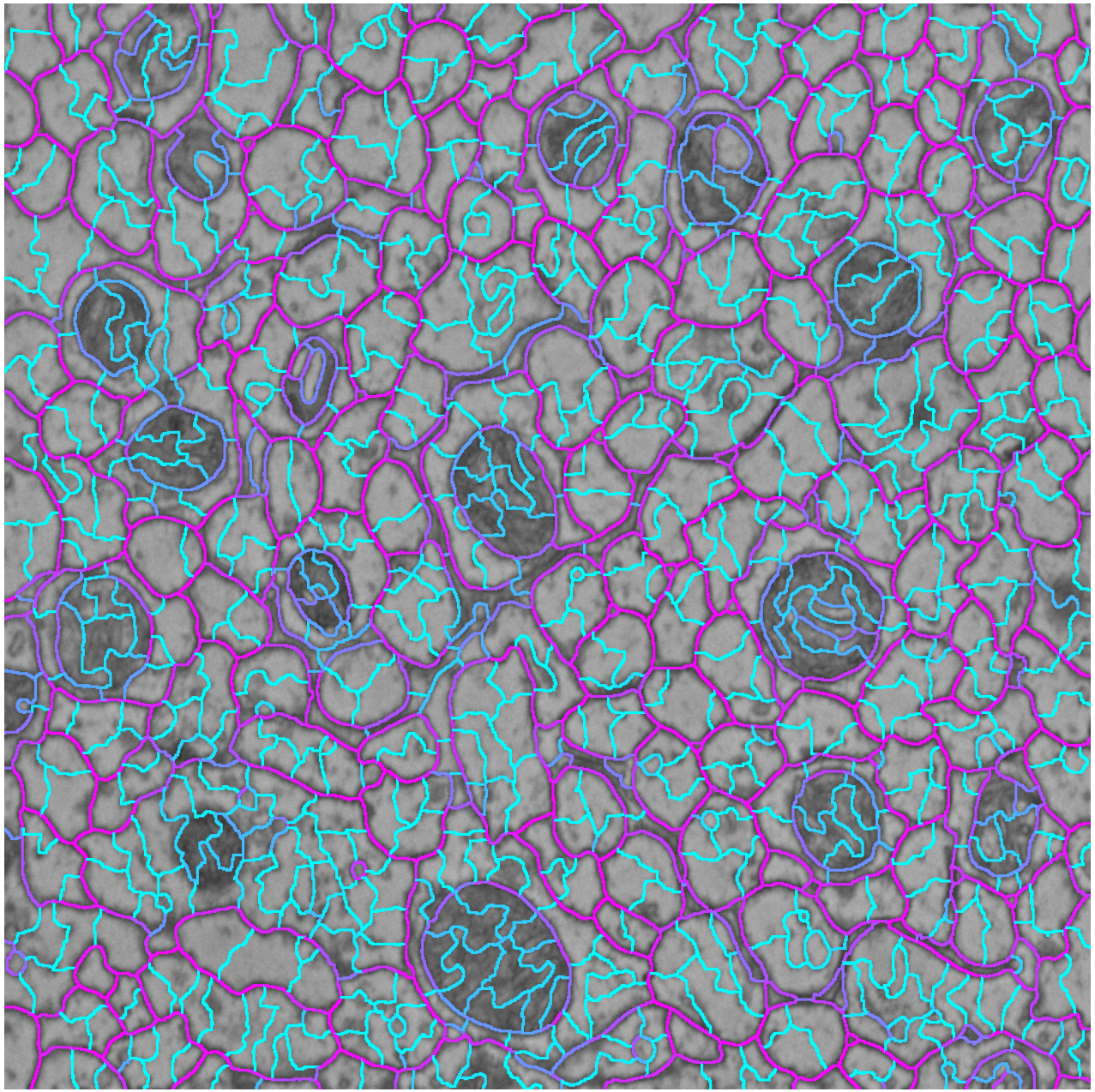


Figure: Vanilla RF on junctions

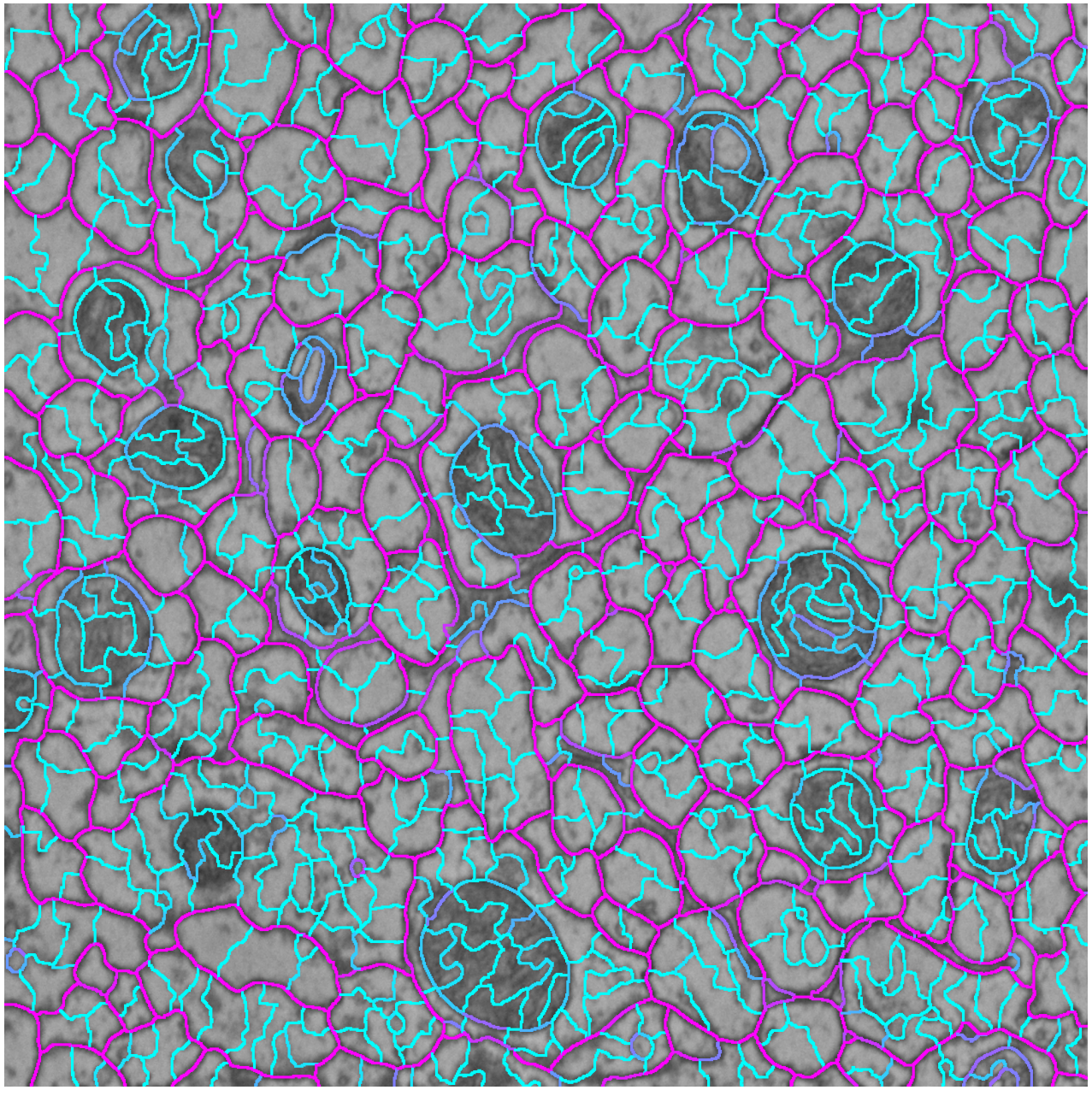


