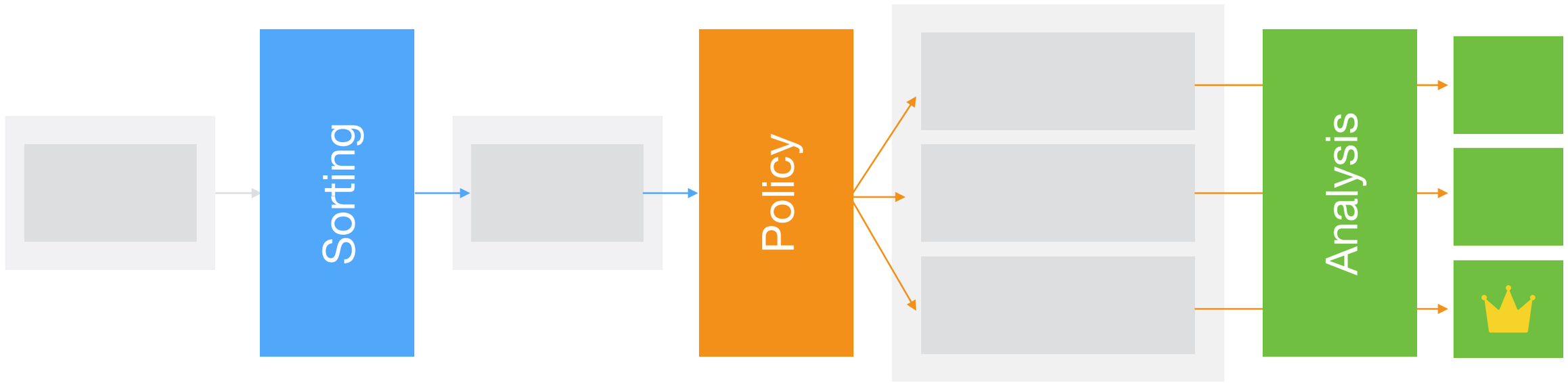


Reinforced Sorting Networks for Particle Physics Analyses

Martin Erdmann, Benjamin Fischer, Dennis Noll, Yannik Rath,
Marcel Rieger, David Schmidt, Marcus Wirtz

SPONSORED BY THE
Federal Ministry of Education and Research

ACAT 2019

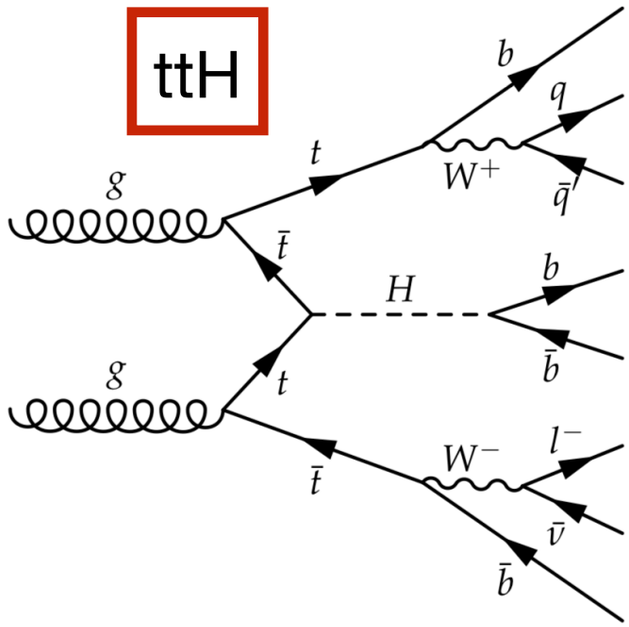
14 March 19

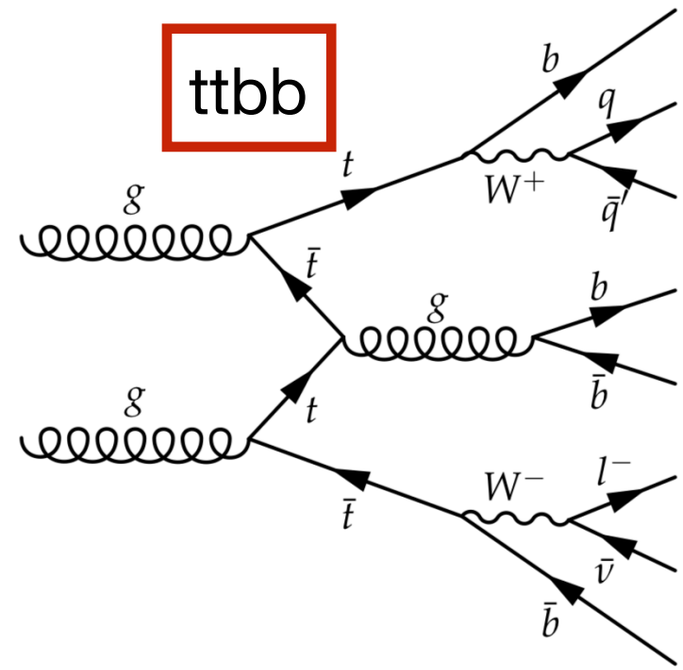
RWTH AACHEN UNIVERSITY

Motivation

Dennis Noll - 14 March 19

- Physics analyses often highly depend on the order of their input variables
  - Example: Classification - ttH vs ttbb (6 jets, lepton, neutrino, Pythia+Delphes)
  - Usual approach: p_T order





Order	Accuracy [%]
Gen	87.3
p_T	69.5

Not optimal

talk by Yannik

Approach

- Idea: Train mechanism (Neural Network) which outputs the best permutation

Problem 1: Many possible permutations (n!)

Solution: Divide & conquer

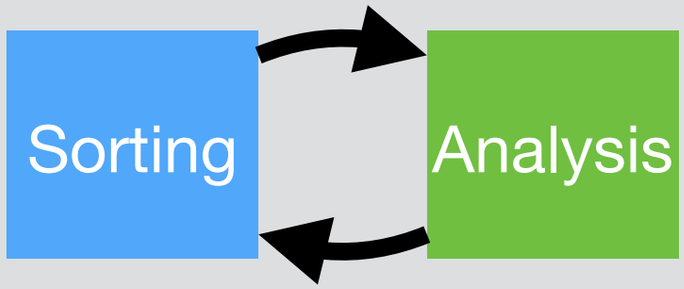
1. Look at a subset of possible permutations
2. Explore subset
3. Choose next subset

=> Continue until best permutation is found

Problem 2: Best permutation generally not known

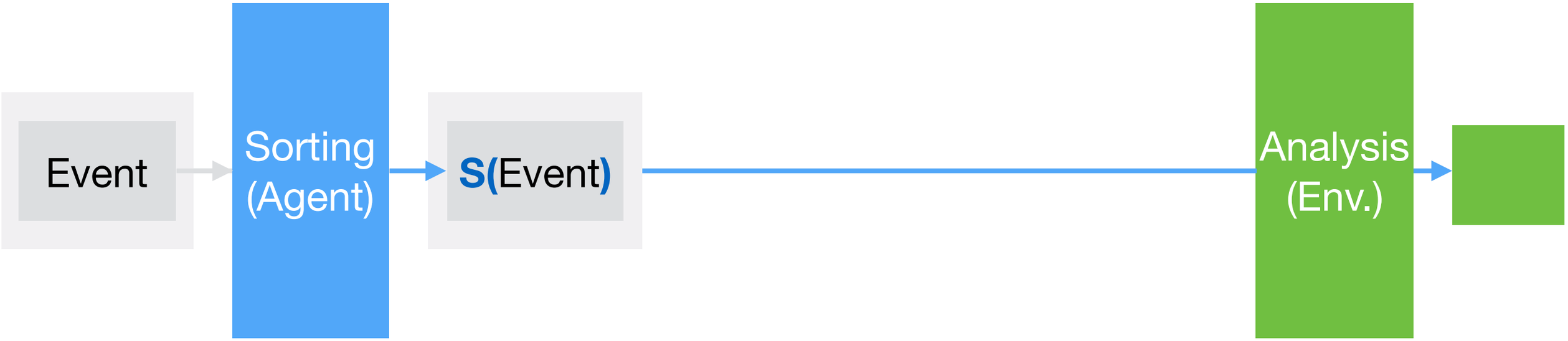
Solution: Reinforcement approach

- Simultaneously train analysis and sorting
  - Train sorting with analysis feedback
  - Train analysis with sorting feedback



