Continuous Deployment of Machine Learning Pipelines

Behrouz Derakhshan behrouz.derakhshan@dfki.de
Alireza Rezaei Mahdiraji alireza.rm@dfki.de
Tilmann Rabl rabl@tu-berlin.de
Volker Markl volker.markl@tu-berlin.de

Database Systems and Information Management Group
TU Berlin

https://www.dima.tu-berlin.de/

Intelligent Analytics for Massive Data
German Research Center for Artificial Intelligence

https://www.dfki.de/

- Life cycle of ML application does not end with training

- Life cycle of ML application does not end with training
- Models and Pipelines must be deployed to answer prediction queries

Life Cycle of Machine Learning Applications

- Life cycle of ML application does not end with training

- Models and Pipelines must be deployed to answer prediction queries

- Deployed models and pipelines should be monitored and trained further

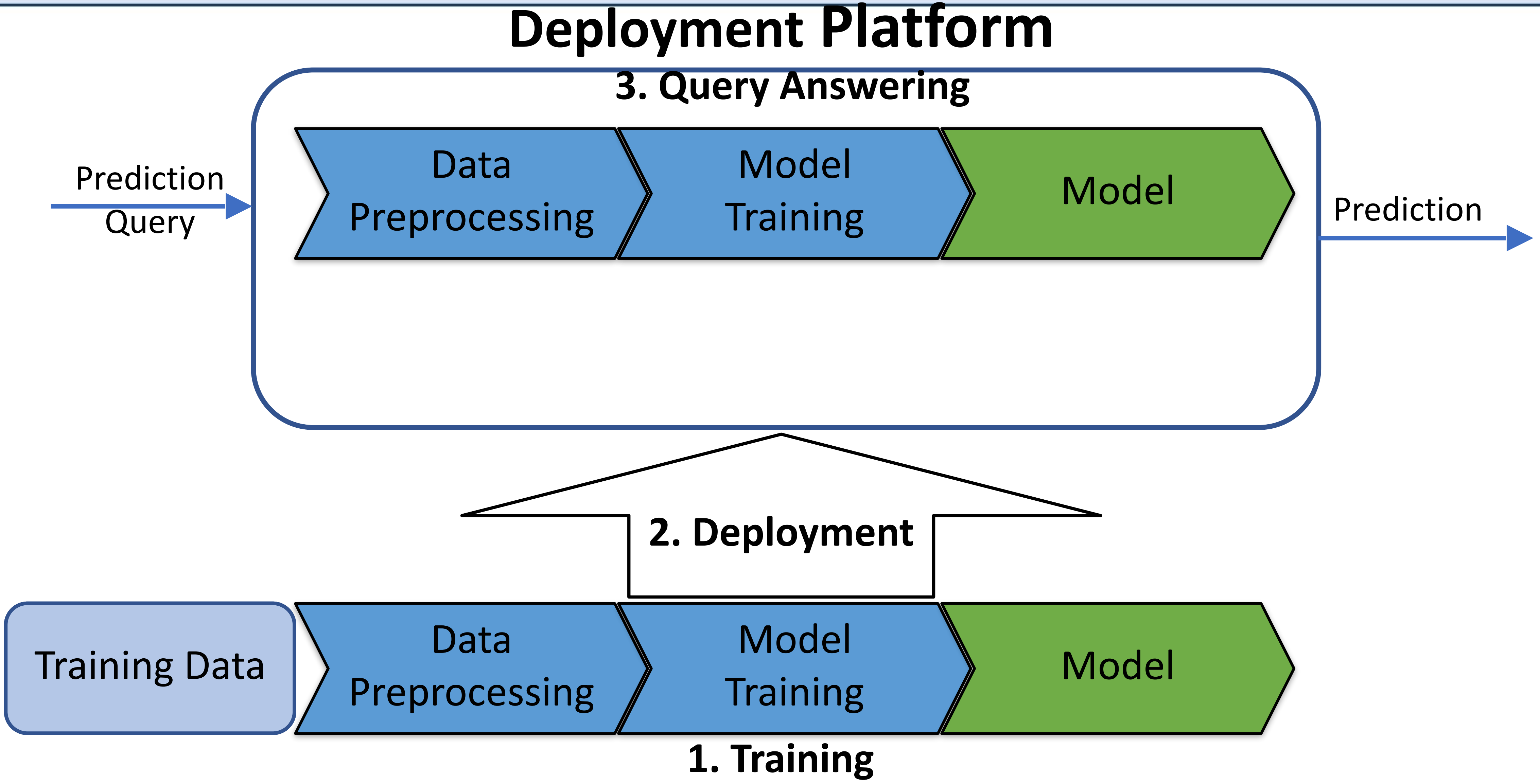
Life Cycle of Machine Learning Applications

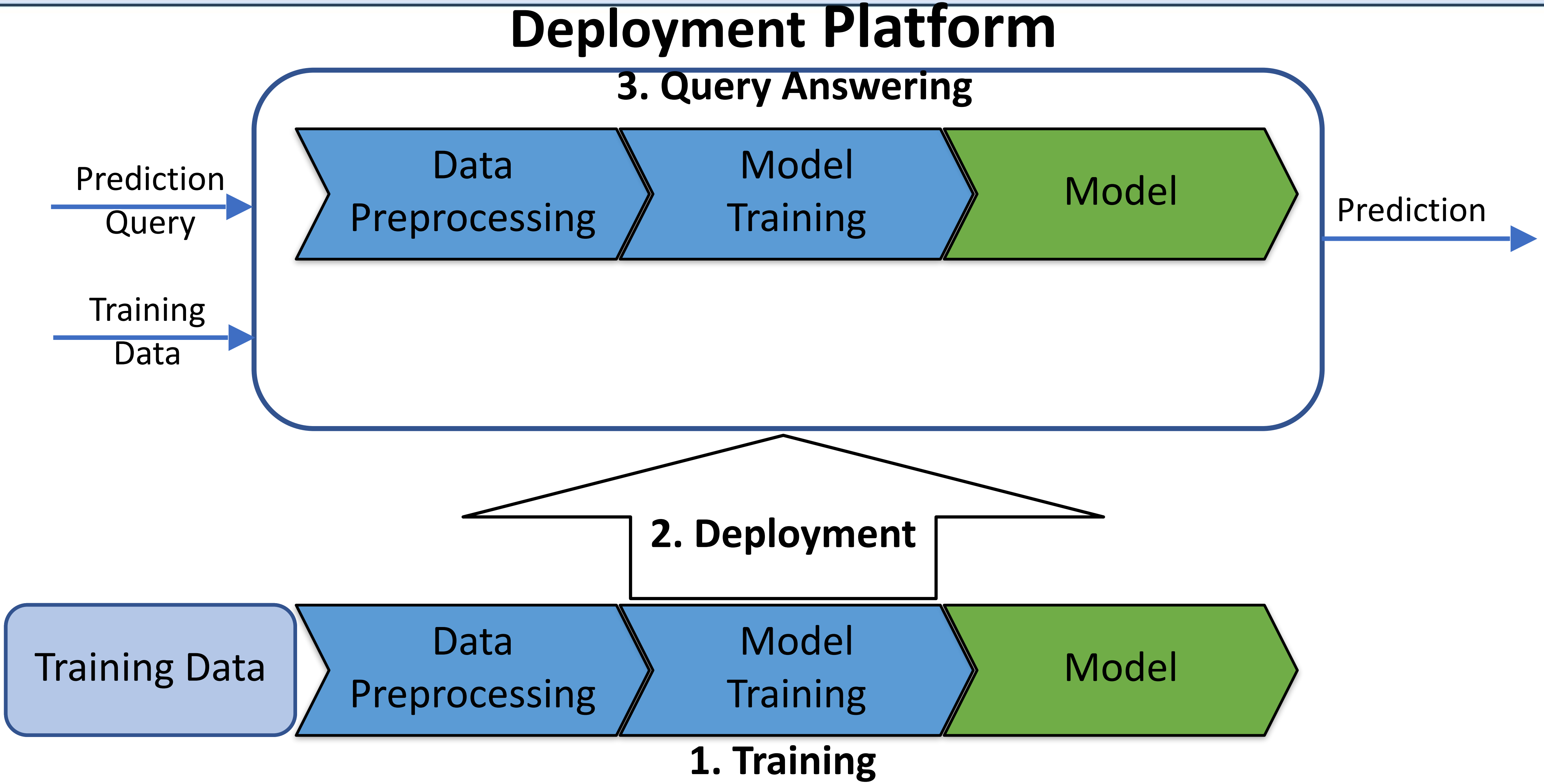
- Life cycle of ML application does not end with training

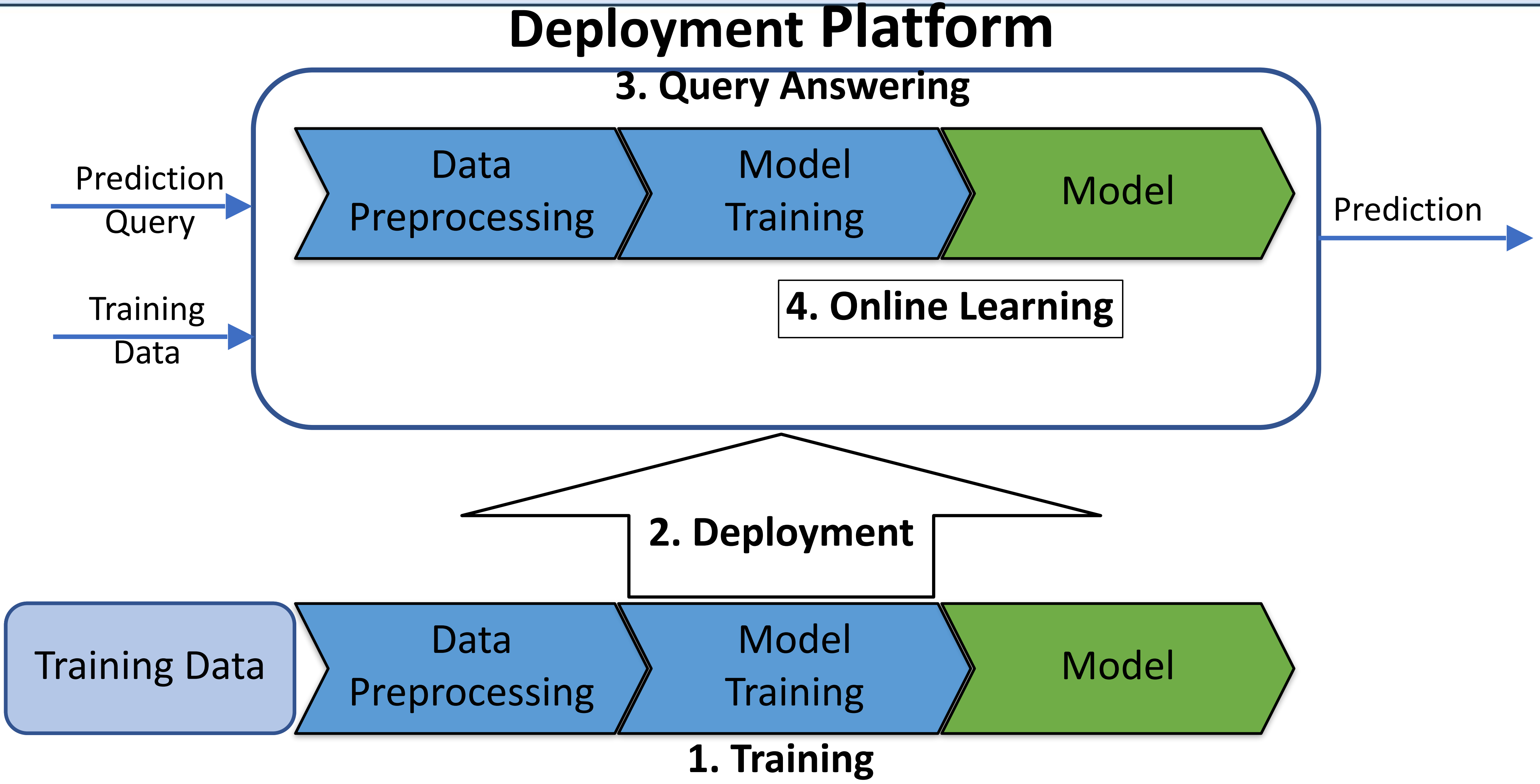
- Models and Pipelines must be deployed to answer prediction queries

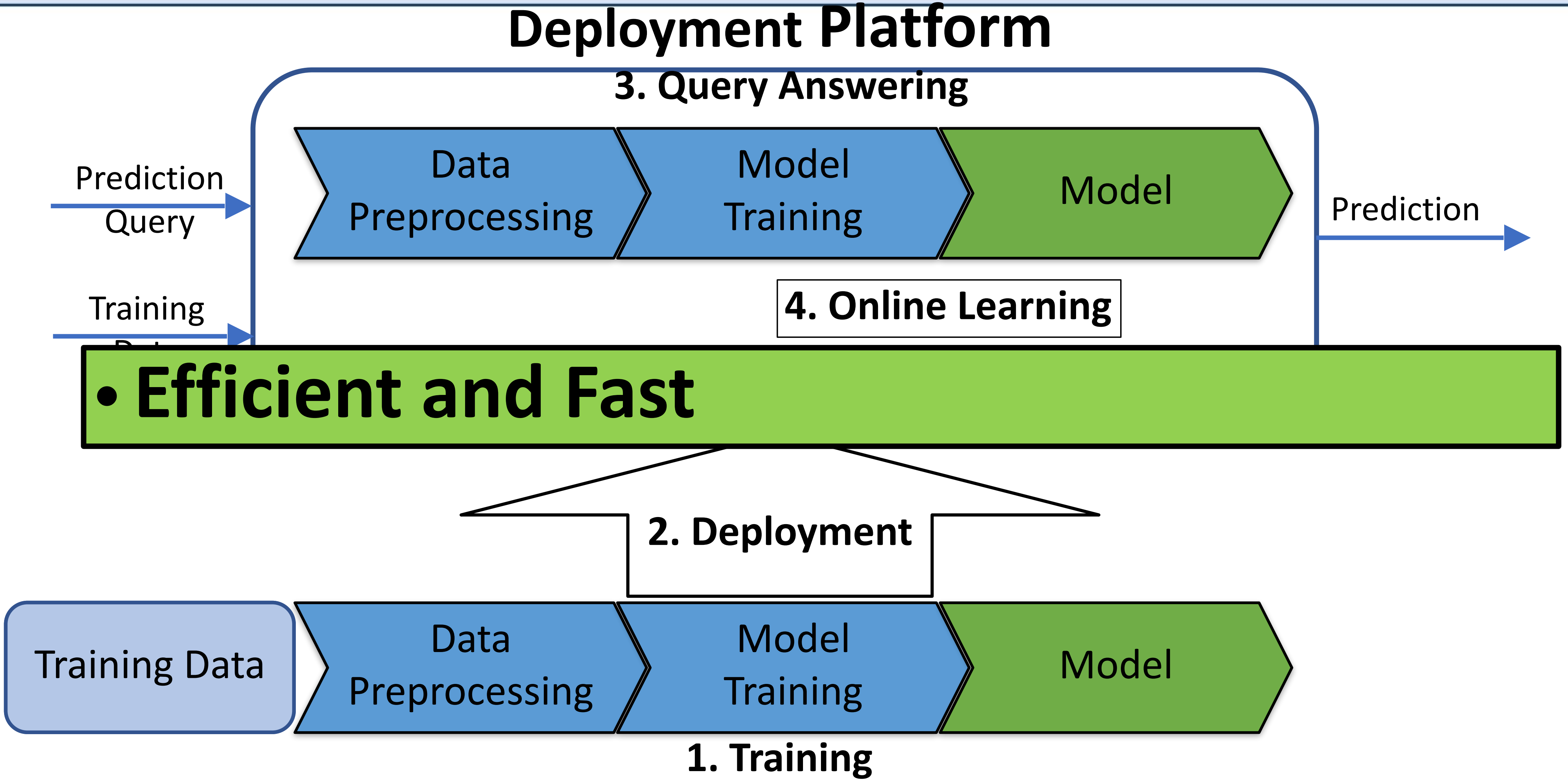
Focus of this talk

- Deployed models and pipelines should be monitored and trained further

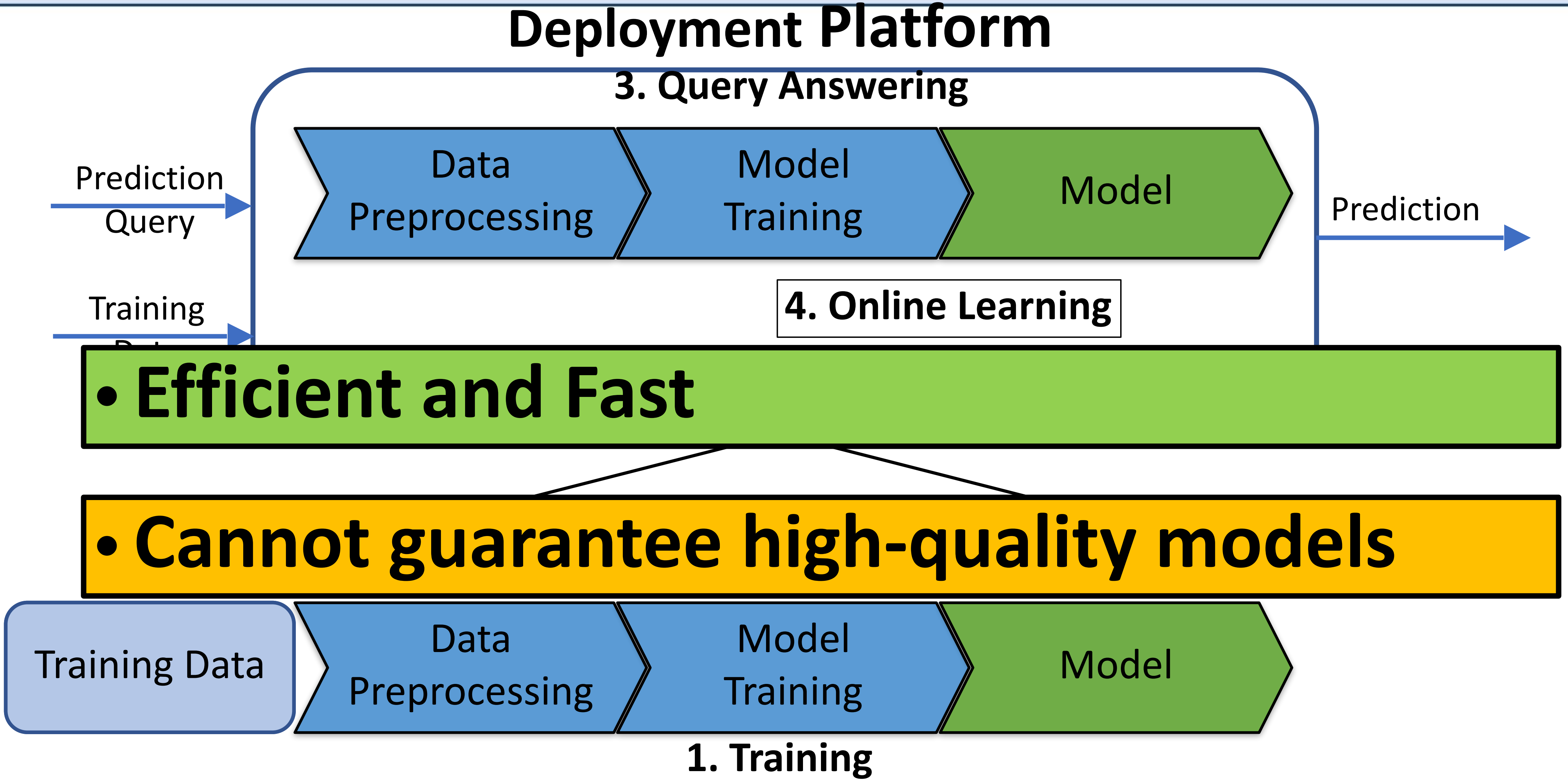

Deployment Platform
3. Query Answering
Prediction
Query
Data
Preprocessing
Model
Training
Model
Prediction
2. Deployment
Training Data
Data
Preprocessing
Model
Training
Model
1. Training