Figure: Autocontext RF on edges

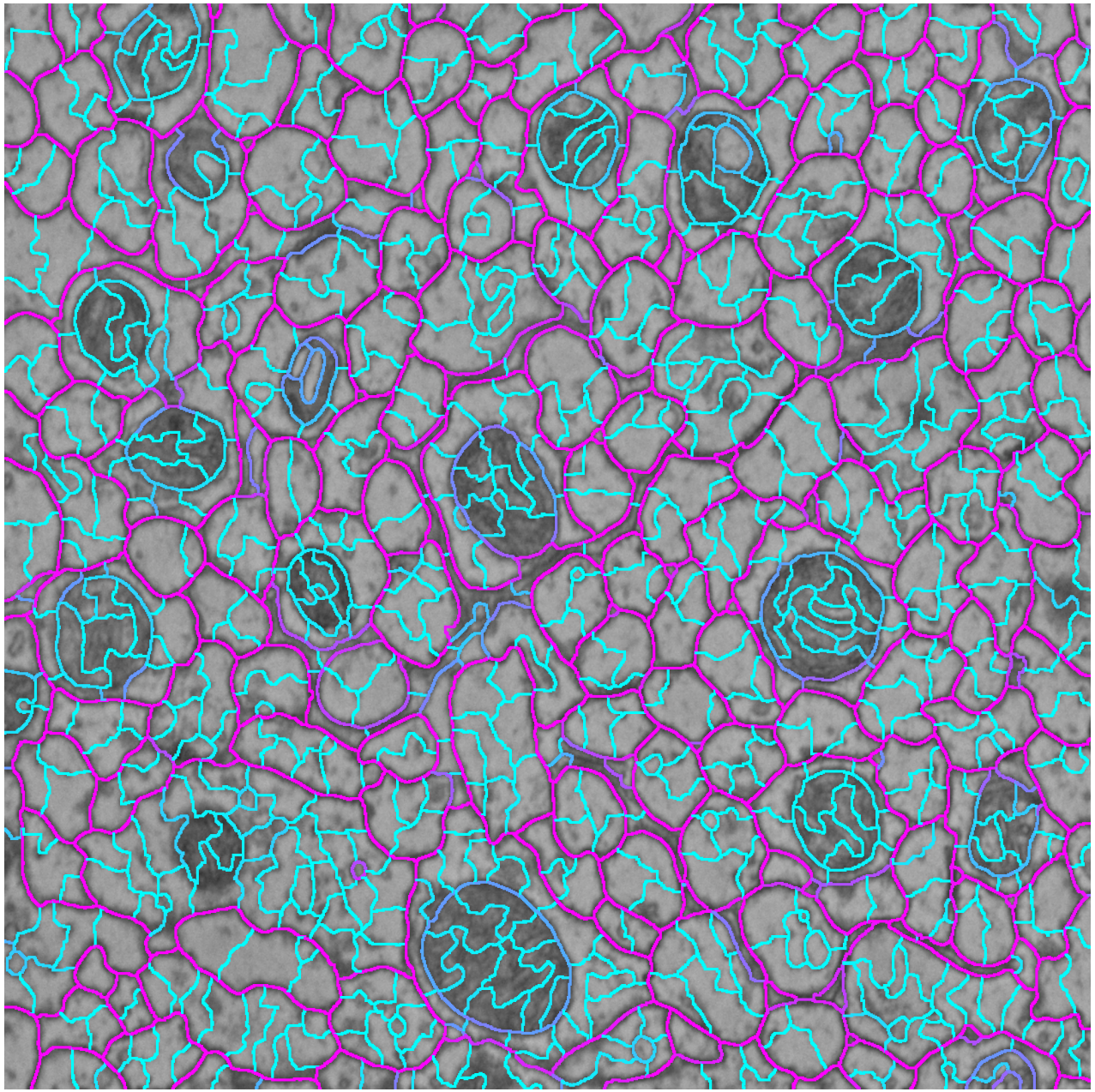


Figure: Autocontext RF on junctions