Complete Mechanism - Evaluation

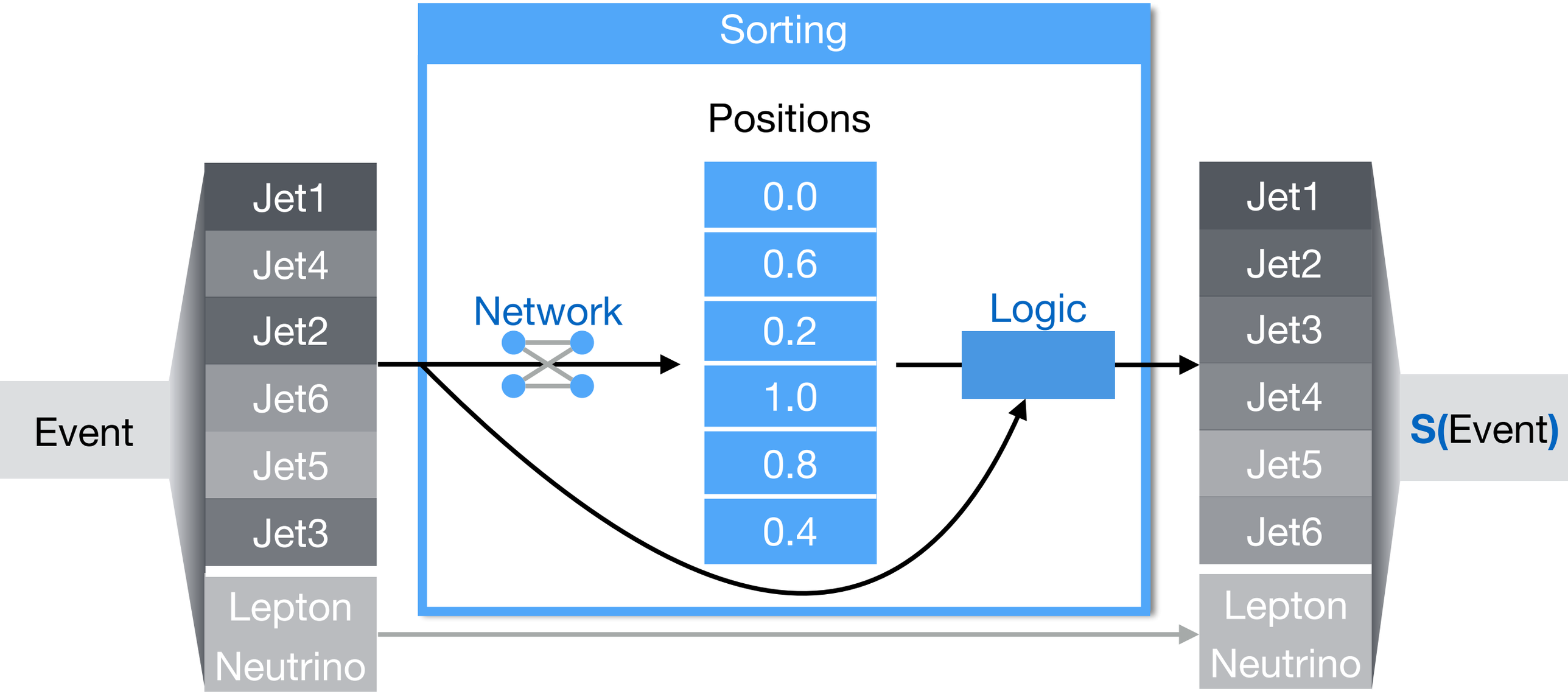
RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19



- Evaluation:
  - Input p_T sorted events
  - Sorting network sorts particles **S(**Event**)**
  - Analysis evaluates sorted events (trainable)

Sorting Mechanism

RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19



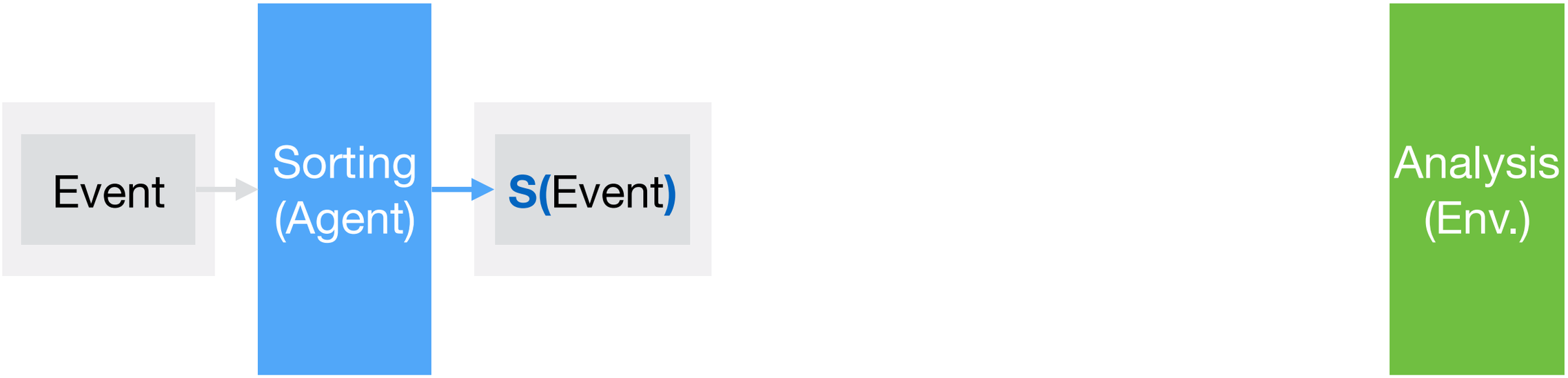
Network outputs position for every jet:

- Small position: front
- Large position: end
- Uniformly between 0 and 1

Logic algorithm applies the positions and rearranges the jets

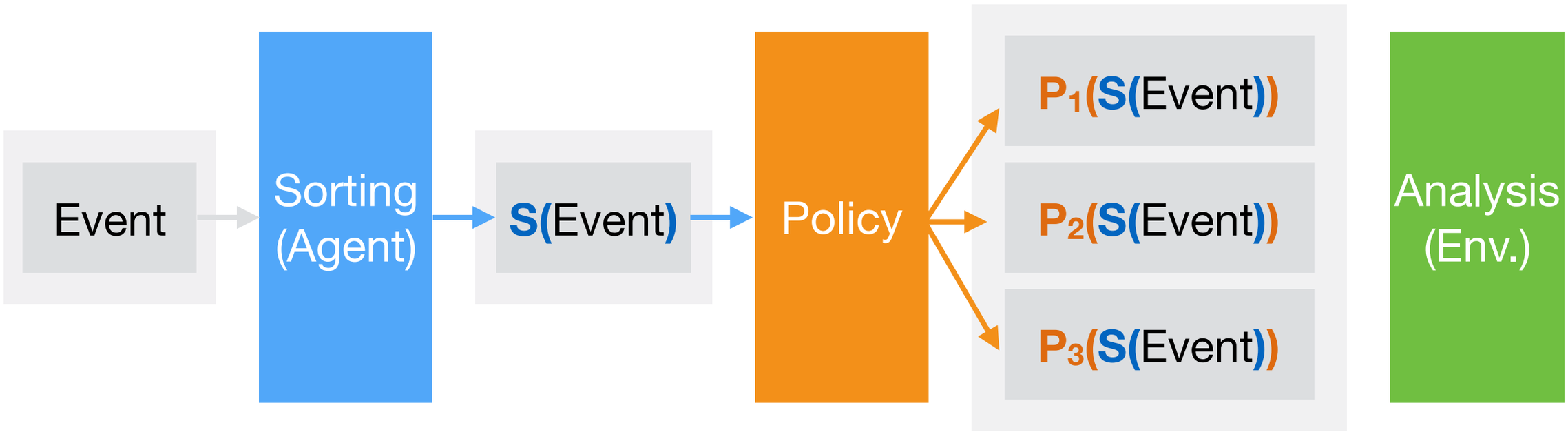
Complete Mechanism - Reinforced Training

RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19



- Training:
  - Sorting network sorts particles S(Event)
  - Policy suggests new orderings P_1(S(Event)), P_2(S(Event)), P_3(S(Event))
    - Subset of possible orderings based on current ordering
  - Analysis evaluates sorted (and permuted) events

Complete Mechanism - Reinforced Training

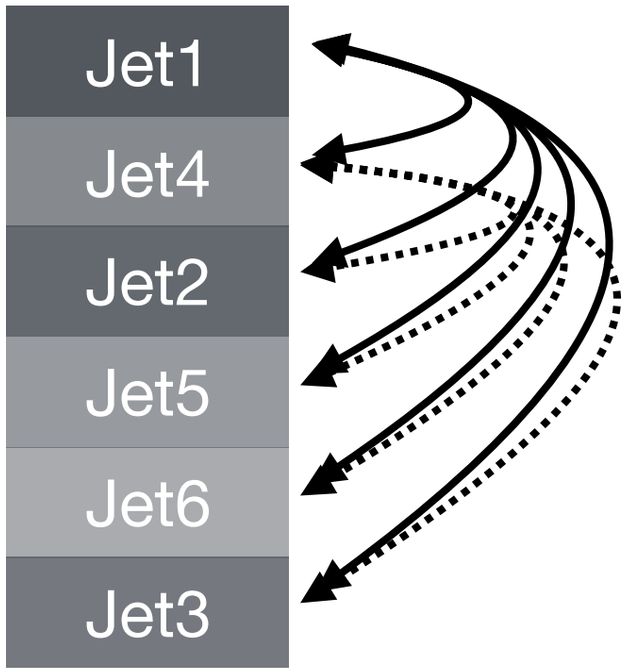


- Training:
  - Sorting network sorts particles S(Event)
  - Policy suggests new orderings P_1(S(Event)), P_2(S(Event)), P_3(S(Event))
    - Subset of possible orderings based on current ordering
  - Analysis evaluates sorted (and permuted) events