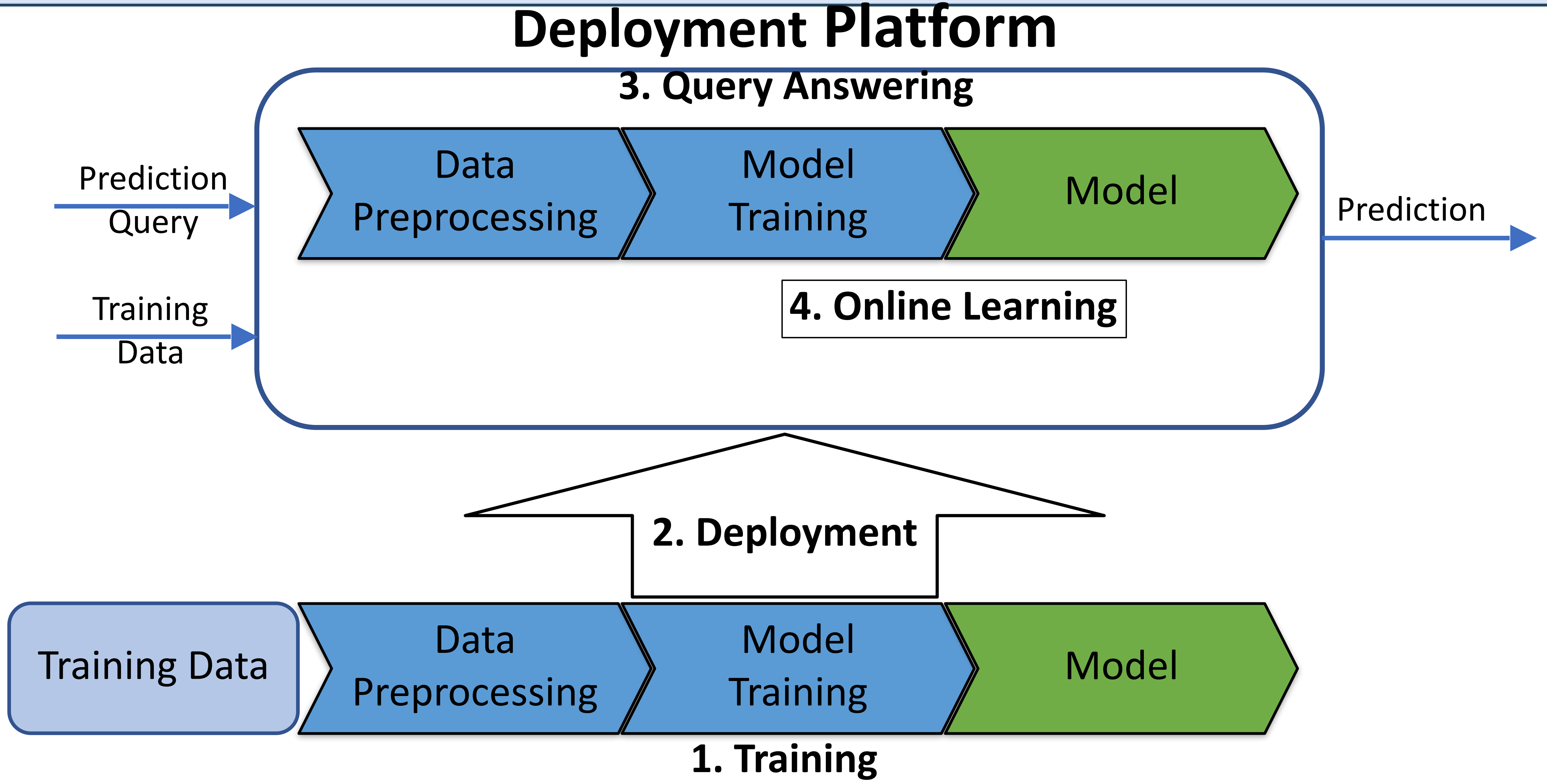

Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
2. Deployment
Training Data
Data Preprocessing
Model Training
Model
1. Training

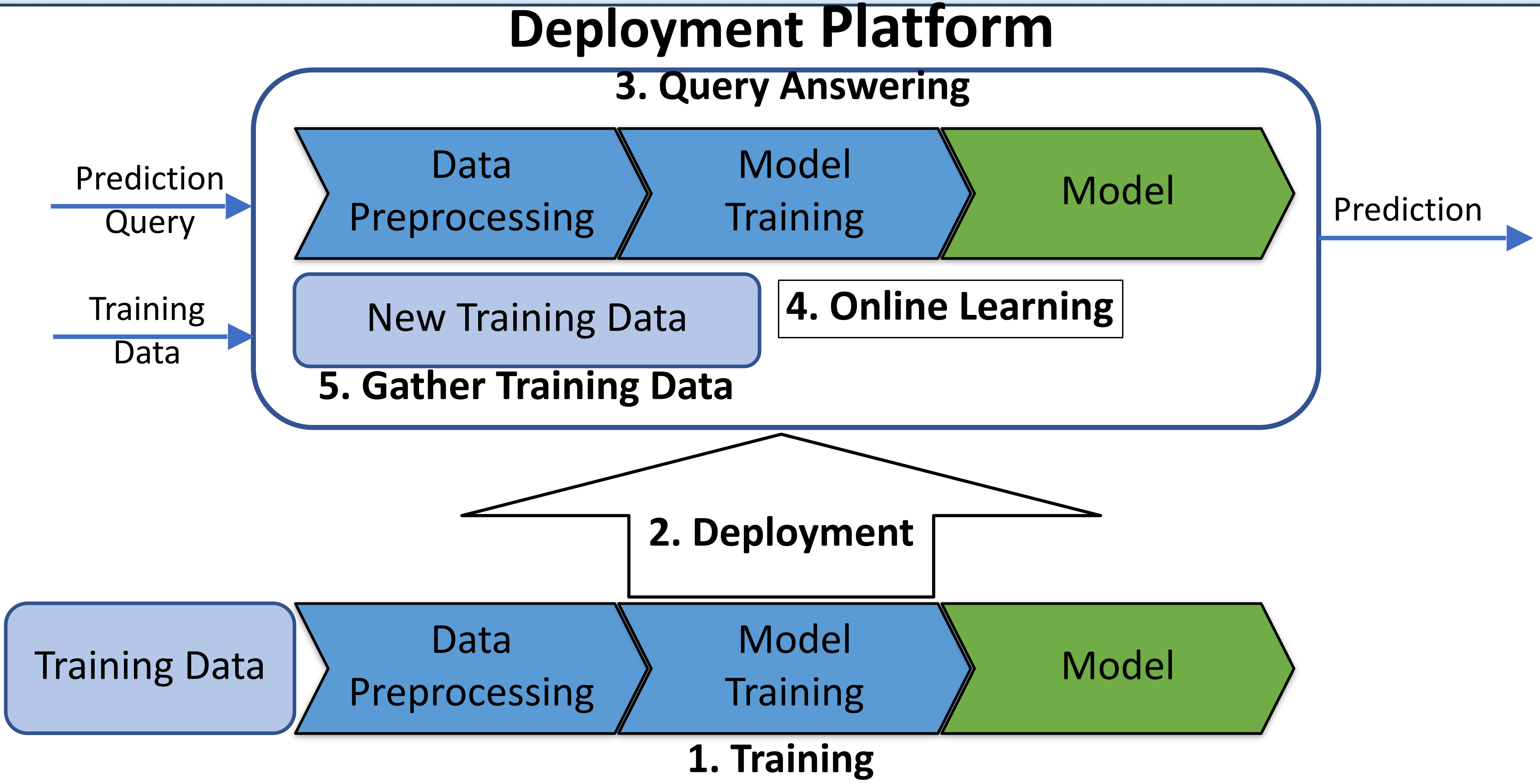

Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
4. Online Learning
2. Deployment
Training Data
Data Preprocessing
Model Training
Model
1. Training