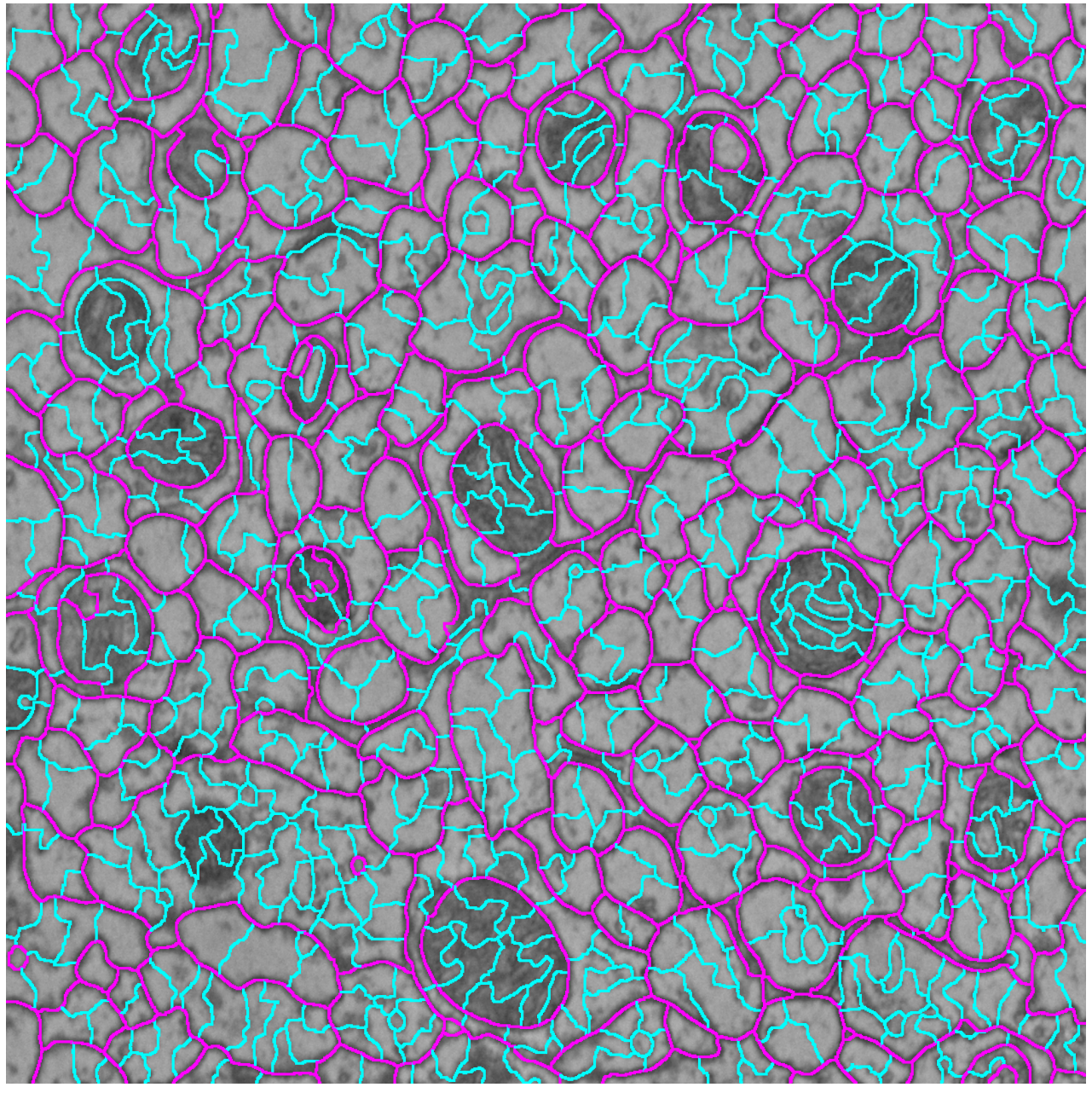


Figure: Vanilla RF on edges + Multicut

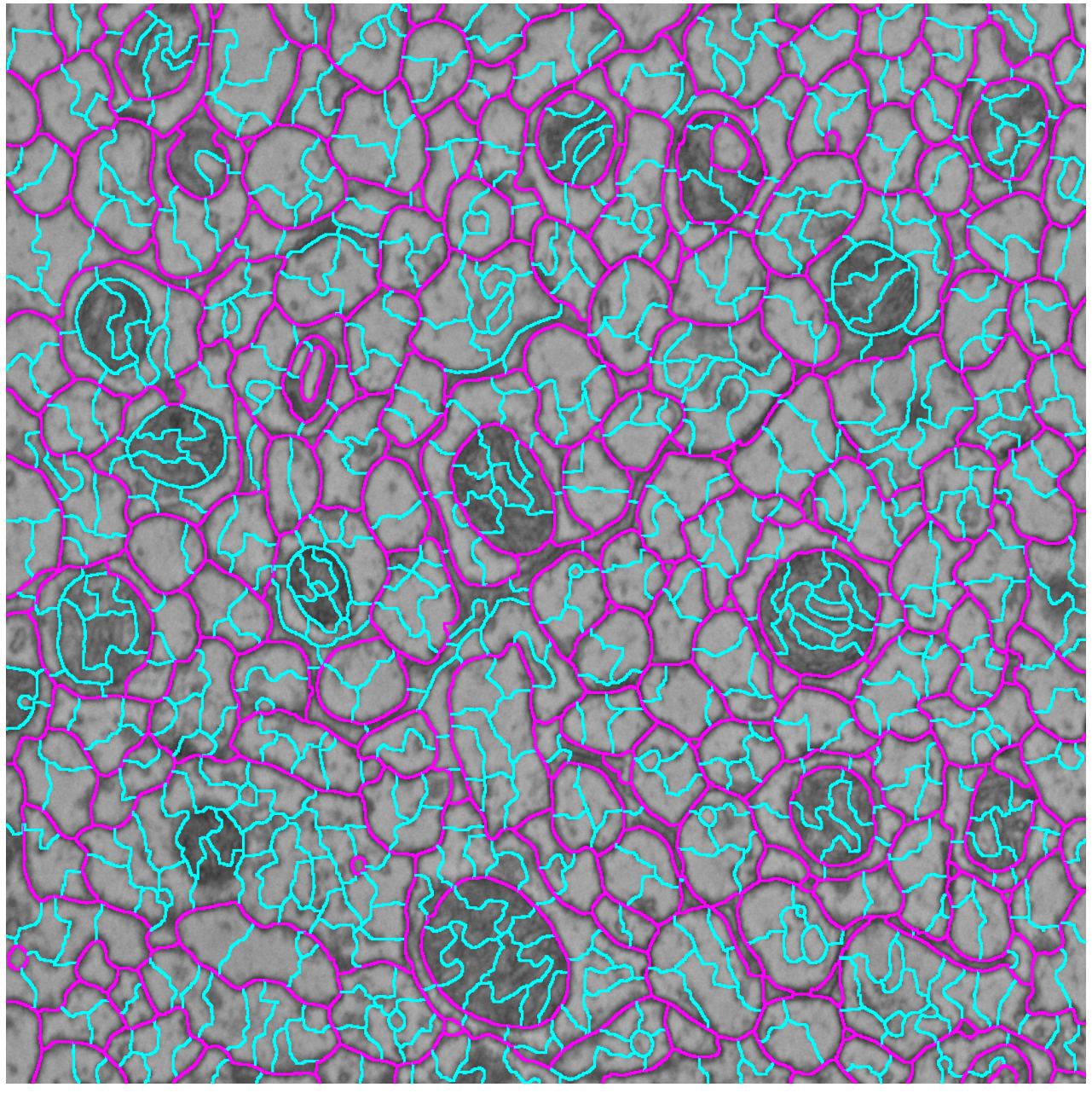