Permutation Policy

- Two different permutation policies:
  - Basic approach (Pool) and extension (Tree)

Pool $\mathbb{P}$



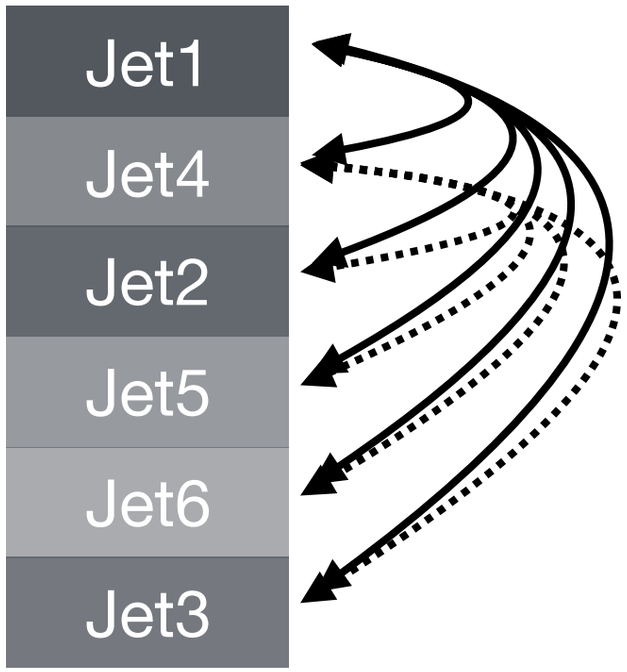
- All pair-wise switches
  - $N_{pool}(n) = \mathcal{O}(n^2)$
  - $N_{pool}(6) = 15$
- Policy (Pool):
  - Draw from pool $P_i \in \mathbb{P}$

Permutation Policy

RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19

- Two different permutation policies:
  - Basic approach (Pool) and extension (Tree)

Pool $\mathbb{P}$

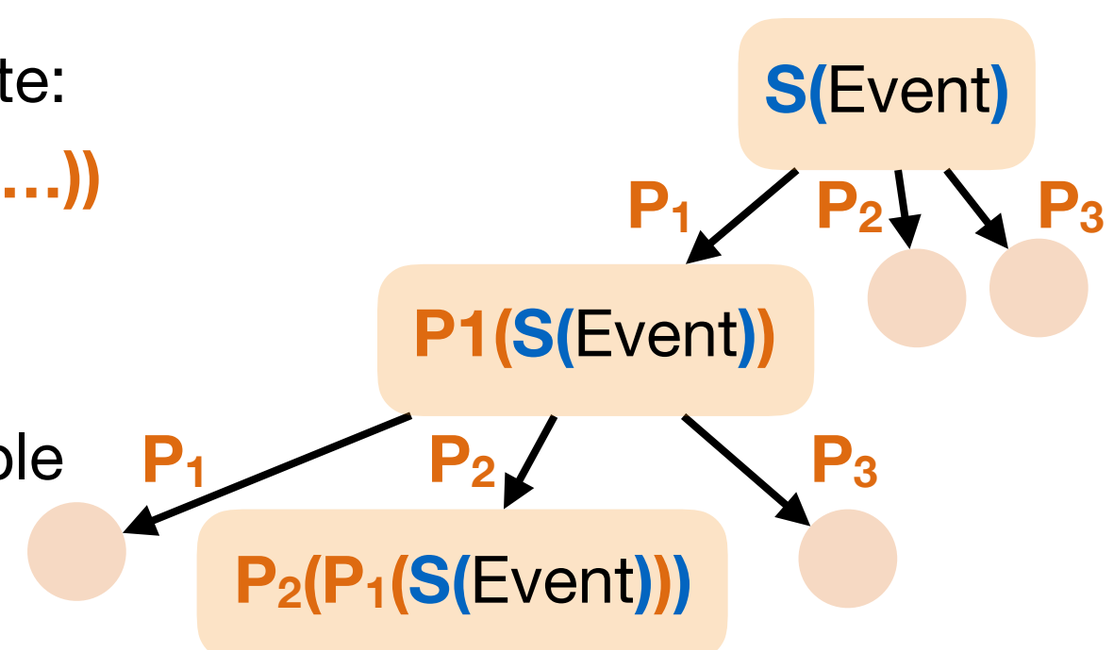


- All pair-wise switches
  - $N_{pool}(n) = \mathcal{O}(n^2)$
  - $N_{pool}(6) = 15$
- Policy (Pool):
  - Draw from pool $P_i \in \mathbb{P}$

Tree

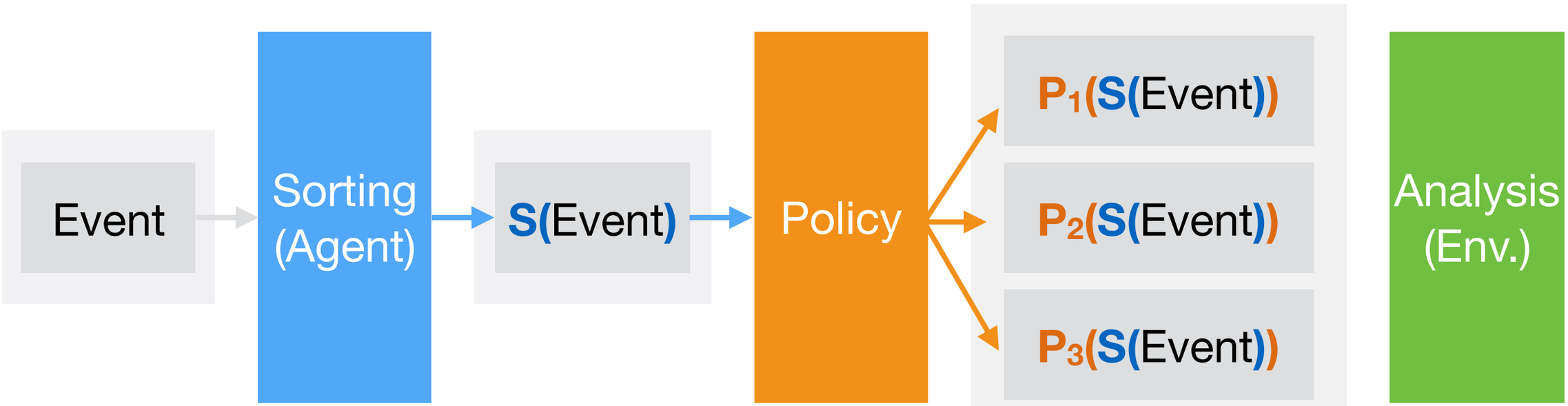
- Chain permutations to reach every state:

  $P_{any}(S(\text{Event})) = P_z(P_y(\ldots(P_x(S(\text{Event})))\ldots))$
- Policy (Tree):
  - Draw from pool ($\mathbb{P}$) in every node
  - Use tree mechanics to reach valuable nodes