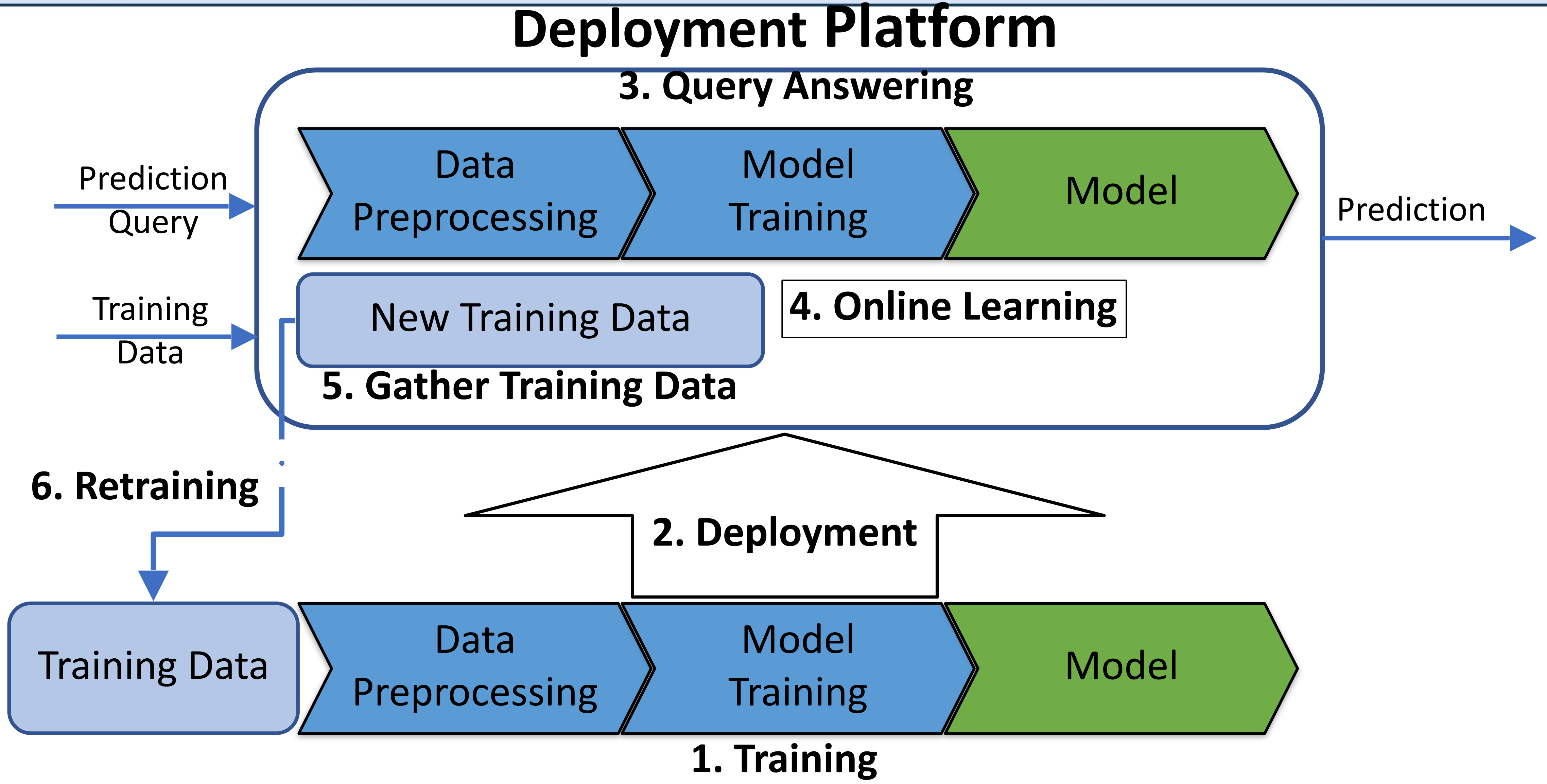


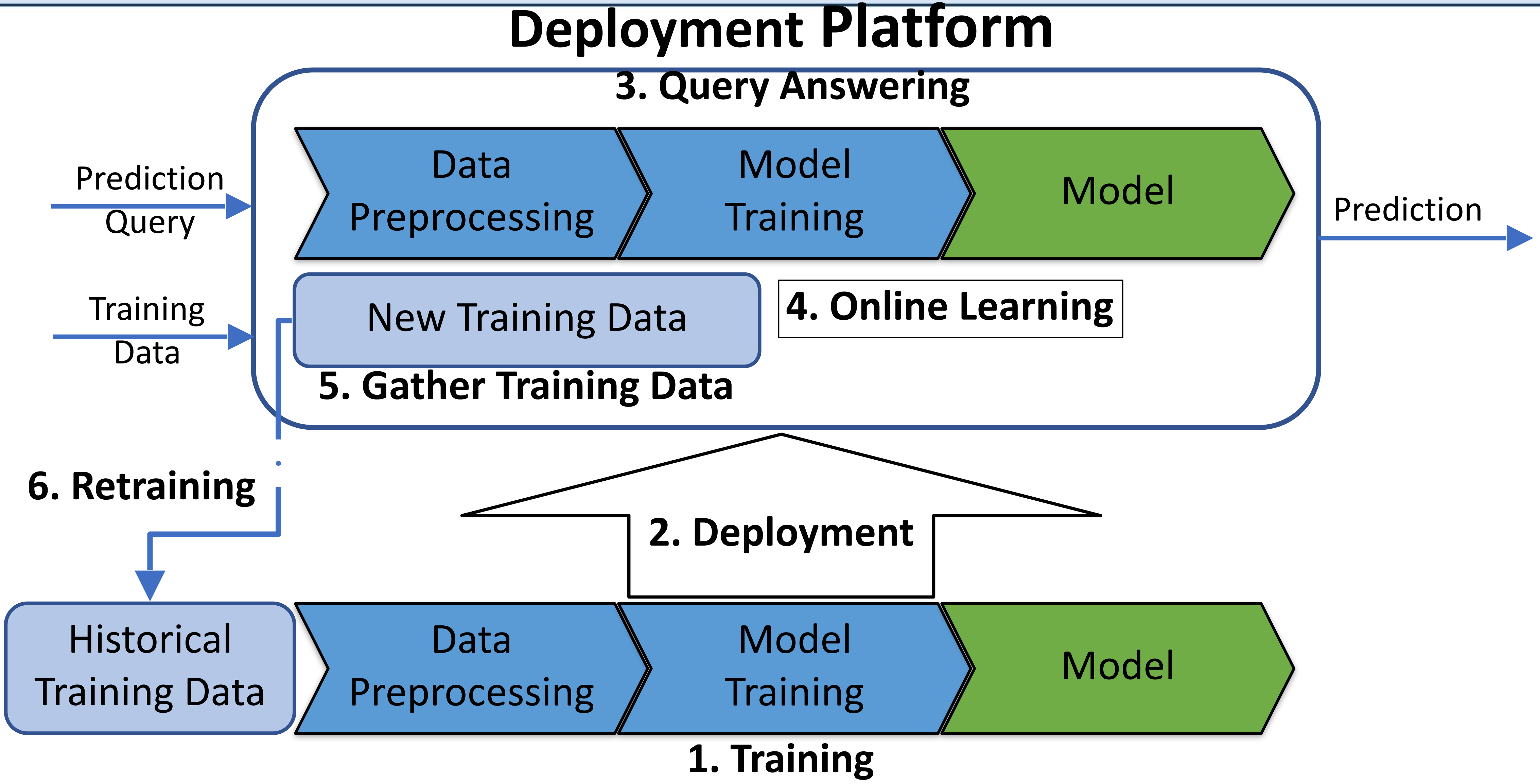
- **Efficient and Fast**

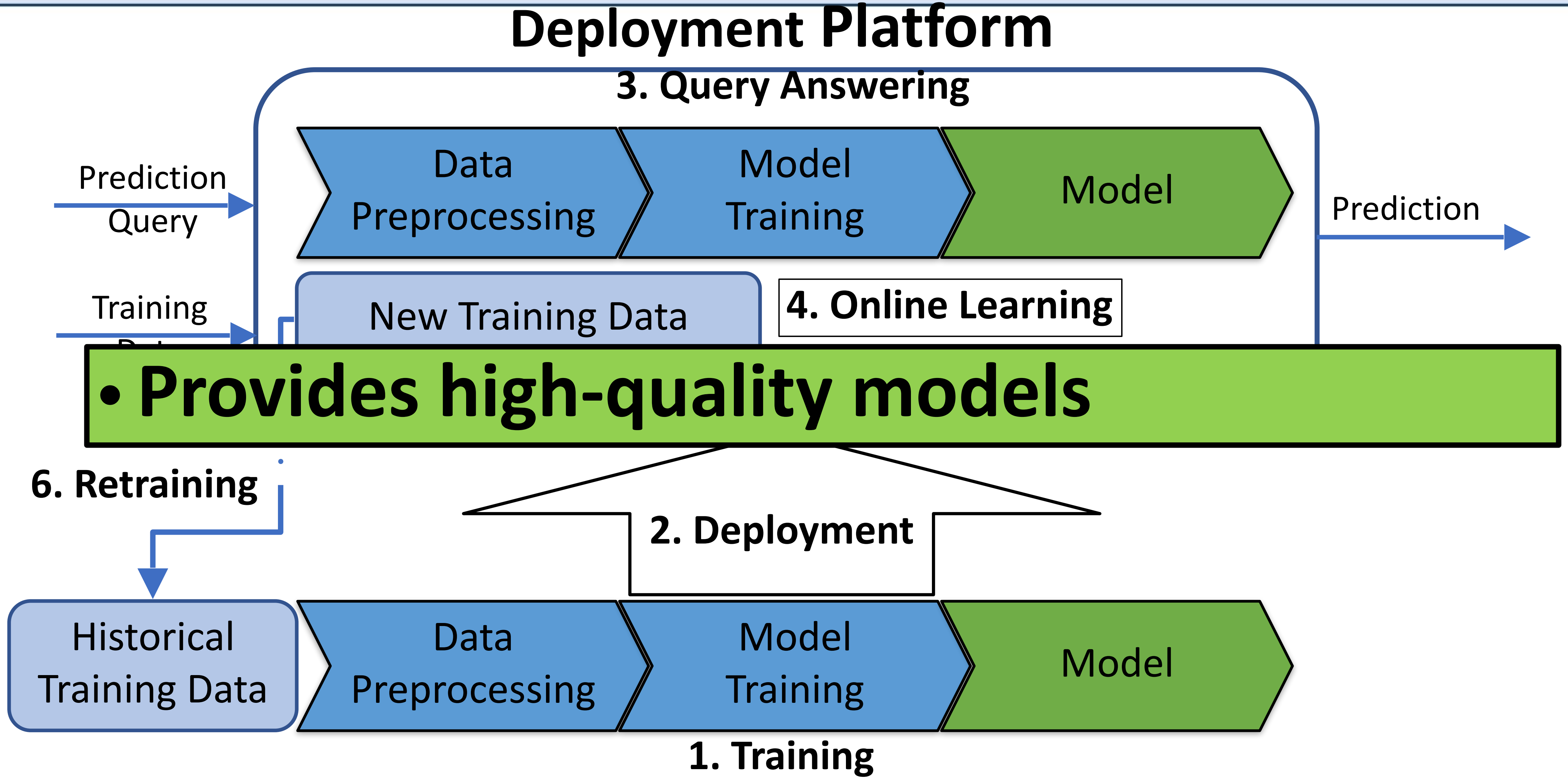

Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
4. Online Learning
• Efficient and Fast
• Cannot guarantee high-quality models
Training Data
Data Preprocessing
Model Training
Model
1. Training


Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
4. Online Learning
2. Deployment
Training Data
Data Preprocessing
Model Training
Model
1. Training


Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
New Training Data
4. Online Learning
5. Gather Training Data
2. Deployment
Training Data
Data Preprocessing
Model Training
Model
1. Training

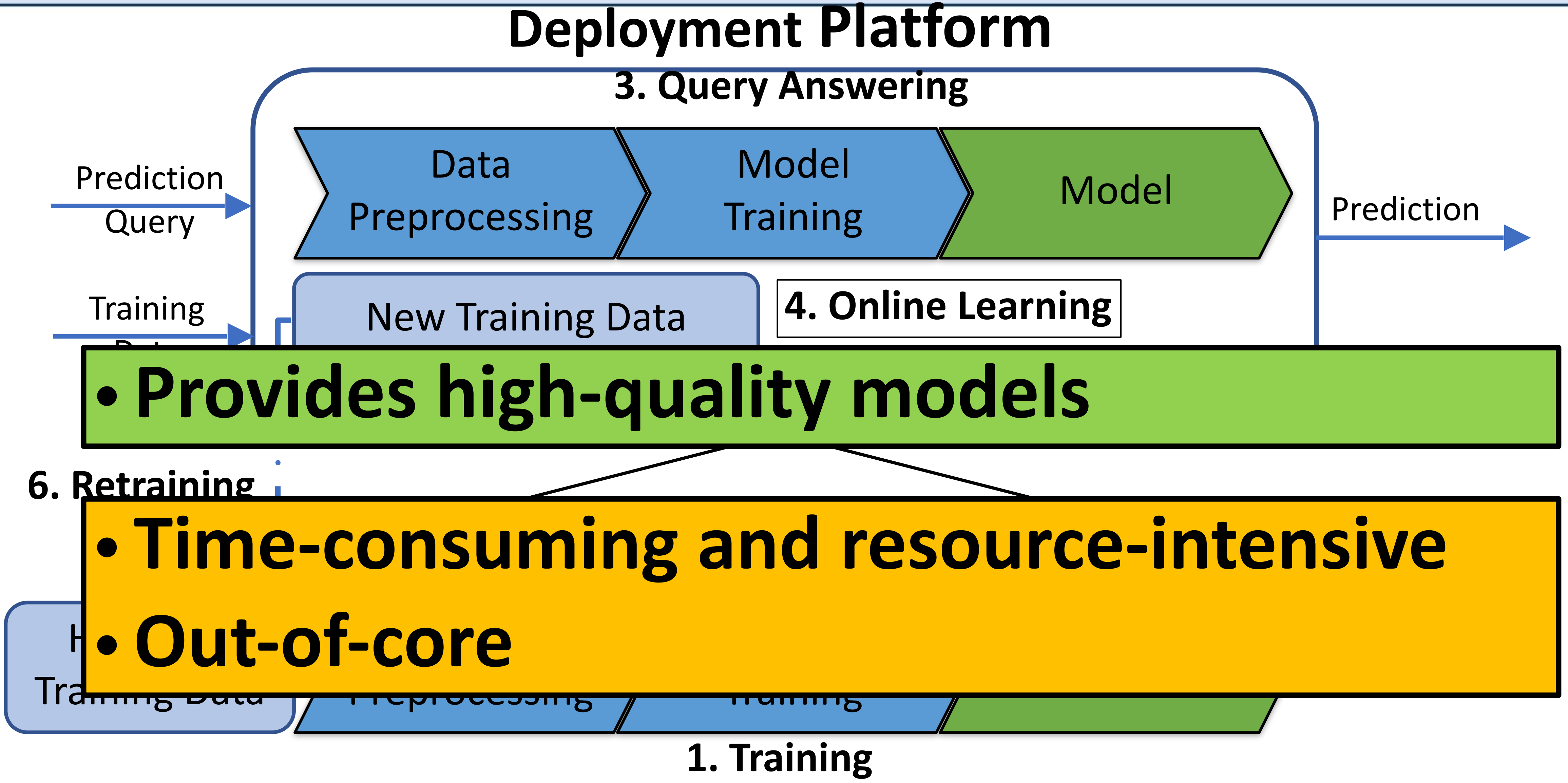

Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
New Training Data
4. Online Learning
5. Gather Training Data
6. Retraining
2. Deployment
Training Data
Data Preprocessing
Model Training
Model
1. Training


Deployment Platform
3. Query Answering
Prediction Query
Training Data
Data Preprocessing
Model Training
Model
Prediction
New Training Data
4. Online Learning
5. Gather Training Data
6. Retraining
2. Deployment
Historical Training Data
Data Preprocessing
Model Training
Model
1. Training



- **Provides high-quality models**

Improvement By Retraining



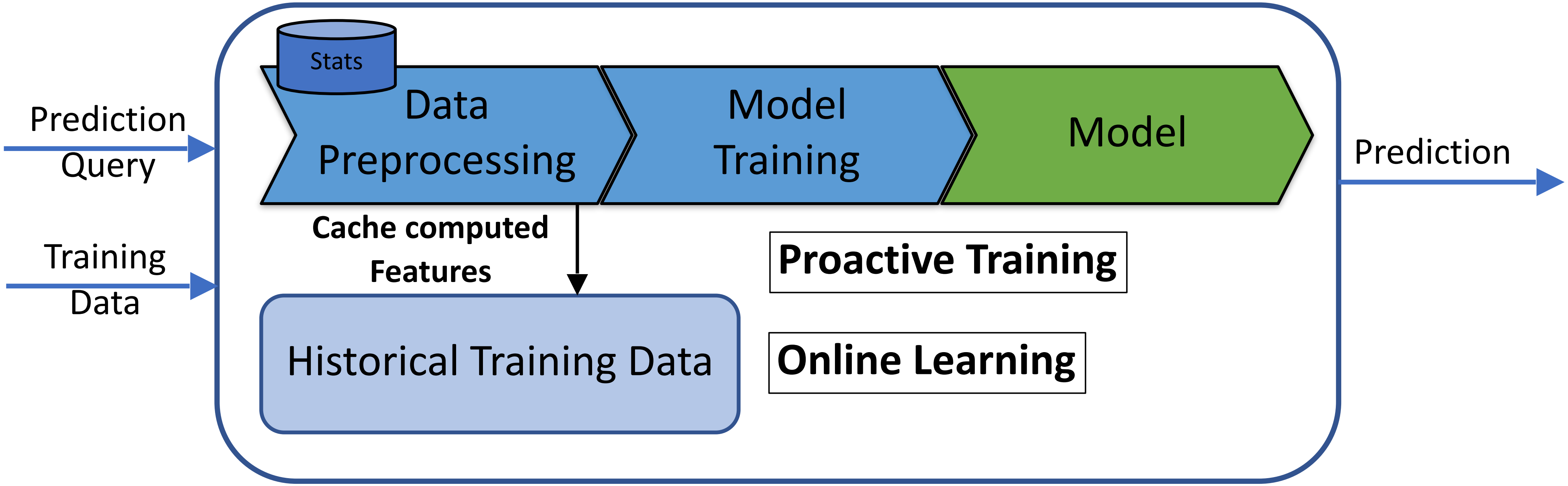
- **Provides high-quality models**

- **Time-consuming and resource-intensive**
- **Out-of-core**

Ideal Deployment Platform

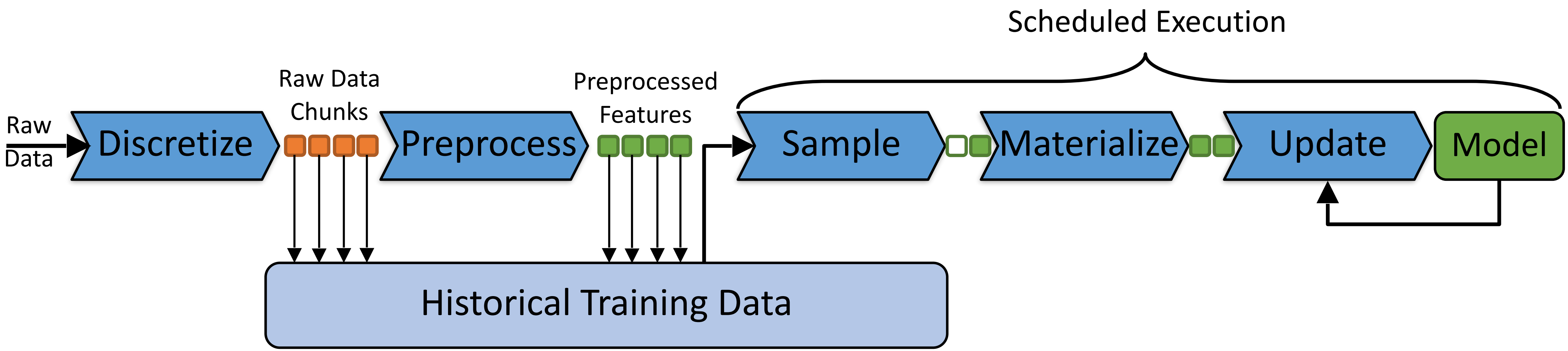
Can a platform provide the same level of **quality** as Retraining and perform (almost) as **efficiently** as Online Learning?

Continuous Deployment Platform

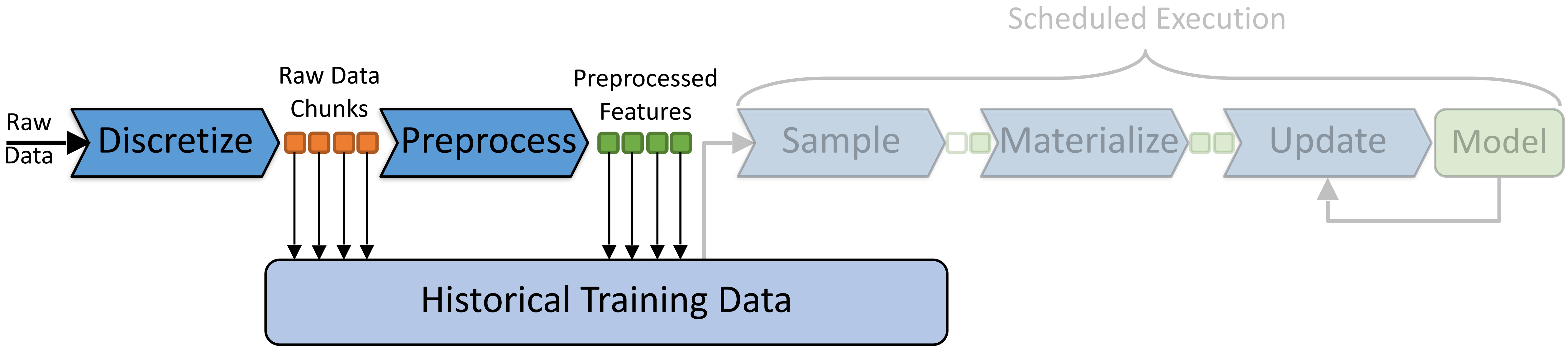


- Train the model inside the platform
- **Compute** features and cache them
- **Update** data preprocessing statistics
- Replace Retraining with **Proactive Training**