Figure: Vanilla RF on junctions + Multicut

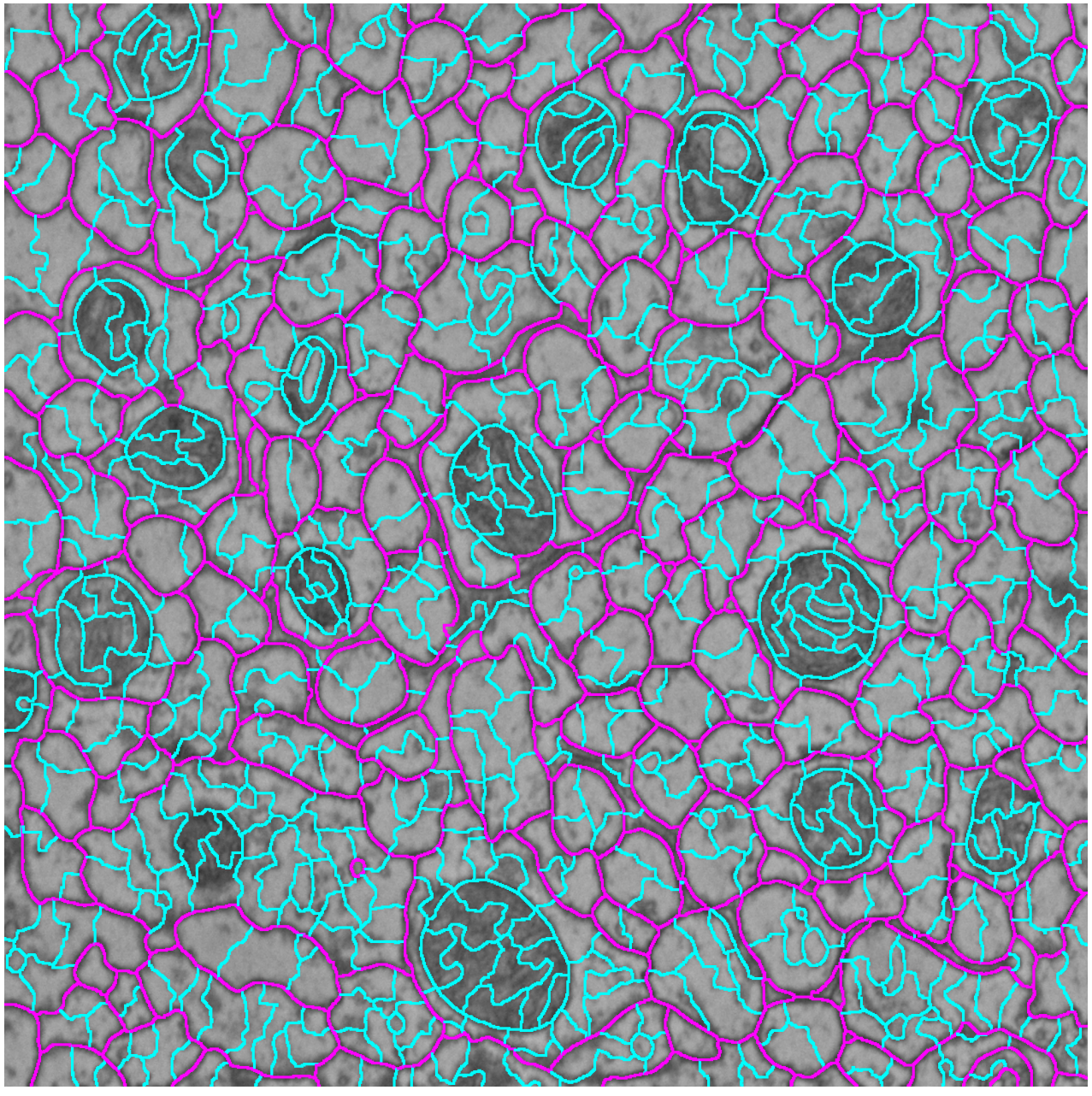