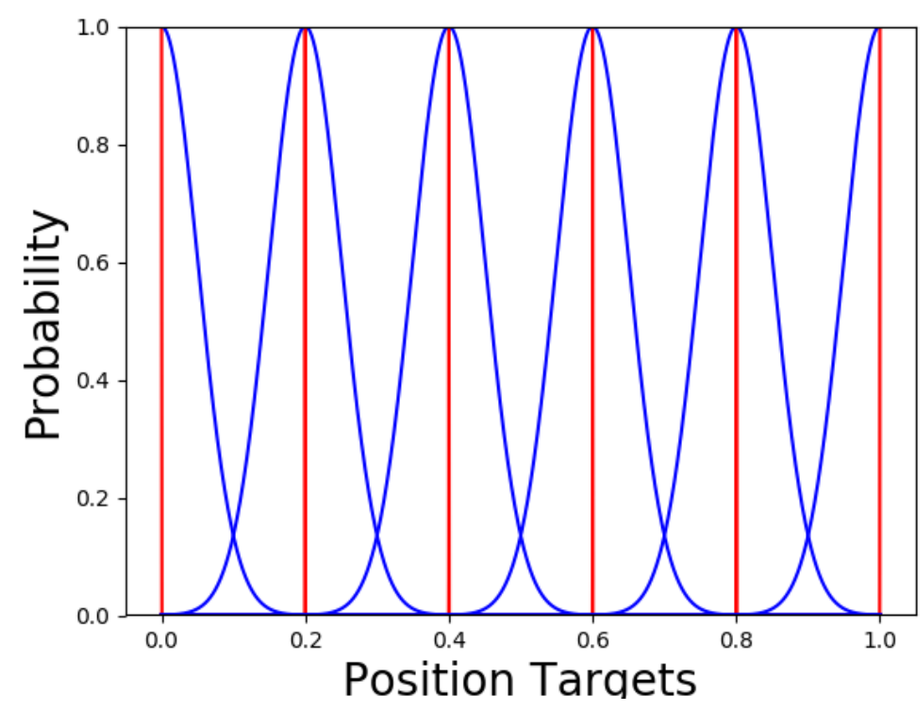


Complete Mechanism - Reinforced Training

Training

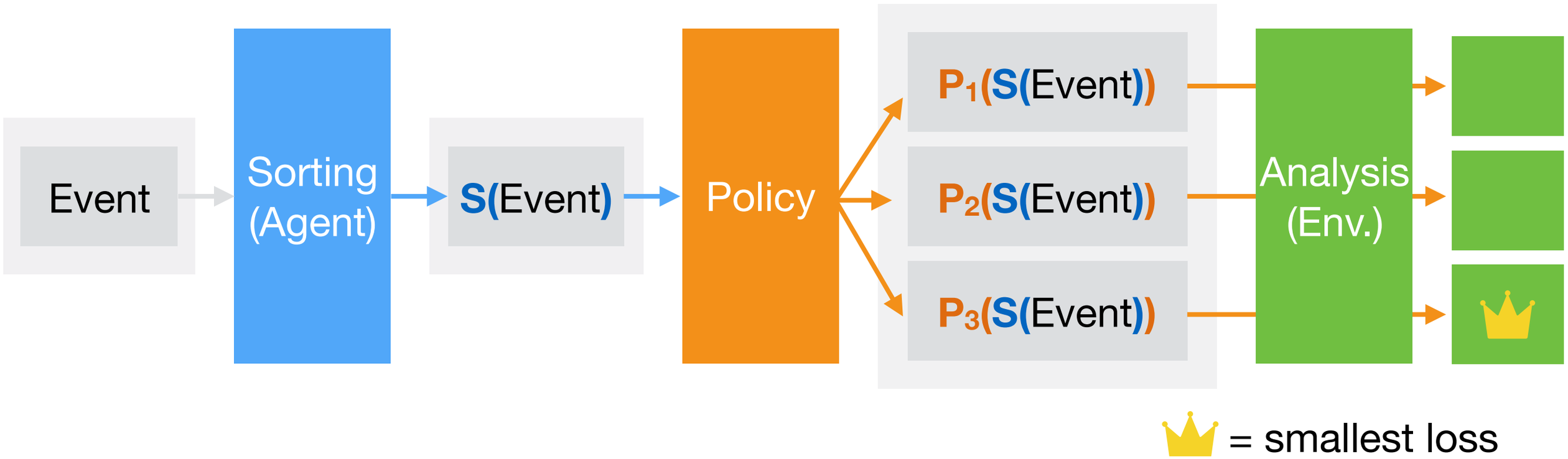


- Train sorting to output $P_3(S(\text{Event}))$:
  - Positions (targets) follow from P_3
  - Gaussian smear
  - Supervised training

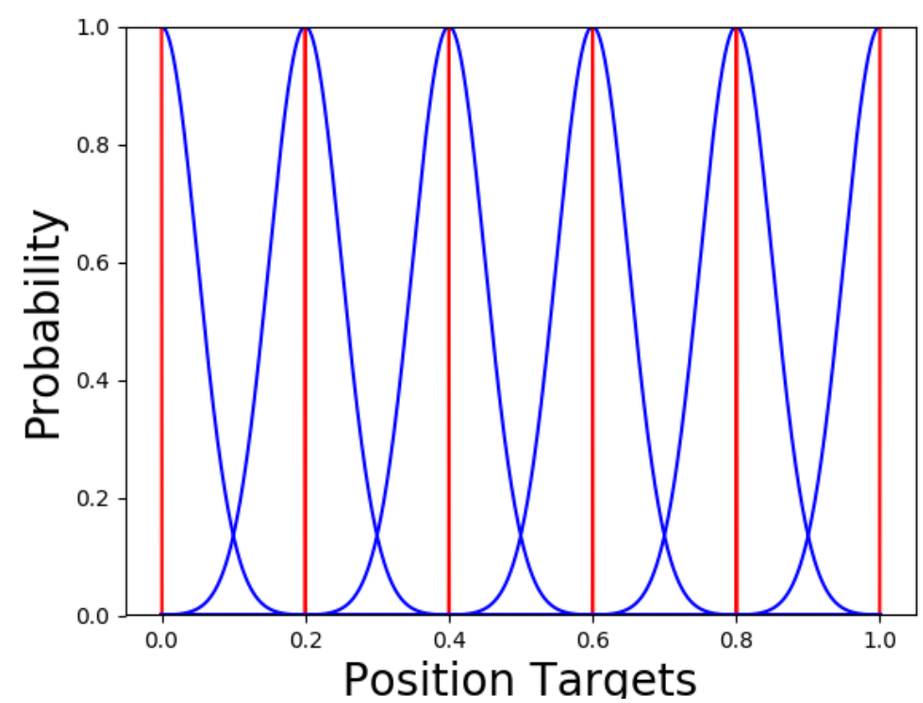


Complete Mechanism - Reinforced Training

Training

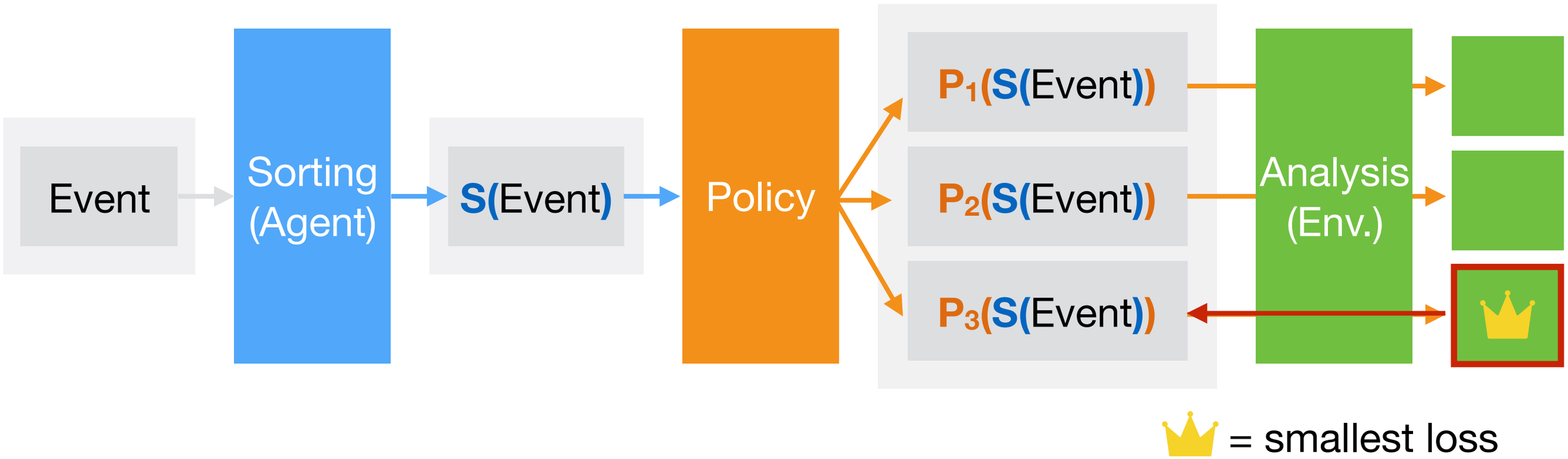


- Train sorting to output $P_3(S(\text{Event}))$:
  - Positions (targets) follow from P_3
  - Gaussian smear
  - Supervised training