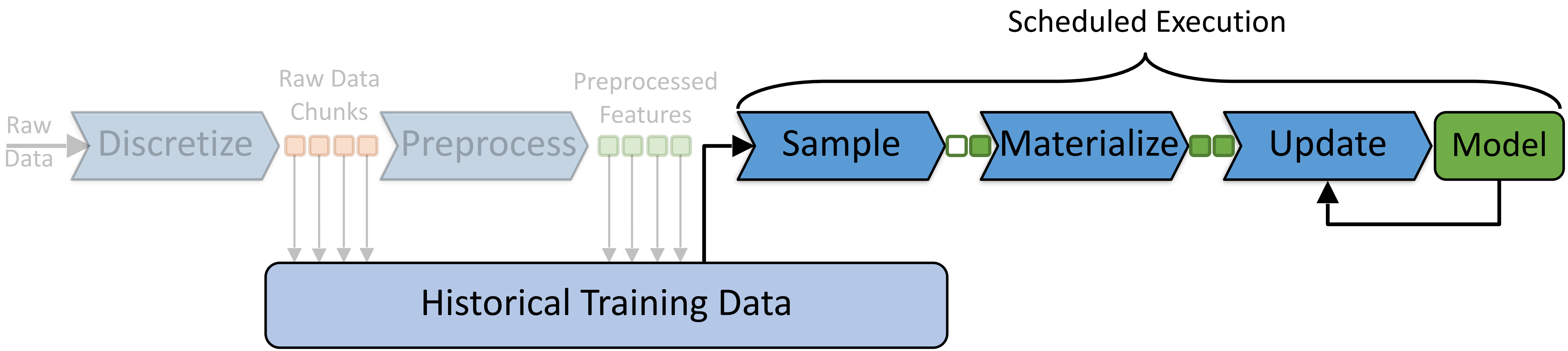
Continuous Deployment Platform



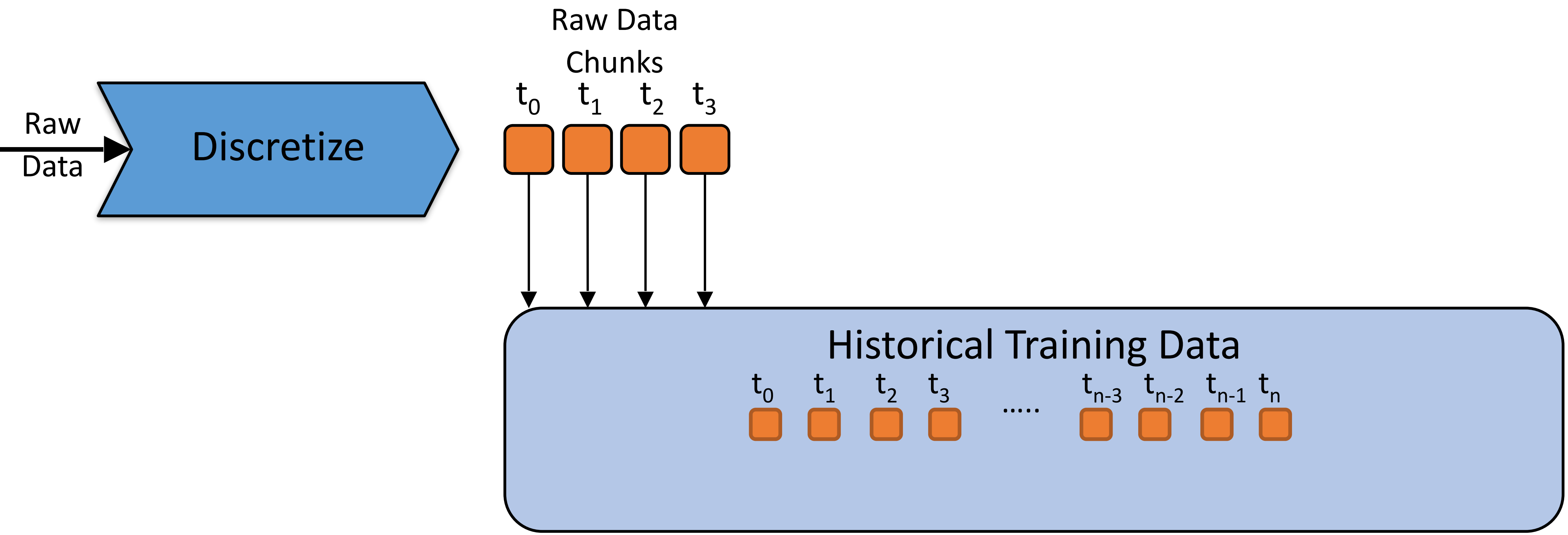
Data Preparation Phase



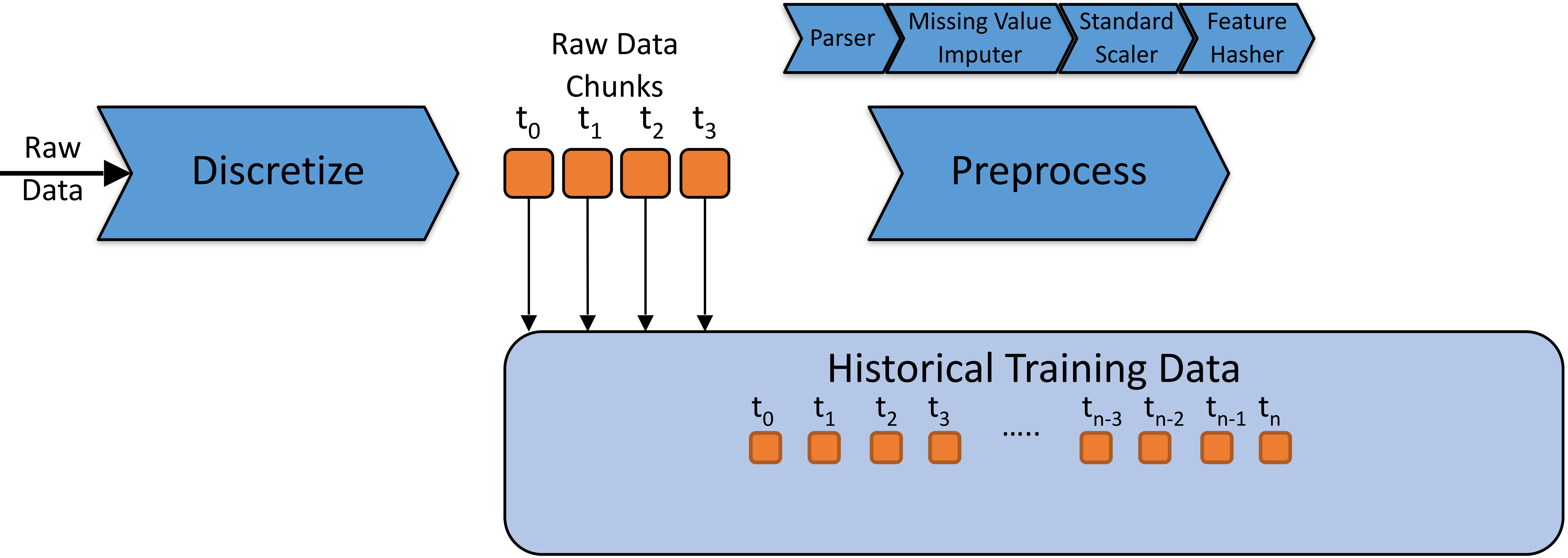
Proactive Training Phase

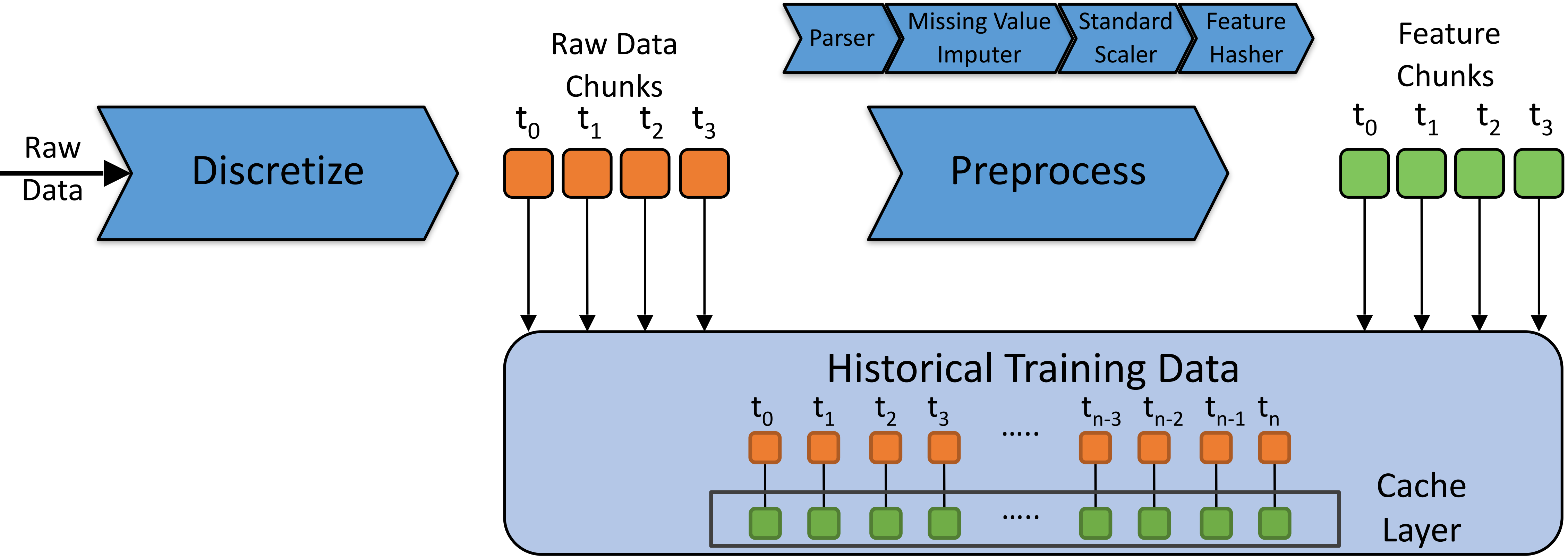


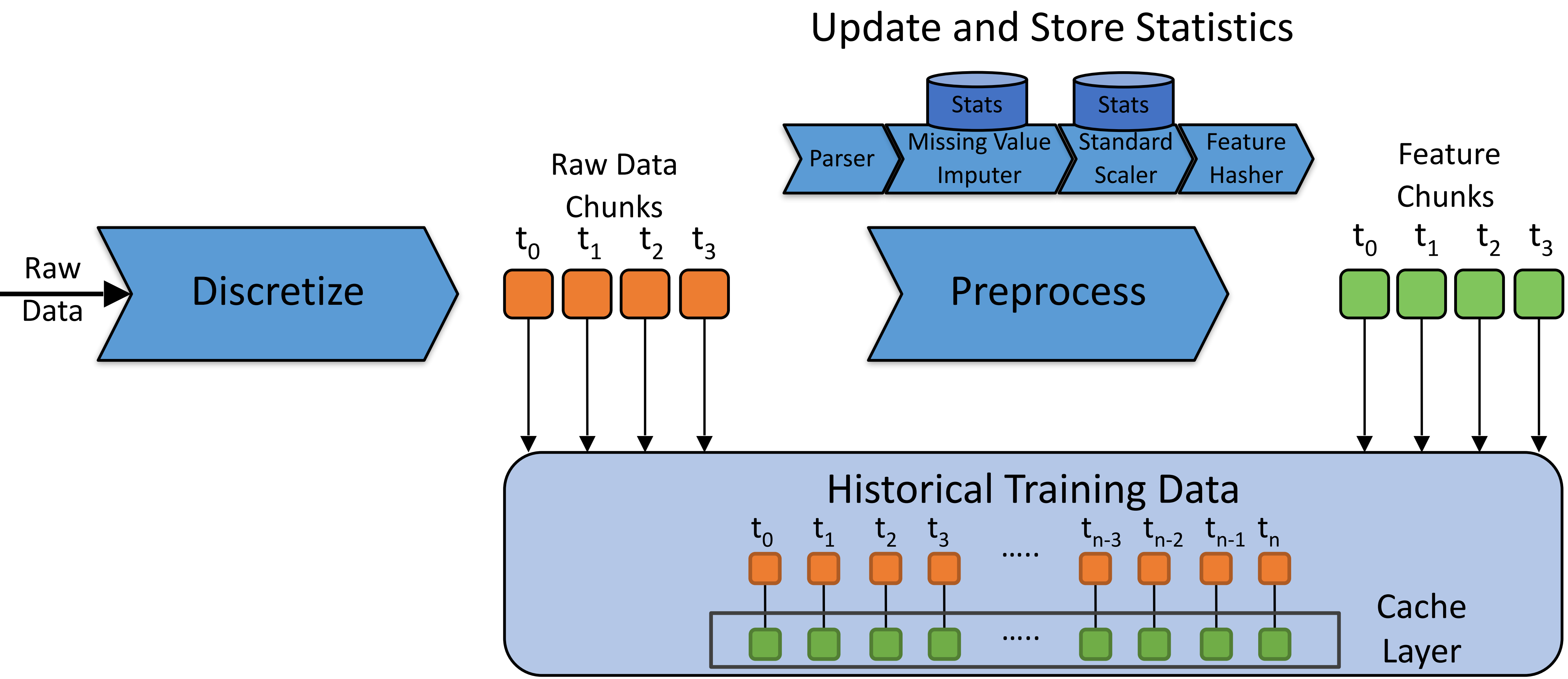
Data Preparation Phase



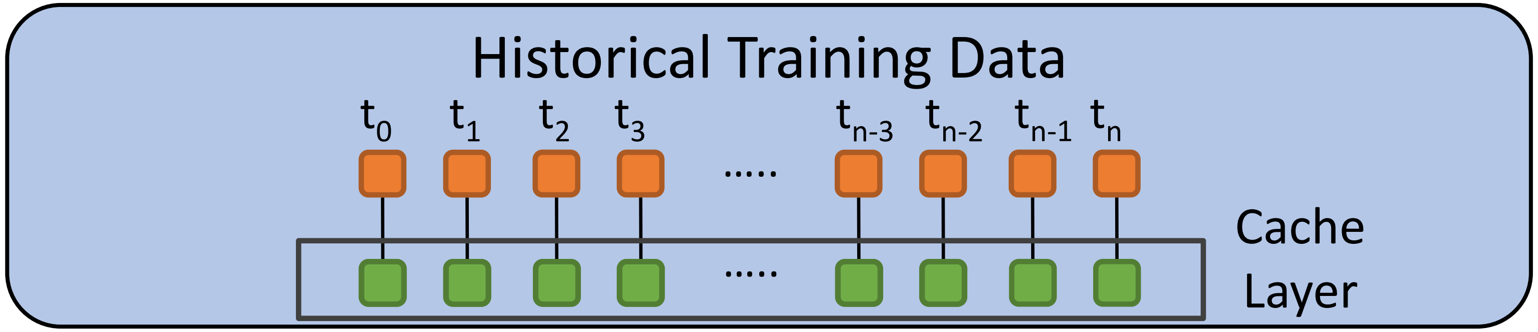
Data Preparation Phase



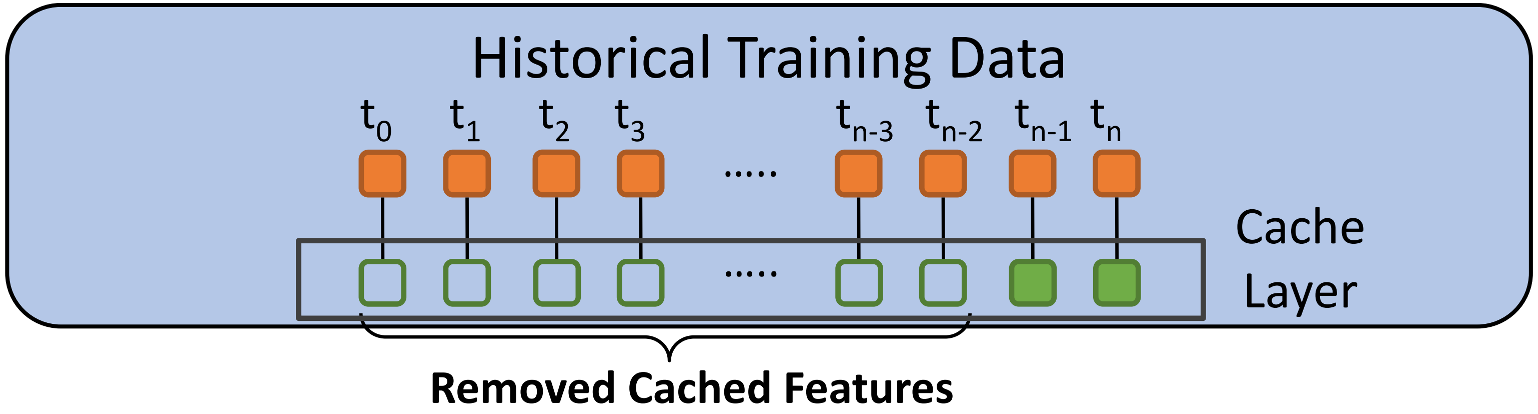

Raw Data
Discretize
Raw Data Chunks
t0 t1 t2 t3
Parser
Missing Value Imputer
Standard Scaler
Feature Hasher
Preprocess
Feature Chunks
t0 t1 t2 t3
Historical Training Data
t0 t1 t2 t3 ..... tn-3 tn-2 tn-1 tn
.....
Cache Layer

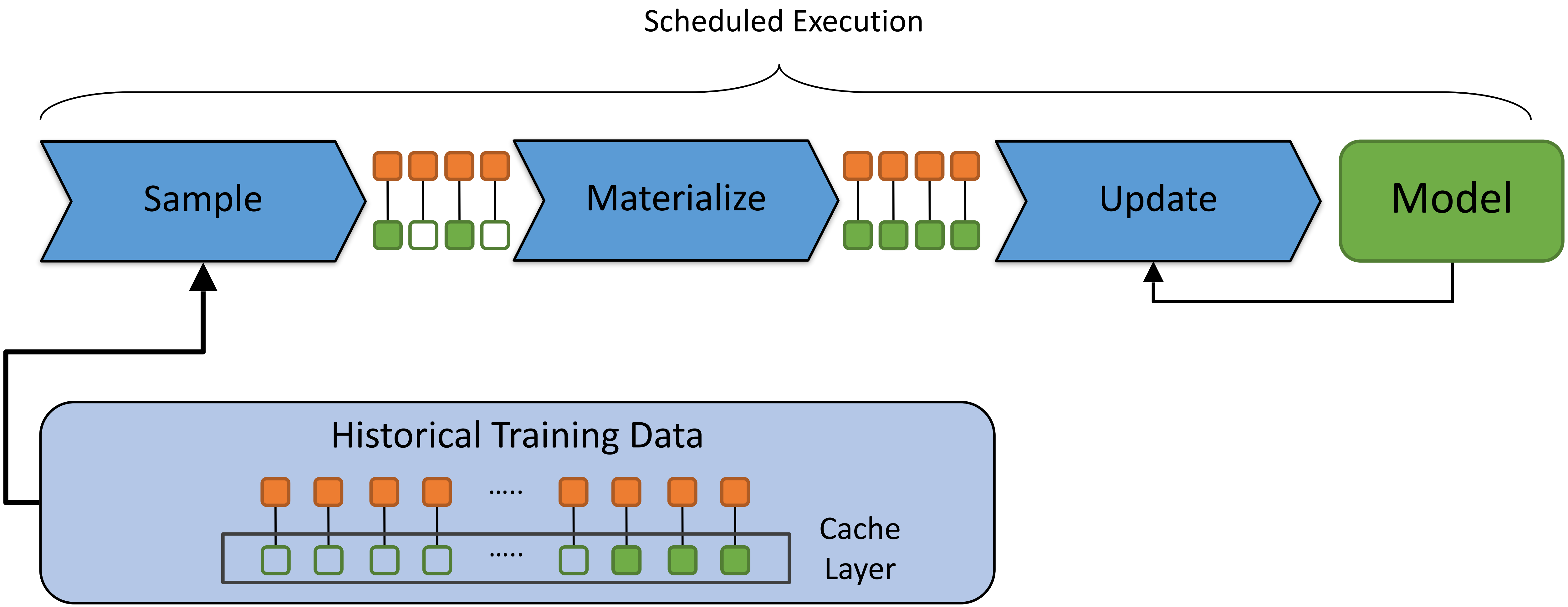

Update and Store Statistics
Stats
Stats
Parser
Missing Value Imputer
Standard Scaler
Feature Hasher
Raw Data Chunks
t_0 t_1 t_2 t_3
Feature Chunks
t_0 t_1 t_2 t_3
Raw Data
Discretize
Preprocess
Historical Training Data
t_0 t_1 t_2 t_3 ..... t_{n-3} t_{n-2} t_{n-1} t_n
.....
Cache Layer