Figure: Autocontext RF on edges + Multicut

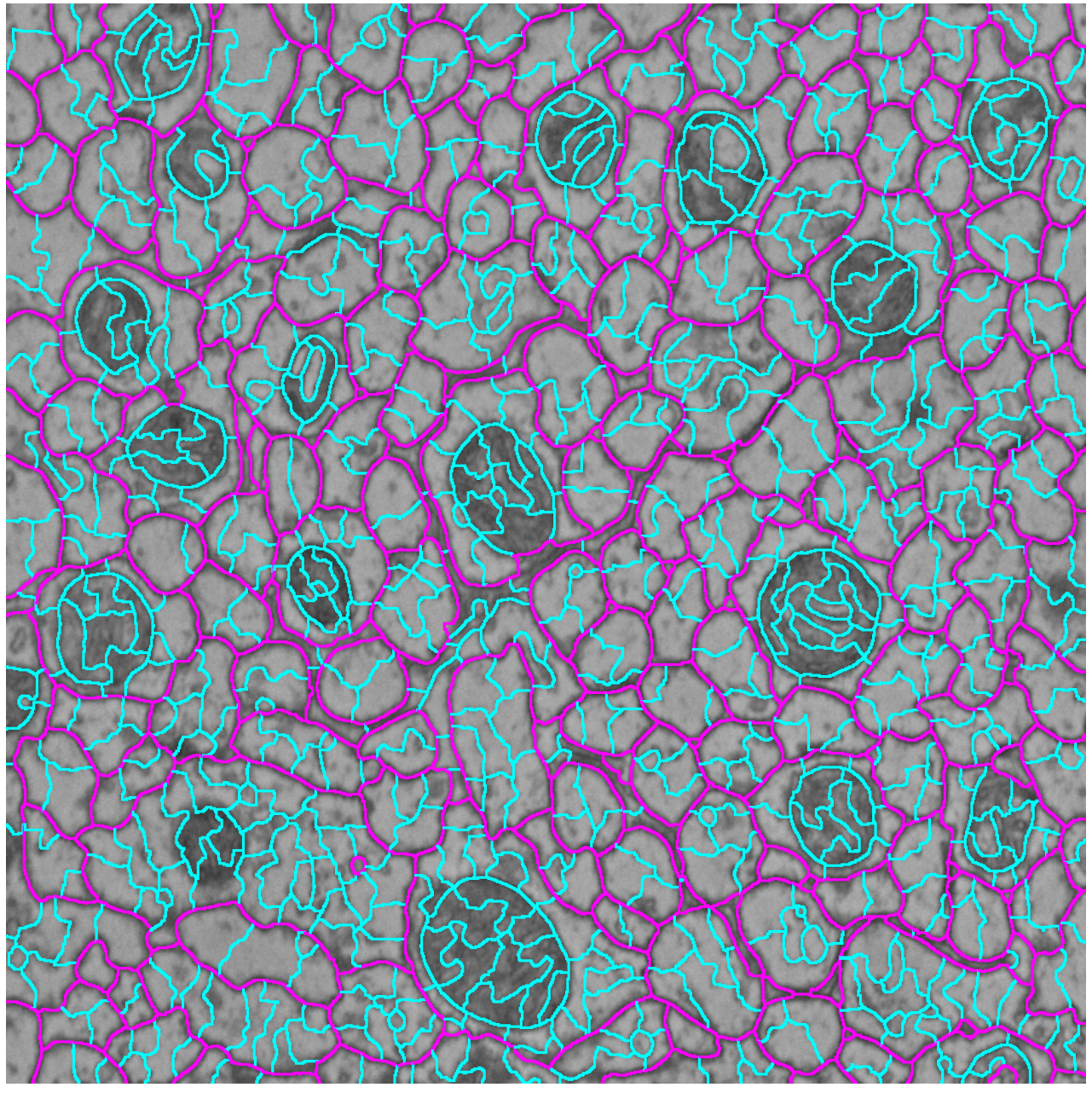
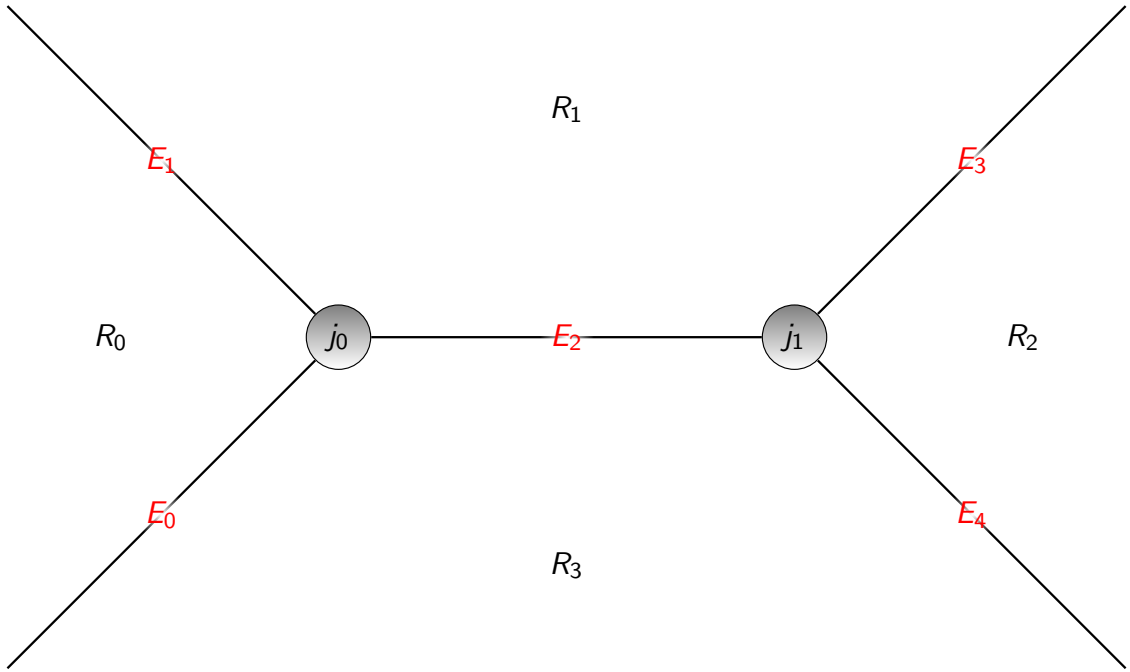