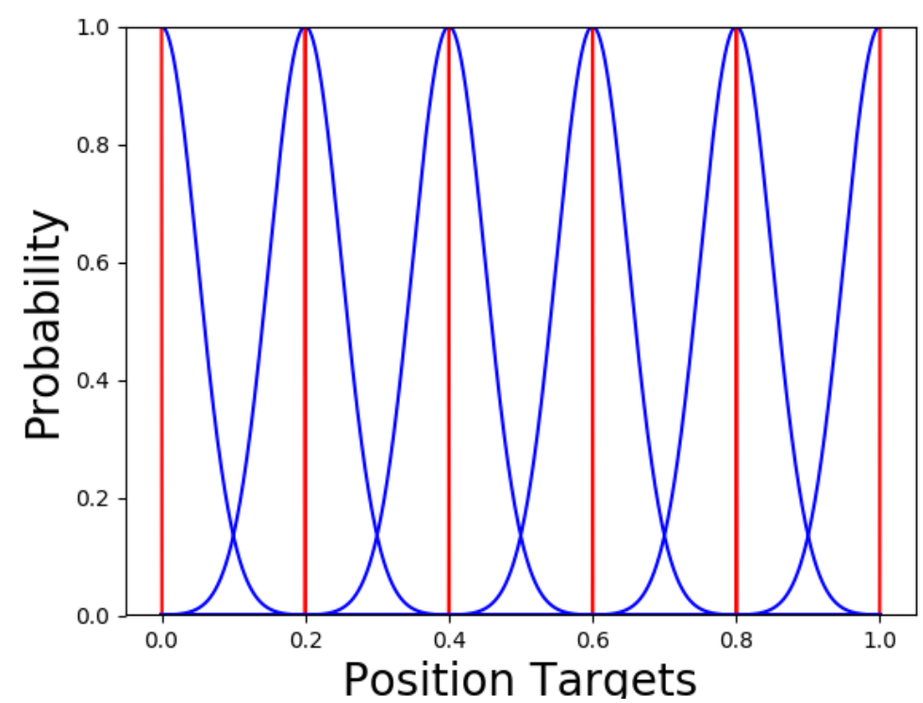


Complete Mechanism - Reinforced Training

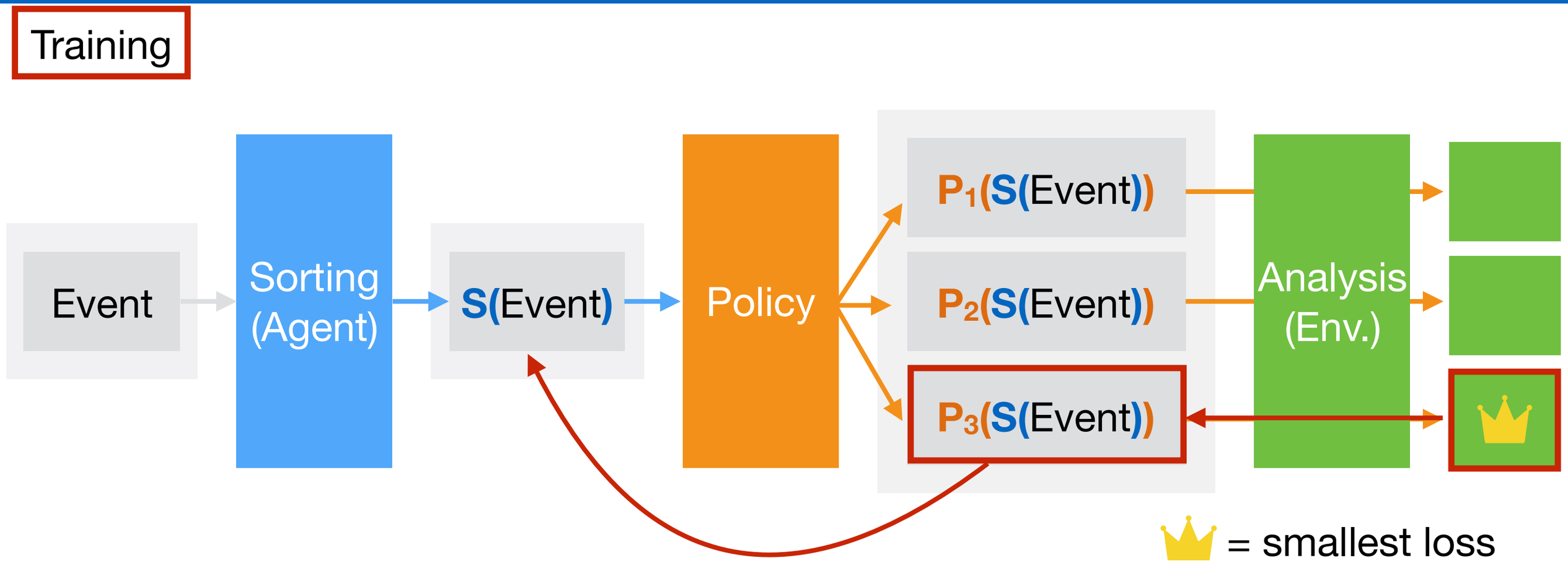
Training



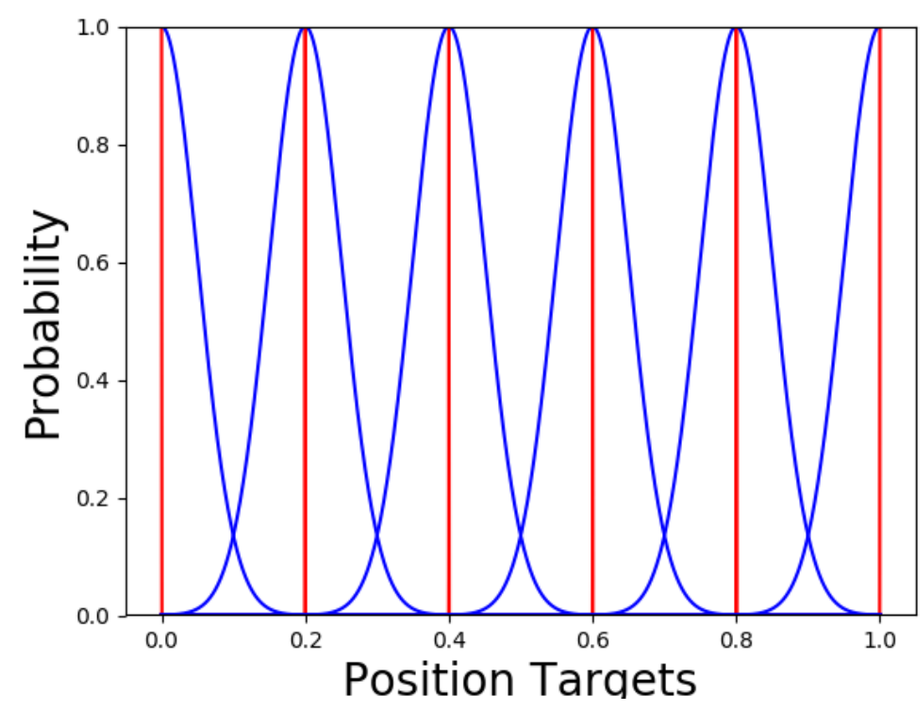
- Train sorting to output $P_3(S(\text{Event}))$:
  - Positions (targets) follow from P_3
  - Gaussian smear
  - Supervised training



Complete Mechanism - Reinforced Training



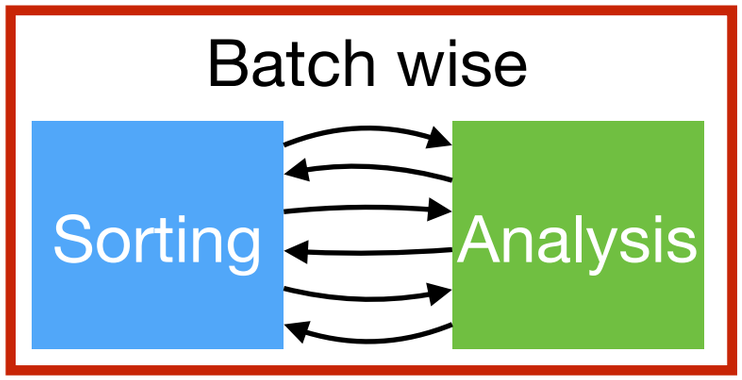
- Train sorting to output $P_3(S(\text{Event}))$:
  - Positions (targets) follow from P_3
  - Gaussian smear
  - Supervised training

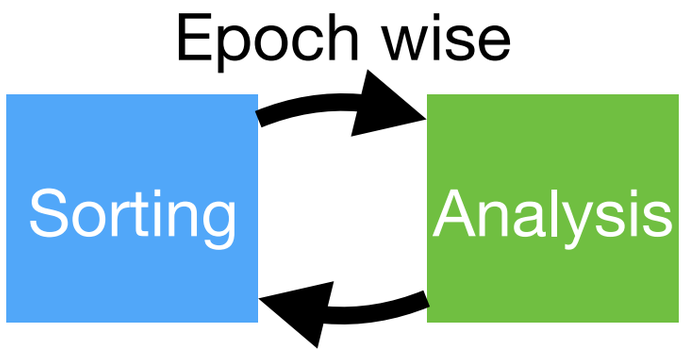


ttH vs ttbb - Hyper Parameters and Training

RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19

- Inputs for both networks: **only four-vectors** (Pythia + Delphes)
- Sorting Network:
  - Feed forward
    - 6 SeLu Layers, 256 Nodes
    - L_2 regularization + Layer Normalization
  - Adam Optimizer
- Analysis Network:
  - Lorentz-Boost Network (1812.09722, talk by Yannik)
- Training:
  - Schedule:



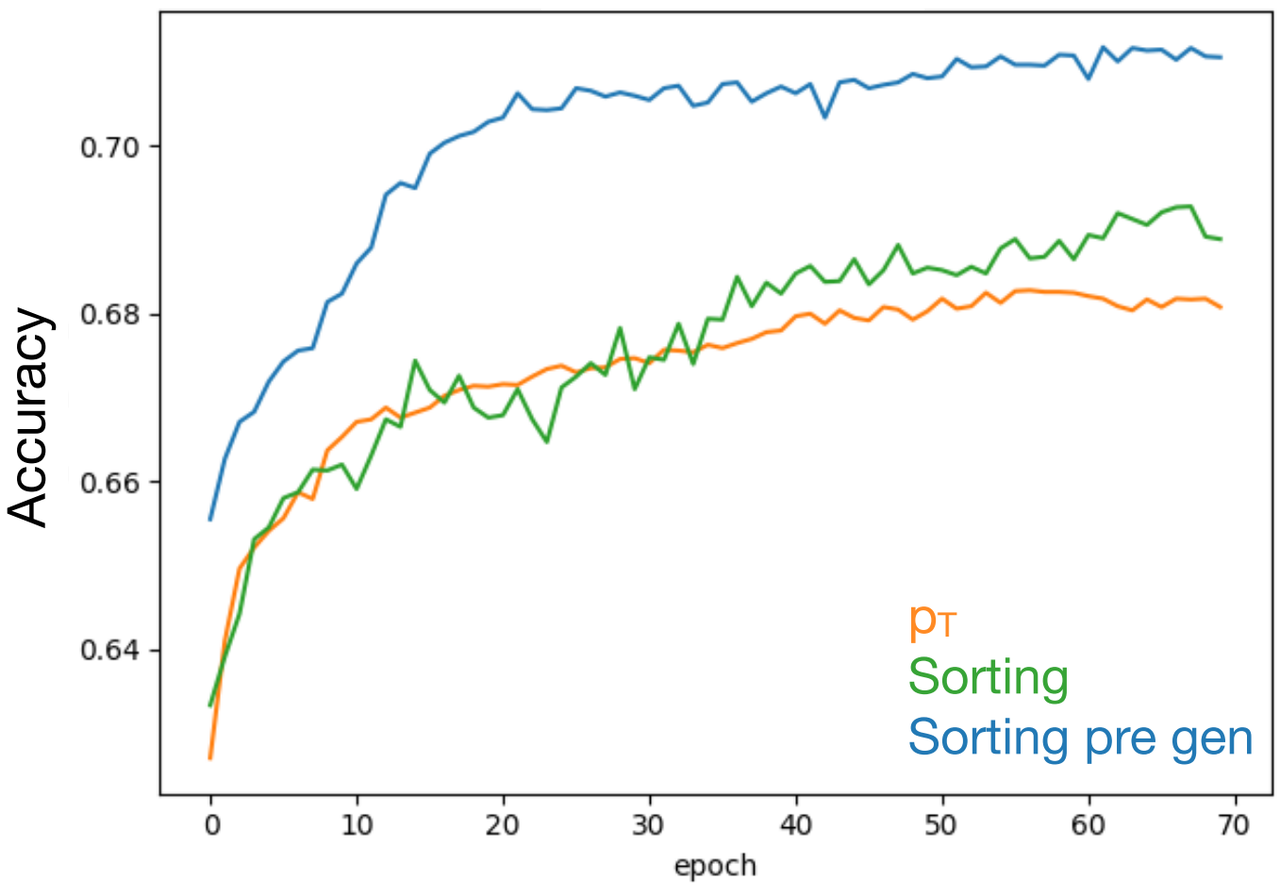


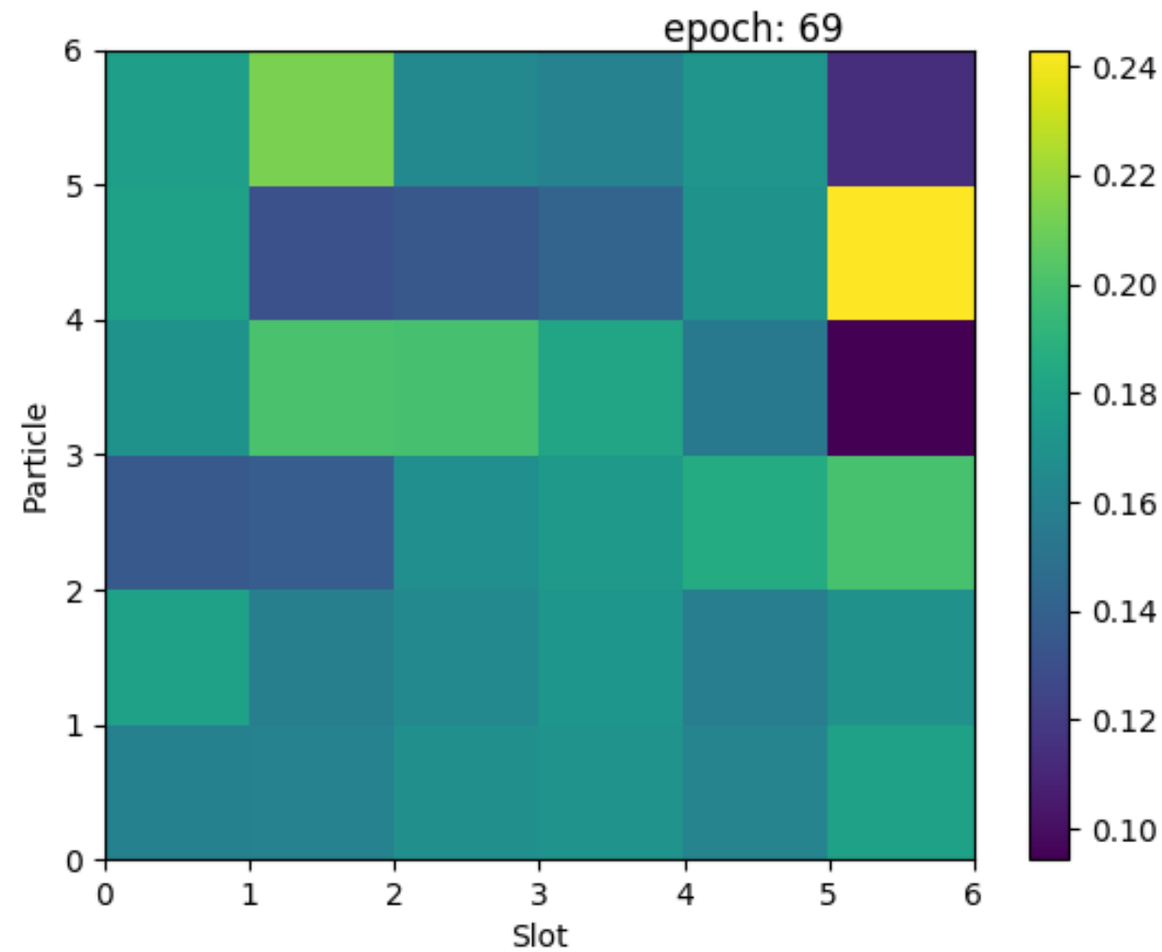
  - Pre-training:
    - Let analysis network converge before starting to train
    - Pre-train sorting to output p_T

ttH vs ttbb - Results (1)

RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19

- Sorting network learns an intrinsic structure
- Performance of classifier increases



