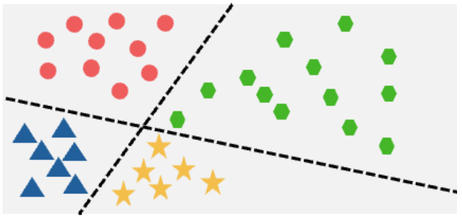
nsdi '26

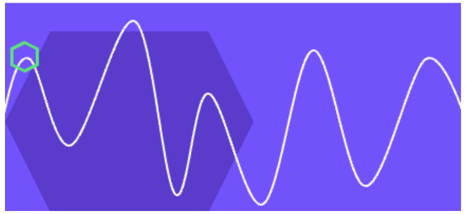
SpliDT: Partitioned Decision Trees for Scalable Stateful Inference at Line Rate

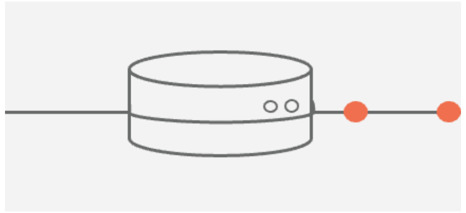
Murayyiam Parvez*, Annus Zulfiqar*, Roman Beltiukov, Shir Landau Feibish, Walter Willinger, Arpit Gupta, Muhammad Shahbaz

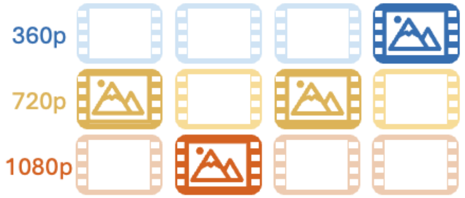
PURDUE UNIVERSITY
UNIVERSITY OF MICHIGAN
UNIVERSITY OF CALIFORNIA SANTA BARBARA
UNIVERSITY OF HAIFA
NORTHWESTERN UNIVERSITY

Machine Learning in Programmable Data Planes


Traffic Classification

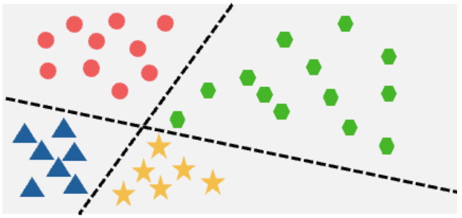

Anomaly Detection


Congestion Control

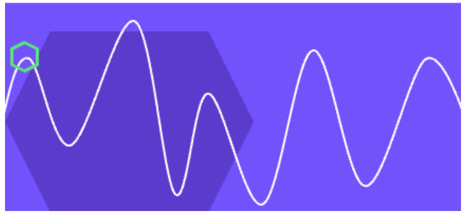


Adaptive Bitrate Streaming

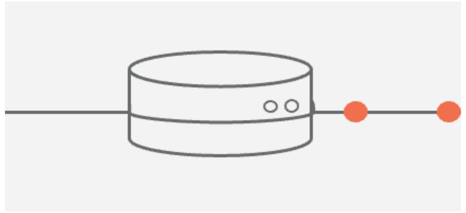
Machine Learning in Programmable Data Planes



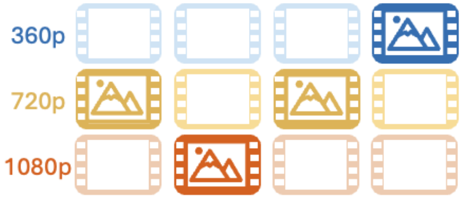
Traffic Classification



Anomaly Detection



Congestion Control



Adaptive Bitrate Streaming

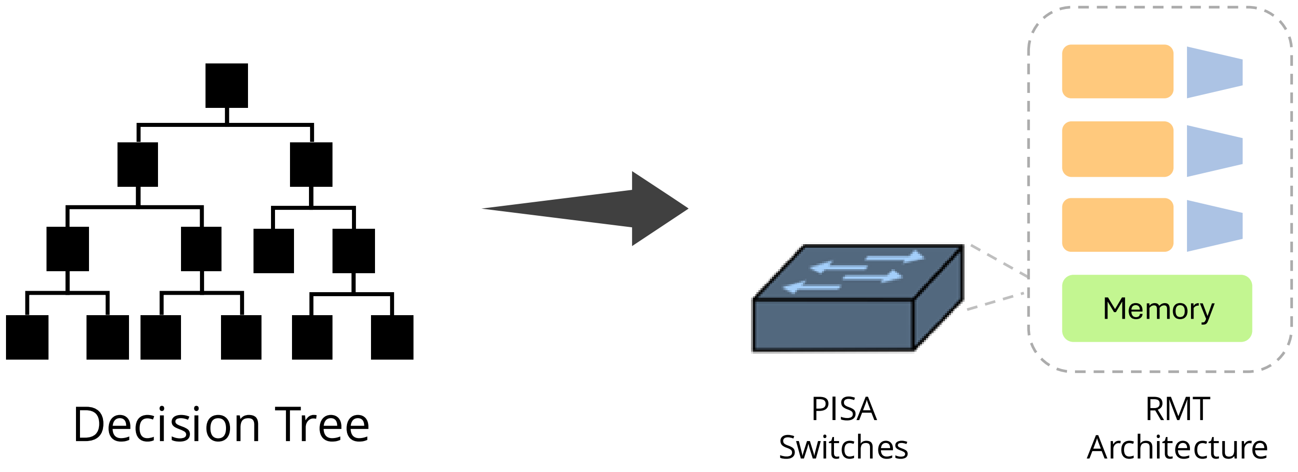
Line rate!

NVIDIA

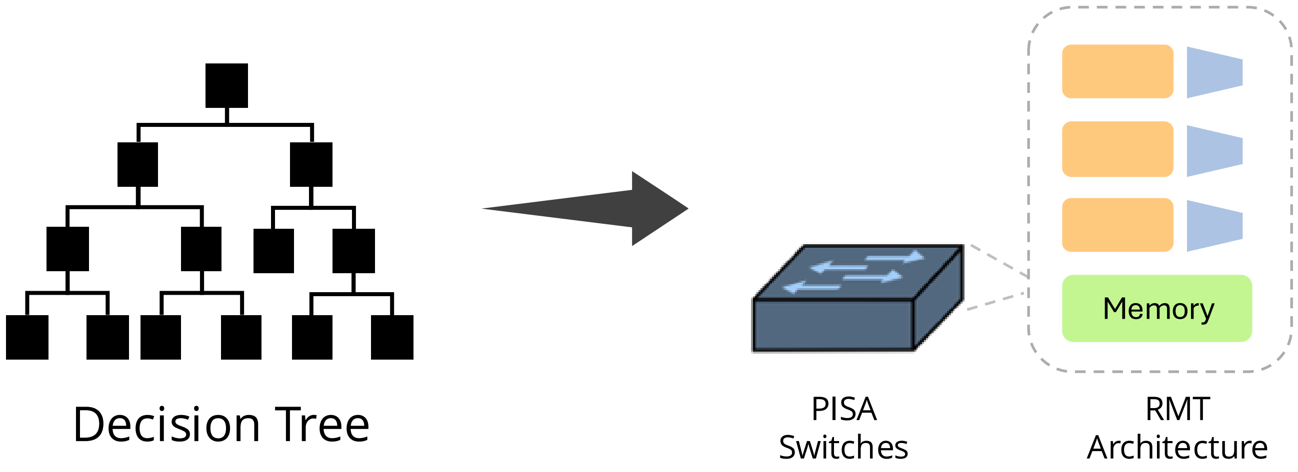
PENSANDO

intel

Decision Trees in Programmable Switches

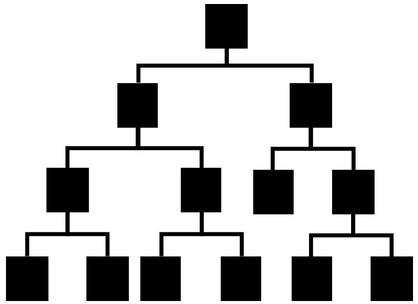


Decision Trees in Programmable Switches

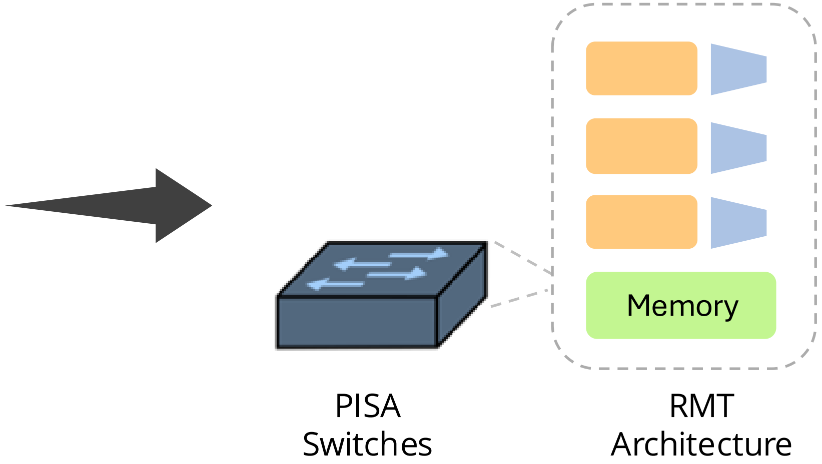


Maps cleanly onto the **RMT's match-action pipeline**

Decision Trees in Programmable Switches



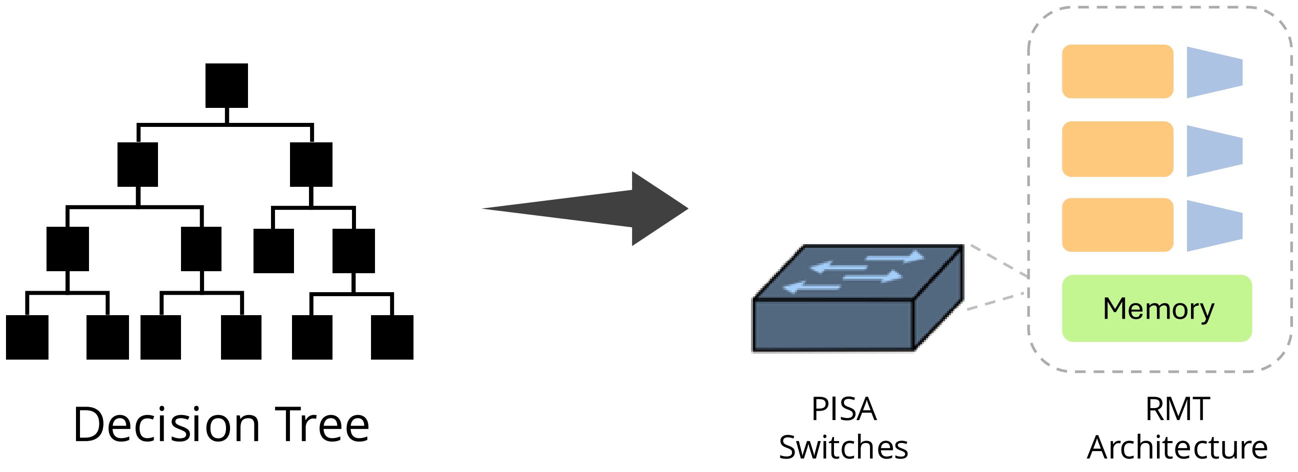
Decision Tree



Maps cleanly onto the **RMT's match-action pipeline**

Interpretable and **easy to verify**

Decision Trees in Programmable Switches

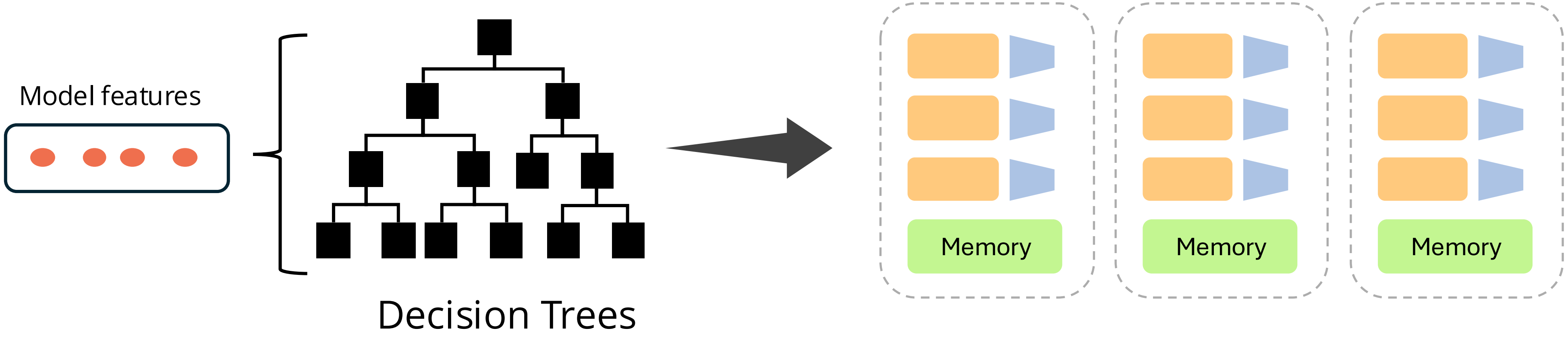


Maps cleanly onto the **RMT's match-action pipeline**

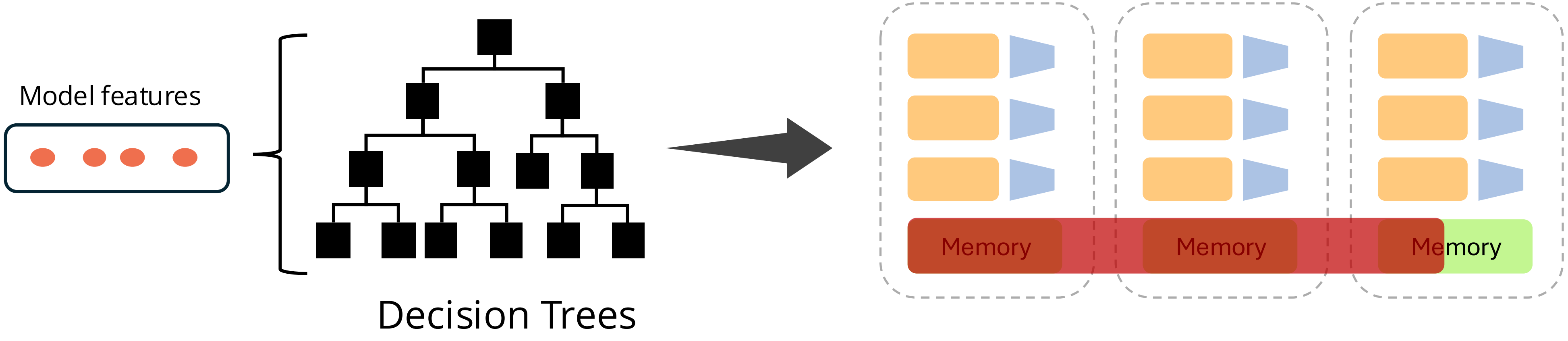
Interpretable and **easy to verify**

Simple comparisons → **ternary** or **exact** matches

DT Feature Collection in Stateful Memory

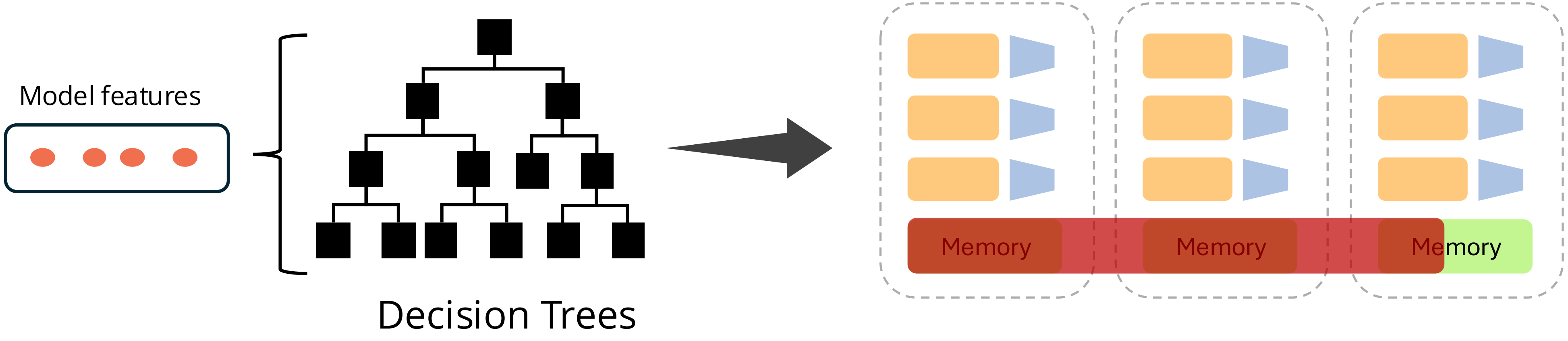


DT Feature Collection in Stateful Memory



Limits scalability to **top-k** features

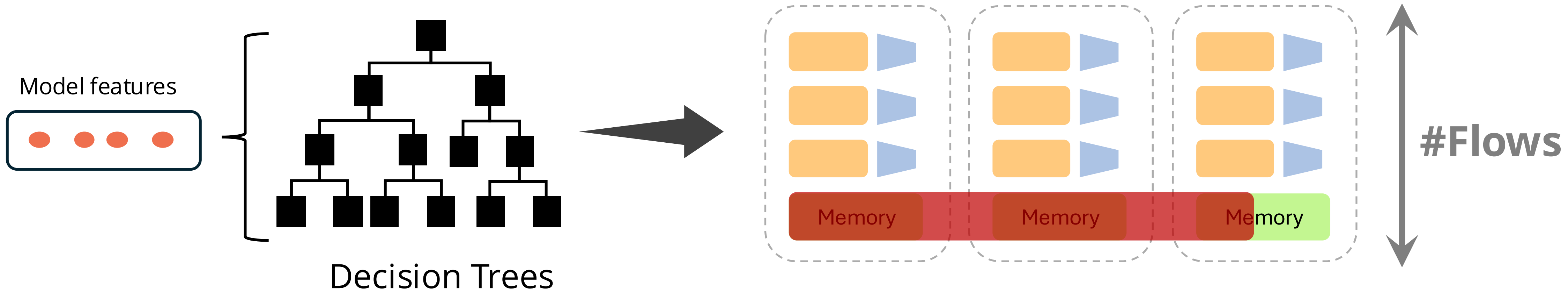
DT Feature Collection in Stateful Memory



Limits scalability to **top-k** features

Prevents the model from reaching its **full accuracy potential**

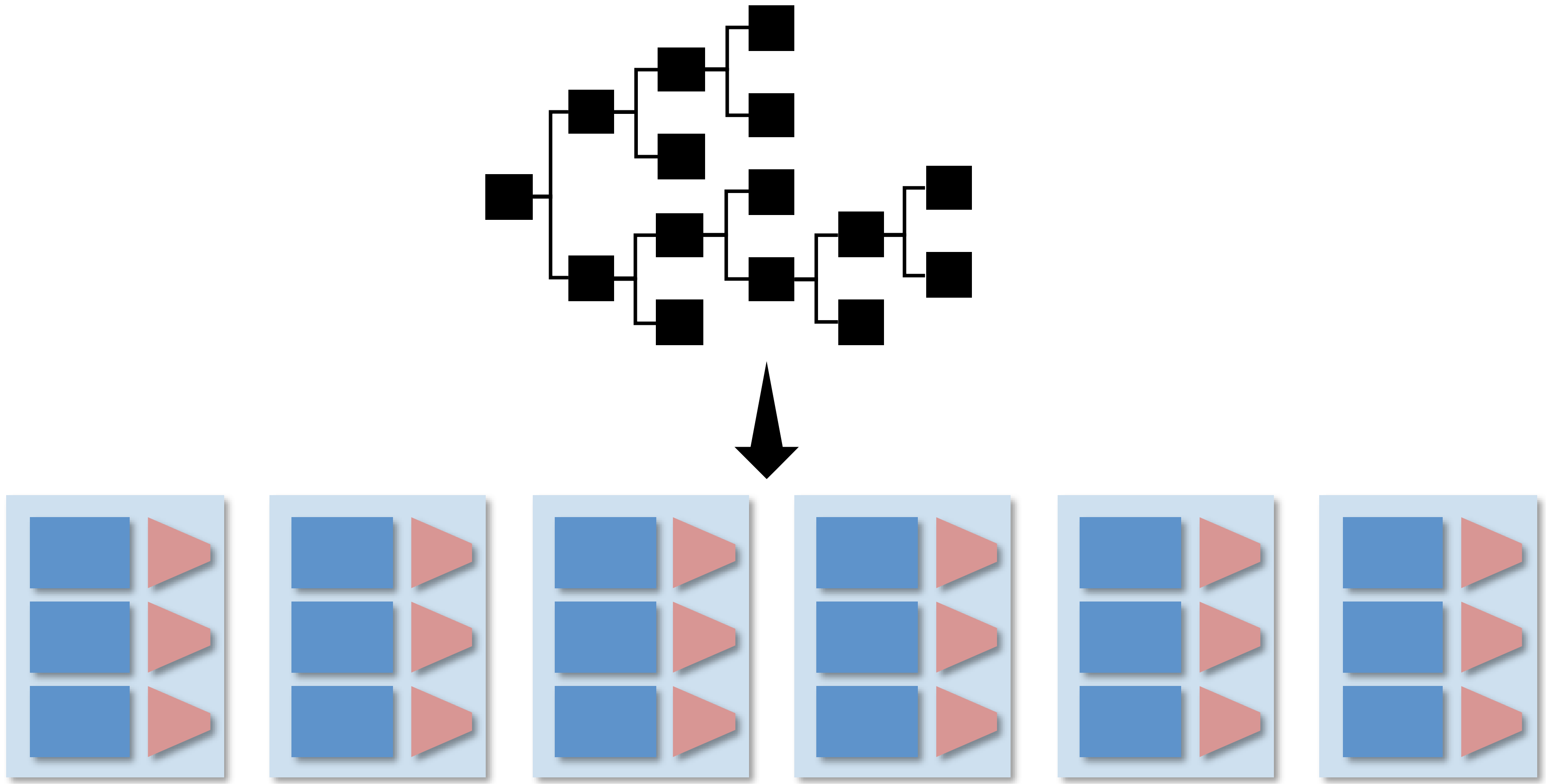
DT Feature Collection in Stateful Memory



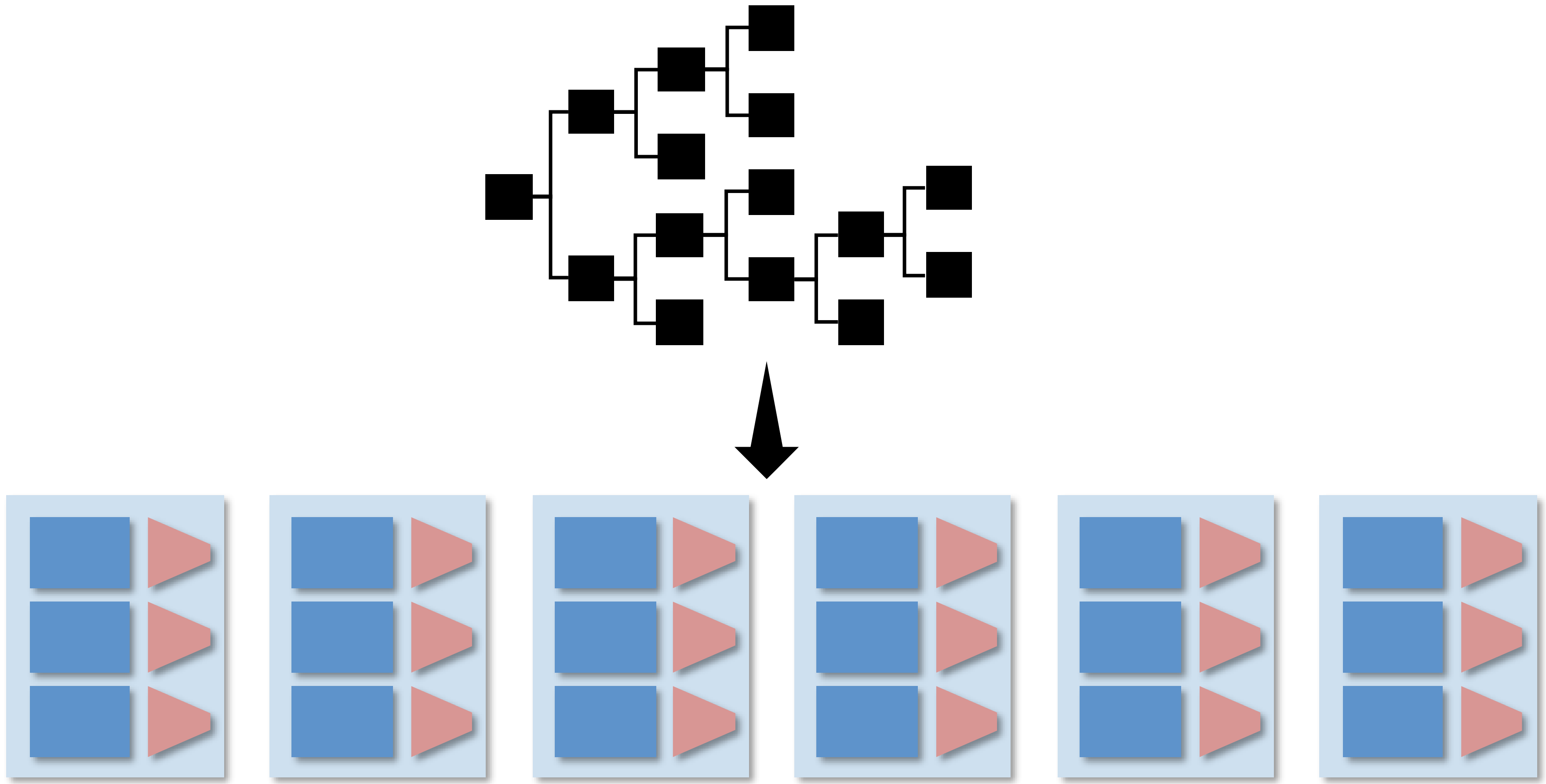
Limits scalability to **top-k** features