Figure: Autocontext RF on junctions + Multicut

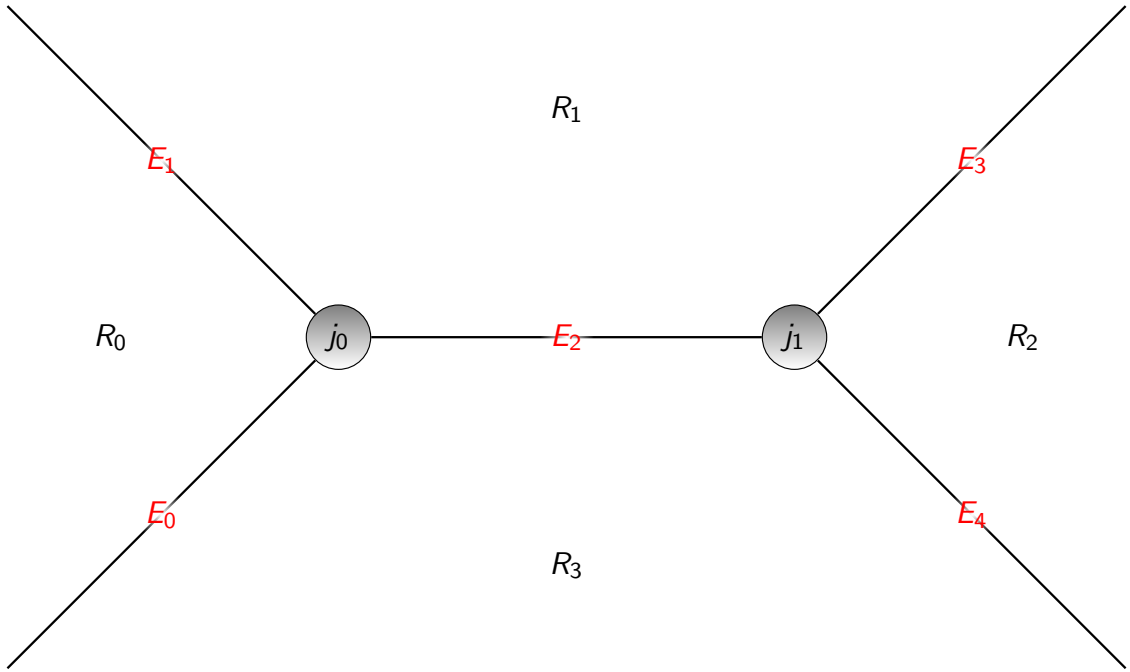
# Outlook

- Joint prediction of junction's edges is helpful: $\rightarrow$ make it deep.