Storage Requirement

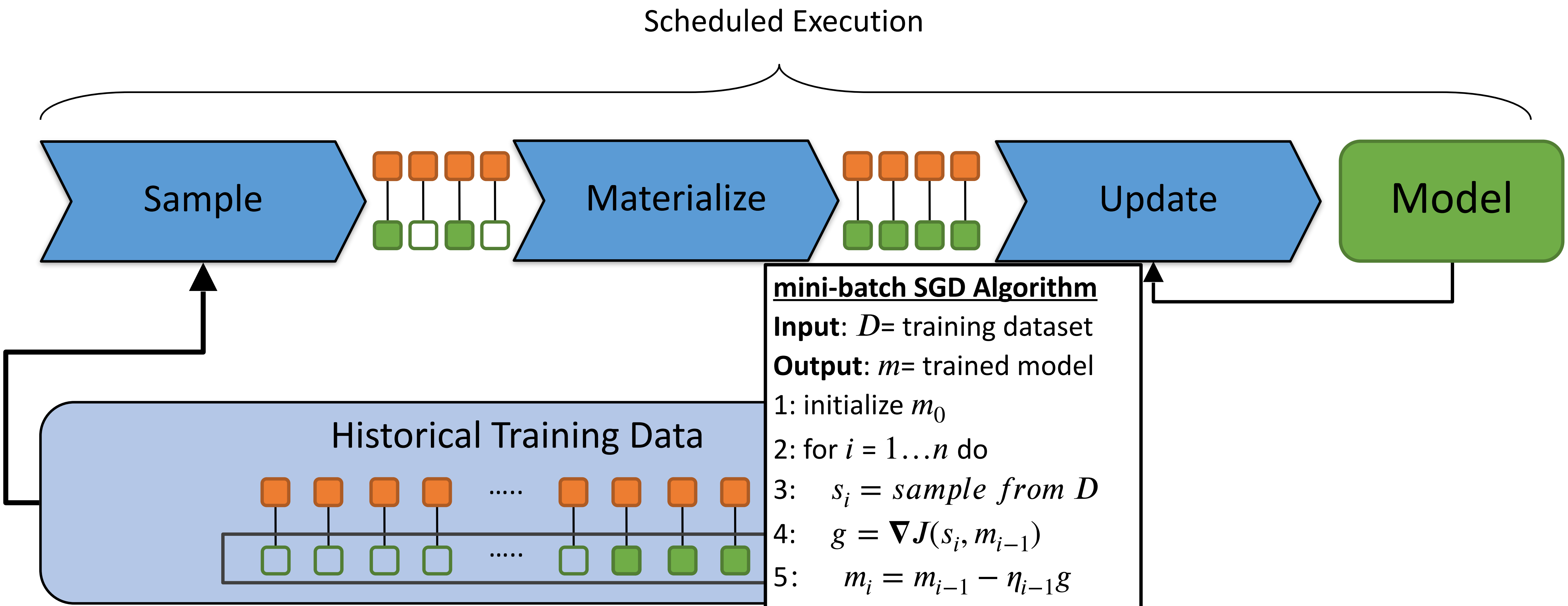


Storage Requirement




Scheduled Execution
Sample
Materialize
Update
Model
Historical Training Data
.....
.....
Cache
Layer