Prevents the model from reaching its **full accuracy potential**

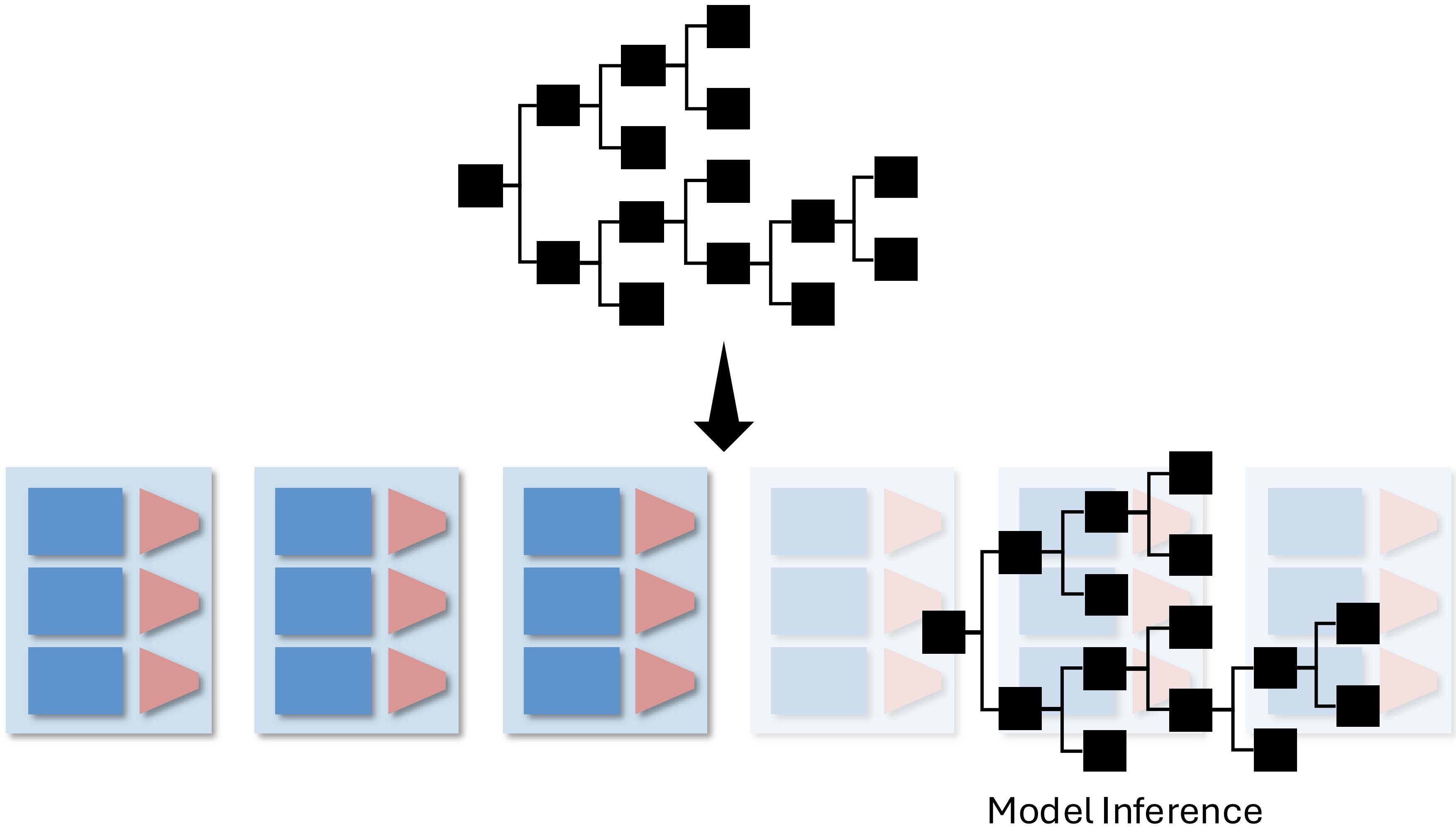
One-Shot Feature Collection and Inference



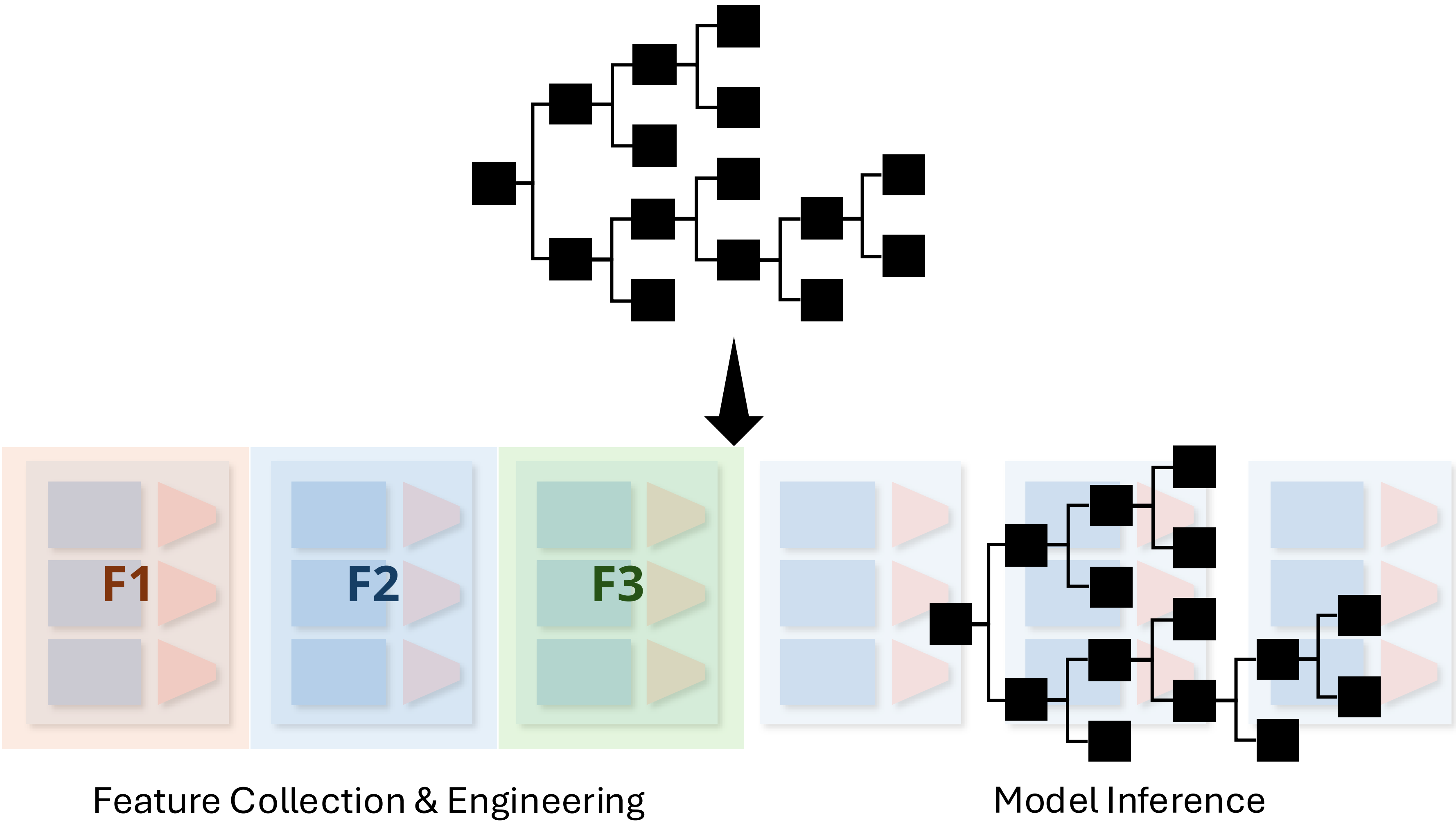
One-Shot Feature Collection and Inference



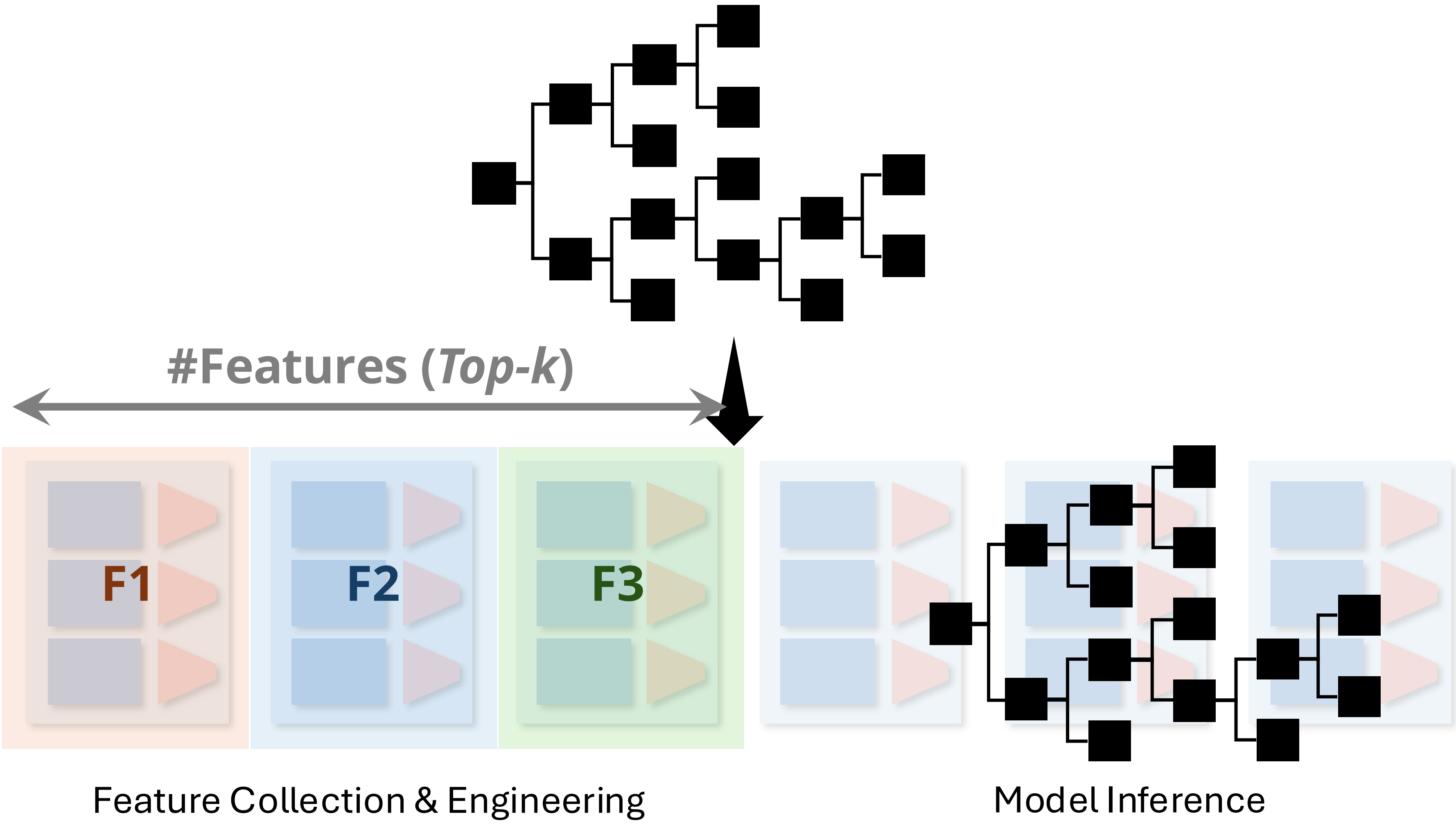
One-Shot Feature Collection and Inference



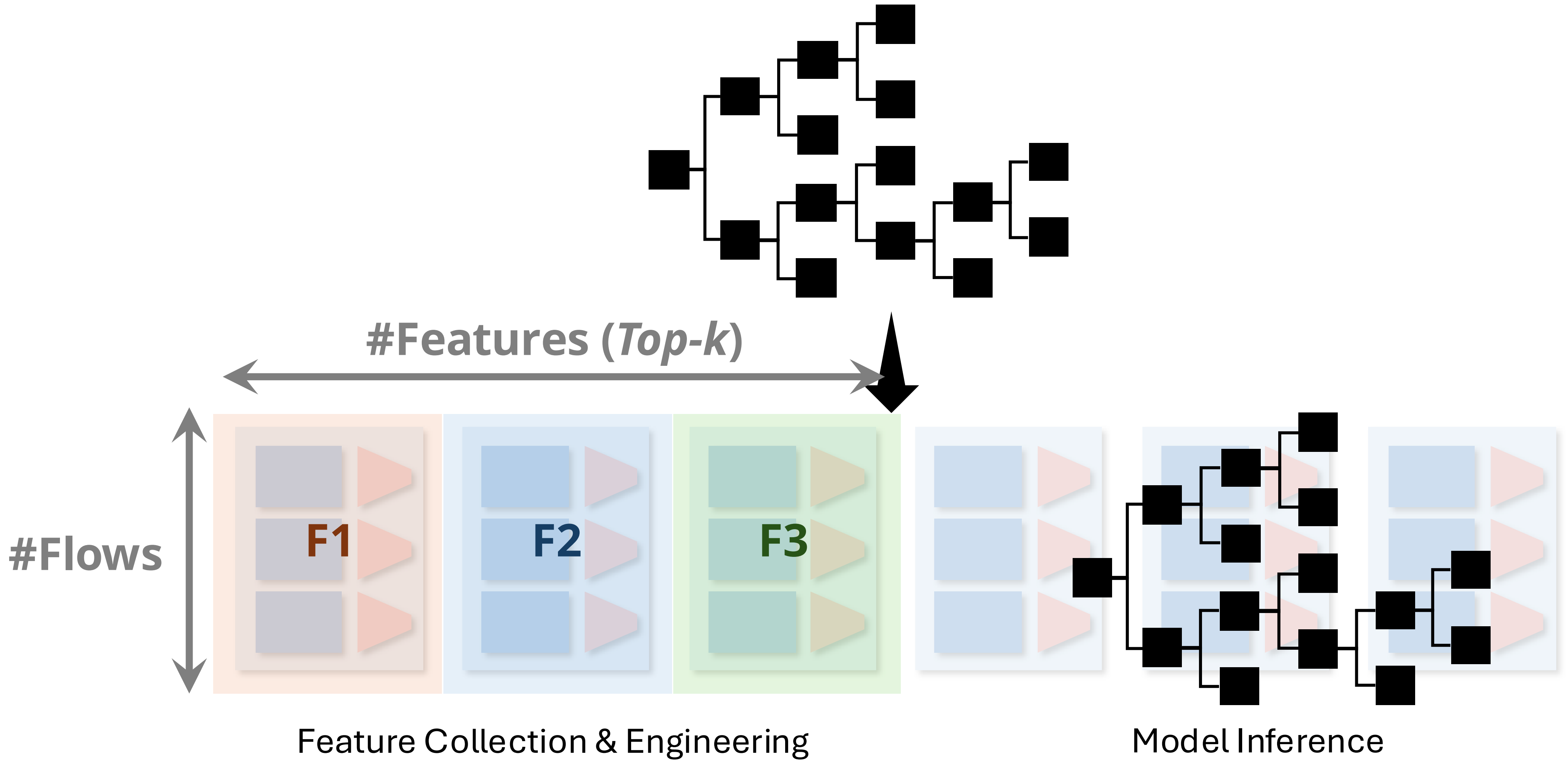
One-Shot Feature Collection and Inference



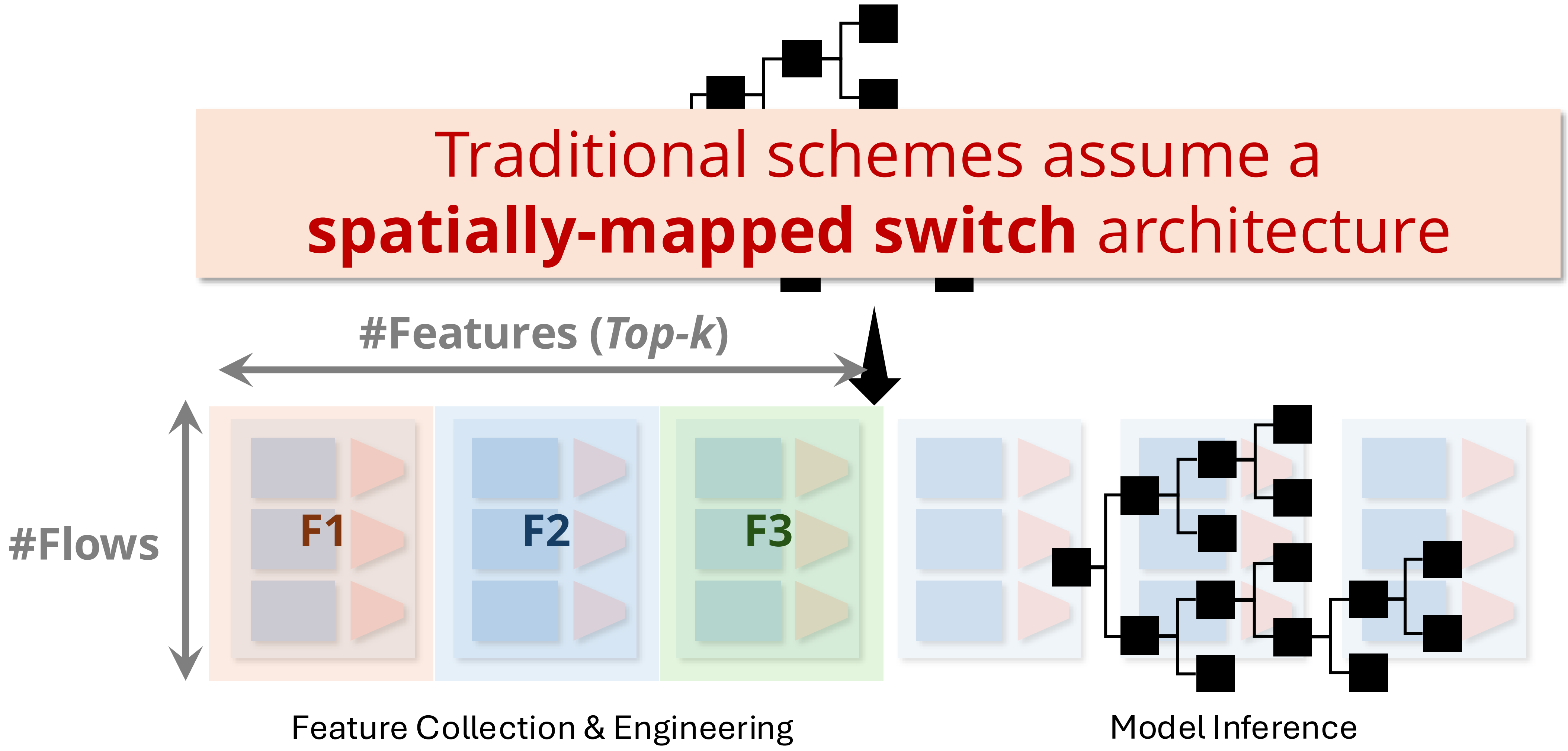
One-Shot Feature Collection and Inference



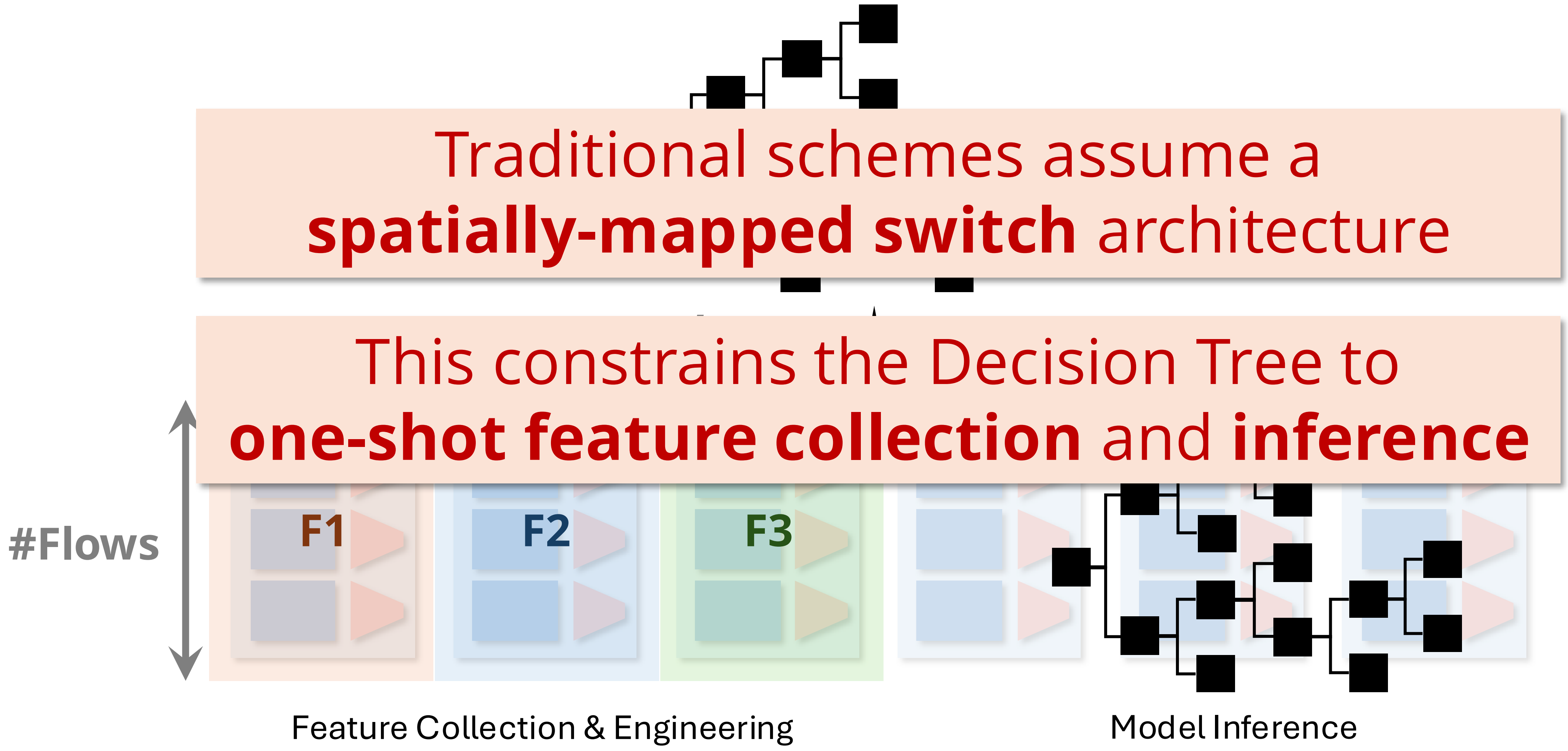
One-Shot Feature Collection and Inference