# Deep Learning of Joint Edge Probability

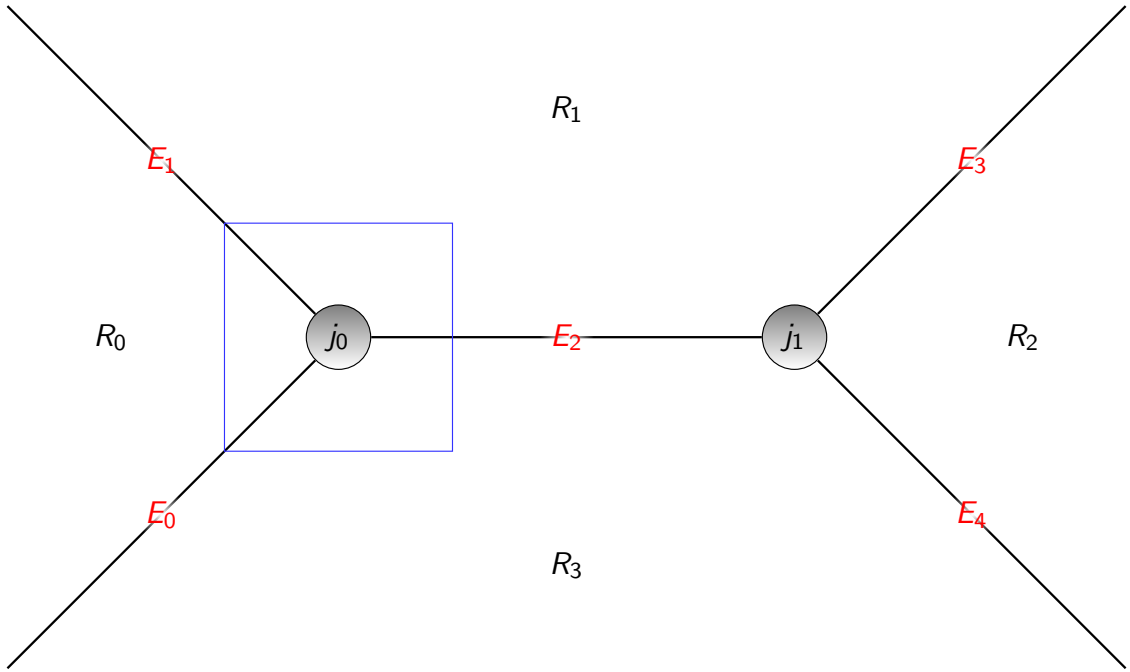


# Deep Learning of Joint Edge Probability



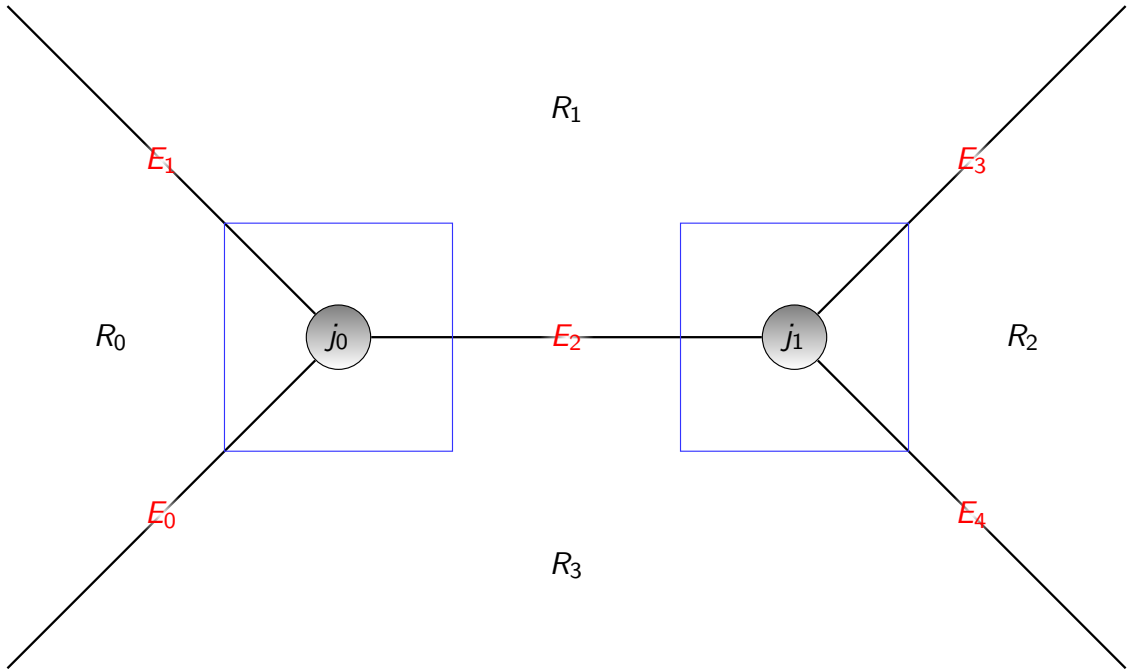

- Use raw data in bounding box around each junction as input

# Deep Learning of Joint Edge Probability



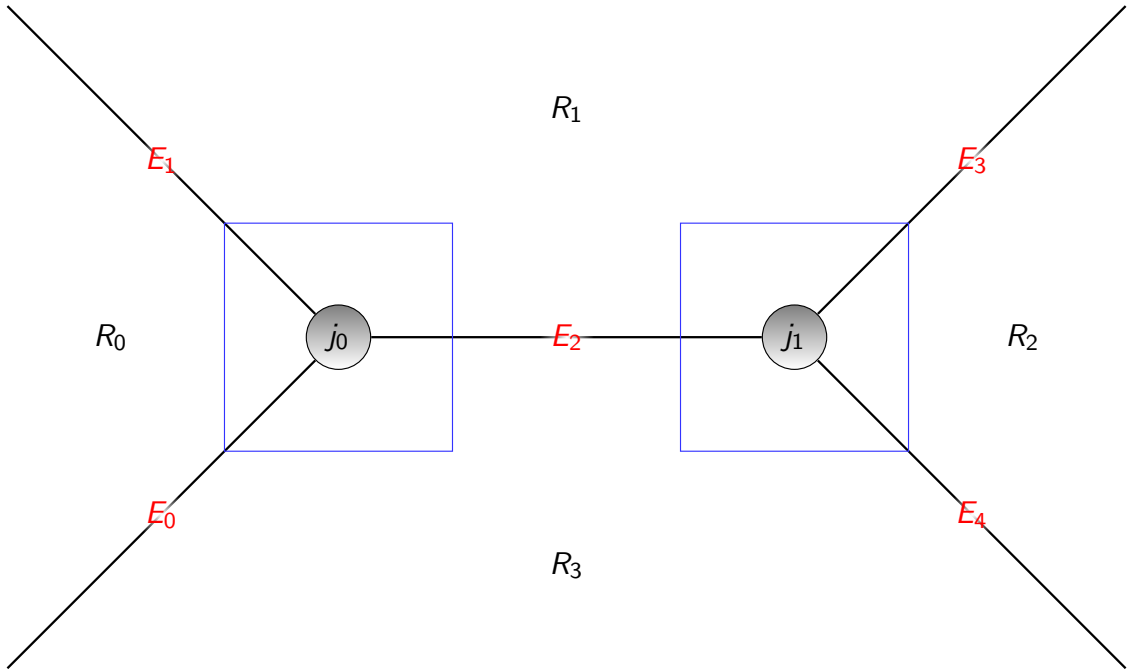

- Use raw data in bounding box around each junction as input

# Deep Learning of Joint Edge Probability



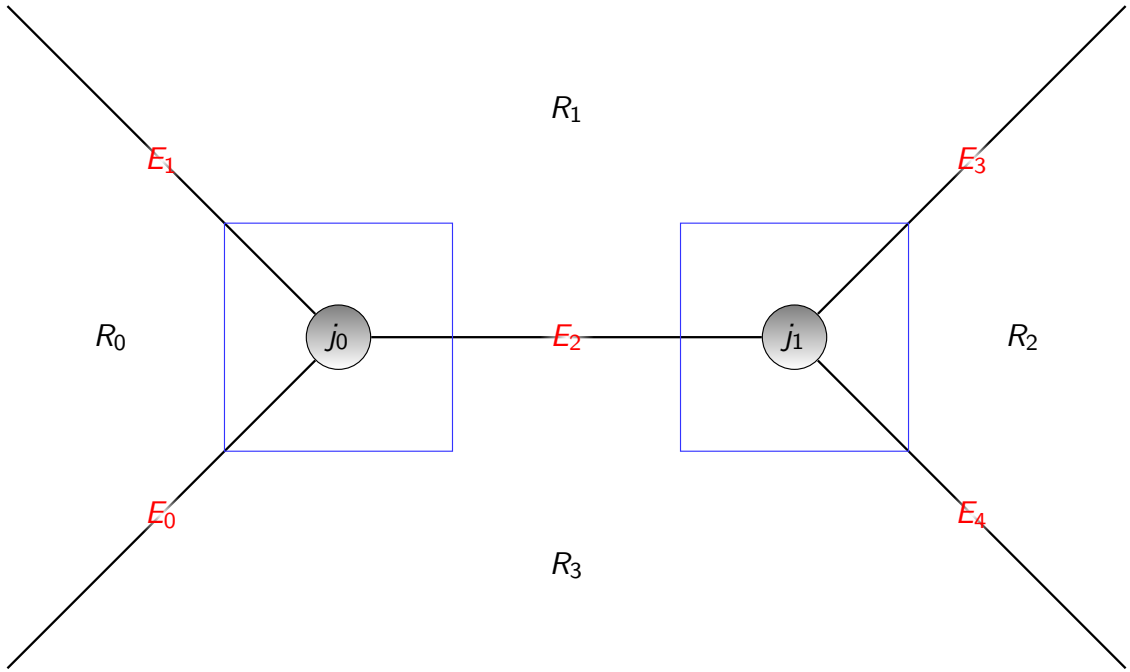

- Use raw data in bounding box around each junction as input

# Deep Learning of Joint Edge Probability




- Use raw data in bounding box around each junction as input
- Predict the joint edge configuration $\{000, 011, 101, 110, 111\}$ for each junction with a CNN (not fully convolutional)

# Deep Learning of Joint Edge Probability




- Use raw data in bounding box around each junction as input
- Predict the joint edge configuration $\{000, 011, 101, 110, 111\}$ for each junction with a CNN (not fully convolutional)