Proactive Training Phase



mini-batch SGD Algorithm

Input: D= training dataset

Output: m= trained model

1: initialize m_0

2: for i = $1 \ldots n$ do

3: $\quad s_i = sample\ from\ D$

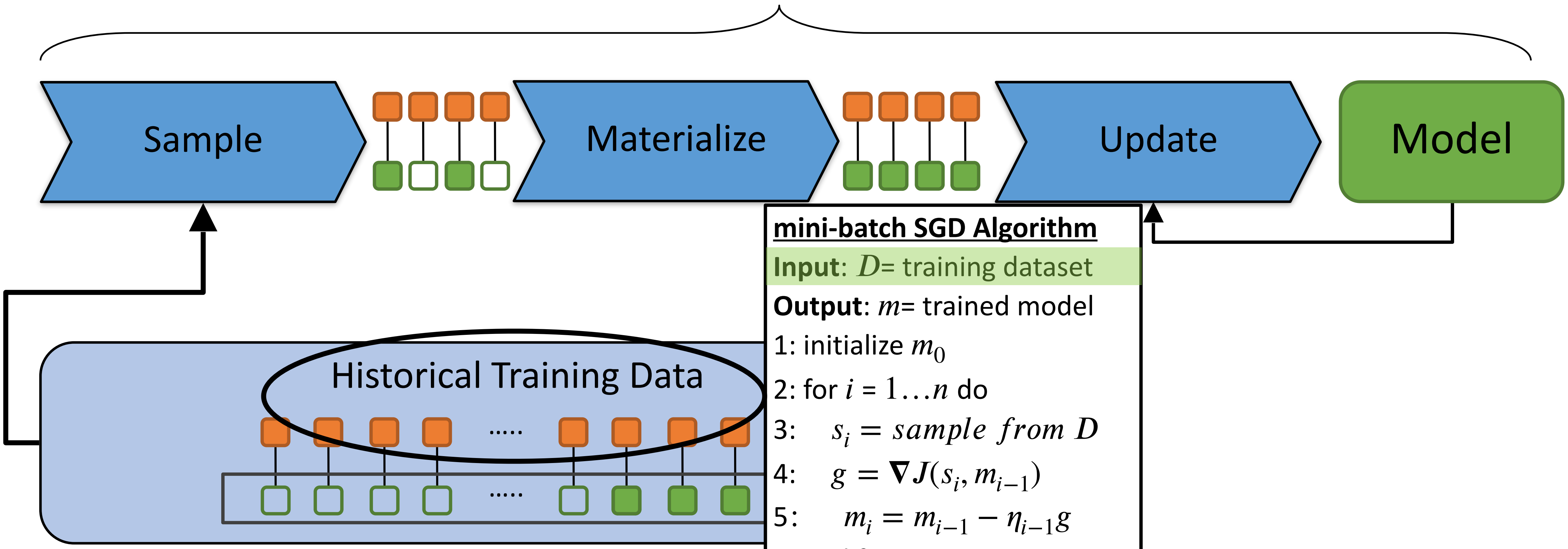
4: $\quad g = \nabla J(s_i, m_{i-1})$

5: $\quad m_i = m_{i-1} - \eta_{i-1} g$

6: end for

7: return m_n

Proactive Training Phase



mini-batch SGD Algorithm

Input: D= training dataset

Output: m= trained model

1: initialize m_0

2: for i = $1 \ldots n$ do

3: $\quad s_i = sample\ from\ D$

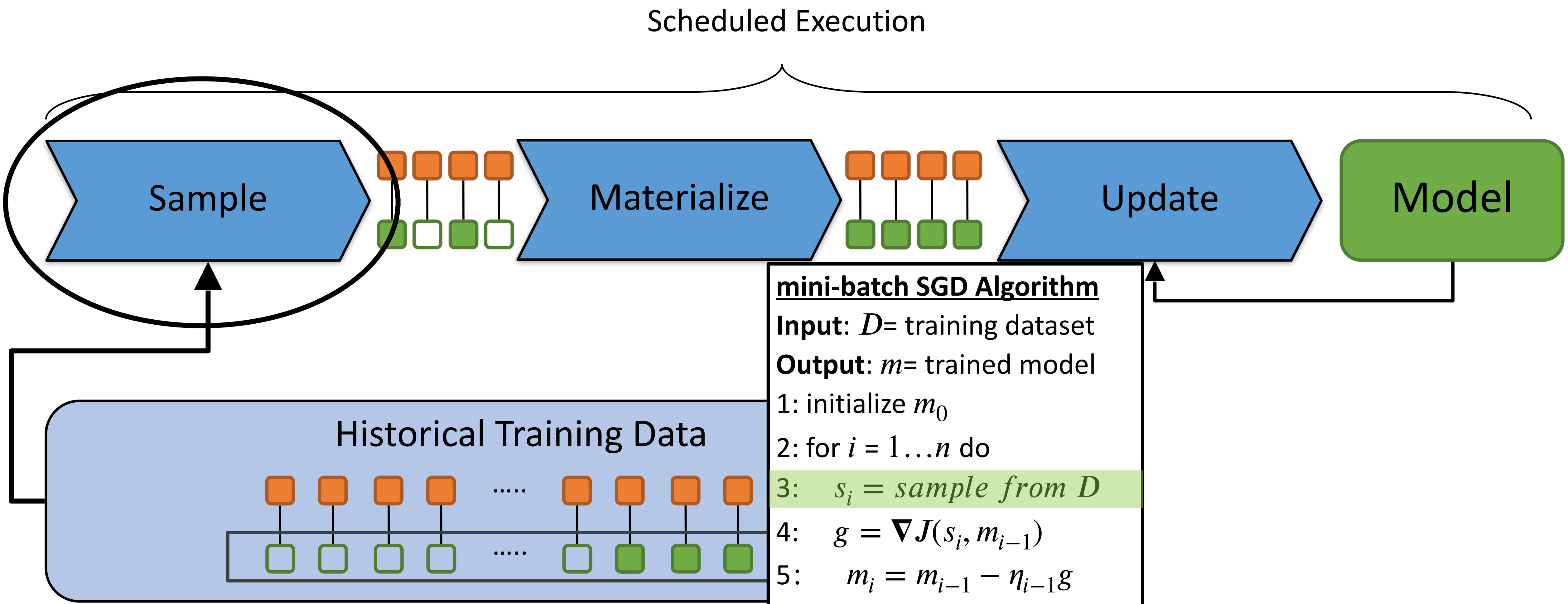
4: $\quad g = \nabla J(s_i, m_{i-1})$

5: $\quad m_i = m_{i-1} - \eta_{i-1} g$

6: end for

7: return m_n

Proactive Training Phase



mini-batch SGD Algorithm

Input: D= training dataset

Output: m= trained model

1: initialize m_0

2: for i = $1 \ldots n$ do

3: $\quad s_i = sample\ from\ D$

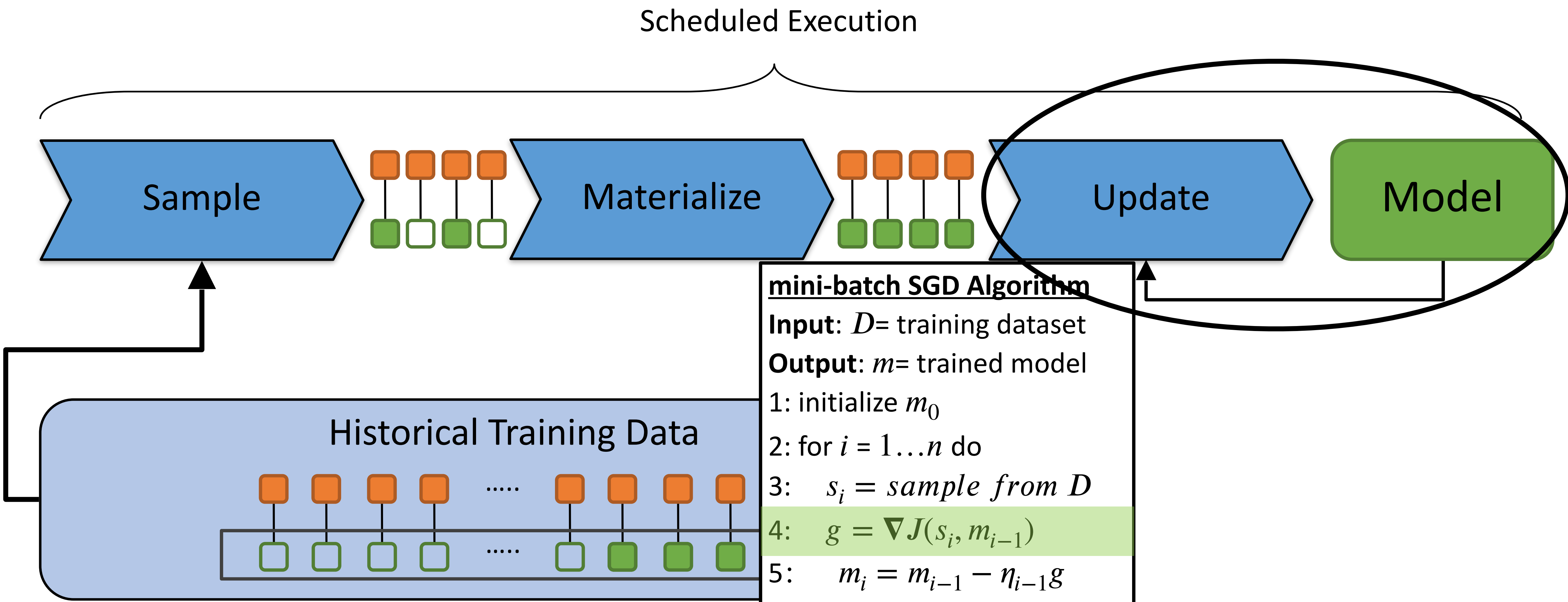
4: $\quad g = \nabla J(s_i, m_{i-1})$

5: $\quad m_i = m_{i-1} - \eta_{i-1} g$

6: end for

7: return m_n

Proactive Training Phase



mini-batch SGD Algorithm

Input: D= training dataset

Output: m= trained model

1: initialize m_0

2: for i = $1 \ldots n$ do

3: $\quad s_i = sample\ from\ D$

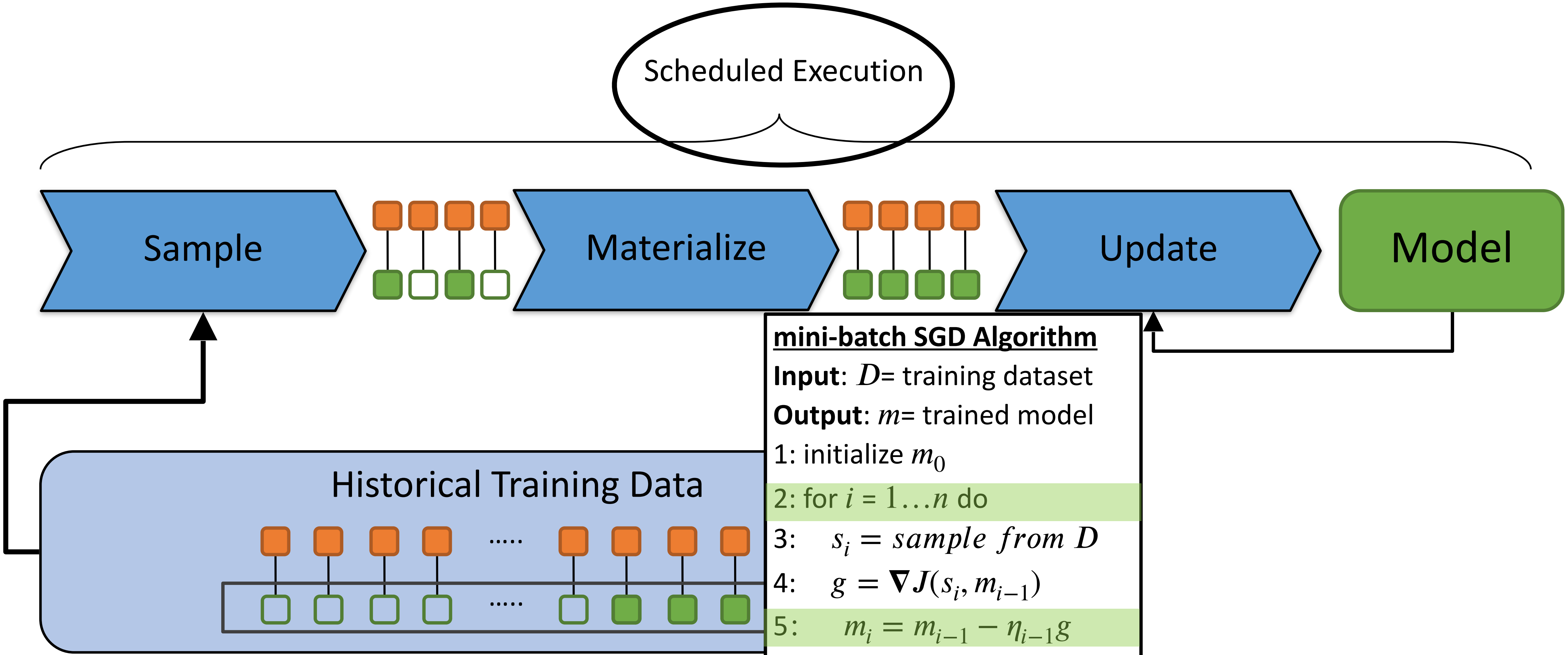
4: $\quad g = \nabla J(s_i, m_{i-1})$

5: $\quad m_i = m_{i-1} - \eta_{i-1} g$

6: end for

7: return m_n

Proactive Training Phase



mini-batch SGD Algorithm

Input: D= training dataset

Output: m= trained model

1: initialize m_0

2: for i = $1 \ldots n$ do

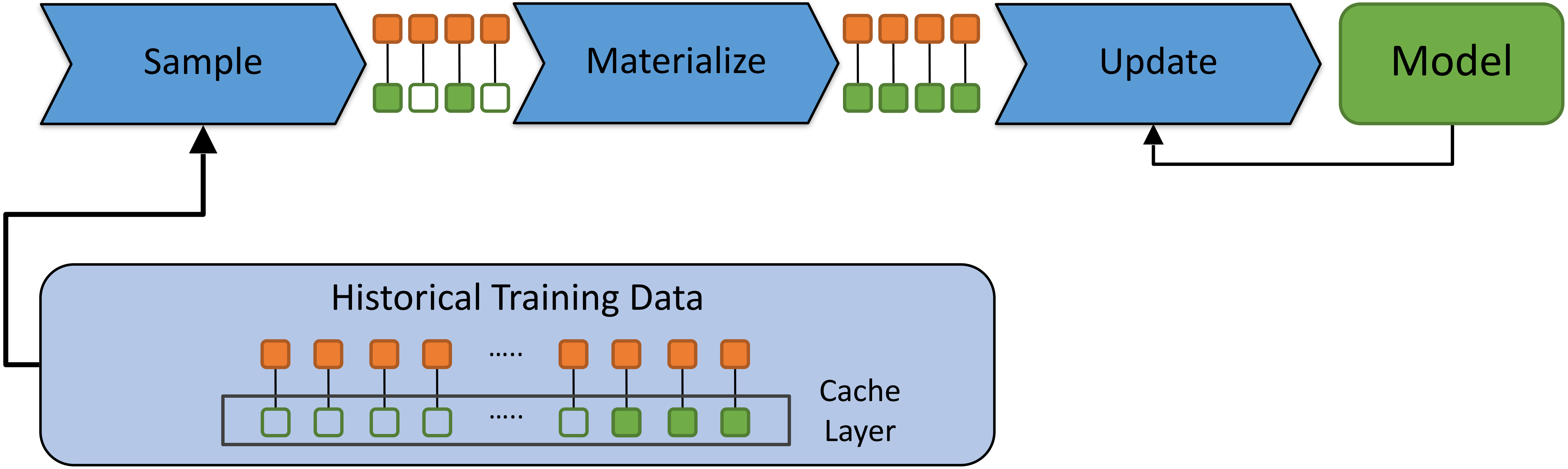
3: $\quad s_i = sample\ from\ D$

4: $\quad g = \nabla J(s_i, m_{i-1})$

5: $\quad m_i = m_{i-1} - \eta_{i-1} g$

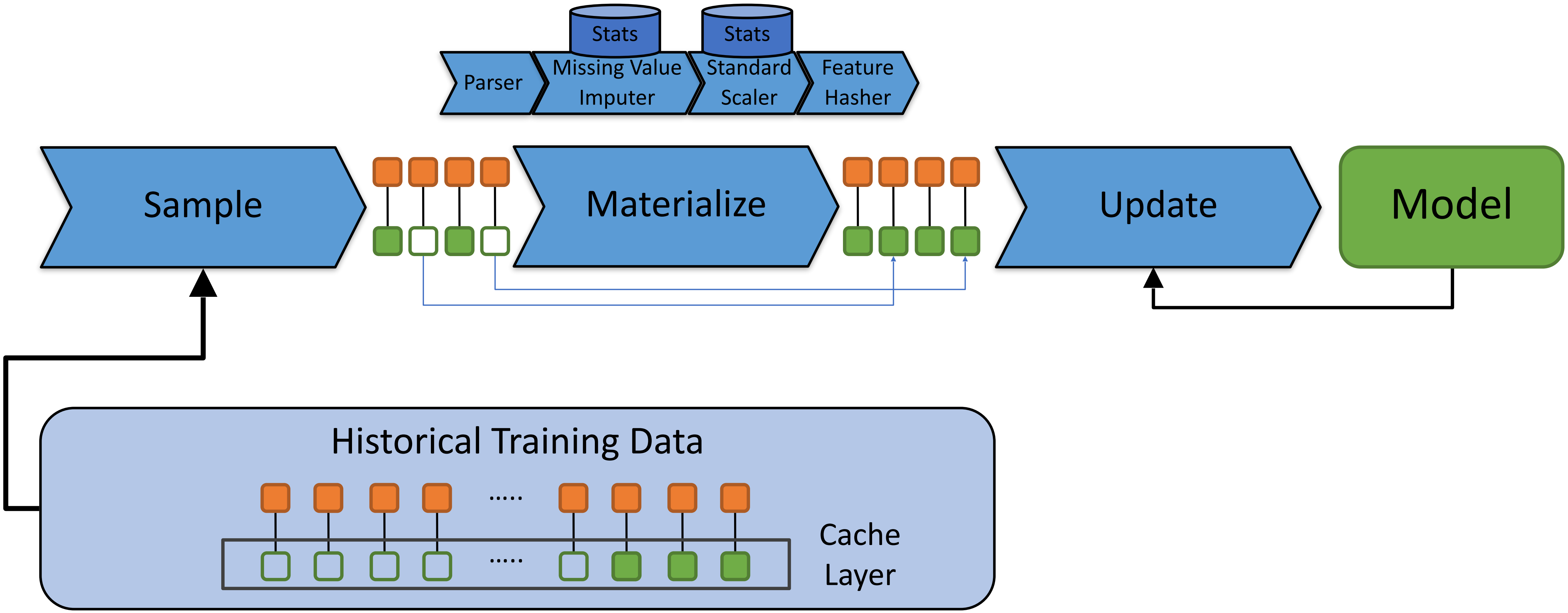
6: end for

7: return m_n


Sample
Materialize
Update
Model
Historical Training Data
.....
.....
Cache
Layer

Data Materializing

Statistics precomputed during the Data Preparation Phase



Evaluation

URL Pipeline

Parser → Missing Value Imputer → Standard Scaler → Feature Hasher → SVM Model

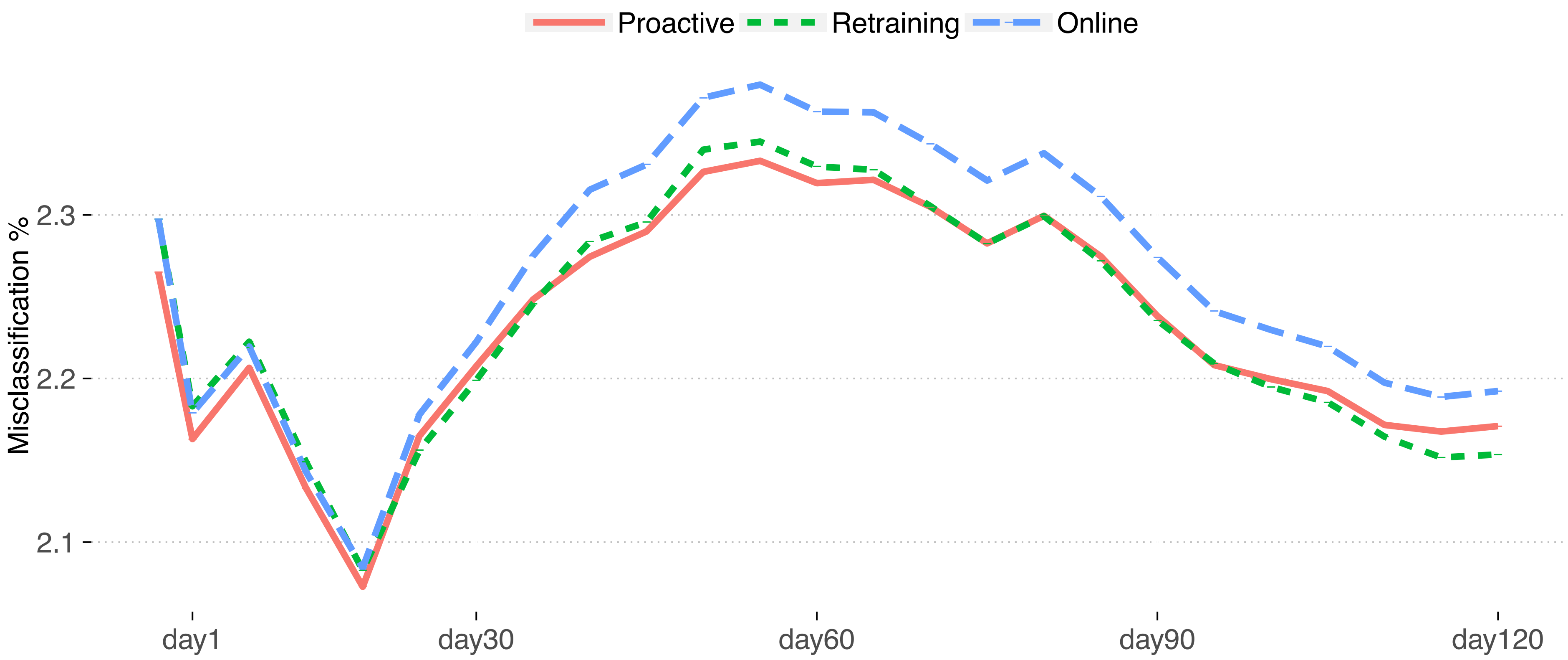
Taxi Pipeline

Parser → Feature Extractor → Anomaly Detector → Standard Scaler → Linear Regression Model

Datasets	Size	#Instances	Initial	Deployment
URL	2.1 GB	2.4 M	Day 0	Day 1-120
Taxi	42 GB	280 M	Jan 15	Feb 15 – Jun 16

Can Proactive Training provide the same level of **quality** as Retraining?

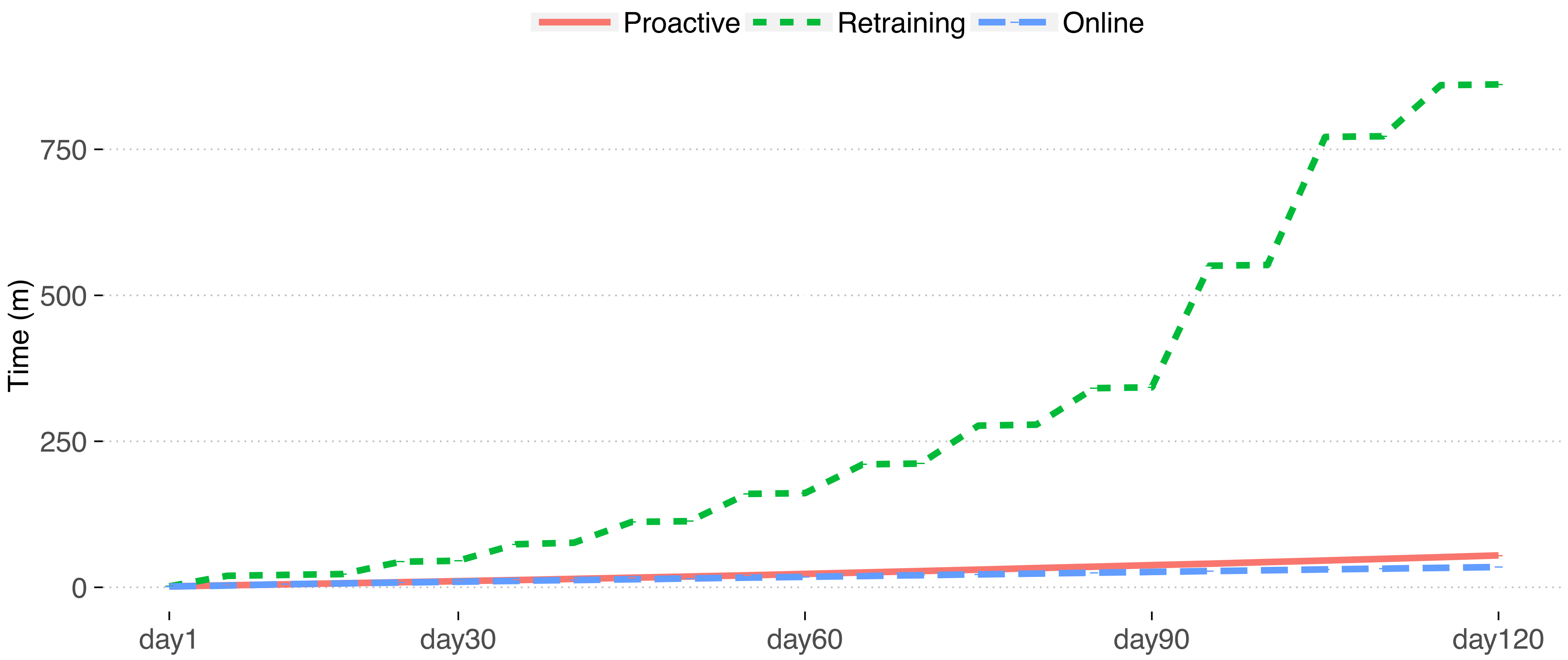
Can Proactive Training provide the same level of **quality** as Retraining?



Cumulative Prequential Prediction Error Rate for the URL Pipeline During the Deployment

Can Proactive Training perform (almost) as **efficiently** as Online Learning?

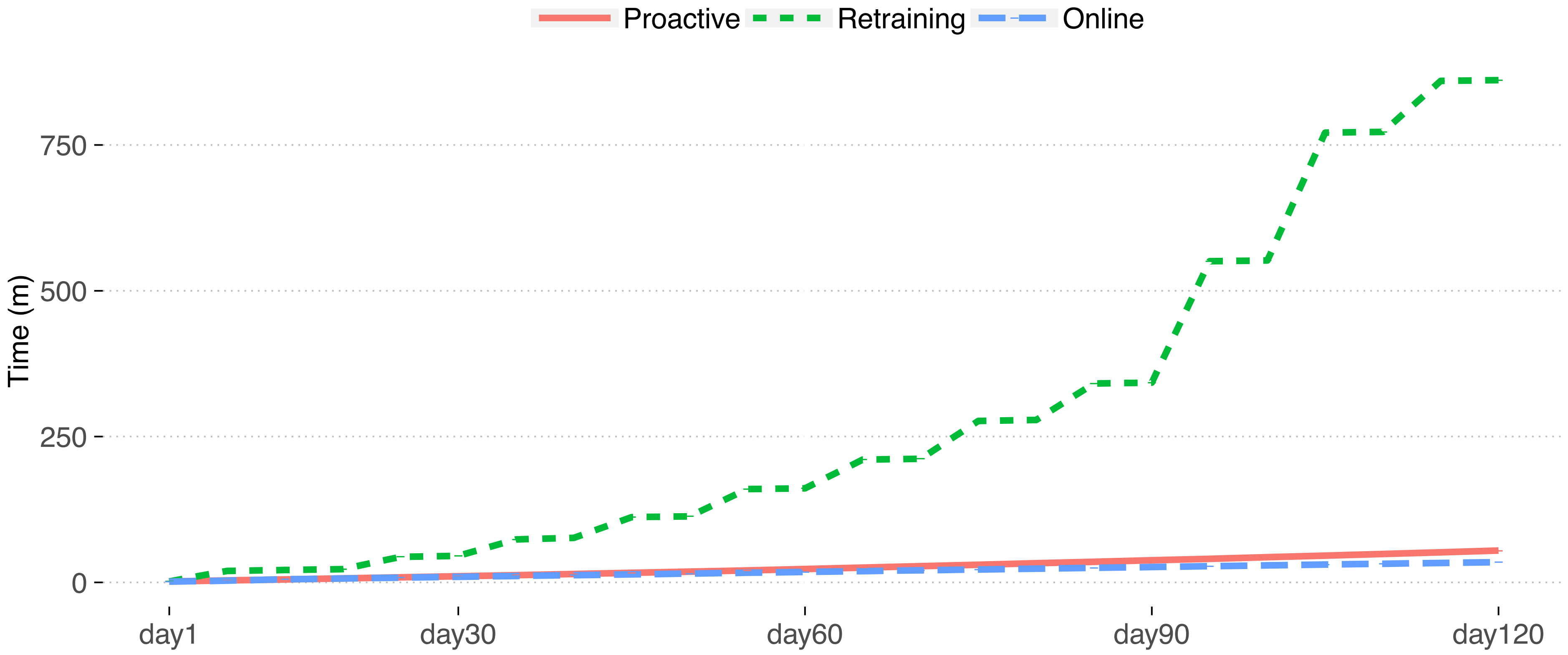
Can Proactive Training perform (almost) as **efficiently** as Online Learning?