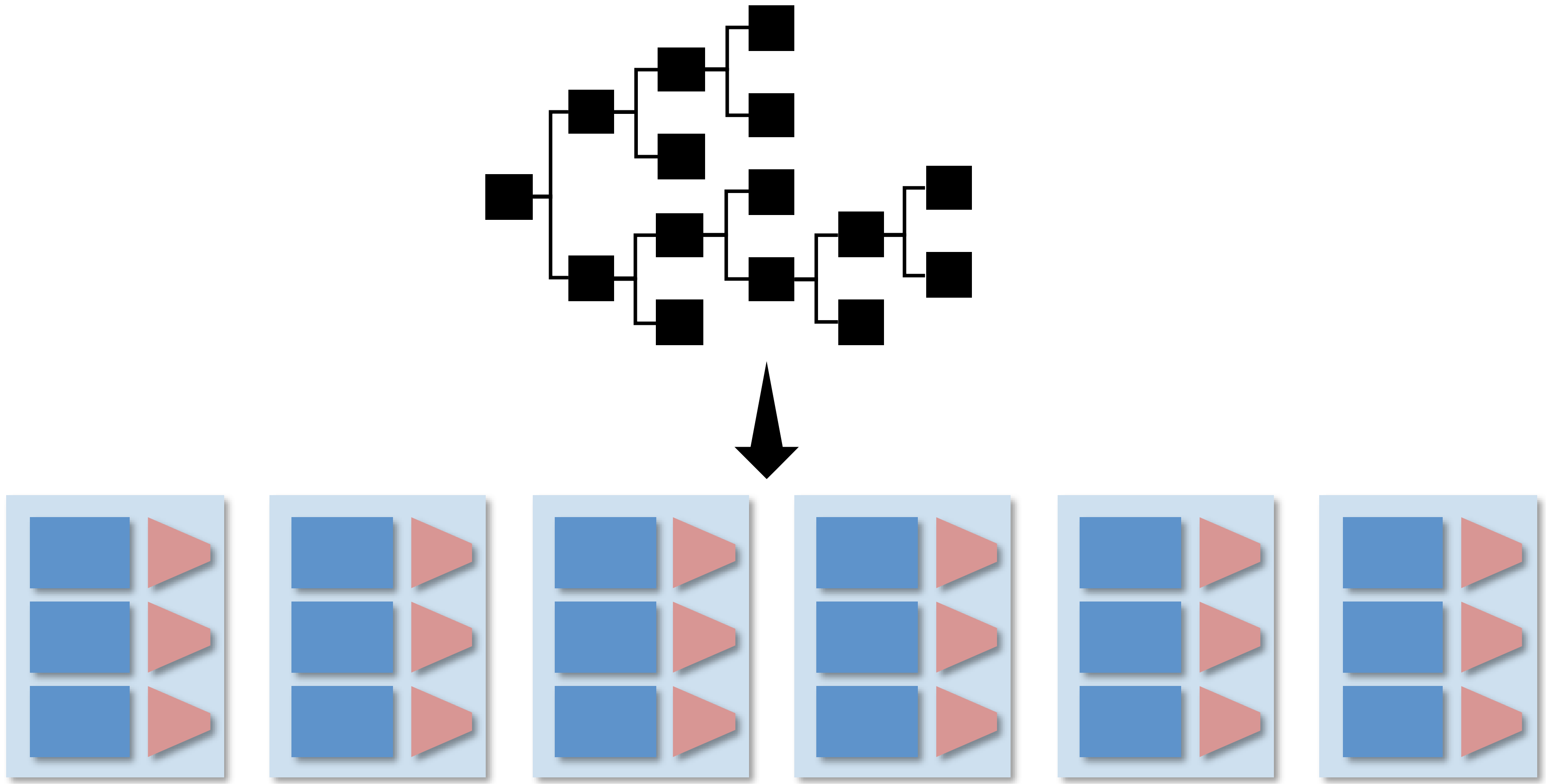
One-Shot Feature Collection and Inference



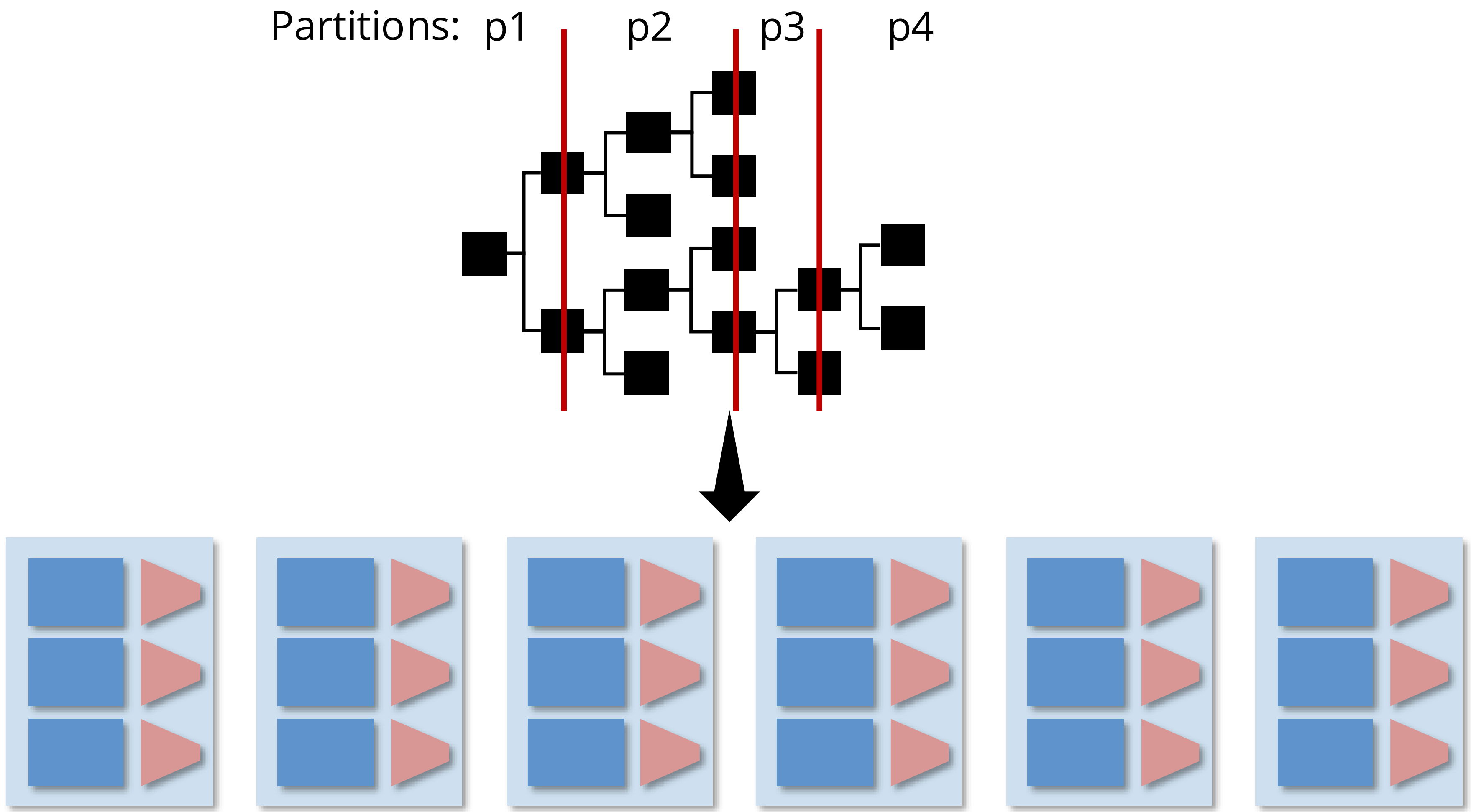
One-Shot Feature Collection and Inference