- Use precomputed edge features as additional input

# Joint Optimization

# Higher order Lifted Multicut

Graph $G = (E, V)$ Lifted Graph $G' = (V, E')$ with $E \subseteq E'$

**Higher order Lifted multicut:**

$$y* = \quad argmin_y \sum_{e \in E} w_e \cdot y_e + \sum_{e \in E' \setminus E} w_e \cdot y_e + \sum_{j=\{y_i, y_j, y_k\} \in J} \phi_j(y_i, y_j, y_k)$$

$$s.t \qquad y \in \text{LMC(G,G')}$$

**Higher order Lifted multicut:**

$$E(y) = \sum_{e \in E} w_e \cdot y_e + \sum_{e \in E' \setminus E} w_e \cdot y_e + \sum_{j=\{y_i, y_j, y_k\} \in J} \phi_j(y_i, y_j, y_k)$$

$$s.t \quad y \in \text{LMC(G,G')}$$

**Solve via Dual Decomposition** $E(y) = G(y) + H(y) \quad y^G = y^H$:

**Lifted Multicut Problem:**

$$G(y^G) = \sum_{e \in E} w_e^G \cdot y_e^G + \sum_{e \in E' \setminus E} w_e \cdot y_e$$

$$s.t \quad y \in \text{LMC(G,G')}$$

**Higher order Multicut Problem:**

$$H(y^H) = \sum_{e \in E} w_e^H \cdot y_e^H + \sum_{j=\{y_i, y_j, y_k\} \in J} \phi_j(y_i, y_j, y_k)$$

$$s.t \quad y \in \text{LMC(G,G')}$$

# The End