Cumulative Training Time for the URL Pipeline During the Deployment

Training Time

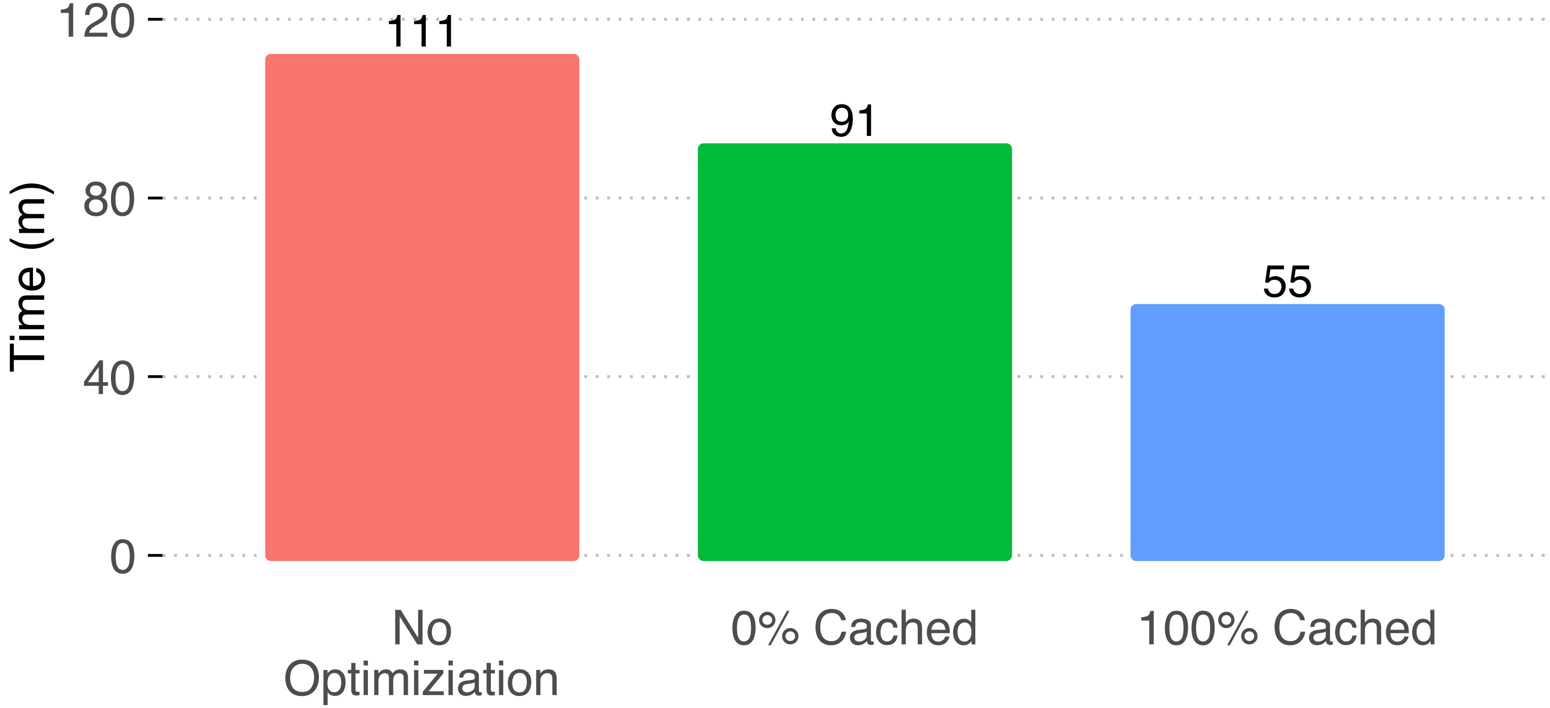
Can Proactive Training perform (almost) as **efficiently** as Online Learning?



Cumulative Training Time for the URL Pipeline During the Deployment

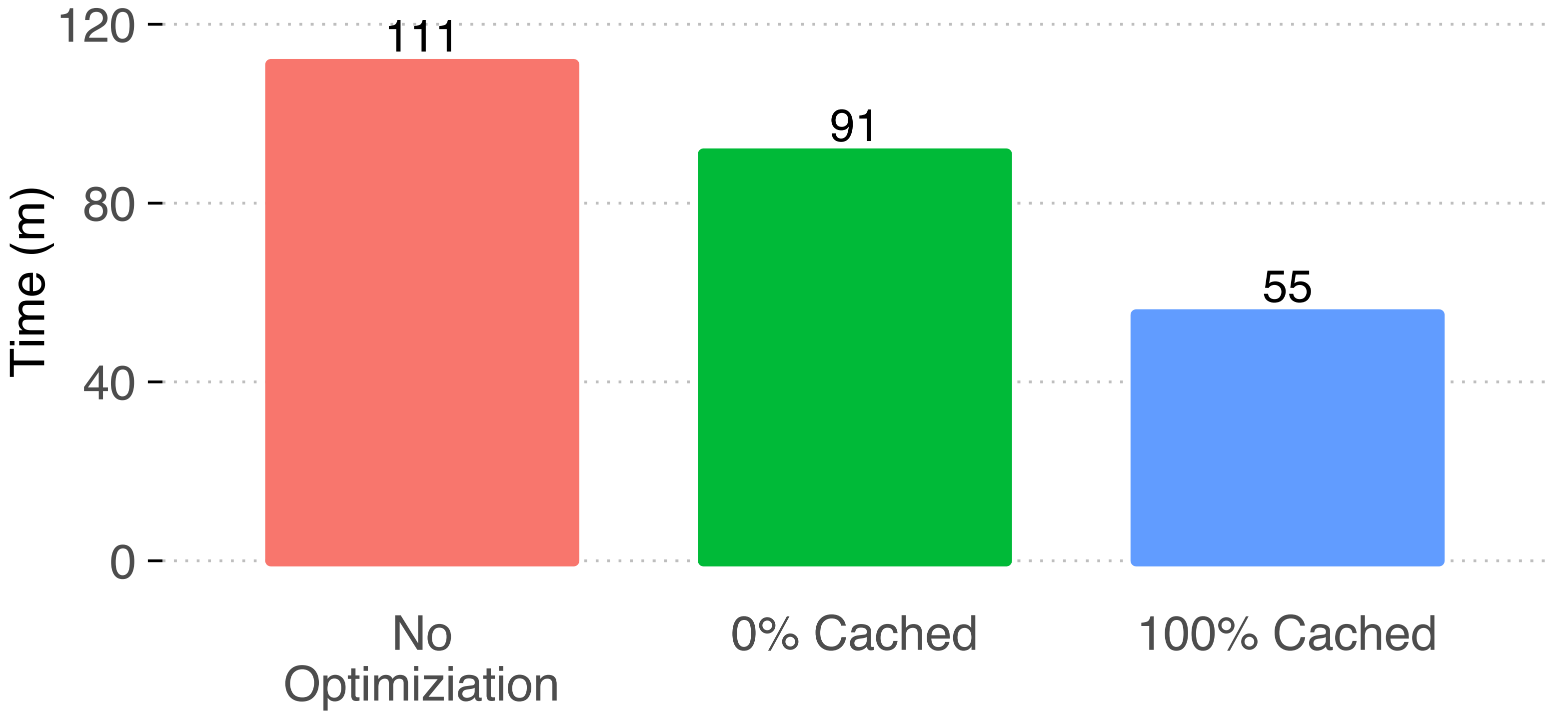
Proactive training provides same level of model accuracy as Retraining, while matching the speed of Online Learning

What are the effects of **Statistics Computation** and **Feature Caching** ?



Total Training Time in Presence of Statistics Computation and Feature Caching

What are the effects of **Statistics Computation** and **Feature Caching** ?

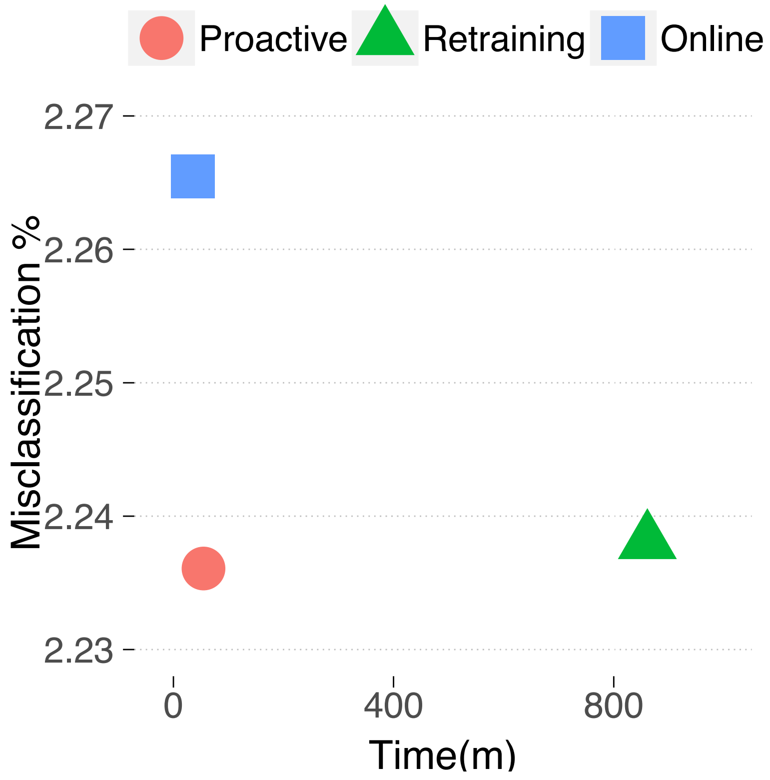


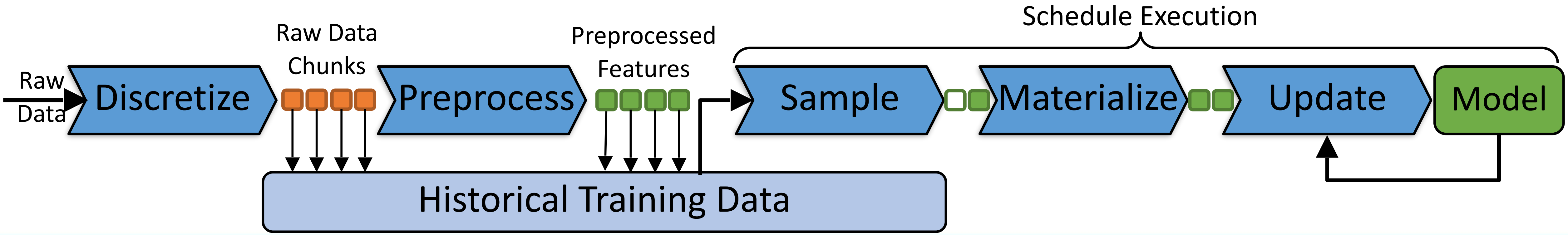
Total Training Time in Presence of Statistics Computation and Feature Caching

Statistics Computation and Feature Caching improves the performance of Proactive training by a factor of 2

Continuous Deployment Platform

- Proactive Training, instead of Offline Retraining
- Feature Caching
- Online Statistics Computation
- Reduces the total training time
- Achieves high quality





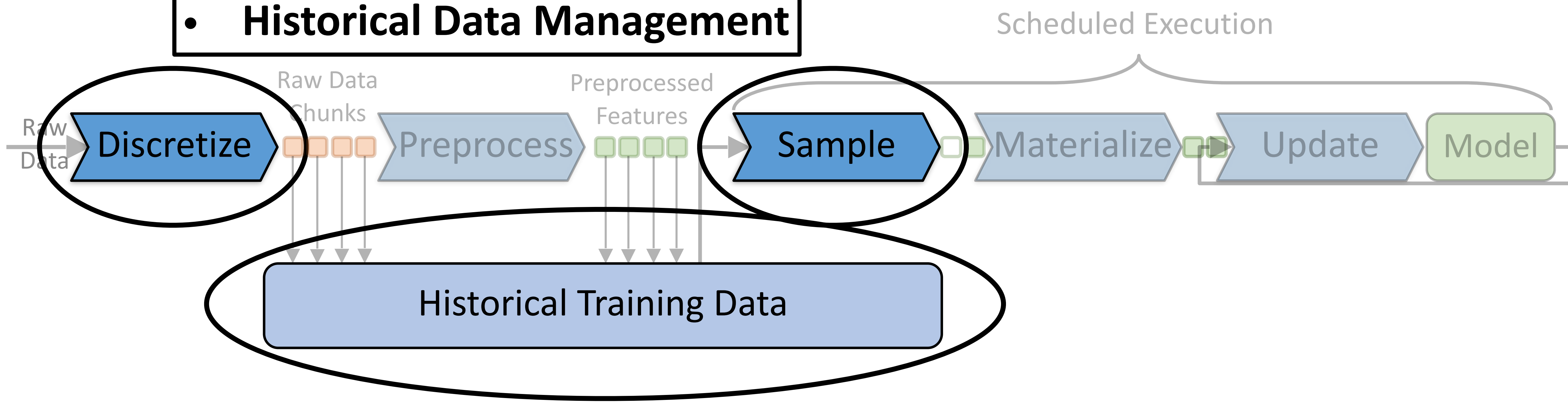
References

1. D. Crankshaw, X. Wang, G. Zhou, M. Franklin, et al. 2016. Clipper: A Low-Latency Online Prediction Serving System. arXiv preprint arXiv:1612.03079 (2016).
2. D. Crankshaw, P. Bailis, J. Gonzalez, H. Li, et al. 2014. The missing piece incomplex analytics: Low latency, scalable model management and serving with velox.
3. D. Baylor, E. Breck, H. Cheng, N. Fiedel, et al. 2017. TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1387–1395.
4. L. Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010. Springer, 177–186.
5. M. Zaharia, M. Chowdhury, M. Franklin, S. Shenker, and I. Stoica. 2010. Spark: cluster computing with working sets. HotCloud 10 (2010), 10–10.
6. O. Chapelle. [n. d.]. NYC Taxi & Lomousine Commision Trip Record Data. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. [Online;accessed 10-April-2018].
7. J. Ma, L. Saul, S. Savage, and G. Voelker. 2009. Identifying suspicious URLs: an application of large-scale online learning. In Proceedings of the 26th annual international conference on machine learning. ACM, 681–688.
8. D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
9. M. Zeiler. 2012. ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012).
10. T. Tieleman and G. Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4, 2 (2012), 26–31.