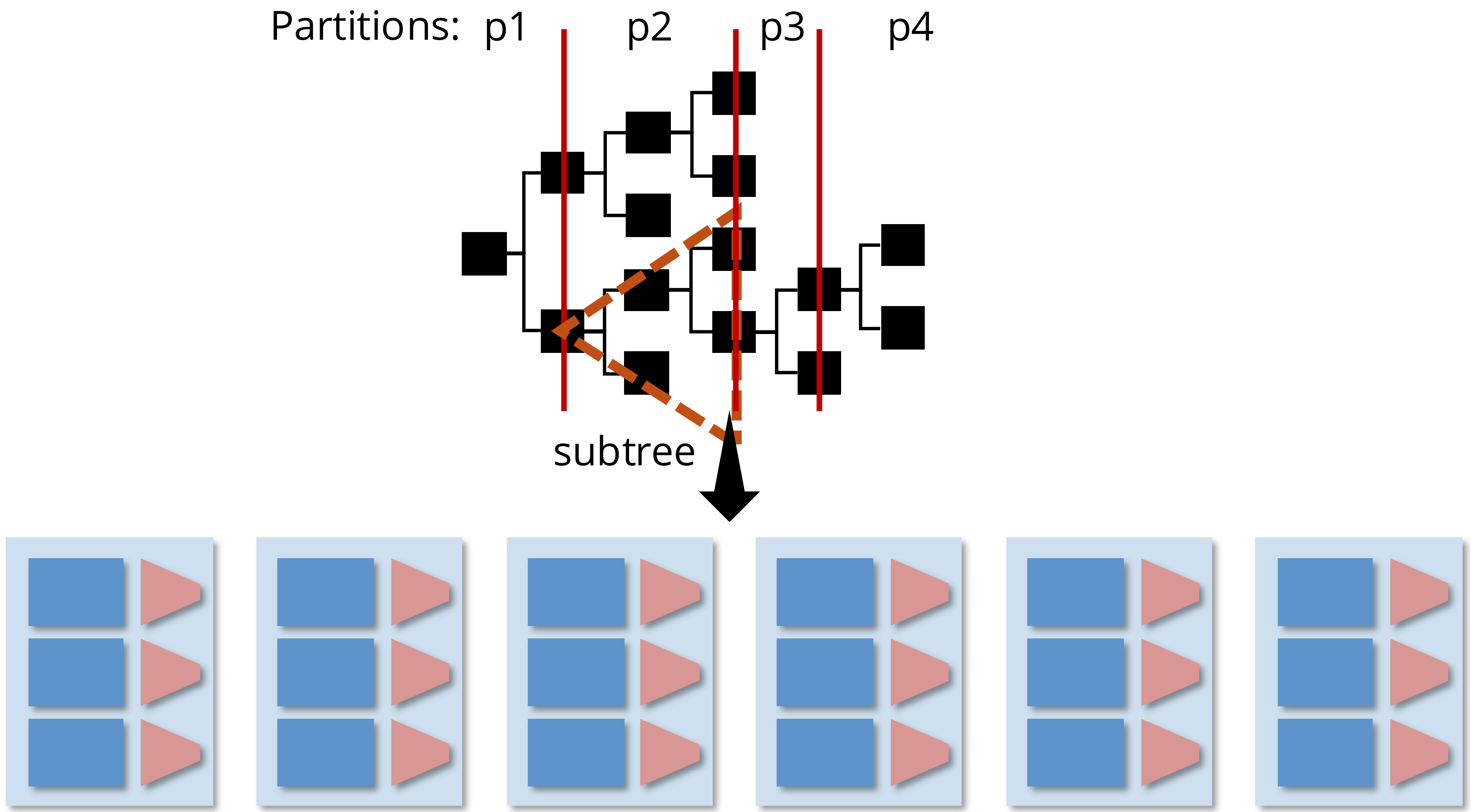
Partitioned Inference Architecture



Partitioned Inference Architecture

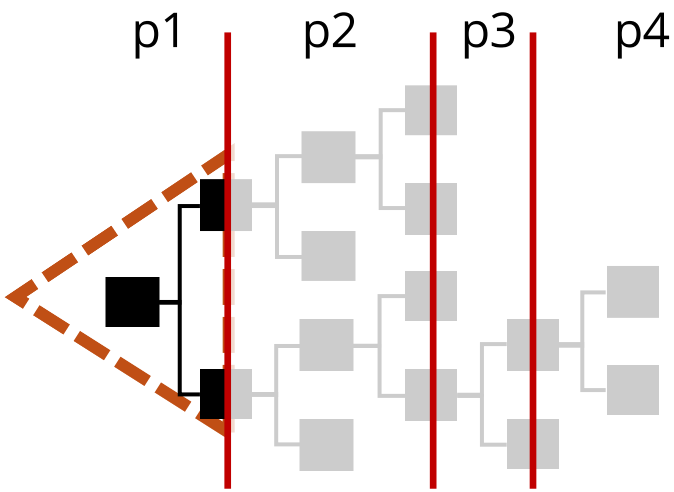


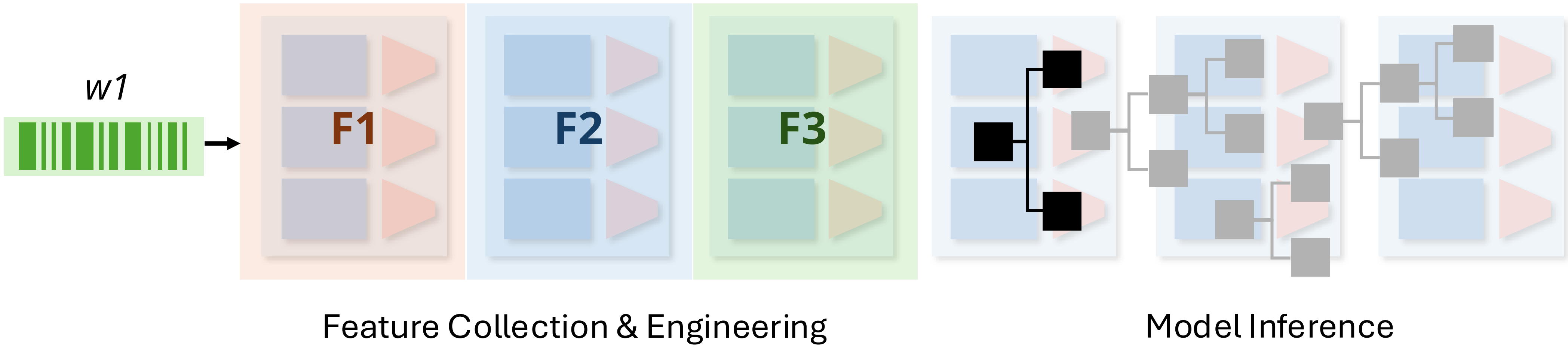
Partitioned Inference Architecture



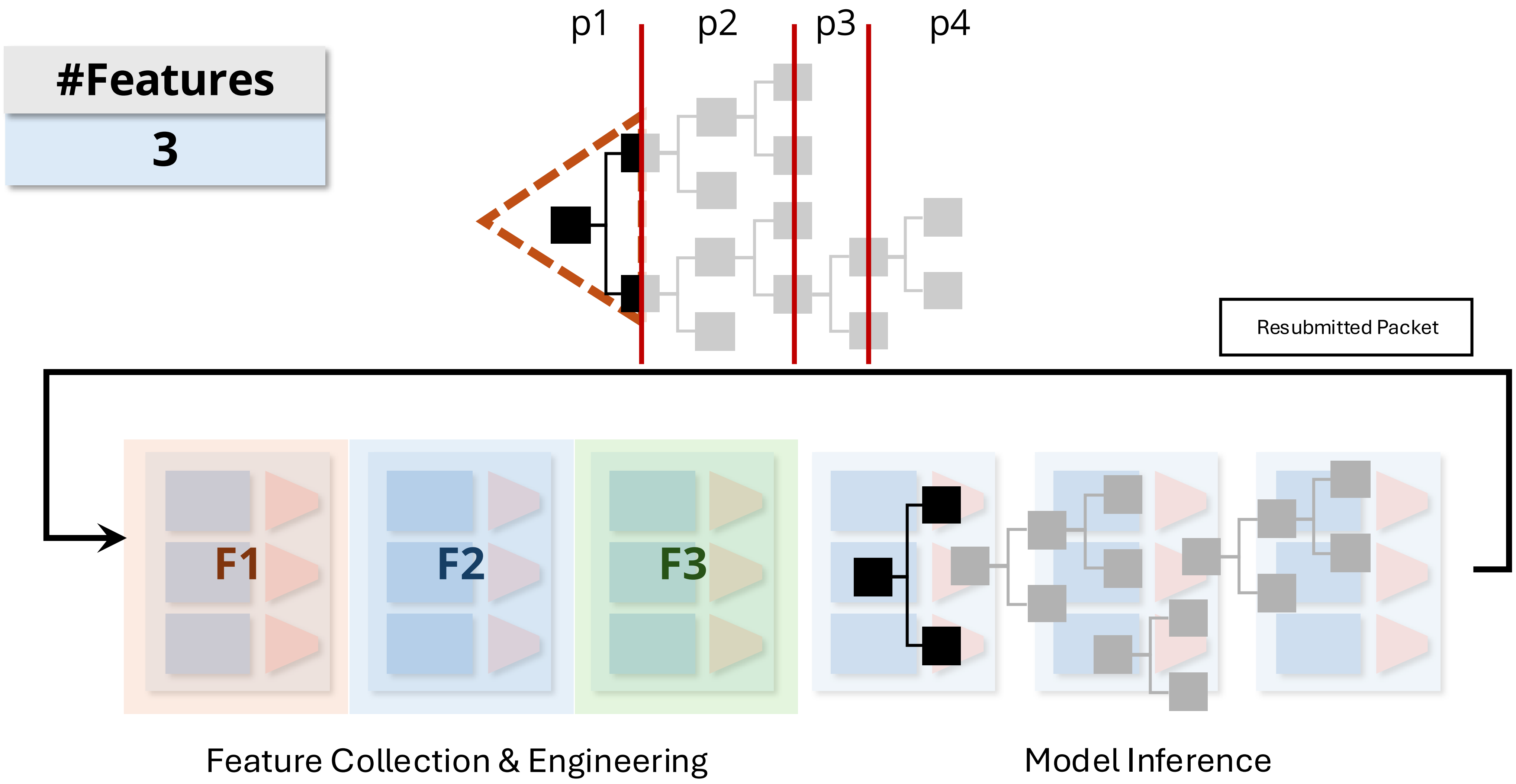
Partitioned Inference Architecture

#Features
3



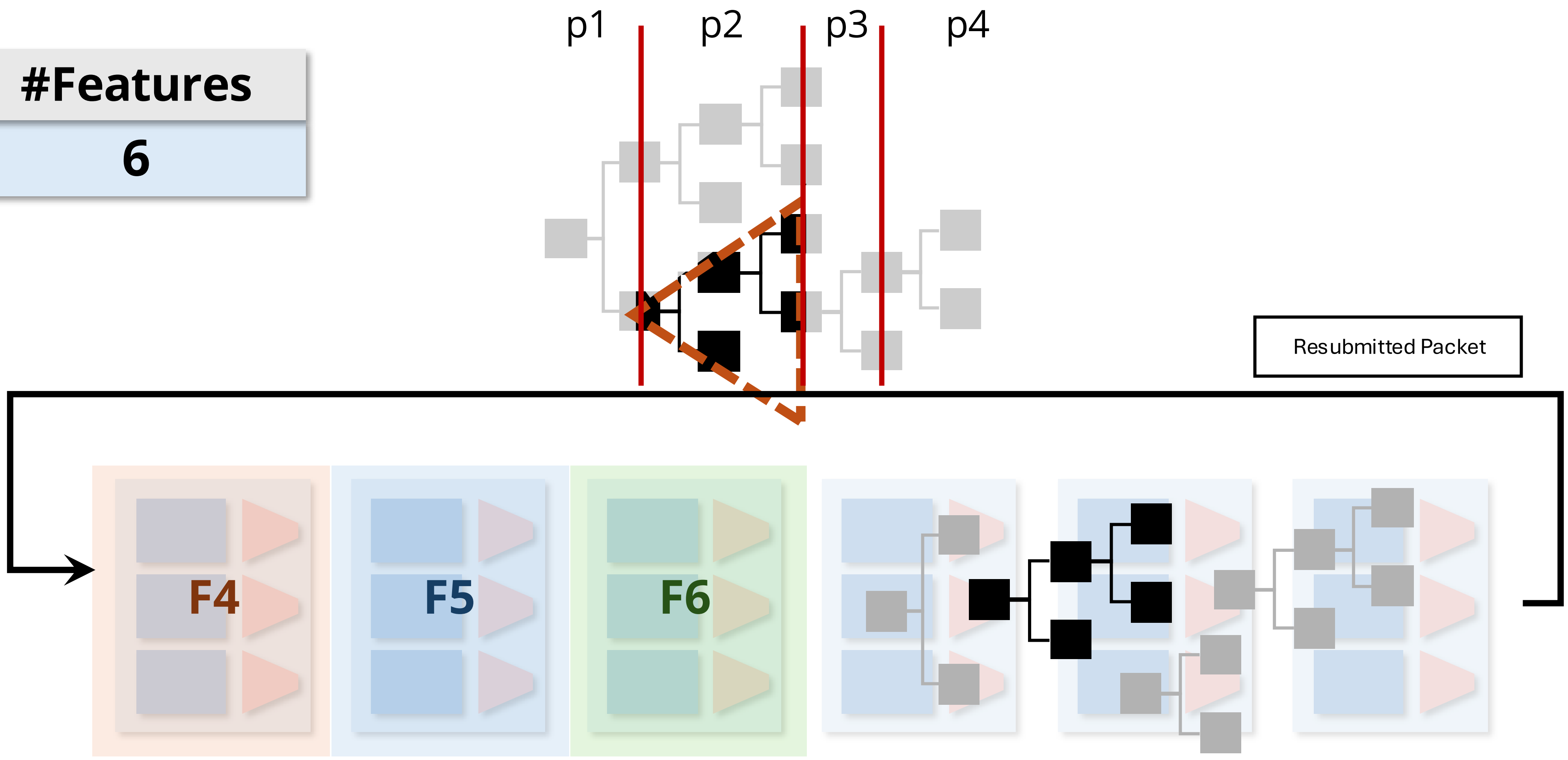


Partitioned Inference Architecture



Partitioned Inference Architecture

#Features
6

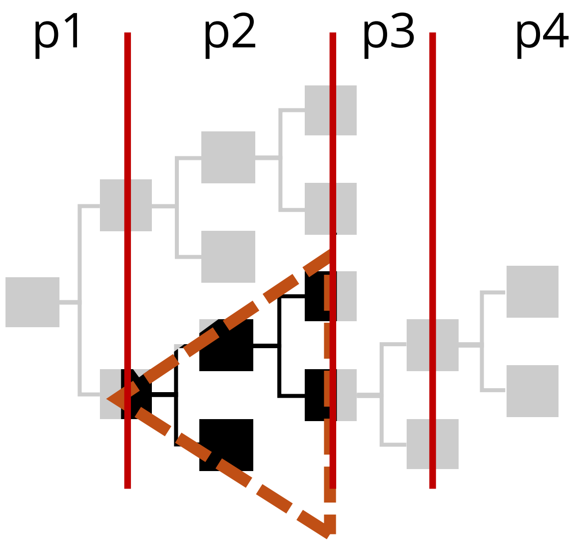


Feature Collection & Engineering

Model Inference

Partitioned Inference Architecture

#Features
6

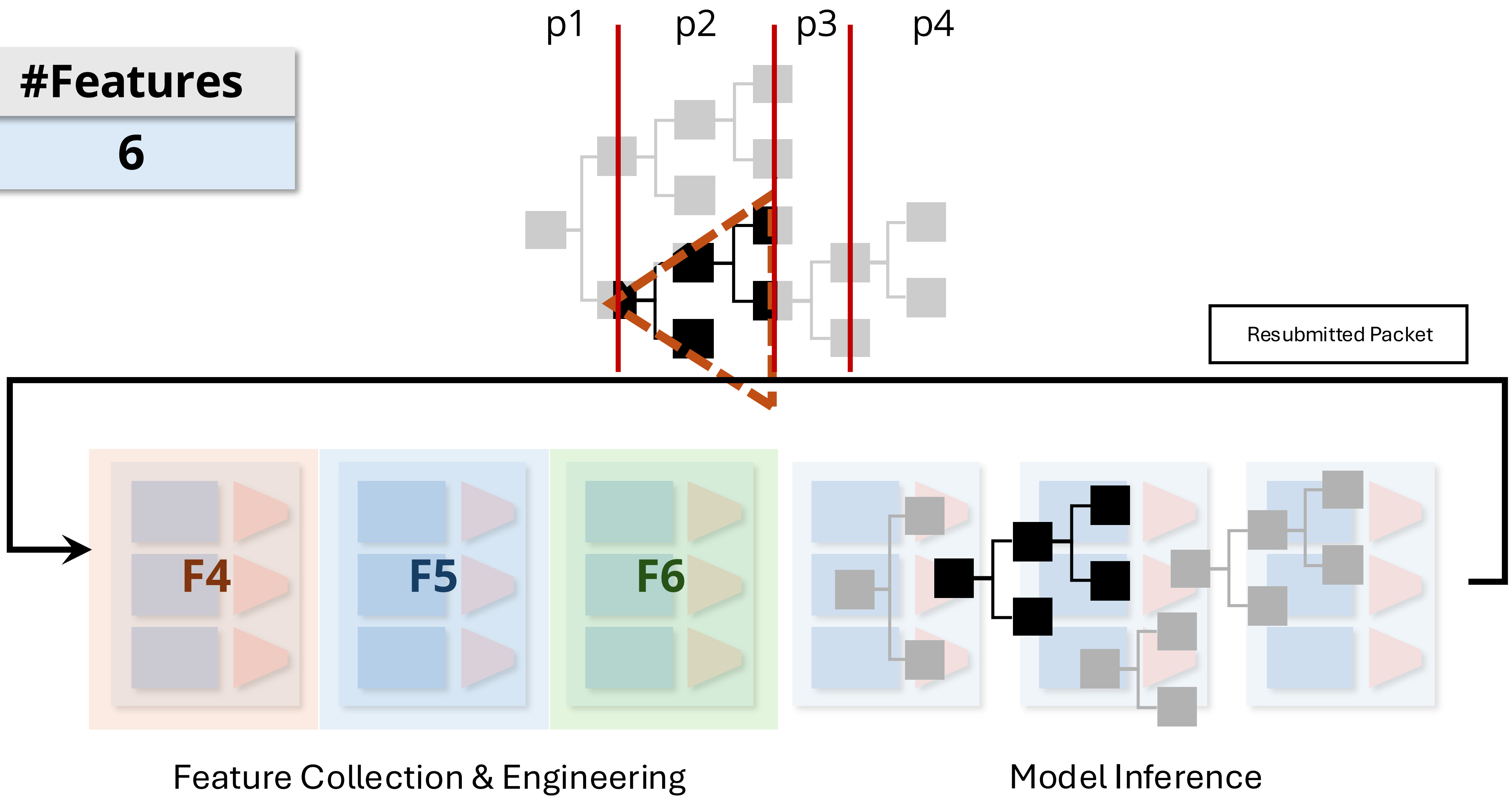


Feature Collection & Engineering

Model Inference

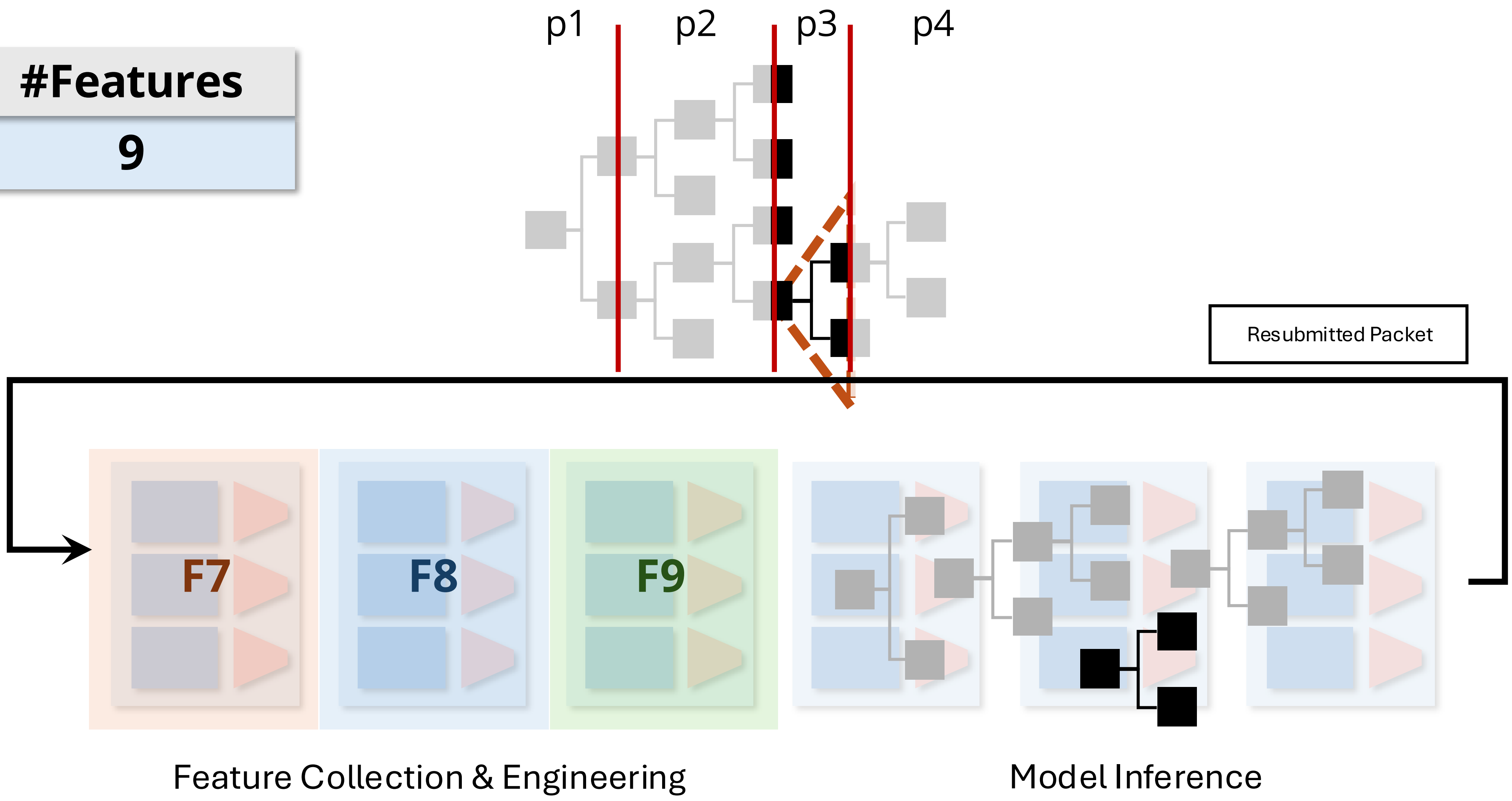
Partitioned Inference Architecture

#Features
6



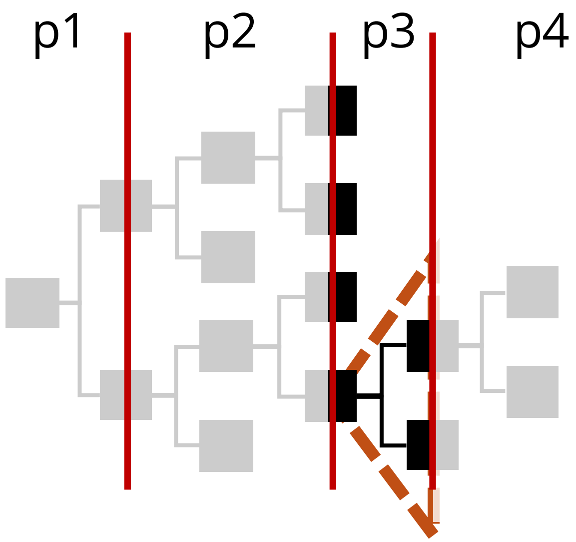
Partitioned Inference Architecture

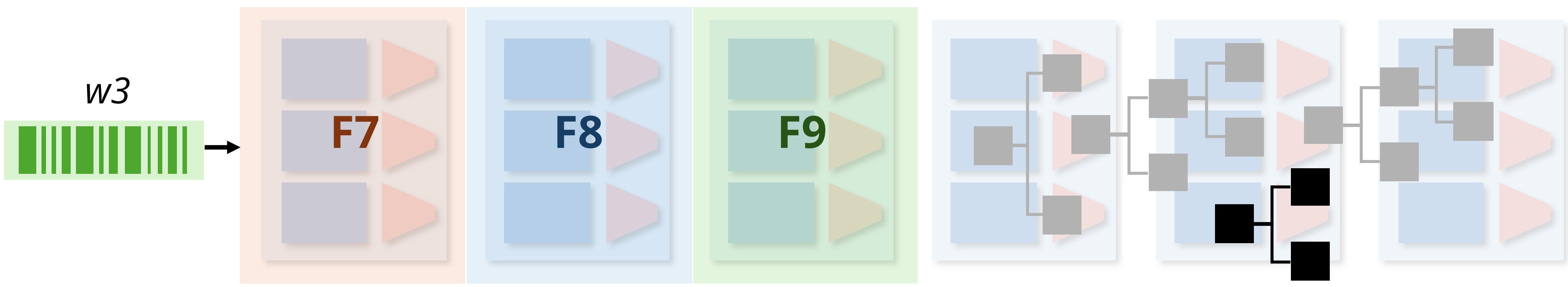
#Features
9



Partitioned Inference Architecture

#Features
9



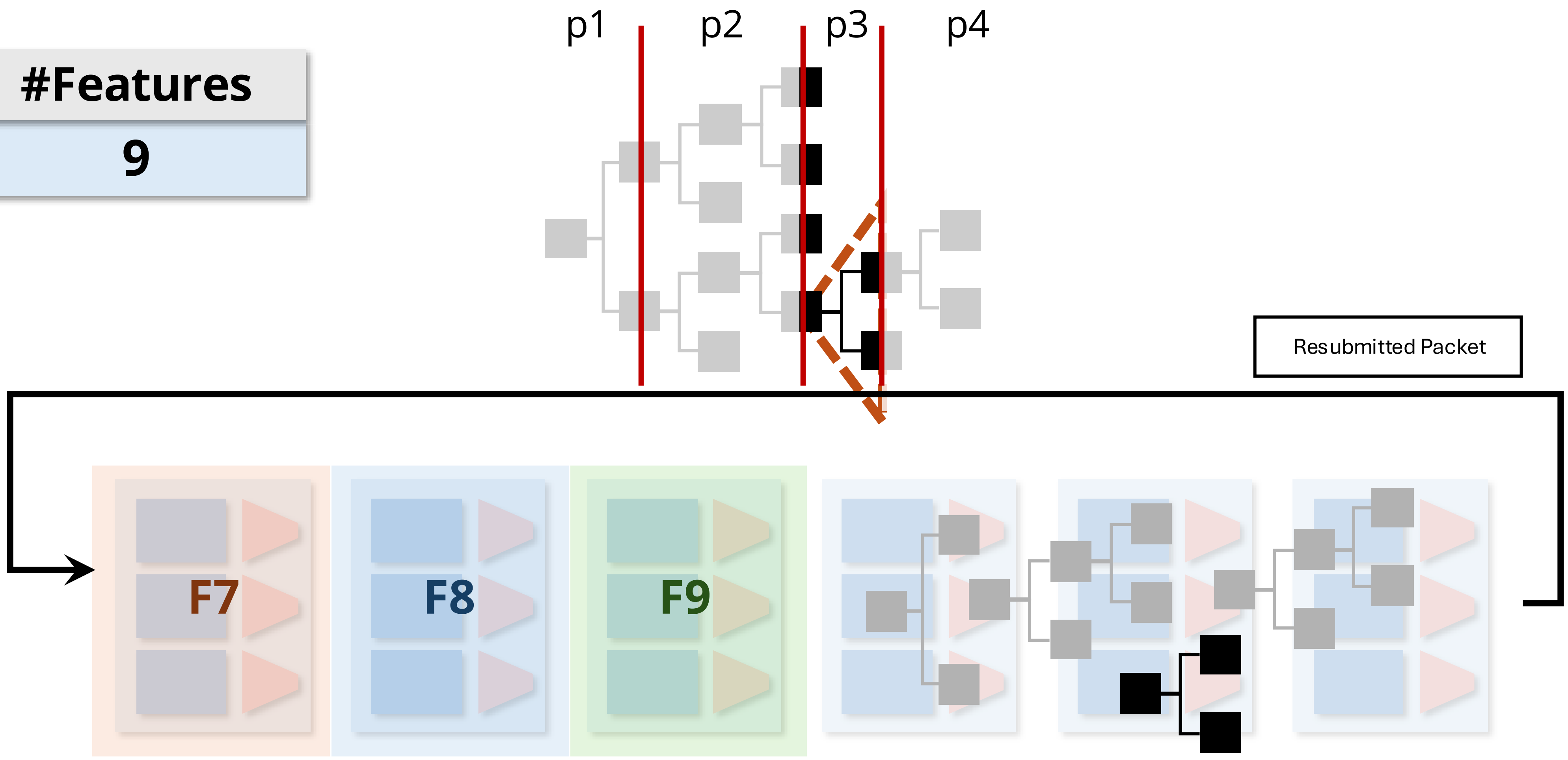


Feature Collection & Engineering

Model Inference

Partitioned Inference Architecture

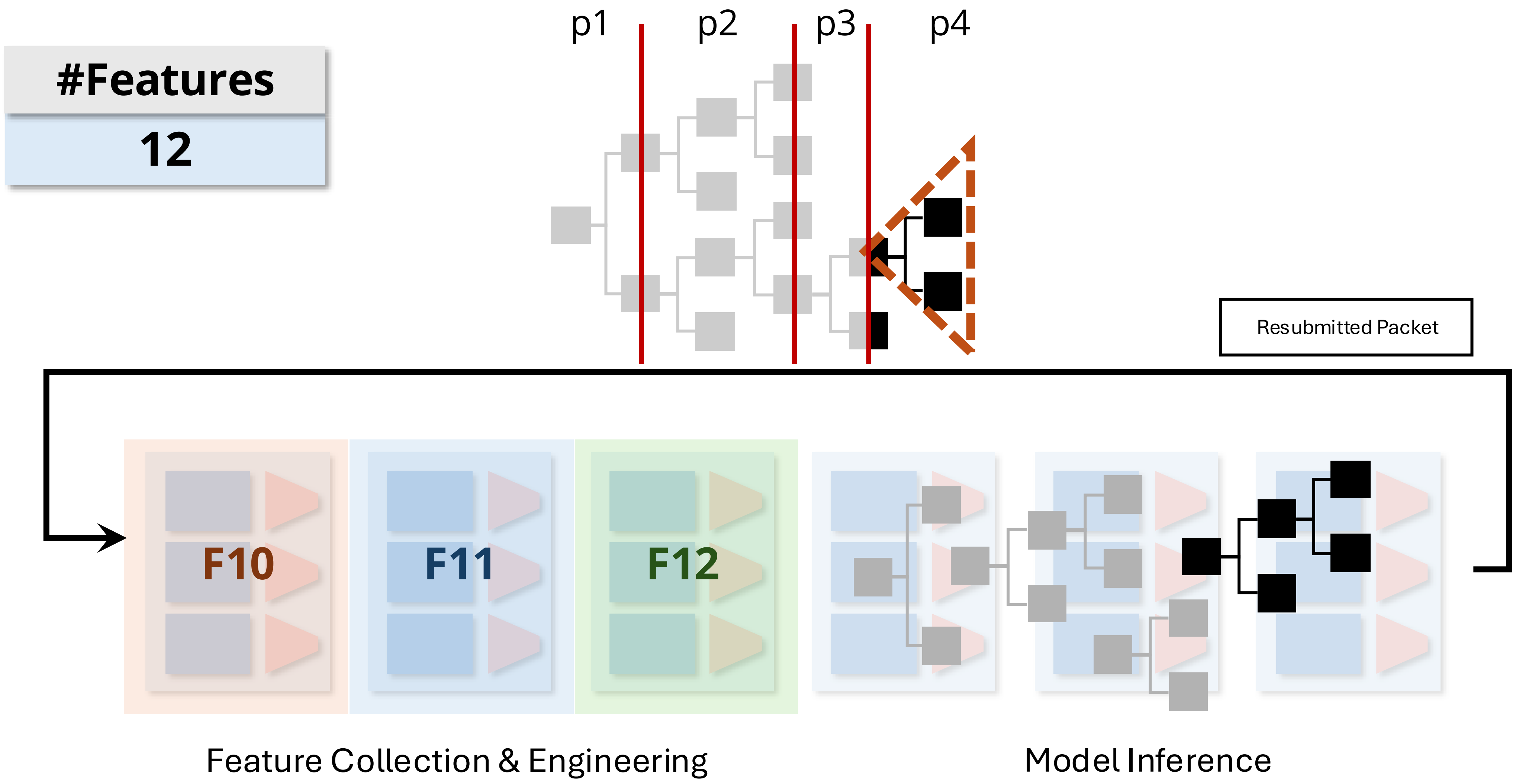
#Features
9



Feature Collection & Engineering

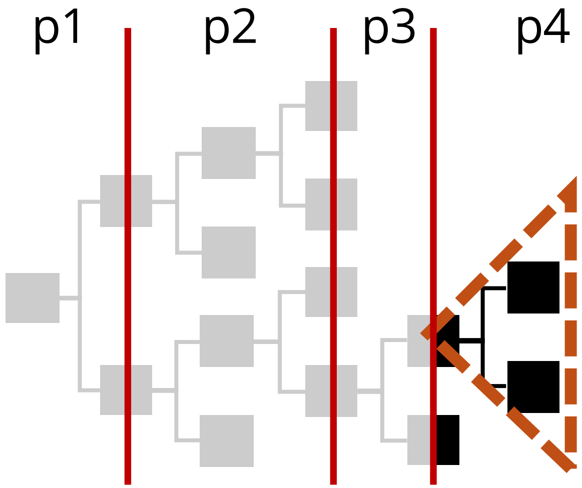
Model Inference

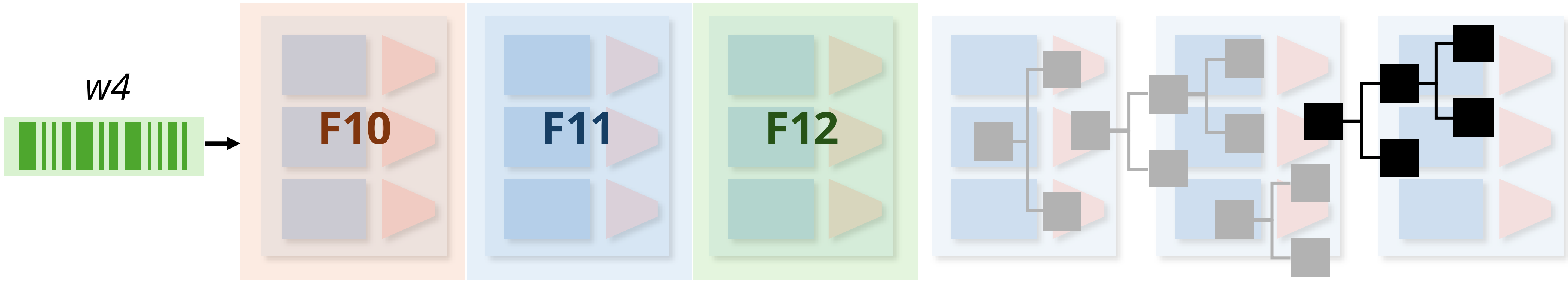
Partitioned Inference Architecture



Partitioned Inference Architecture

#Features
12



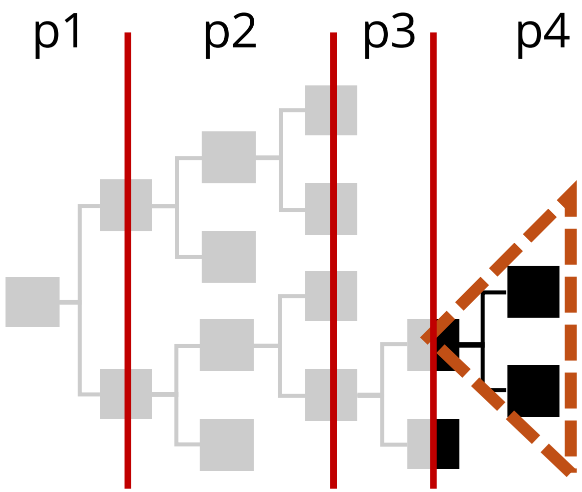


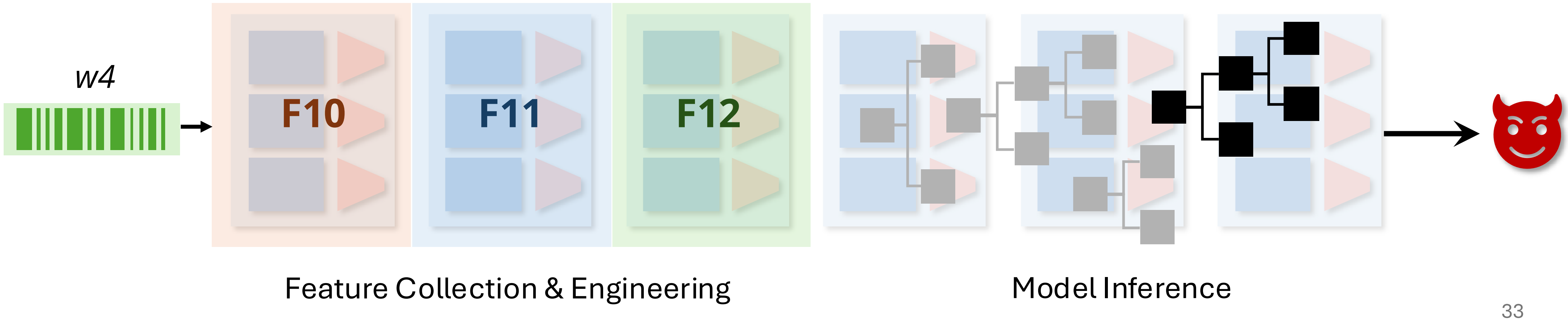
Feature Collection & Engineering

Model Inference

Partitioned Inference Architecture

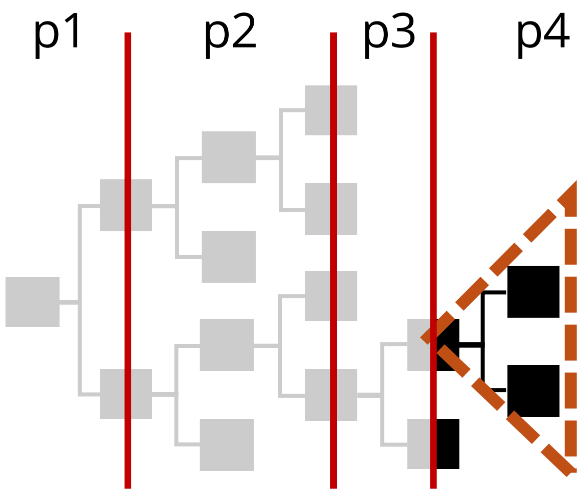
#Features
12

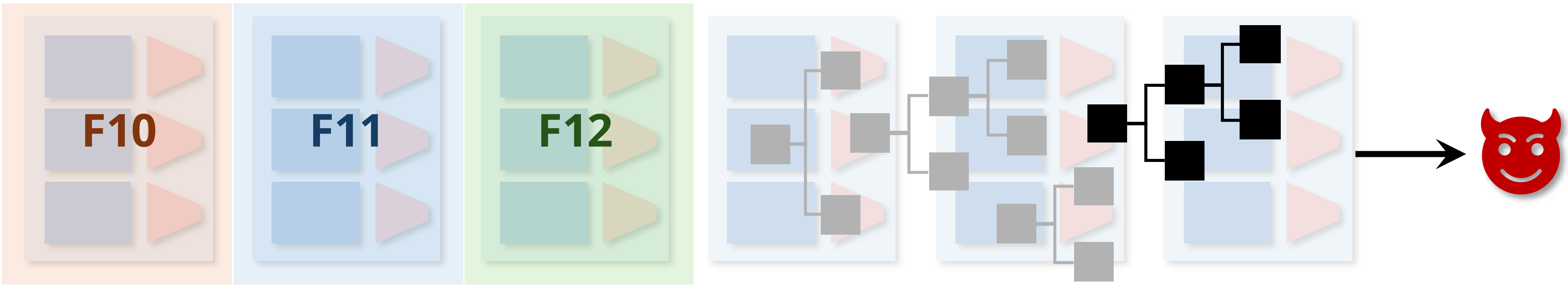




Partitioned Inference Architecture

#Features
12



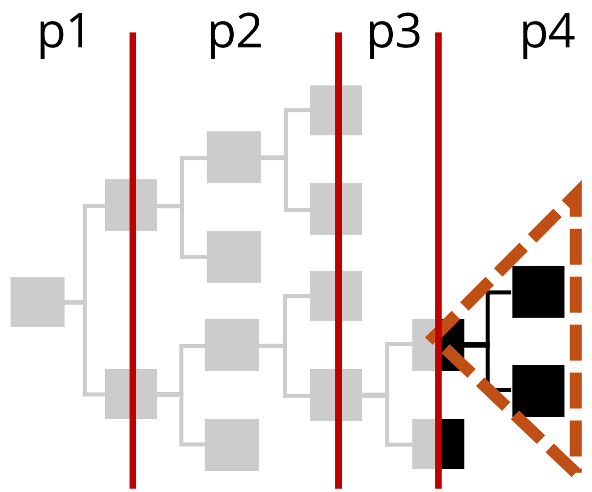