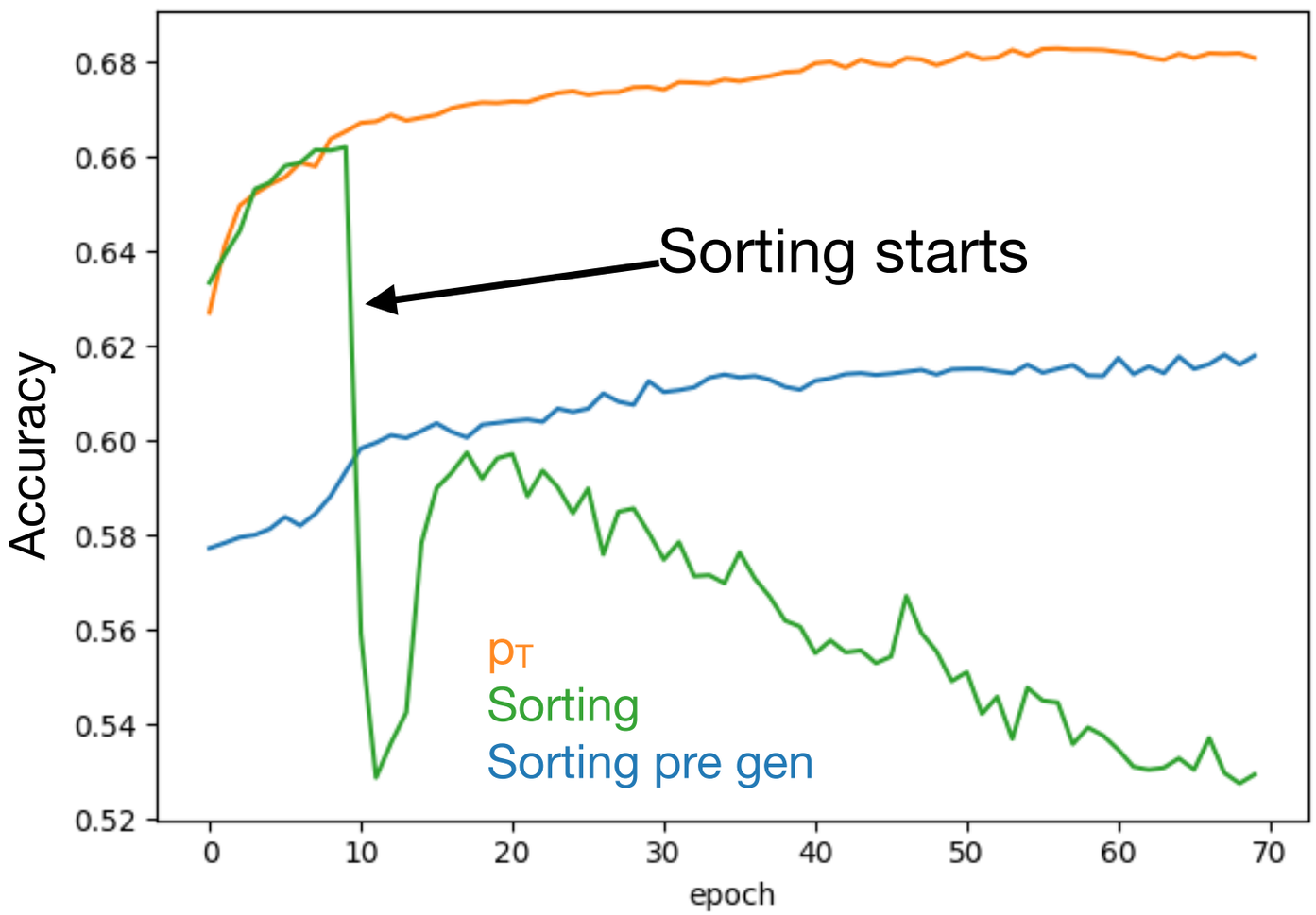


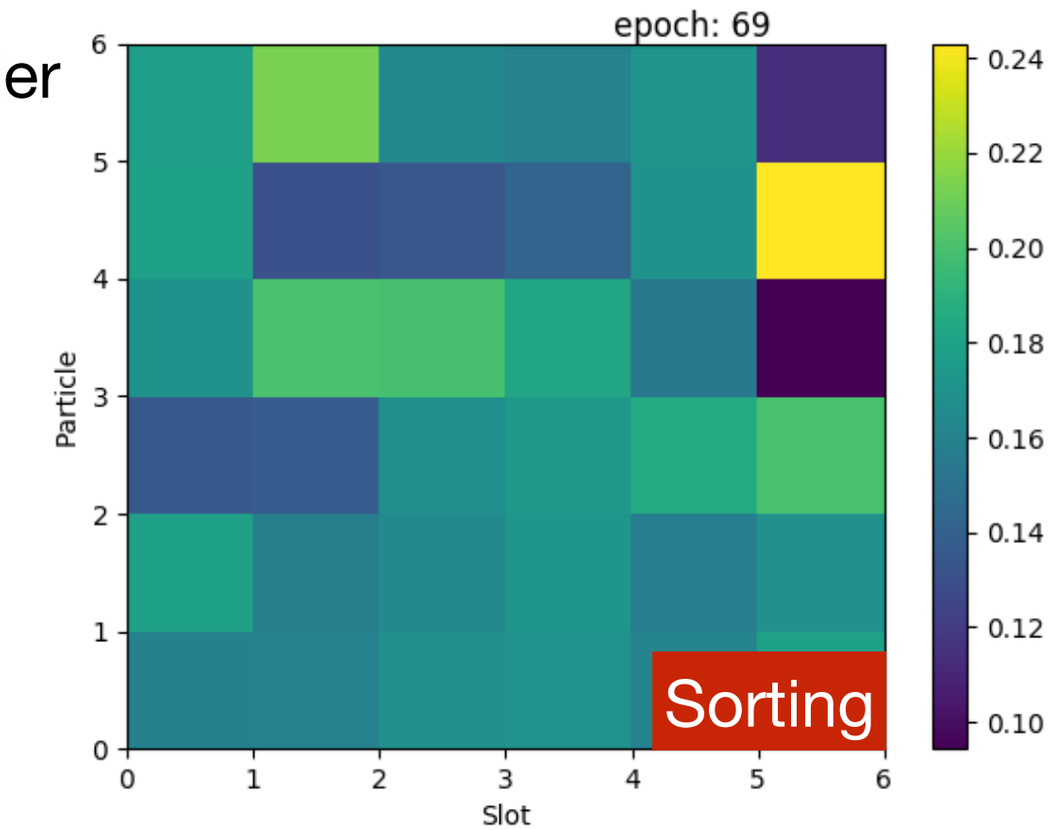
ttH vs ttbb - Results (2)

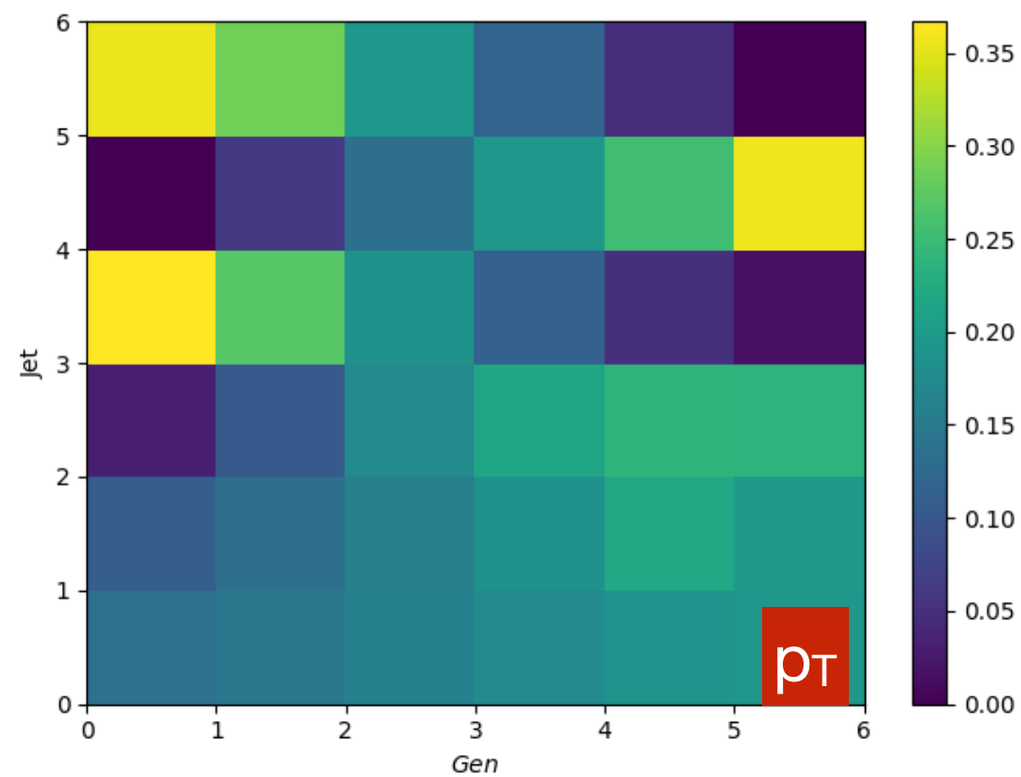
RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19

- Train analysis with sorting, evaluate on p_T order
  - Accuracy drops to 50%
  - Analysis network relies on sorting