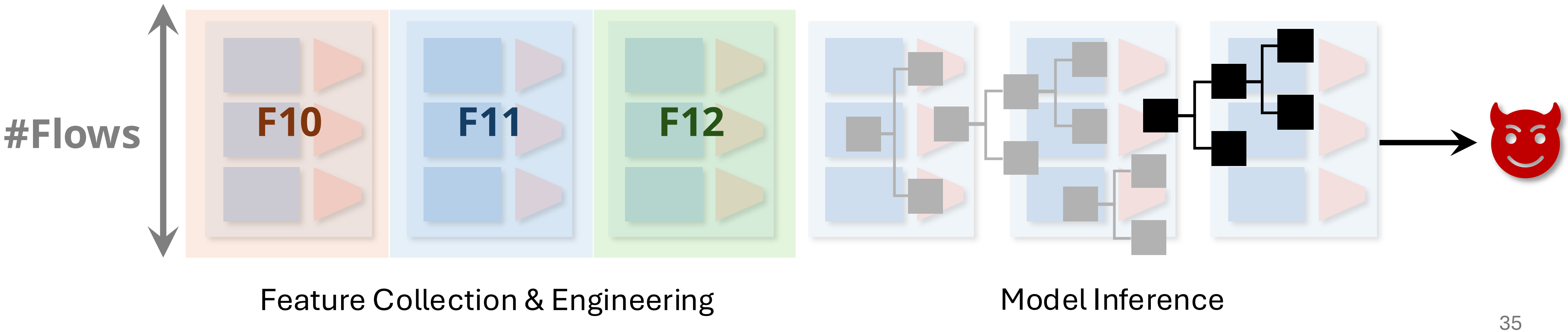
Feature Collection & Engineering

Model Inference

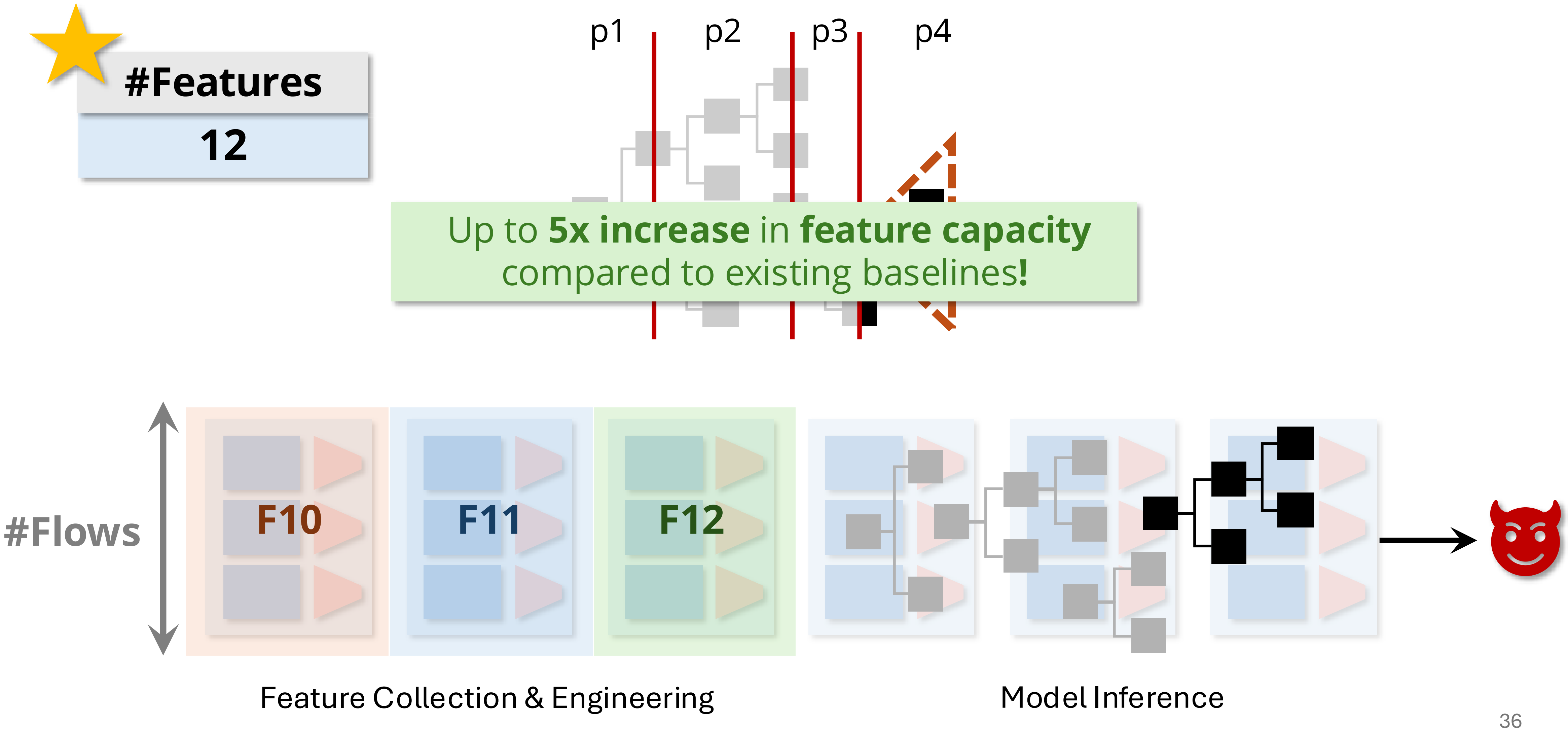
Partitioned Inference Architecture

#Features
12



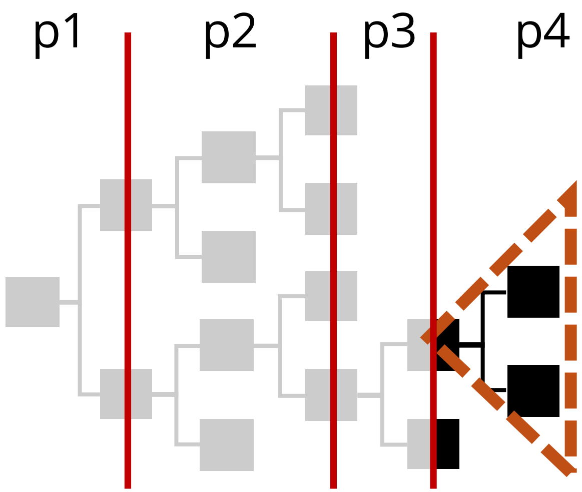


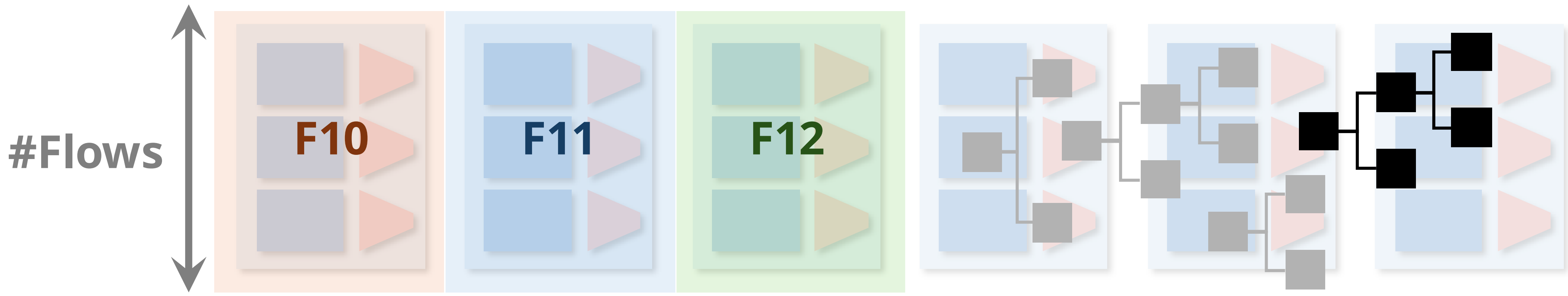
Partitioned Inference Architecture



Design Challenges

#Features
12

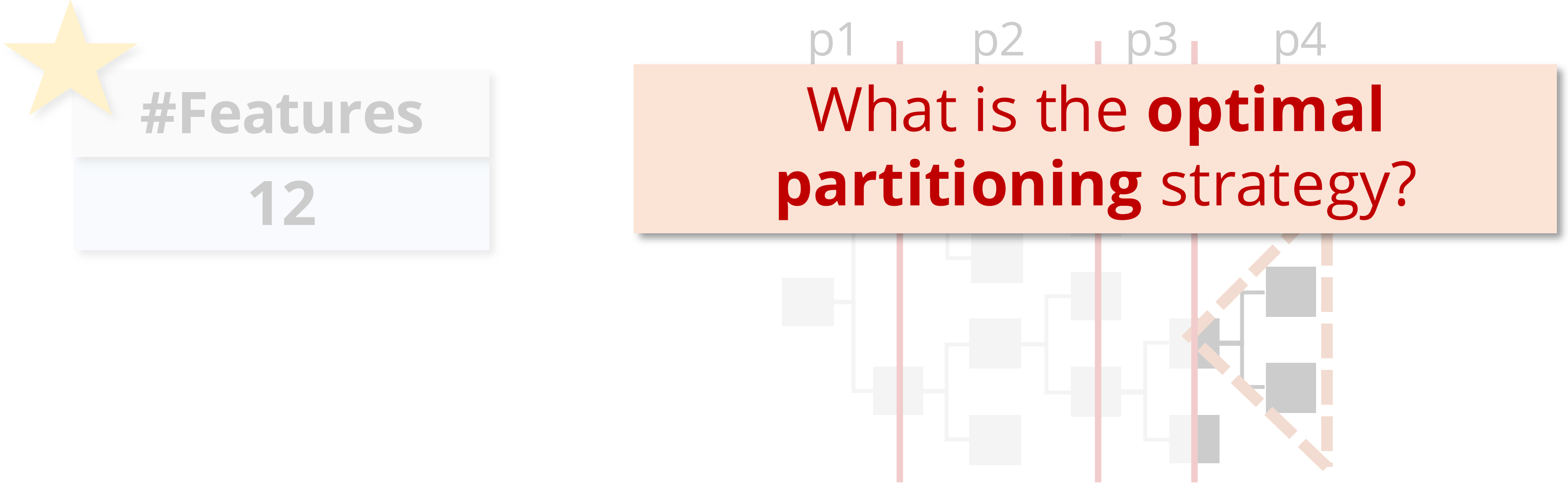




Feature Collection & Engineering

Model Inference

Design Challenges



What is the **optimal partitioning** strategy?

Design Challenges

#Features

12

p1 p2 p3 p4

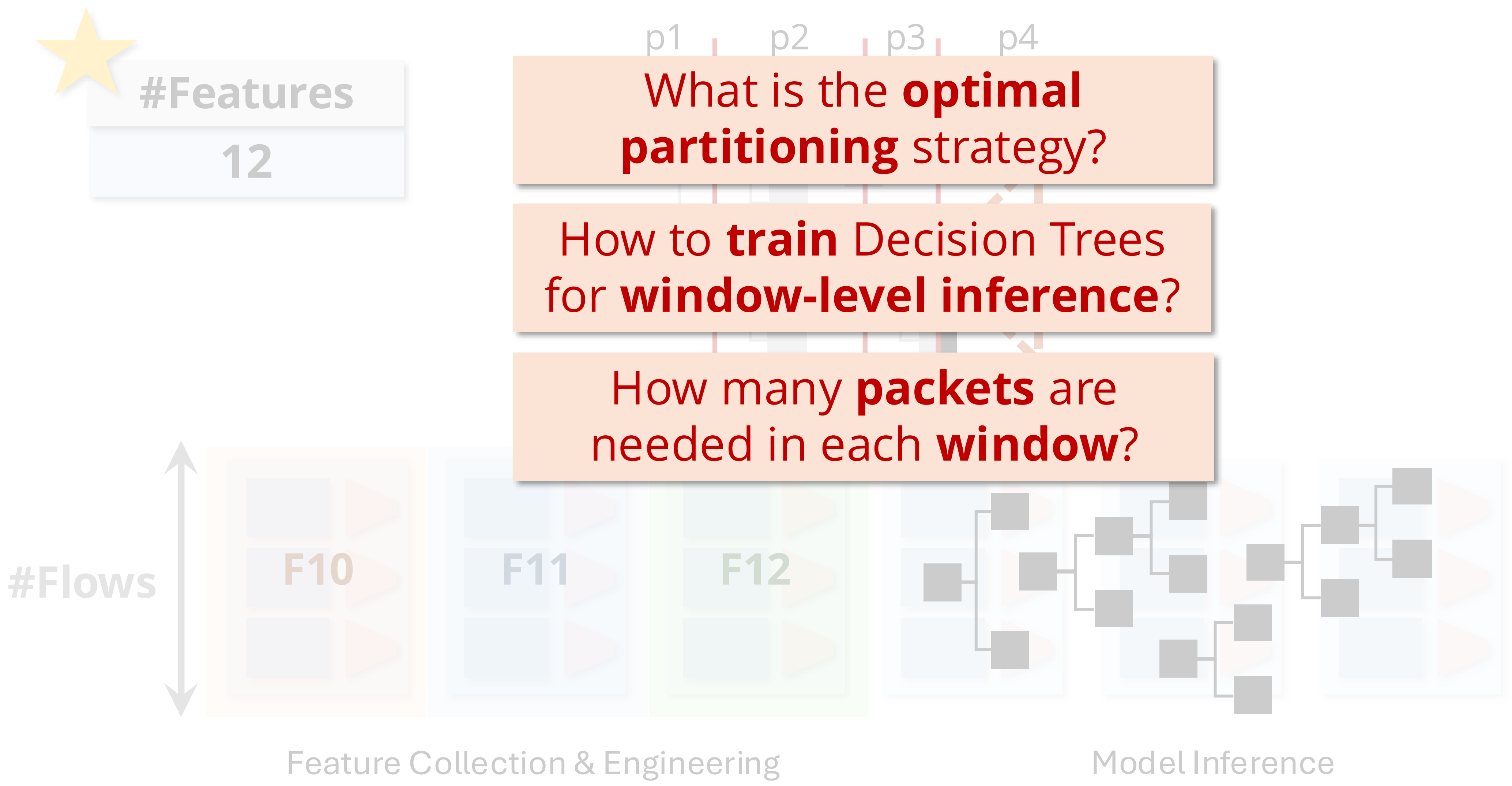
What is the **optimal partitioning** strategy?

How to **train** Decision Trees for **window-level inference**?

Feature Collection & Engineering

Model Inference

Design Challenges

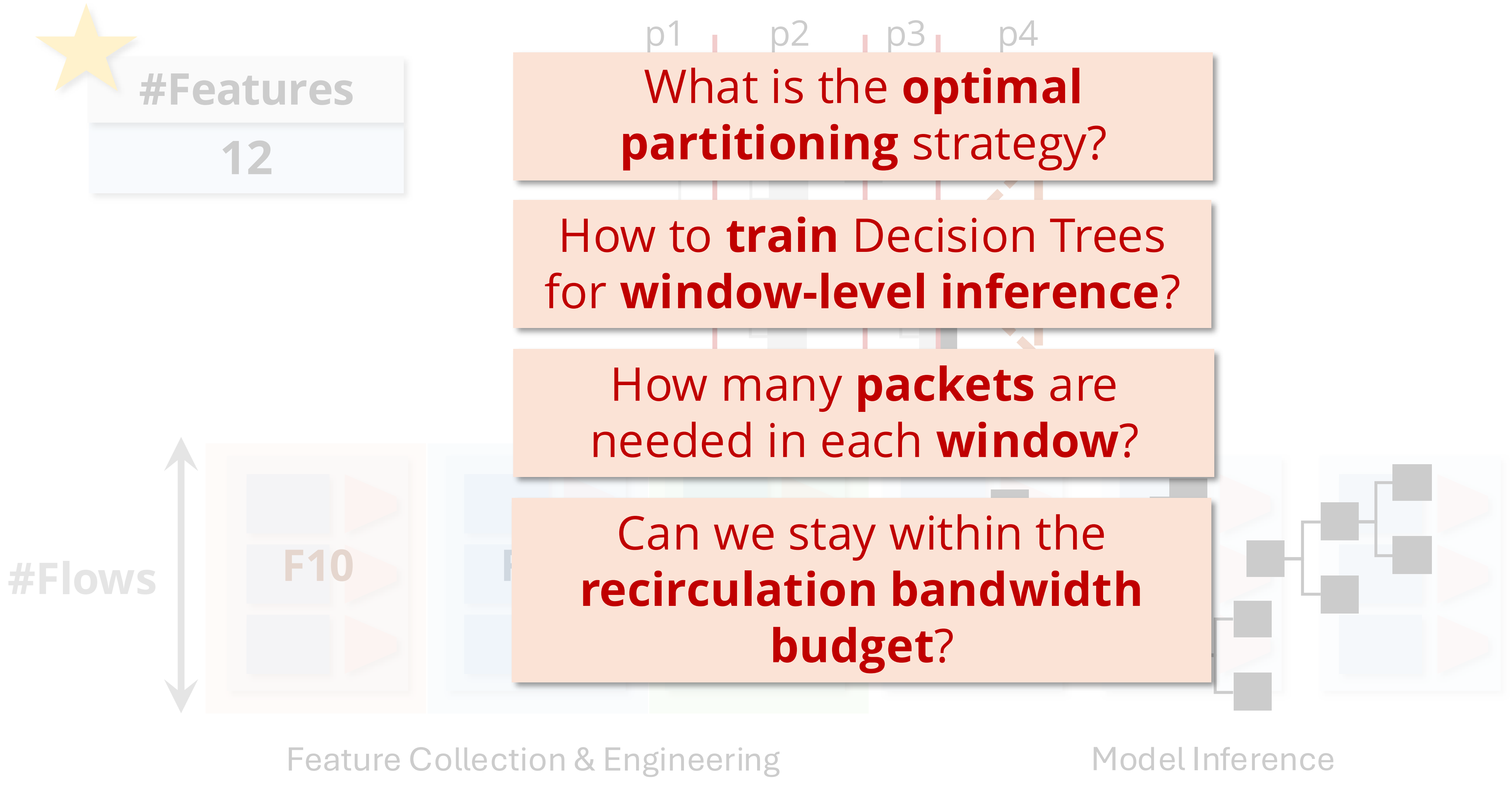


What is the **optimal partitioning** strategy?

How to **train** Decision Trees for **window-level inference**?

How many **packets** are needed in each **window**?

Design Challenges



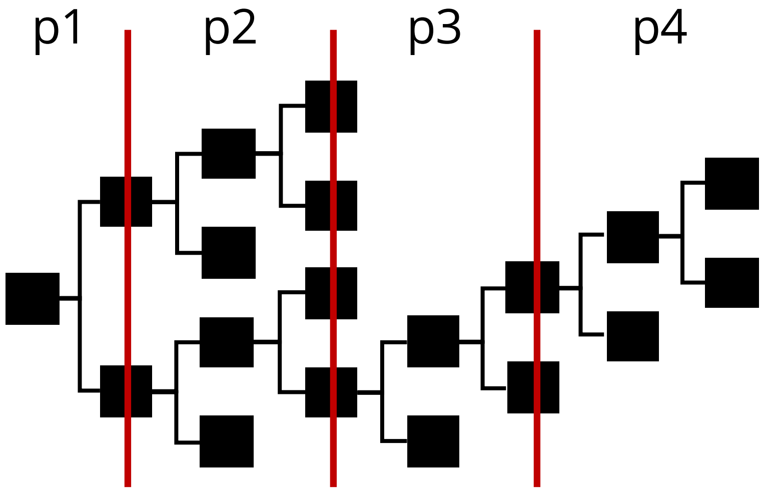
What is the **optimal partitioning** strategy?

How to **train** Decision Trees for **window-level inference**?

How many **packets** are needed in each **window**?

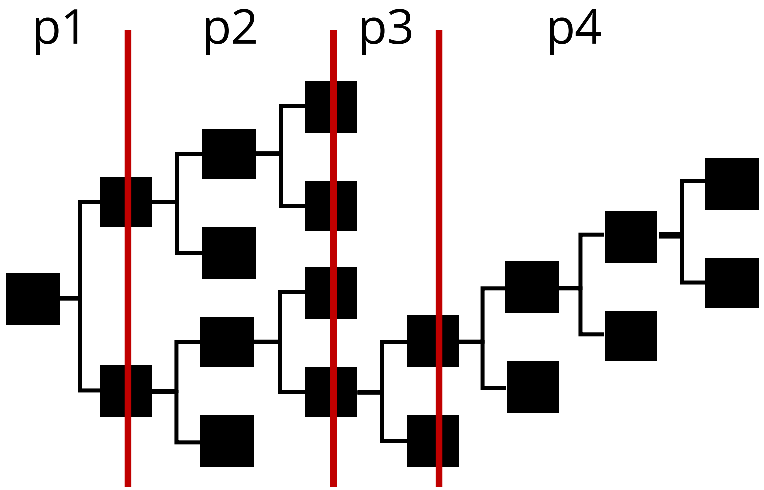
Can we stay within the **recirculation bandwidth budget**?

Partitioning Strategy



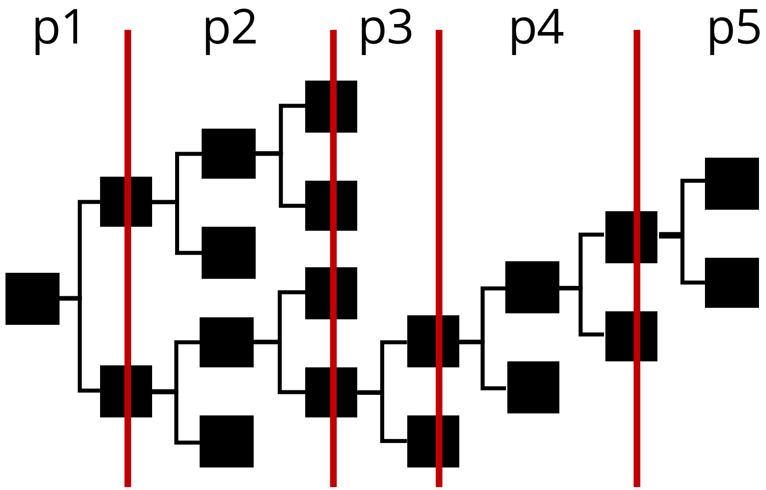
What is the **optimal partitioning** strategy?

Partitioning Strategy



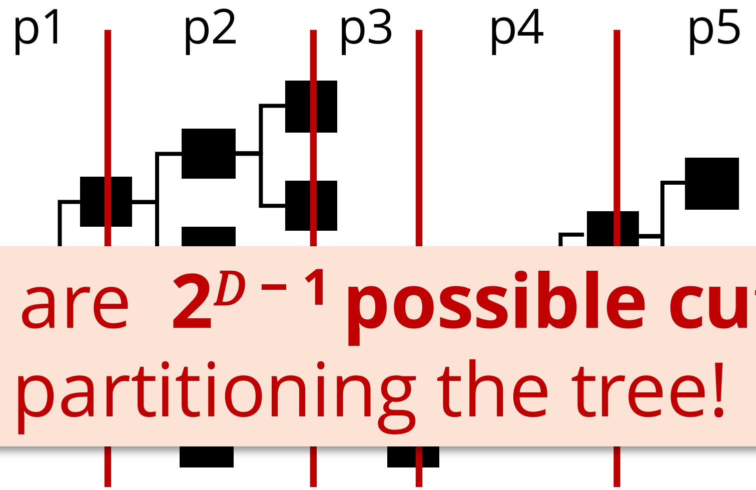
What is the **optimal partitioning** strategy?

Partitioning Strategy



What is the **optimal partitioning** strategy?

Partitioning Strategy



There are $\mathbf{2^{D-1}}$ **possible cuts** for partitioning the tree!

What is the **optimal partitioning** strategy?

Design Space Exploration

Tree Depth

#Partitions /Cuts

Max Features per Subtree

Parameter Space

Design Space Exploration

Parameter Space
- Tree Depth
- #Partitions /Cuts
- Max Features per Subtree

Hardware Constraints
- Registers
- #TCAM entries
- Recirculation Bandwidth

Optimization Objectives
- Maximize F1 score
- Maximize #Flows

Design Space Exploration

Tree Depth

#Partitions

Max Features per Subtree

Parameter Space

Registers

Recirculation Bandwidth

Hardware Constraints

Maximize F1 score

#Flows

Optimization Objectives

The goal is to **maximize model accuracy** and **flow scalability** while operating within **hardware constraints**

Design Space Exploration

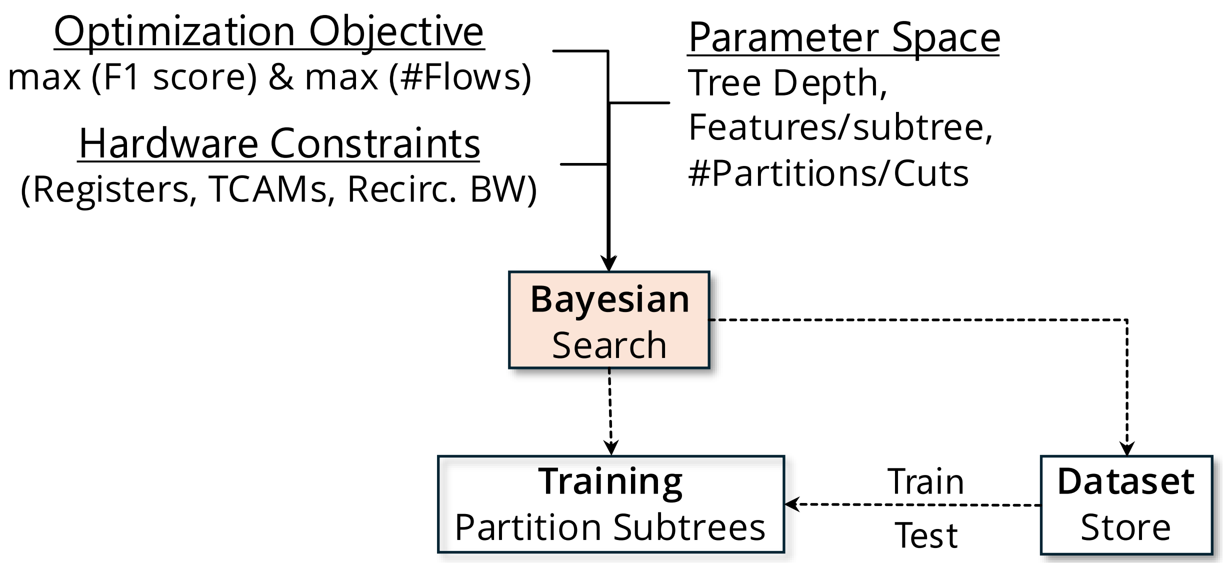
Optimization Objective
max (F1 score) & max (#Flows)

Hardware Constraints
(Registers, TCAMs, Recirc. BW)

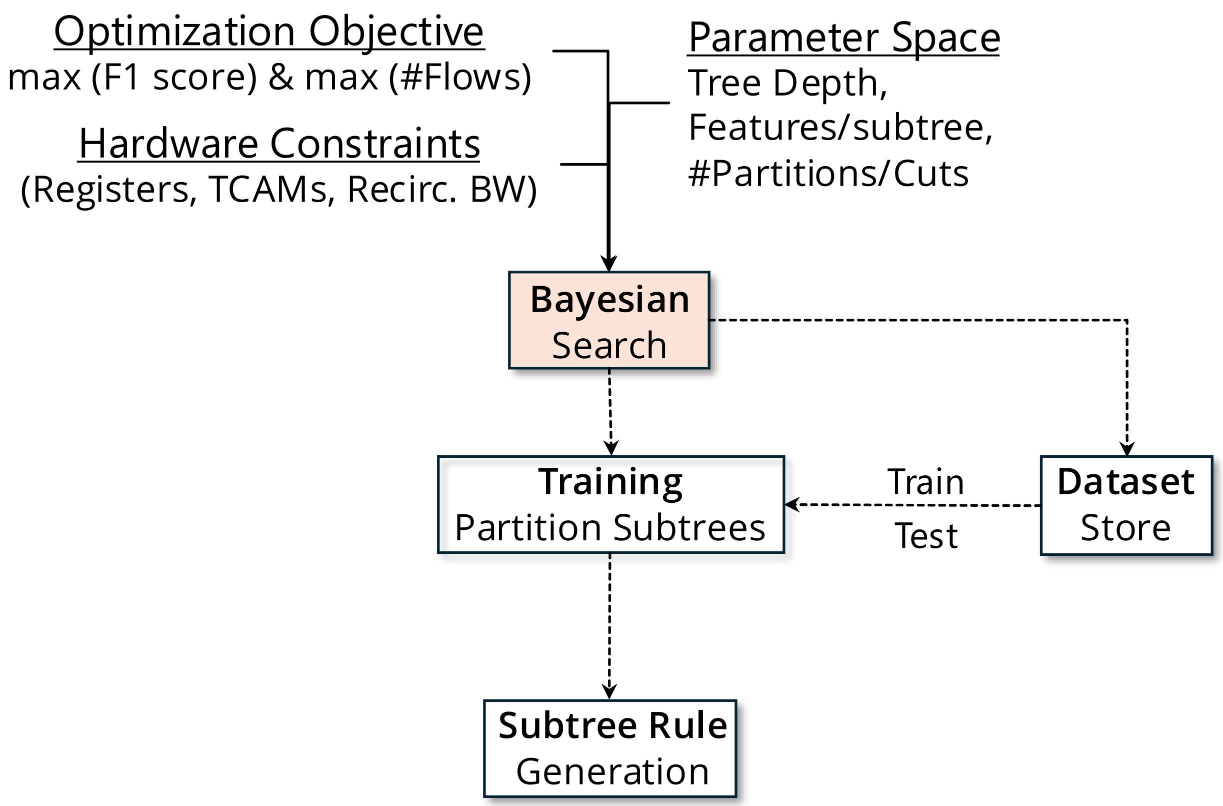
Parameter Space
Tree Depth,
Max Features/subtree,
#Partitions/Cuts

Bayesian Search

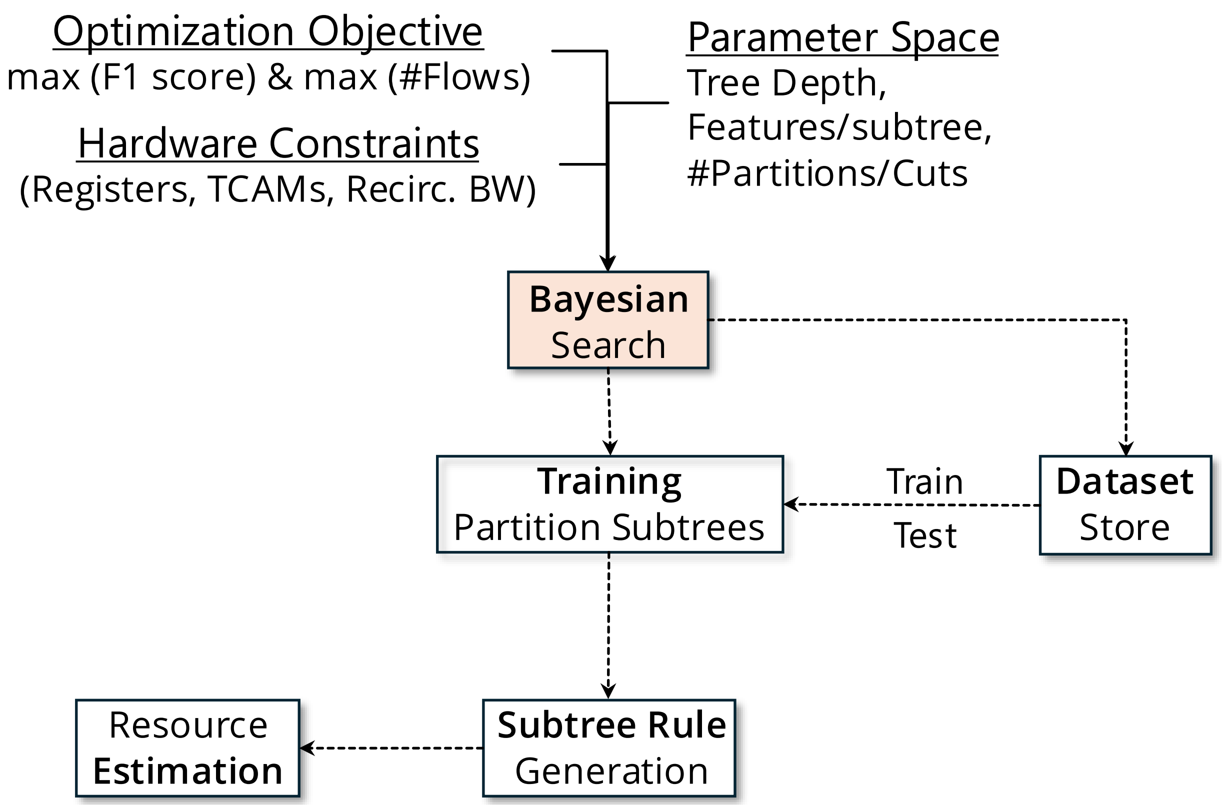
Design Space Exploration



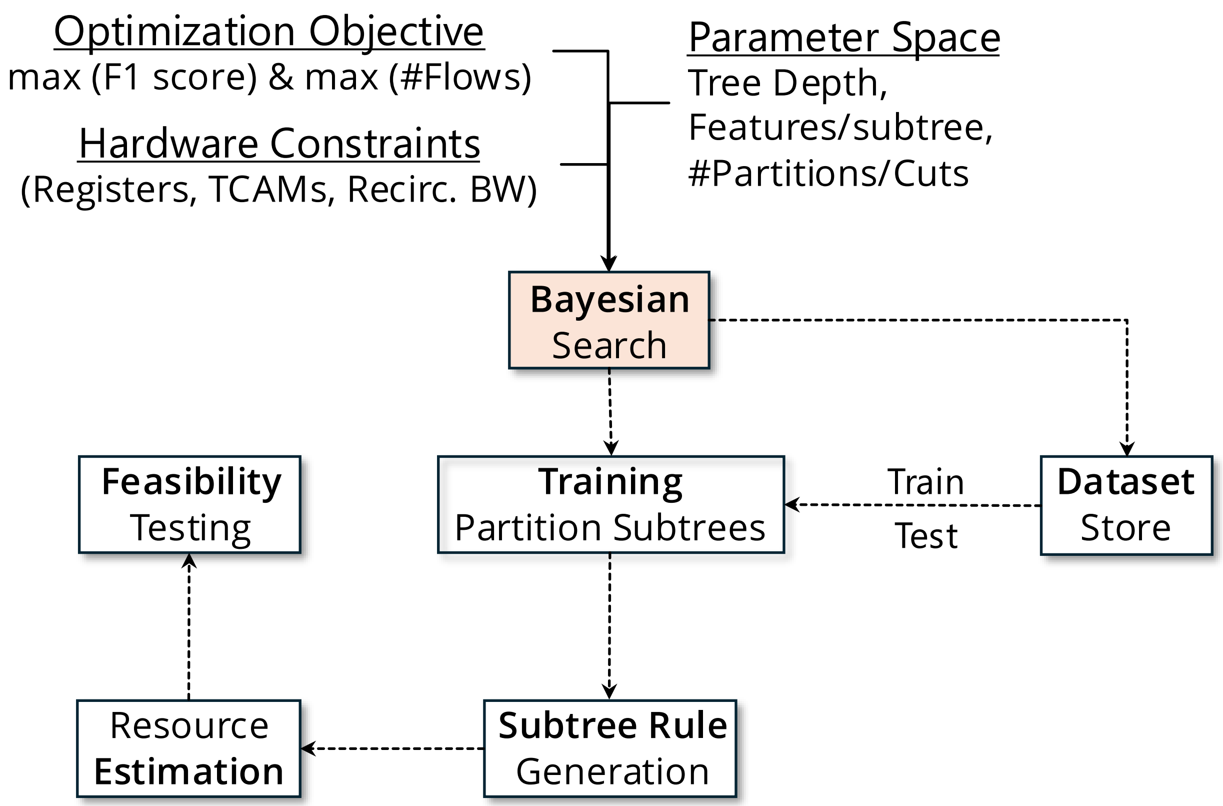
Design Space Exploration



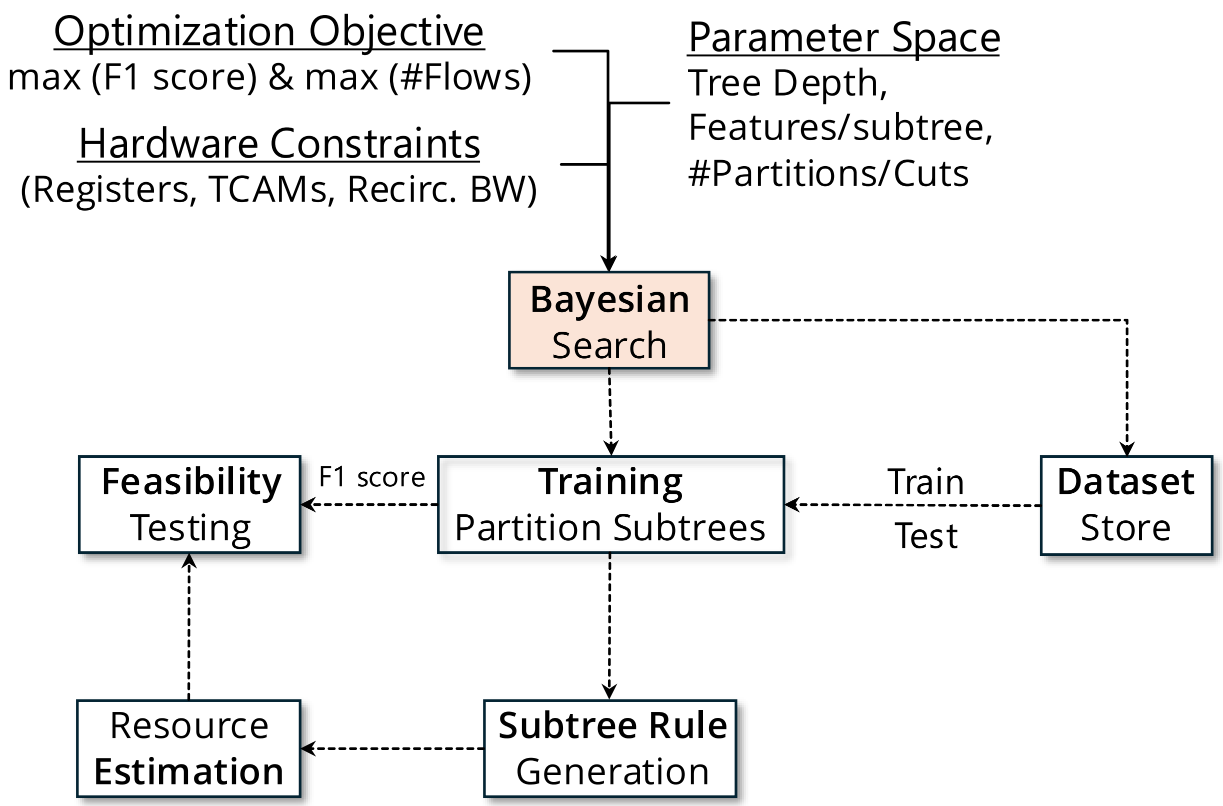
Design Space Exploration



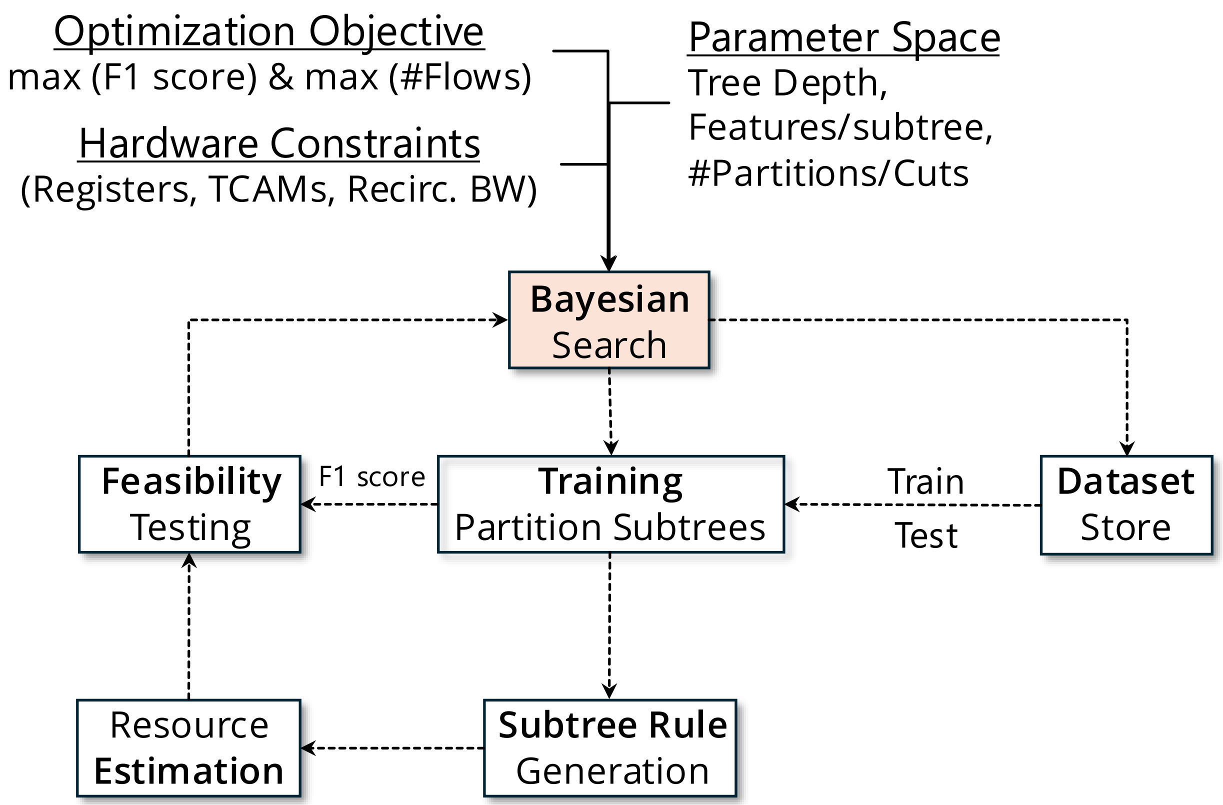
Design Space Exploration



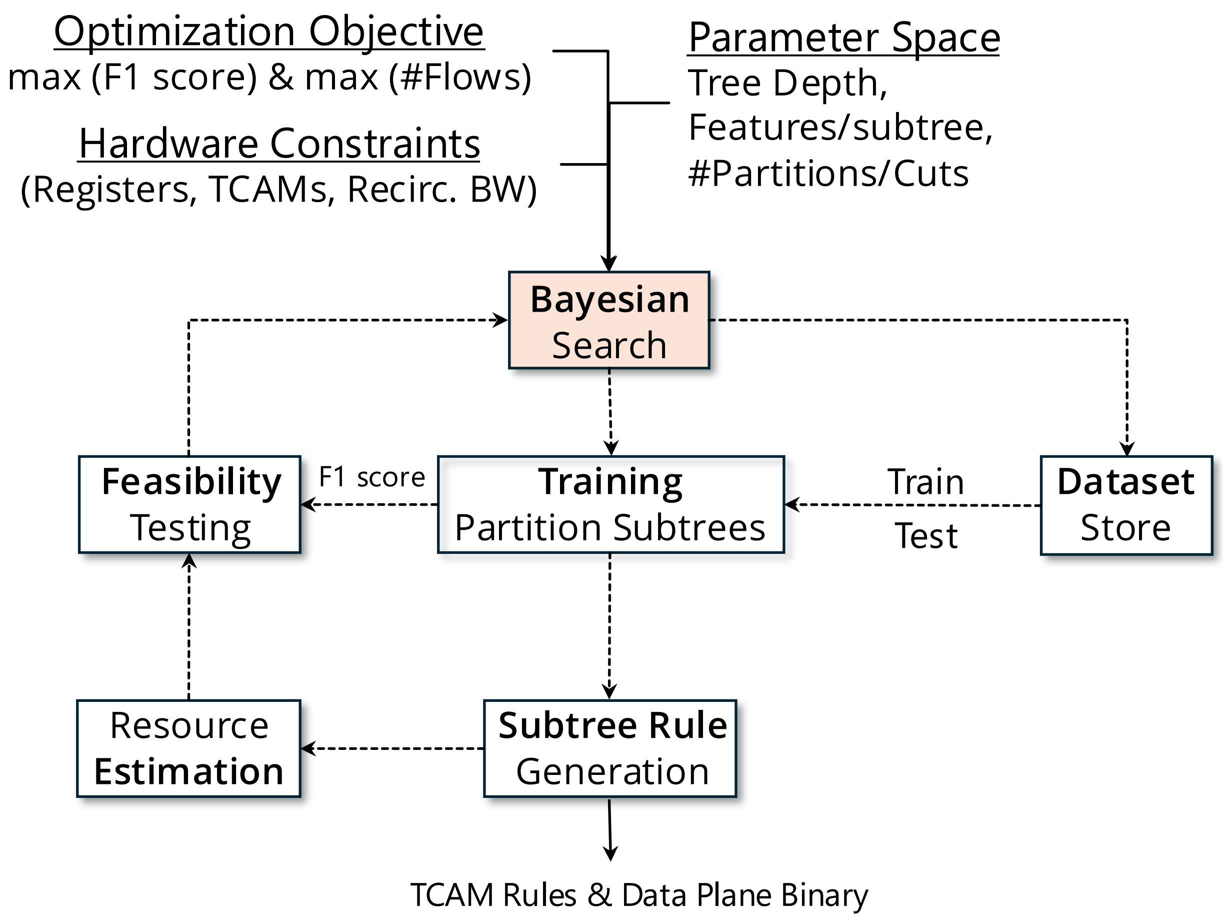
Design Space Exploration



Design Space Exploration



Design Space Exploration



Design Search Performance

Stages	Mean Time
Data Fetch	0.38s
Training	268.14s
Optimizer	37.57s
Rulegen	0.87s
Backend	44.6 μs
Total Time	306.29s

Average per-iteration breakdown of the various stages of SpliDT framework

Design Search Performance

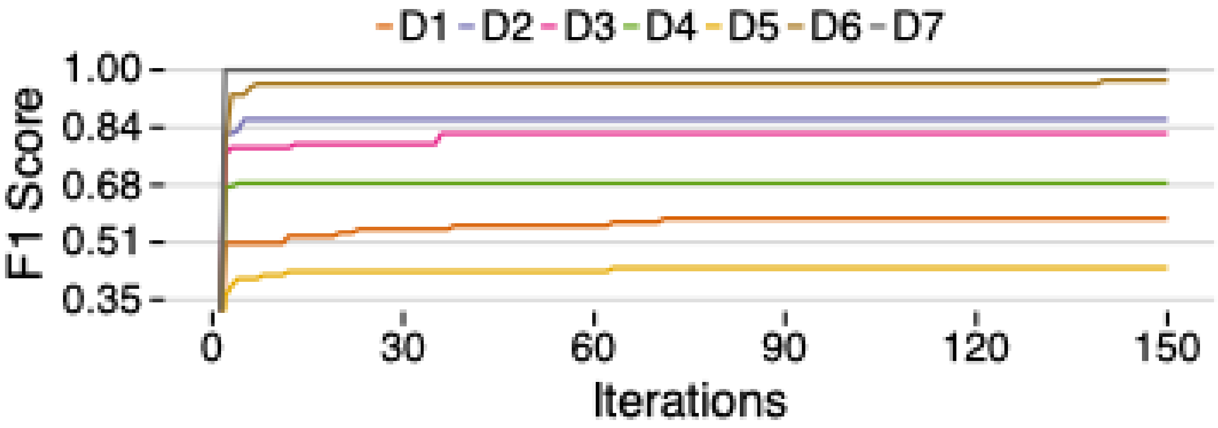
Stages	Mean Time
Data Fetch	0.38s
Training	268.14s
Optimizer	37.57s
Rulegen	0.87s
Backend	44.6 μs
Total Time	306.29s

Average per-iteration breakdown of the various stages of SpliDT framework

Design Search Performance

Stages	Mean Time
Data Fetch	0.38s
Training	268.14s
Optimizer	37.57s
Rulegen	0.87s
Backend	44.6 μs
Total Time	306.29s

Average per-iteration breakdown of the various stages of SpliDT framework

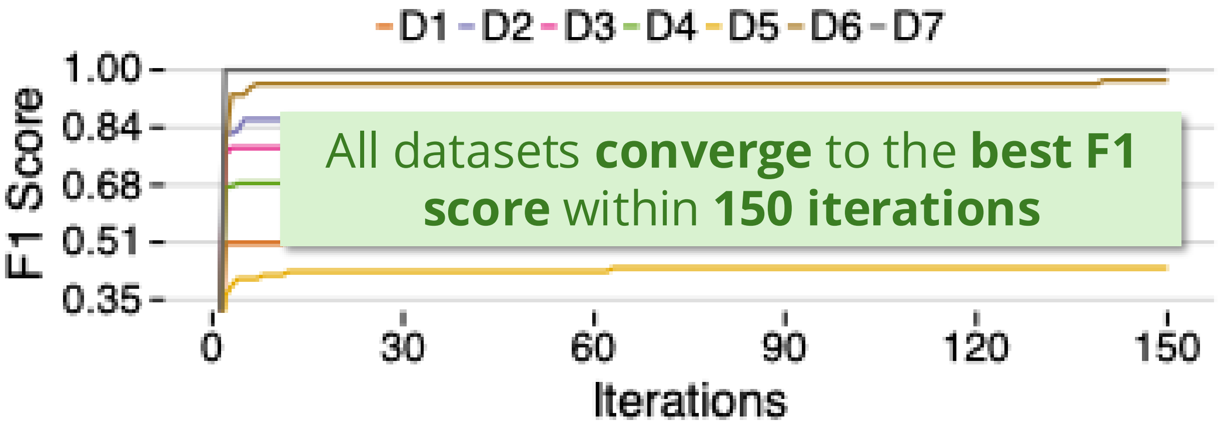


Number of BO search iterations required to reach best F1 score

Design Search Performance

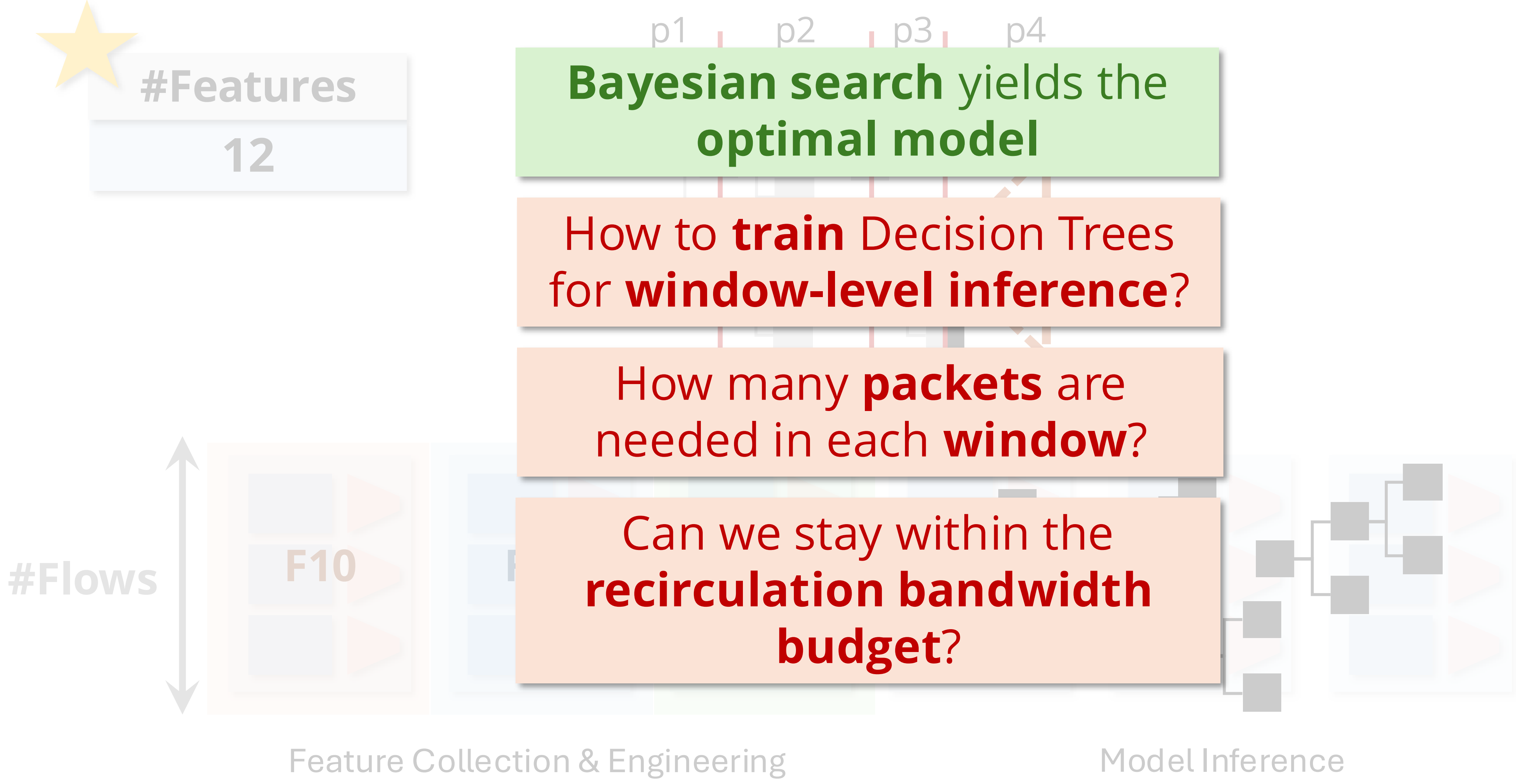
Stages	Mean Time
Data Fetch	0.38s
Training	268.14s
Optimizer	37.57s
Rulegen	0.87s
Backend	44.6 μs
Total Time	306.29s