  → p_T order disfavored



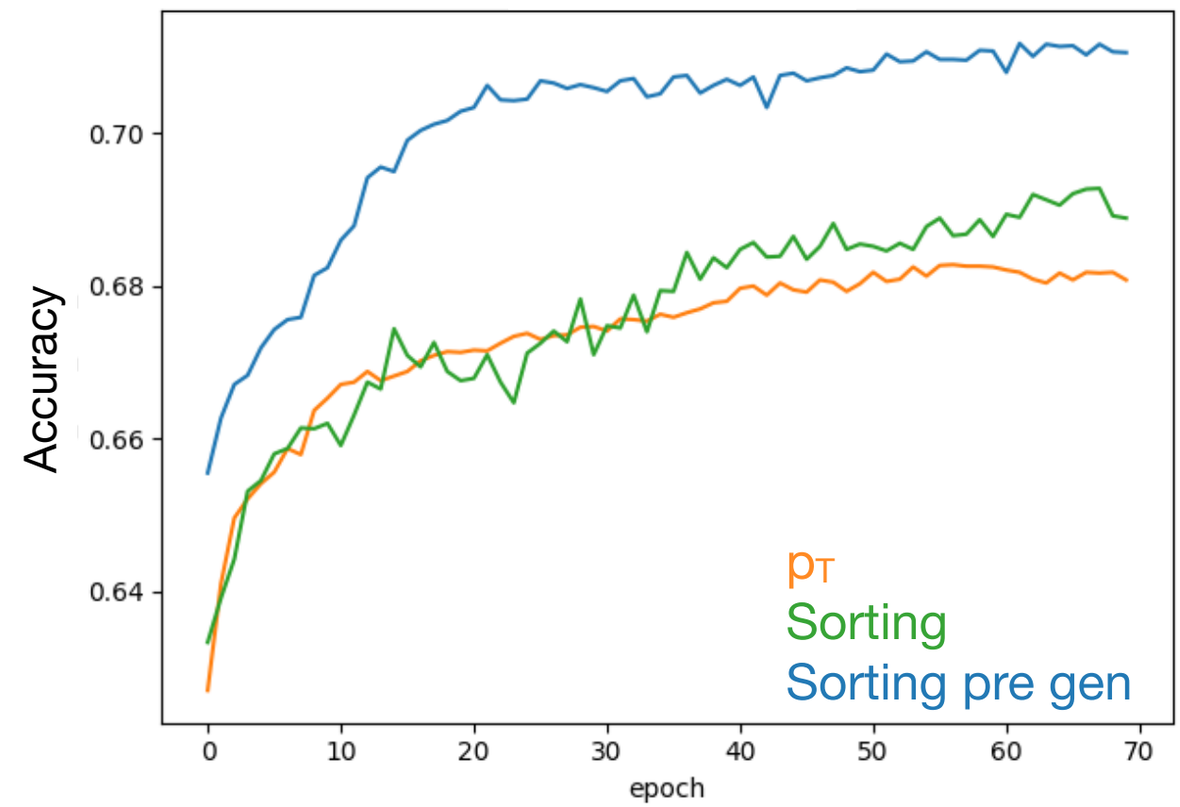




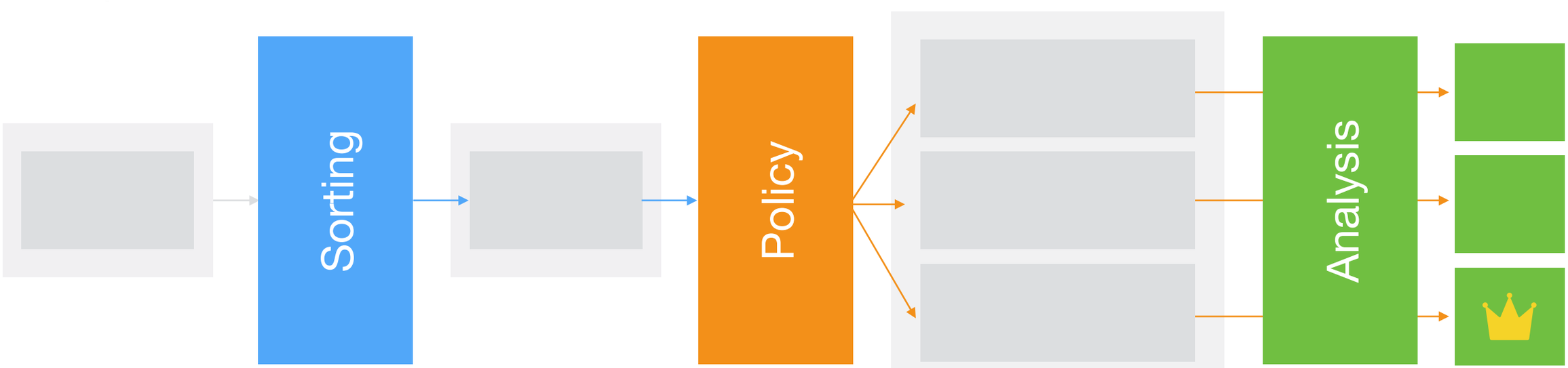
Conclusion

RWTH AACHEN UNIVERSITY
Dennis Noll - 14 March 19

- Reinforced input sorting may benefit various NN applications
  - Agent:
    - Sorting of inputs (e.g. jets)
  - Policy:
    - Classic permutations
    - Tree search
  - Environment:
    - Physics analysis (e.g. LBN)



- Especially promising in scenarios with a-priori unknown “best” order



Backup

Artificial Data - Dataset

- Build artificial dataset - with explicit sorting information:
  - Sorting information: one random number
  - „Features“: random numbers
  - Task: reconstruct weighted alternating sum over the „Features“
    - Easy task, if input order is fixed
    - Nearly impossible if input order is random
    - Regression

Sorted: easy

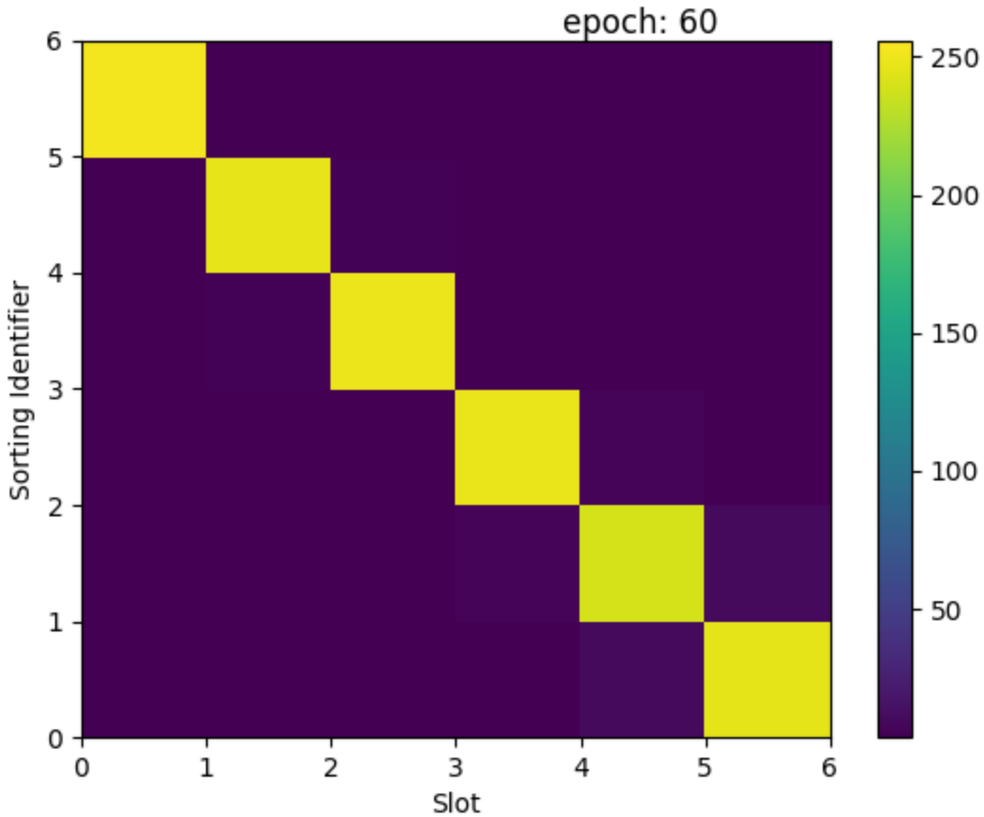
Row1	0.3	+ 0 x 0.3
Row2	0.8	- 1 x 0.8
Row3	0.2	+ 2 x 0.2
Row4	0.5	- 3 x 0.5
Row5	0.9	+ 4 x 0.9
Row6	0.1	-5 x 0.1
		= 1.2

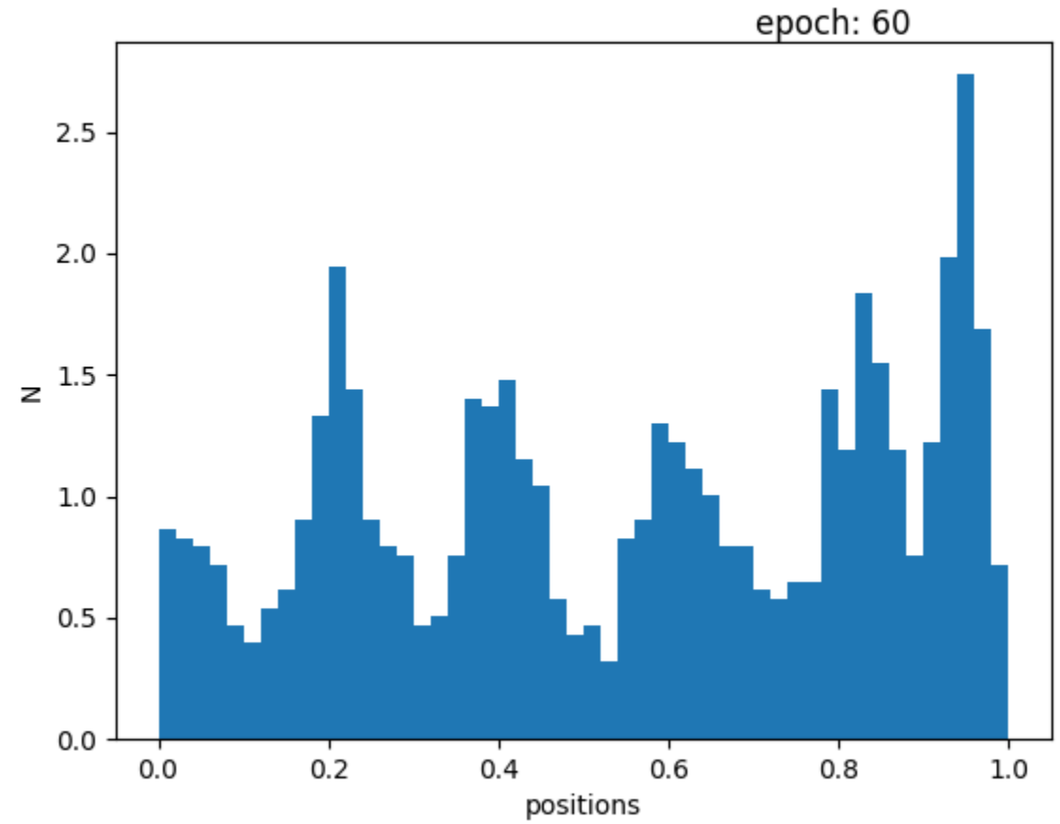
Random: impossible

Row1	0.3
Row4	0.5
Row2	0.8
Row5	0.9
Row6	0.1
Row3	0.2

Artificial Data - Classic - Results

Dennis Noll - 14 March 19





- Sorting model converges fast
- Finds stable sorting
- Outputs gaussian