Average per-iteration breakdown of the various stages of SpliDT framework



Number of BO search iterations required to reach best F1 score

Design Challenges



Bayesian search yields the **optimal model**

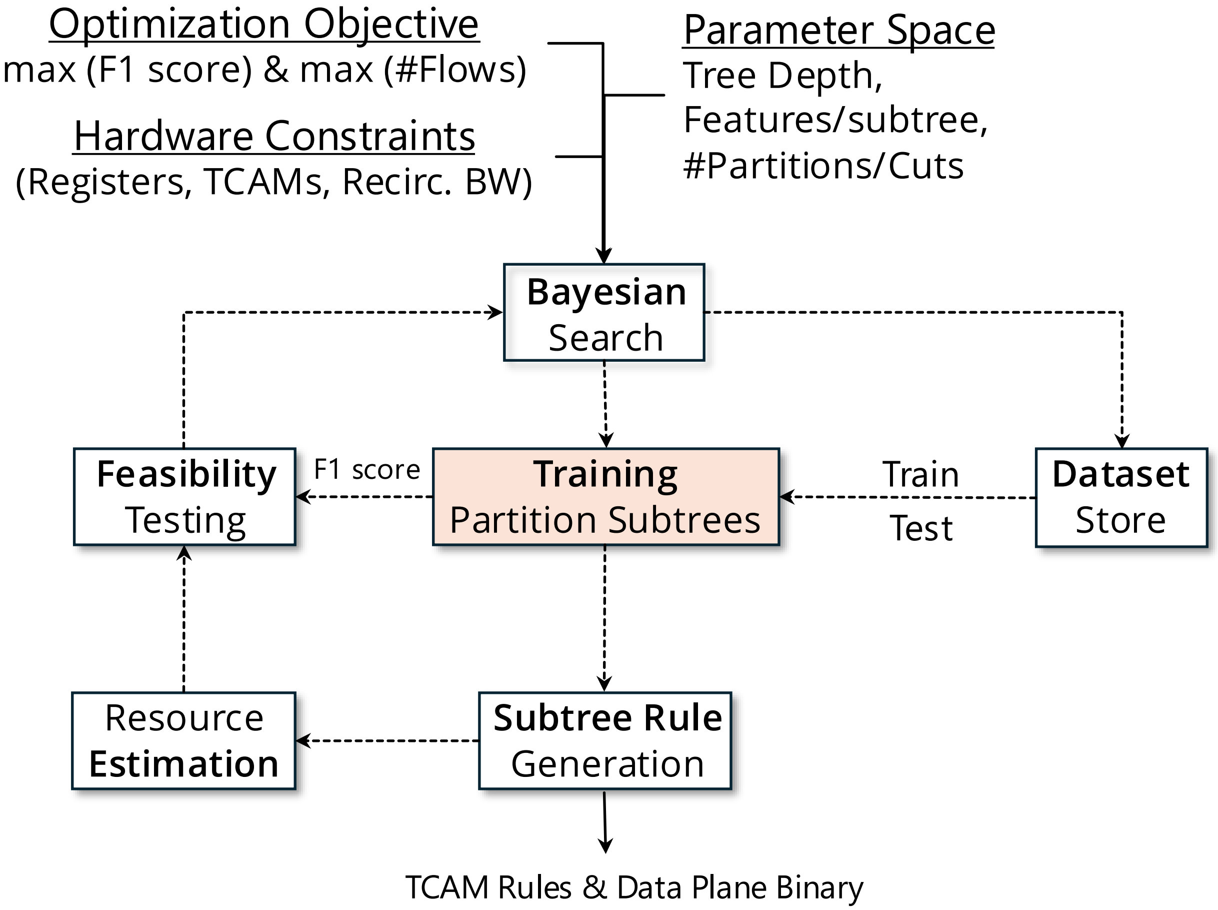
How to **train** Decision Trees for **window-level inference**?

How many **packets** are needed in each **window**?

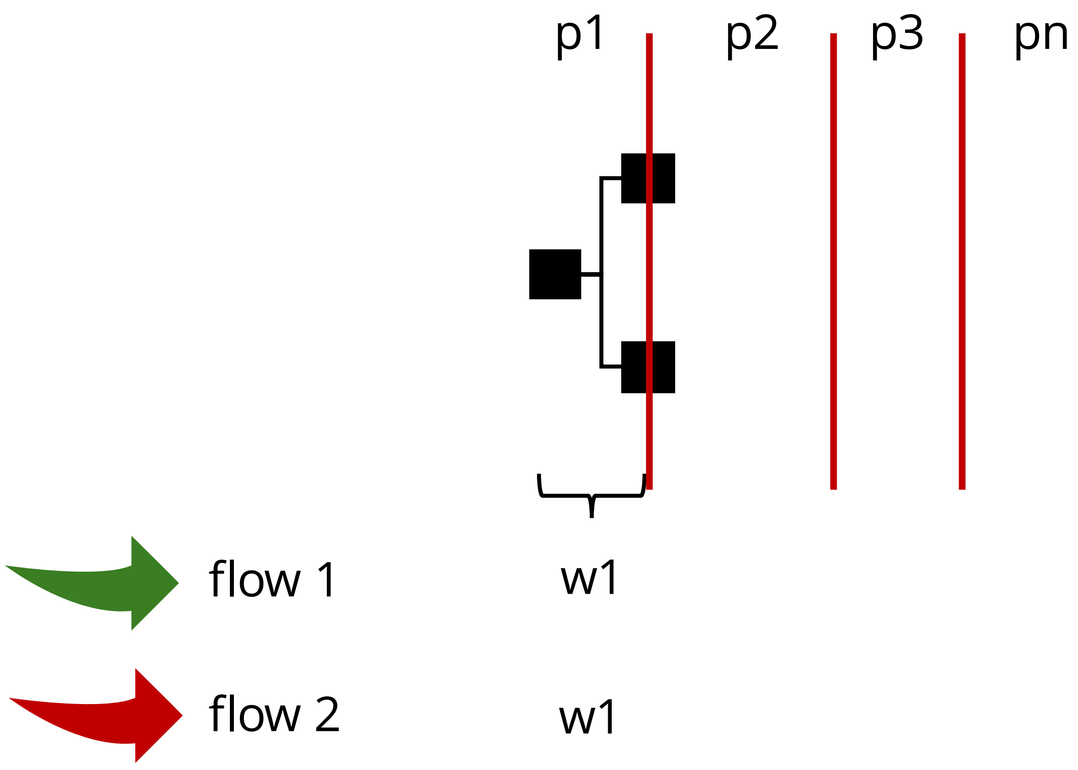
Can we stay within the **recirculation bandwidth budget**?

Training Framework

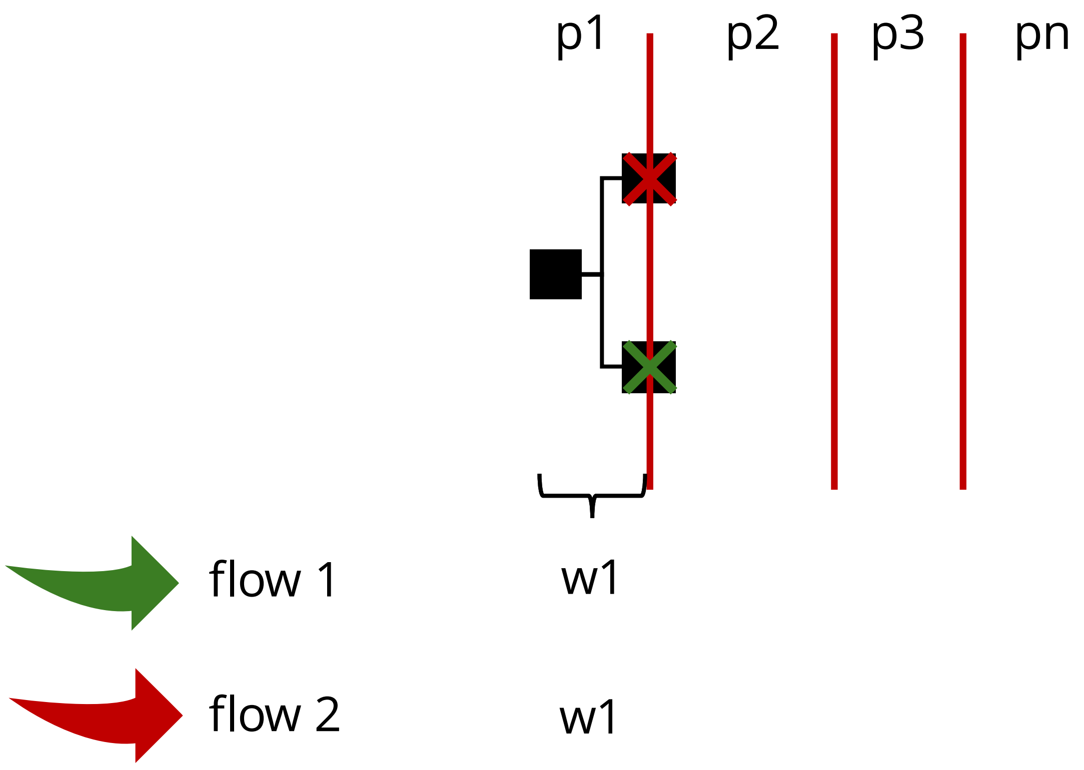
How to **train** Decision Trees for **window-level inference**?



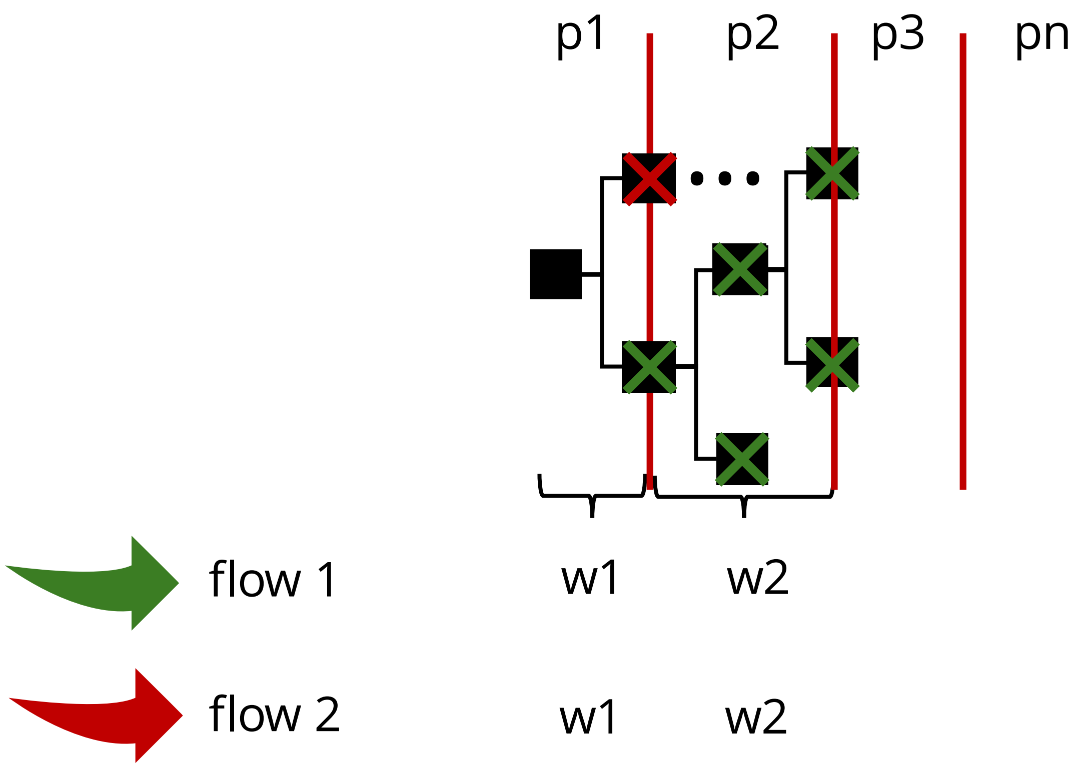
Training Framework



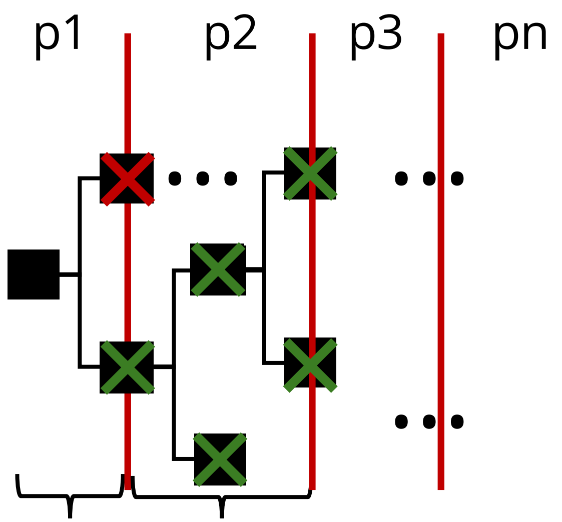
Training Framework



Training Framework



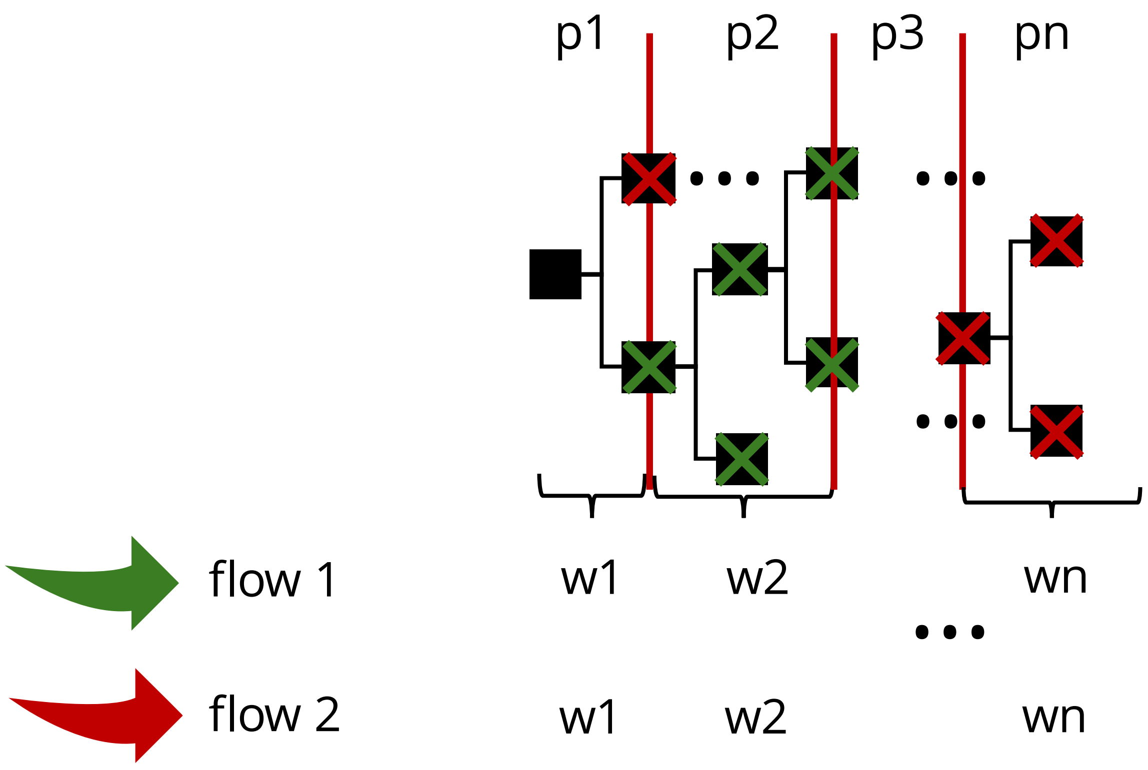
Training Framework



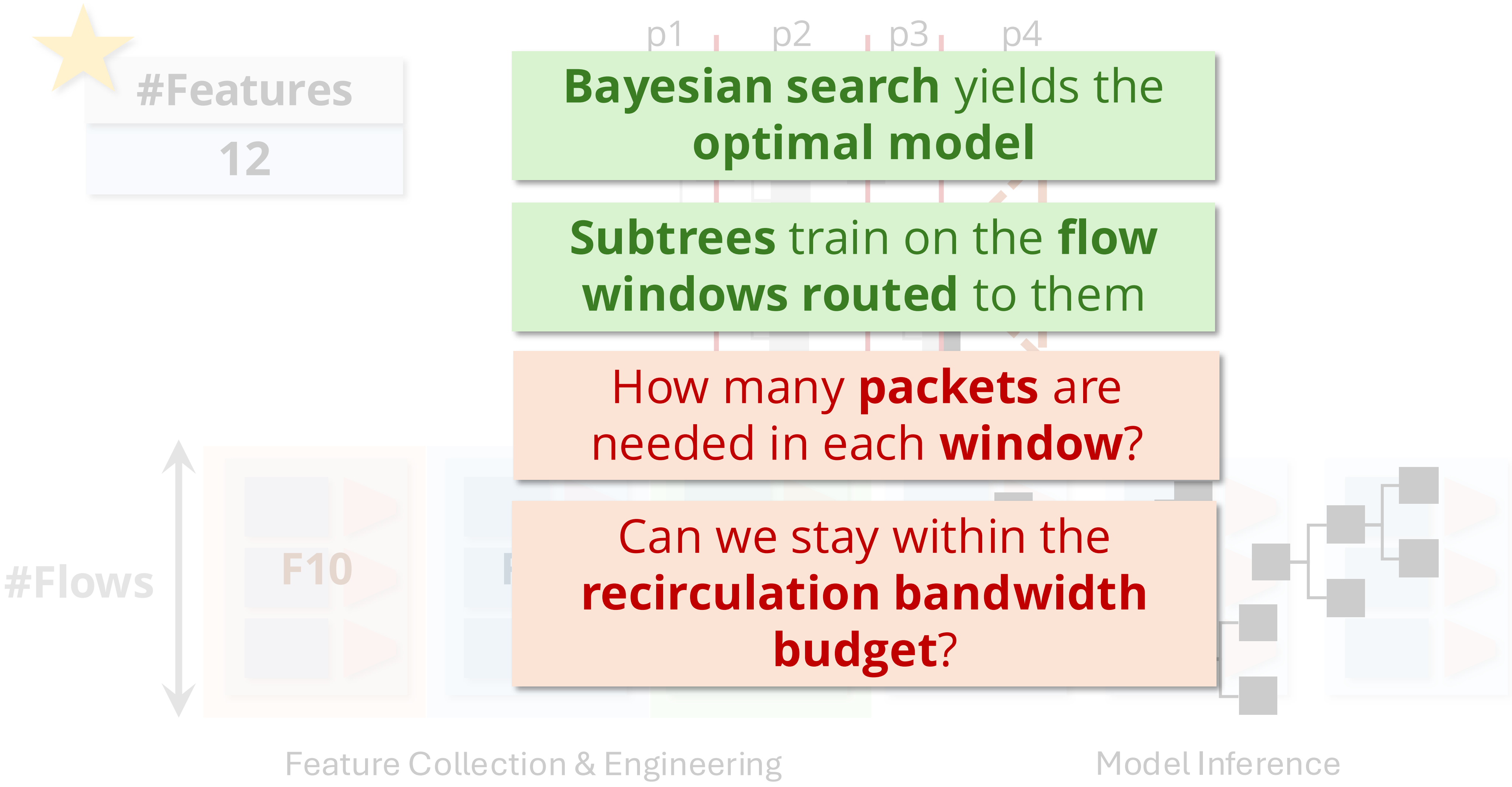
flow 1 w1 w2

flow 2 w1 w2

Training Framework



Design Challenges



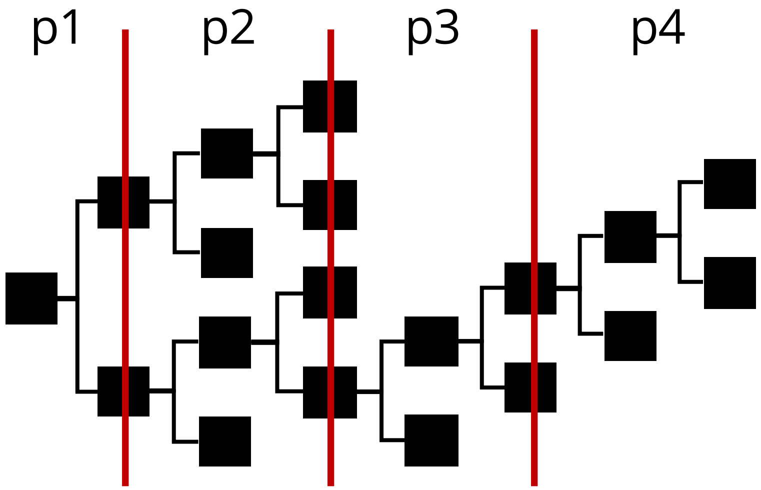
Bayesian search yields the **optimal model**

Subtrees train on the **flow windows routed** to them

How many **packets** are needed in each **window**?

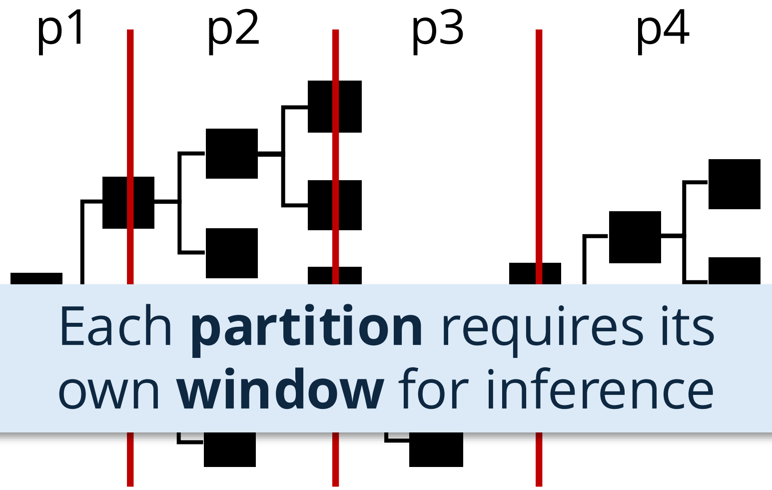
Can we stay within the **recirculation bandwidth budget**?

Windowing Strategy



How many **packets** are needed in each **window**?

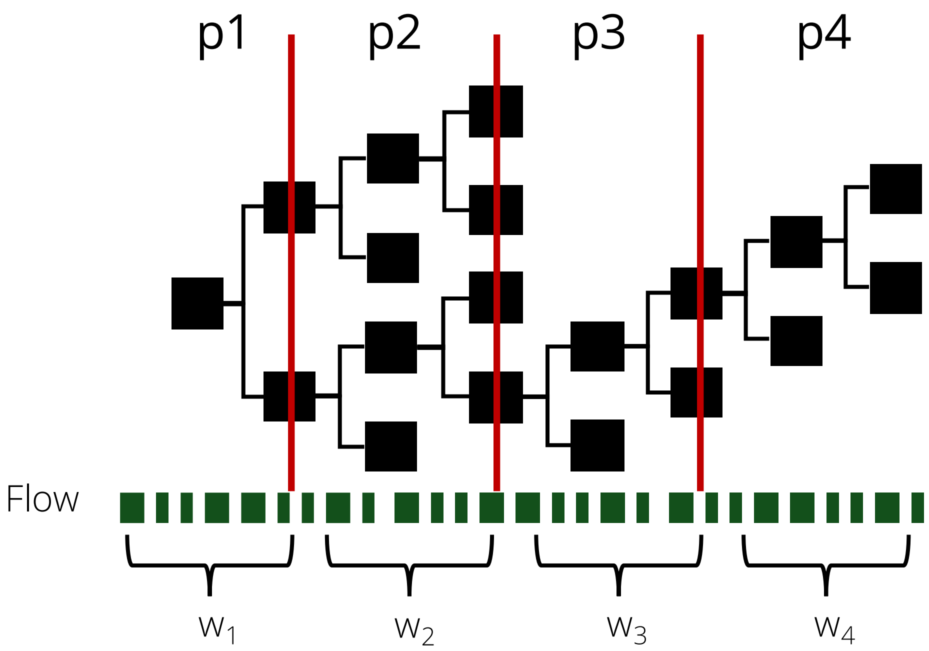
Windowing Strategy



Each **partition** requires its own **window** for inference

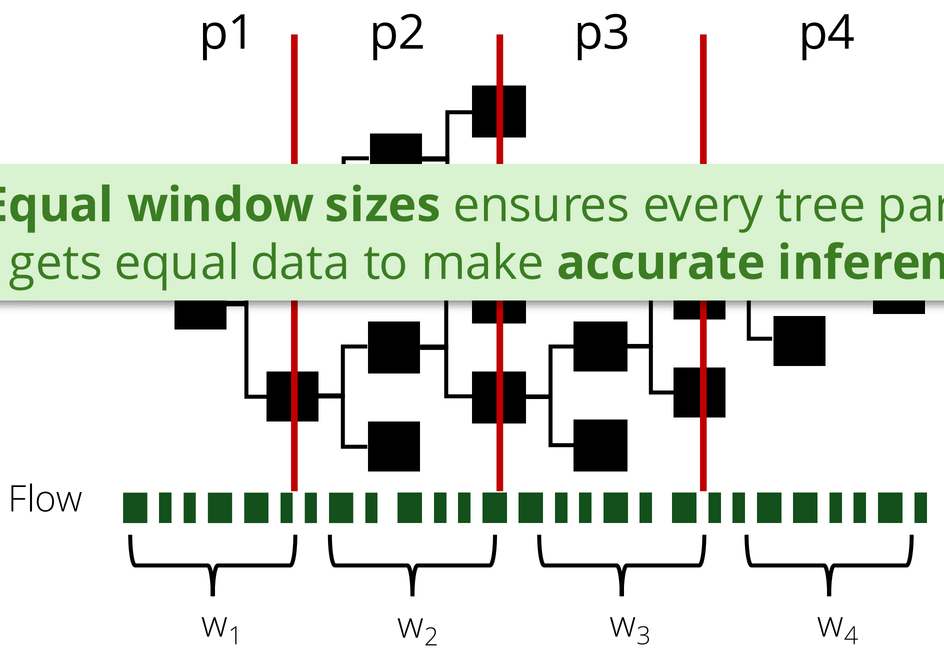
How many **packets** are needed in each **window**?

Windowing Strategy



How many **packets** are needed in each **window**?

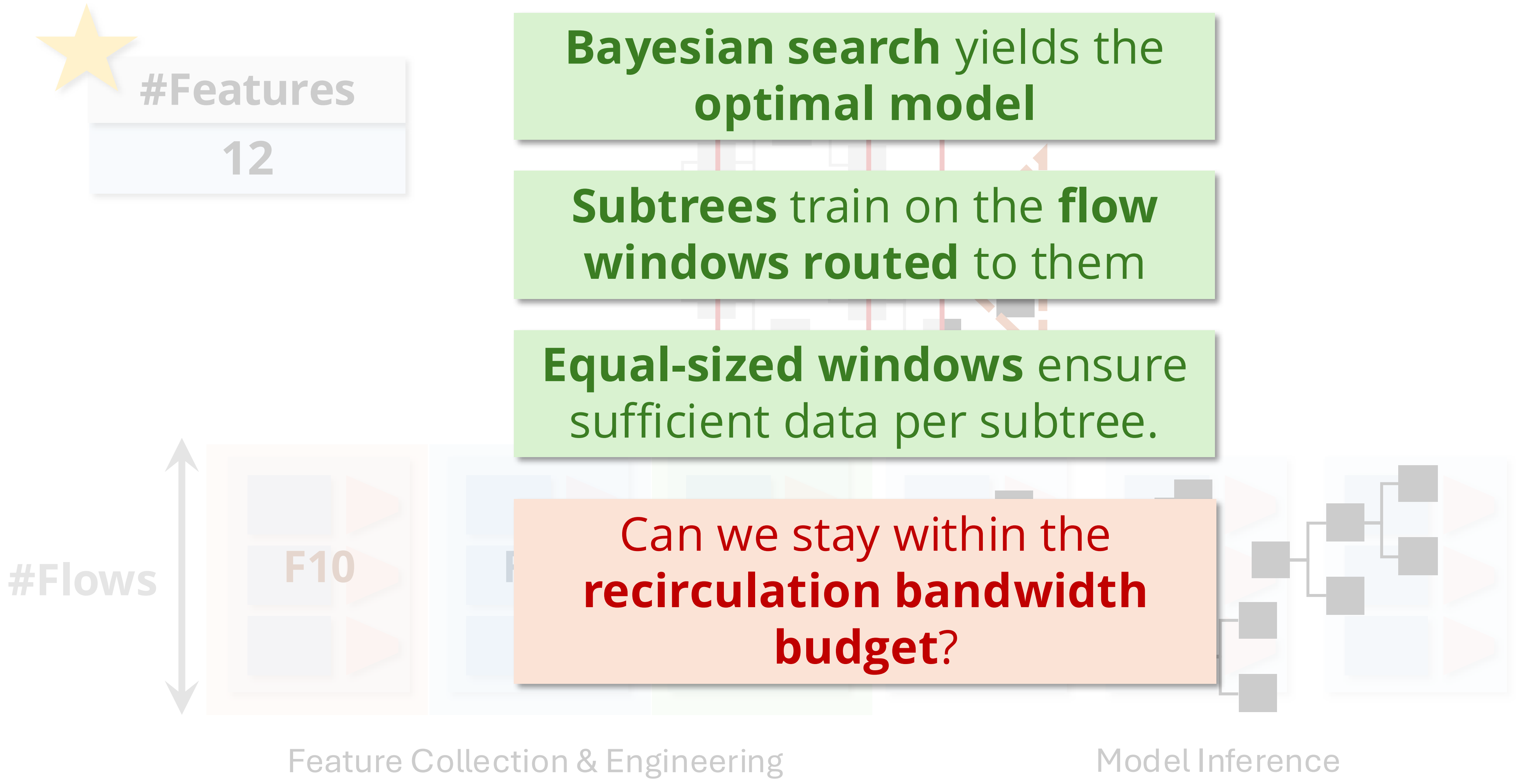
Windowing Strategy



Equal window sizes ensures every tree partition gets equal data to make **accurate inferences!**

How many **packets** are needed in each **window**?

Design Challenges



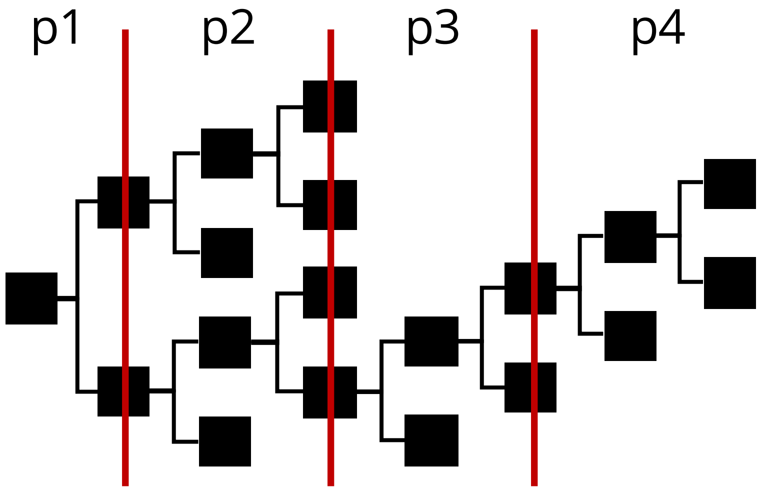
Bayesian search yields the **optimal model**

Subtrees train on the **flow windows routed** to them

Equal-sized windows ensure sufficient data per subtree.

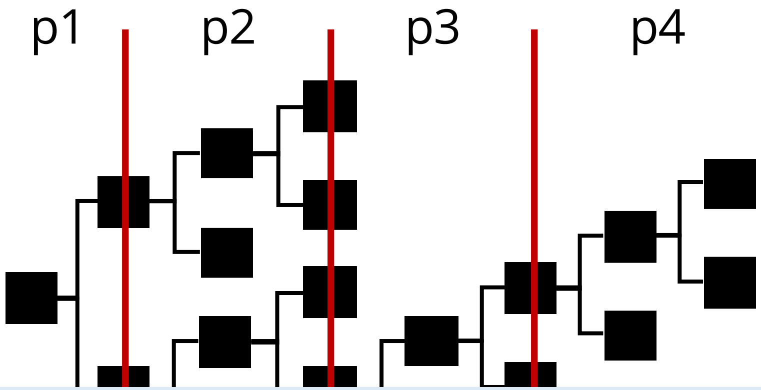
Can we stay within the **recirculation bandwidth budget**?

Recirculation Bandwidth



Can we stay within the **recirculation bandwidth budget**?

Recirculation Bandwidth



Each non-terminal **partition** needs just **one resubmitted packet** per flow!

Can we stay within the **recirculation bandwidth budget**?

Recirculation Bandwidth

Dataset	Recirculation Bandwidth (Mbps)	
	E1: Webserver	**E2: Hadoop**
D1	2.93 ± 2.44	5.99 ± 3.51
D2	6.01 ± 4.01	12.32 ± 5.76
D3	3.58 ± 3.21	7.33 ± 4.62

Maximum recirculation bandwidth (Mbps)

Can we stay within the **recirculation bandwidth budget**?

Recirculation Bandwidth

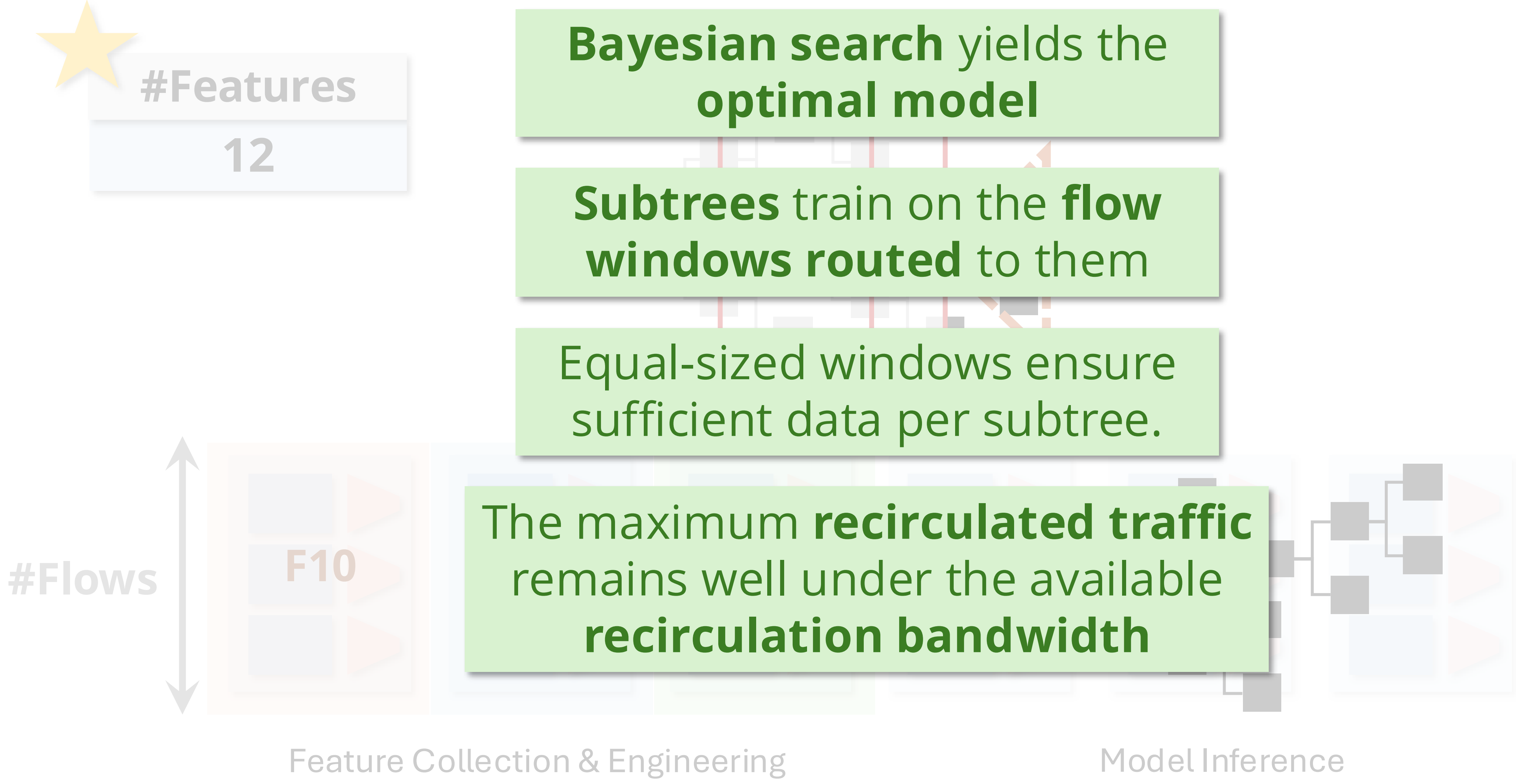
Dataset	Recirculation Bandwidth (Mbps)	
	E1: Webserver	**E2: Hadoop**
D1	2.93 ± 2.44	5.99 ± 3.51
D2	6.01 ± 4.01	12.32 ± 5.76
D3	3.58 ± 3.21	7.33 ± 4.62

Maximum recirculation bandwidth (Mbps)

SpliDT uses at most **~0.012%** of the available **100 Gbps pipeline bandwidth**

Can we stay within the **recirculation bandwidth budget**?

Design Challenges



Bayesian search yields the **optimal model**

Subtrees train on the **flow windows routed** to them

Equal-sized windows ensure sufficient data per subtree.

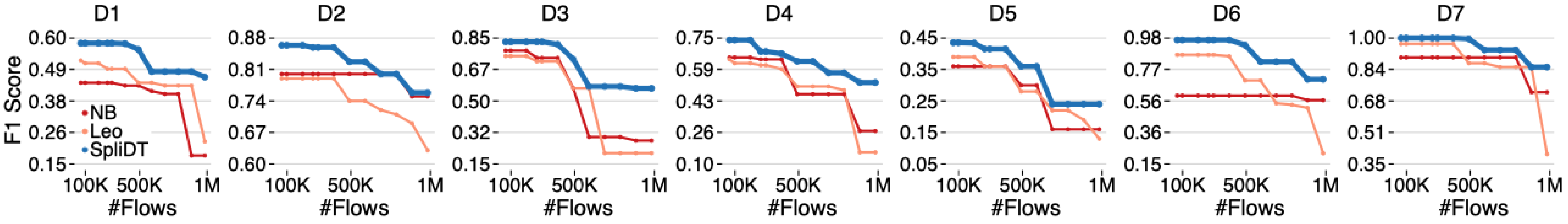
The maximum **recirculated traffic** remains well under the available **recirculation bandwidth**

Datasets

Dataset	Description	#Class
D1: CIC-IoMT2024	IoMT traffic dataset for healthcare intrusion detection.	19
D2: CIC-IoT2023-a	A simplified CIC-IoT-2023 dataset.	4
D3: ISCX-VPN2016	A dataset of VPN and non-VPN traffic for VPN detection.	13
D4: Campus Traffic	UCSB dataset with web, cloud, social, and streaming traffic.	11
D5: CIC-IoT2023-b	A comprehensive CIC-IoT-2023.	32
D6: CIC-IDS2017	A network intrusion detection dataset.	10
D7: CIC-IDS2018	A network intrusion detection dataset.	10

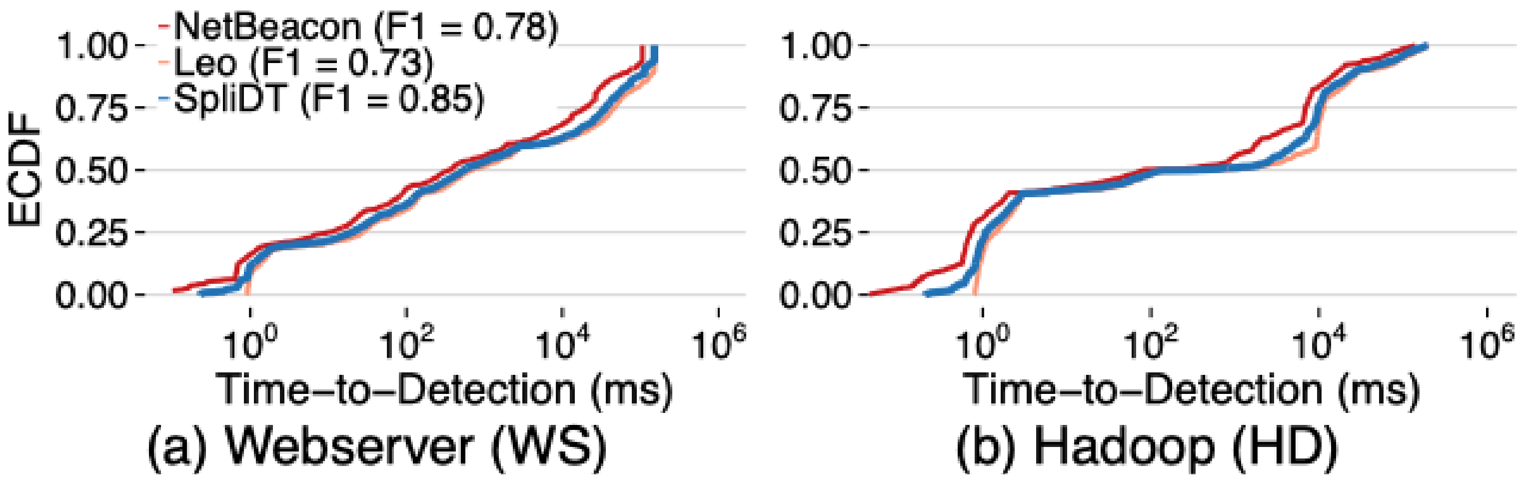
Real-world network traffic datasets used for evaluating SPLIDT across diverse security scenarios.

SpliDT Accuracy–Flows Pareto Frontier



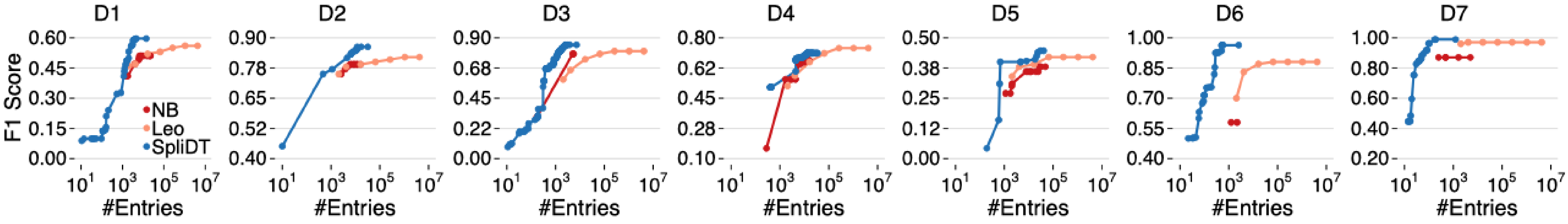
SpliDT achieves a superior **Pareto frontier** of **accuracy (F1 score)** versus **#Flows** compared to baselines.

Flow-level Latency



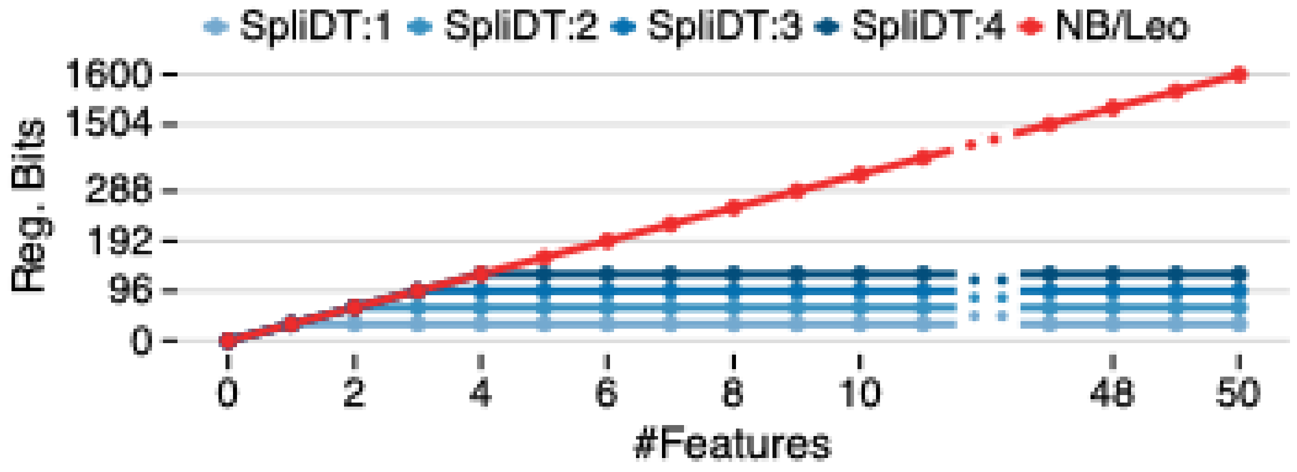
Time-to-detection (TTD) of D3 for environments (E1–E2).
(Other datasets show a similar trend.)

TCAM Entries vs. F1 Score



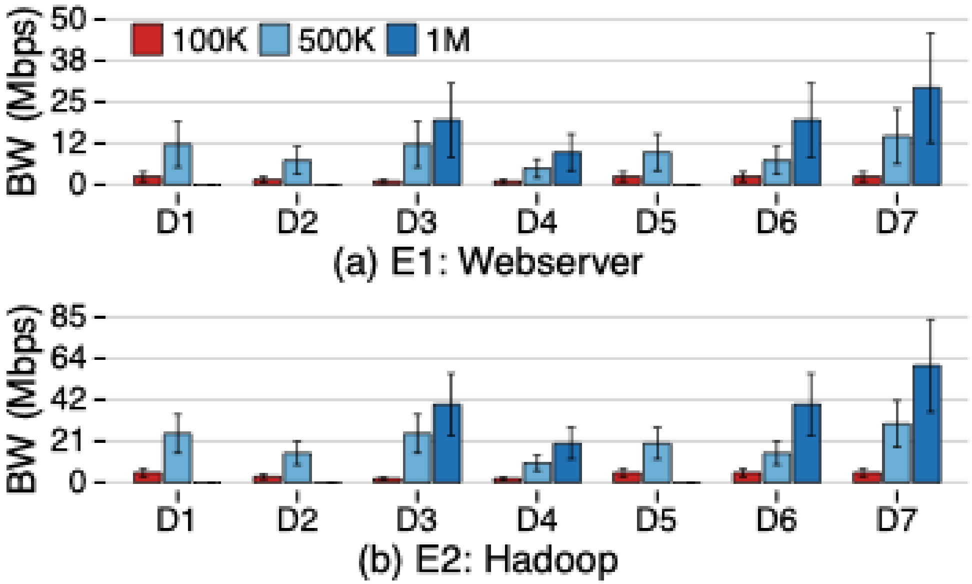
Comparison of **#TCAM entries** against **F1 score** for SpliDT versus baselines.

Feature Scalability vs. Register Size



Register sizes (in bits) versus **number of features** supported by each model. SpliDT:k is a partitioned tree with k features per subtree.

SpliDT Recirculation Bandwidth Across Workloads



Maximum recirculation bandwidth (Mbps) of SpliDT partitioned trees when processing datasets (D1–7) for the two datacenter environments, **E1: Webserver** and **E2: Hadoop**, with a varying number of flows.
A model with a single partition does not recirculate packets (0 Mbps).

SpliDT Performance Summary

- **5× More Stateful Features** than SOTA
- Our architecture remains **scalable to million of flows**
- **Low recirculation overhead** (<50 Mbps)
- Preserves **flow-level time-to-detection**

Thank you!

Contact us: parvezm@purdue.edu

Website: https://splidt-decision-trees.github.io

NextGArch Lab: https://nextgarch.github.io/