Backup Slides

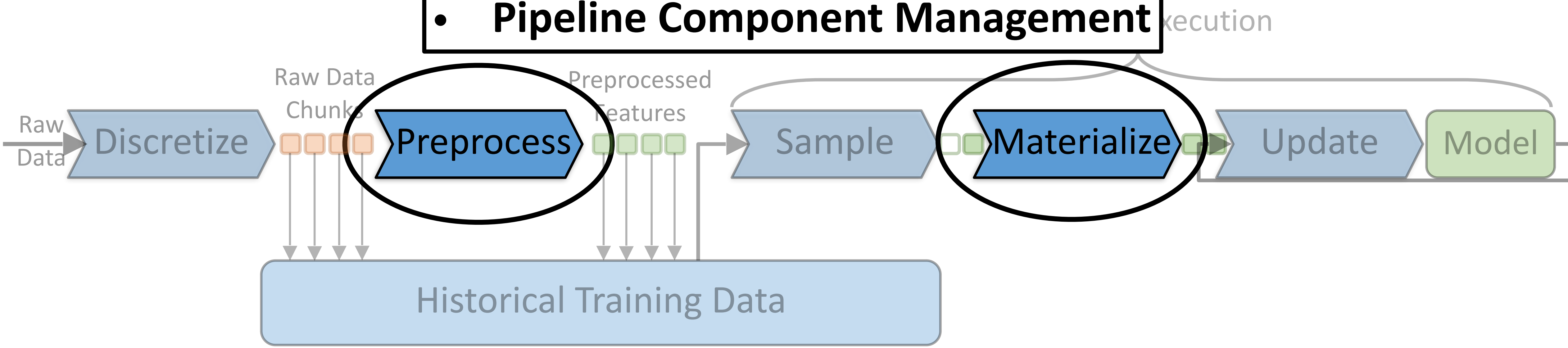
Data Manager

- Data Discretizing
- Data Sampling
- Historical Data Management



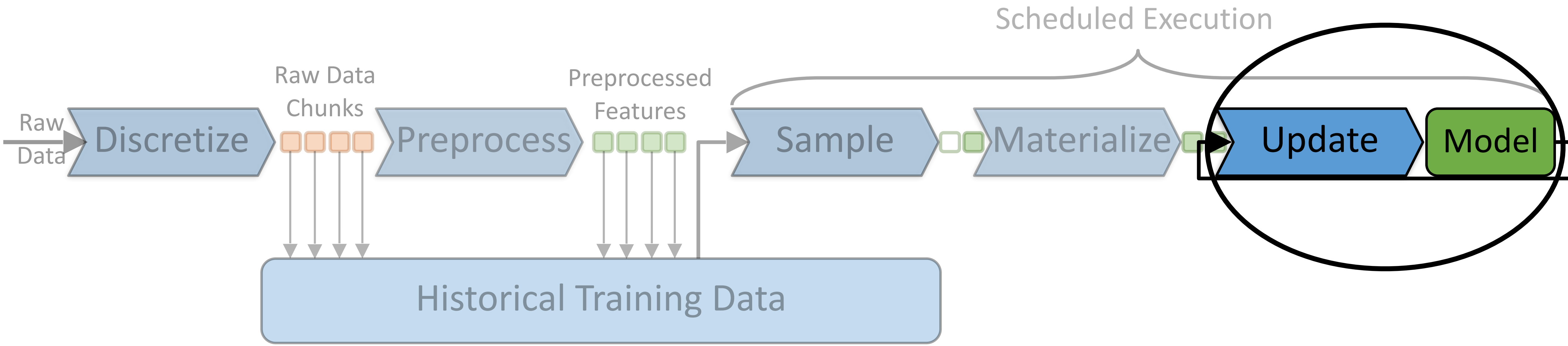
Pipeline Manager

- Data Preprocessing
- Data Materialization
- Pipeline Component Management



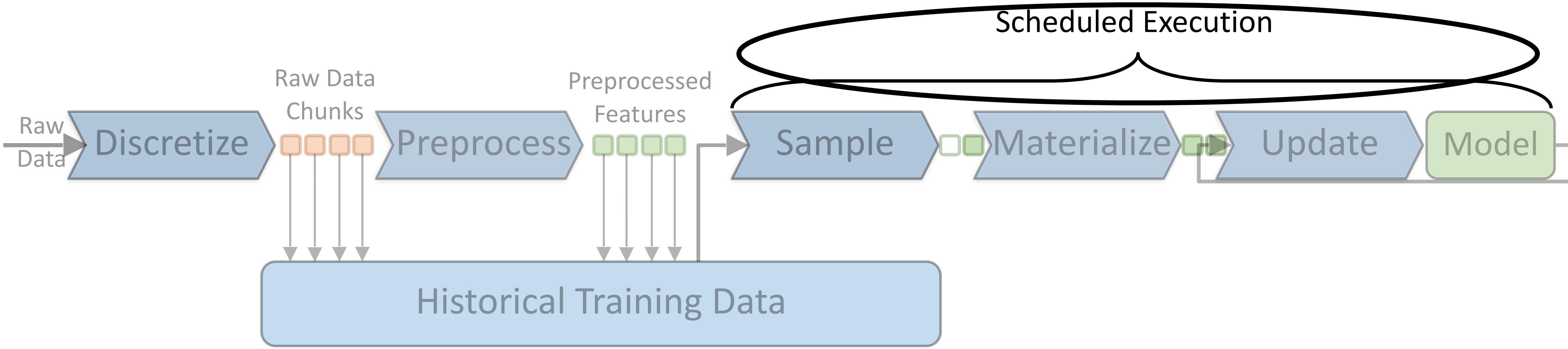
Model Updater

- Online Training
- Proactive Training



Scheduler

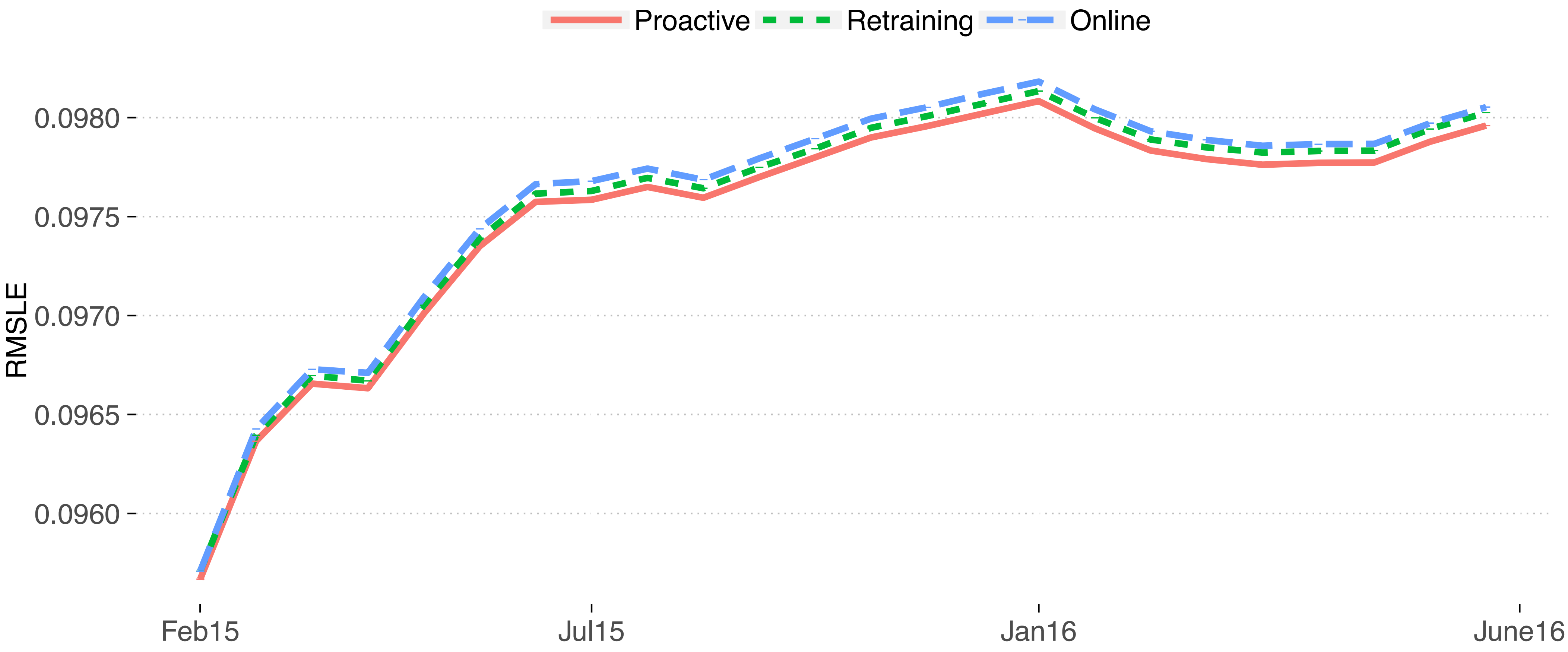
- **Schedule Proactive Training**



Related Work

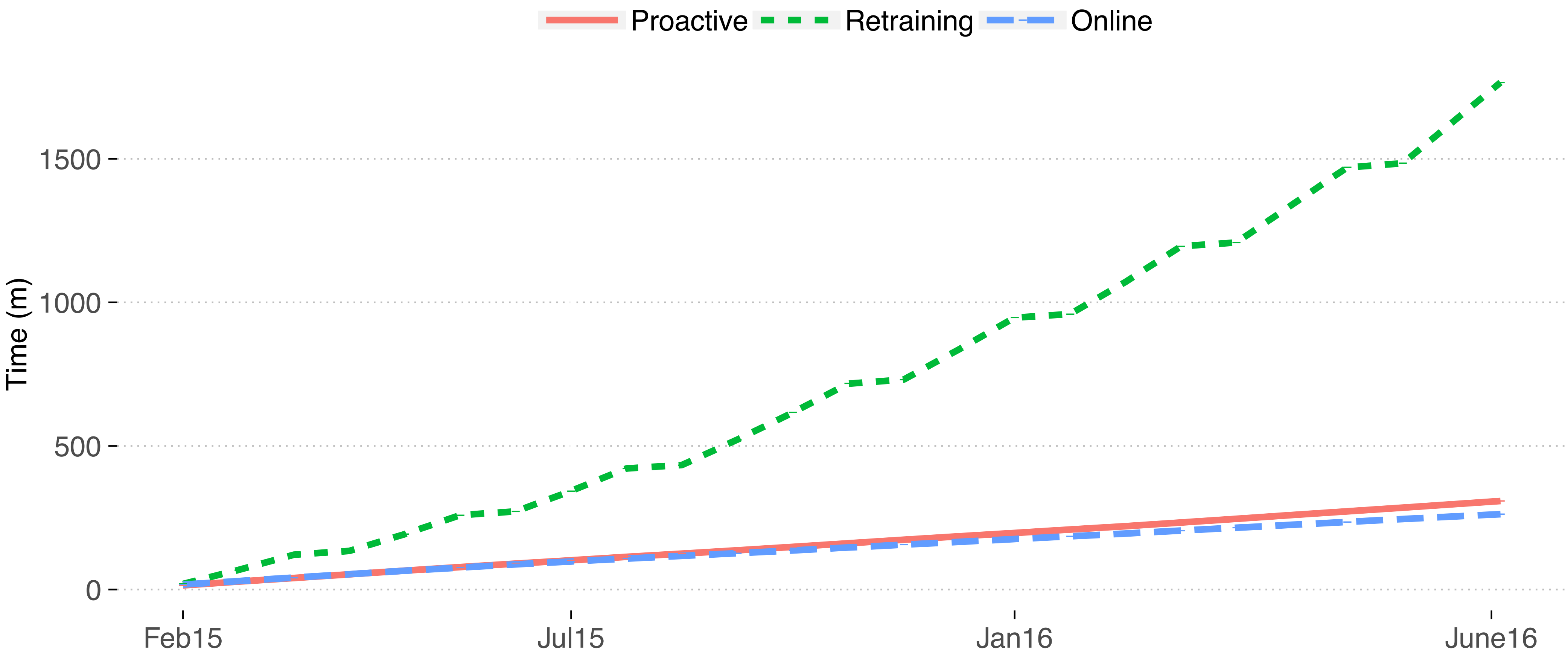
- TFX
  - Manual Retraining
  - No Online Learning
- Velox
  - Automatic Retraining
  - Online Learning
- Clipper
  - No Retraining
  - No Online Learning
  - Ensemble of Models

Proactive Training vs Periodical Retraining (Taxi)



Cumulative Prequential Prediction Error Rate for the Taxi Pipeline During the Deployment

Proactive Training vs Periodical Retraining (Taxi)



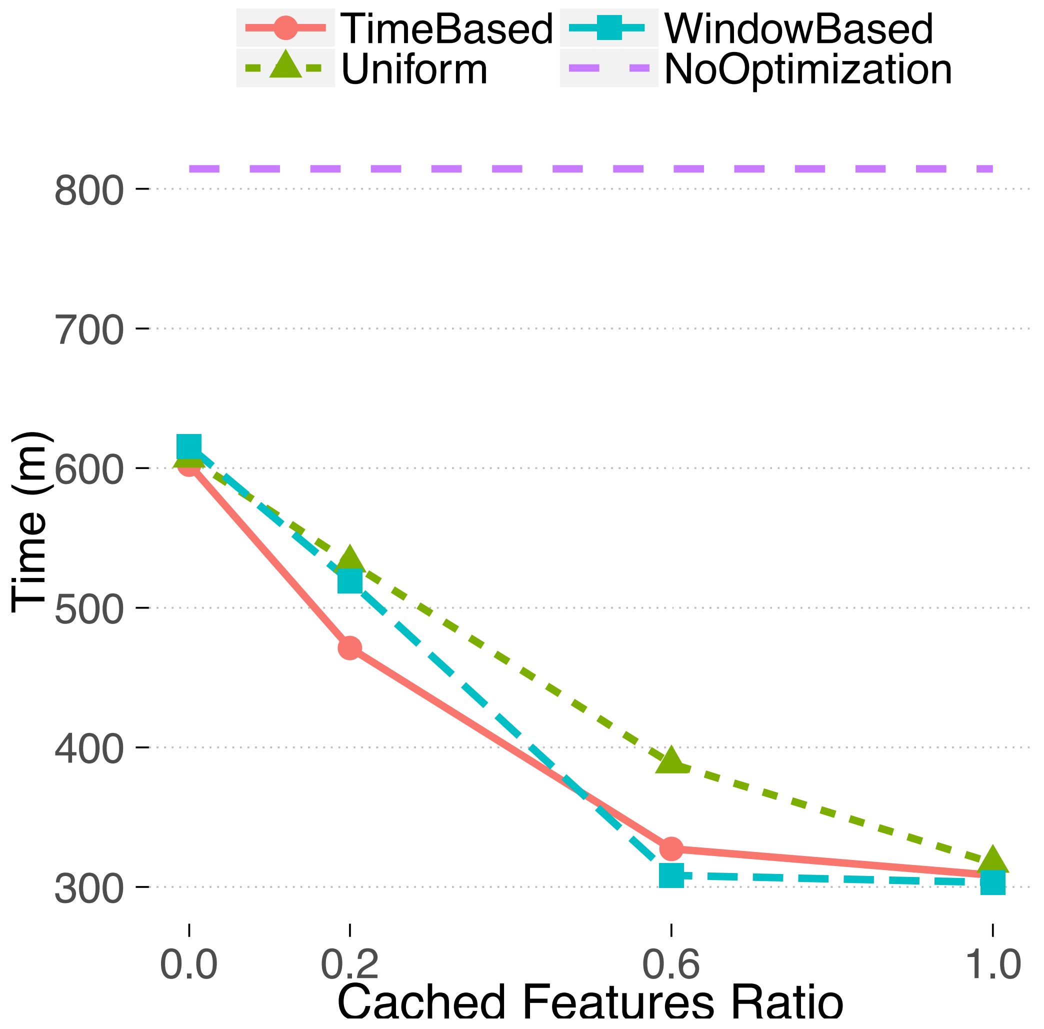
Cumulative Training Time for the Taxi Pipeline During the Deployment

Materialization and Statistics Computation (Taxi)

Materialization Utilization Rate for different ratio of Cached Features Taxi

Sampling	Ratio of Cached Features	
	m = 0.2	m = 0.6
Uniform	0.51	0.90
Window-based	0.57	1.0
Time-based	0.65	0.97

Materialization Utilization Rate:
Ratio of preprocessed features that skipped the materialization step

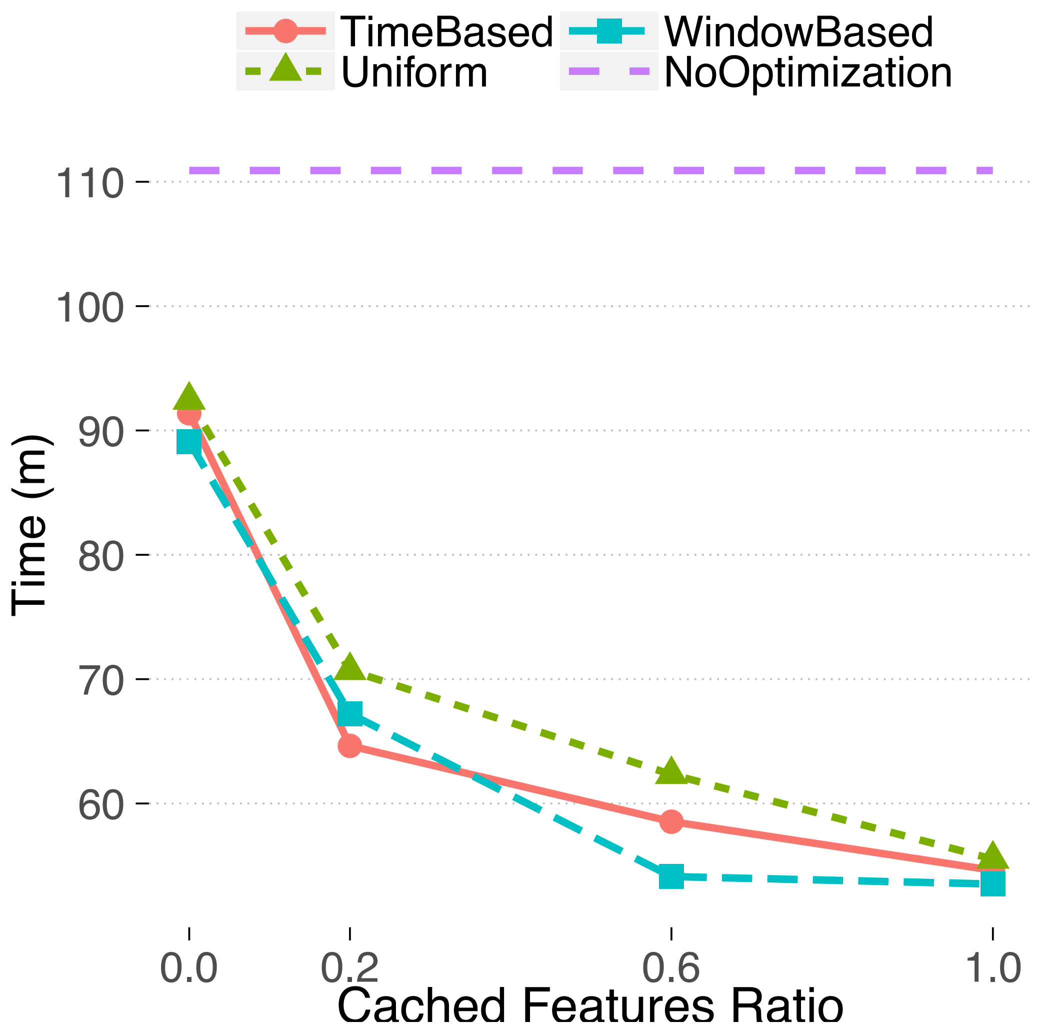


Materialization and Statistics Computation (URL)

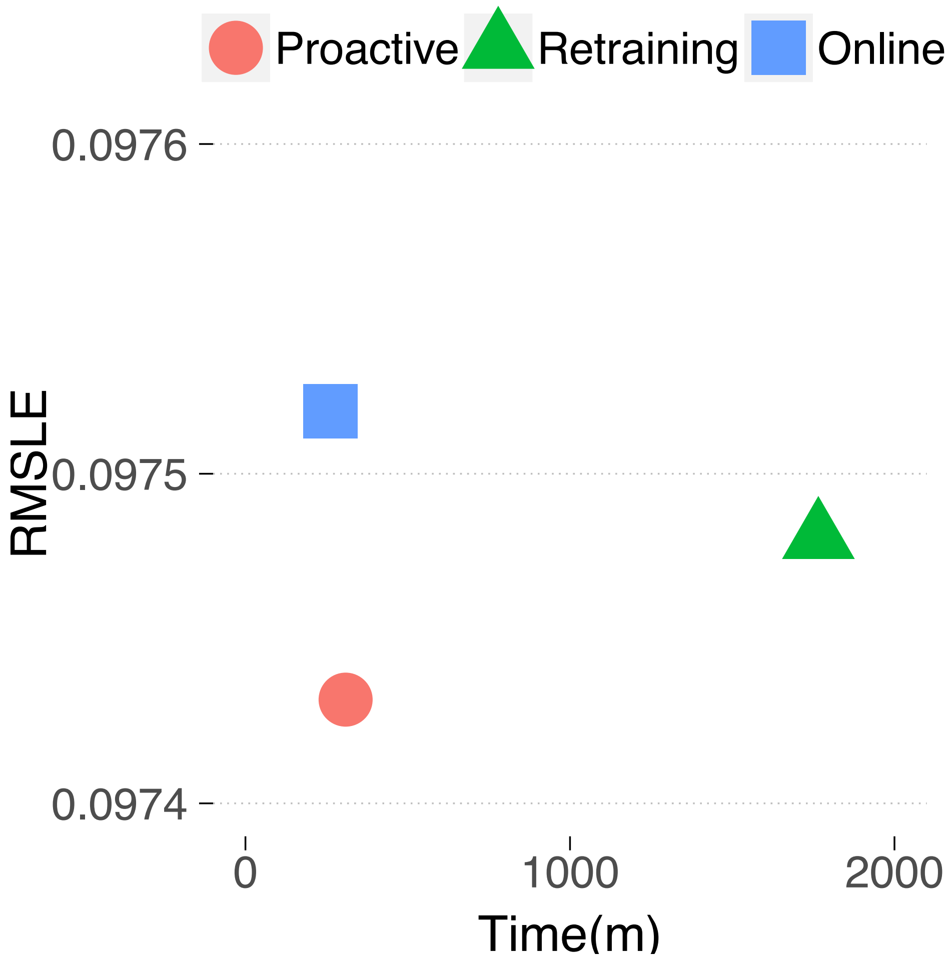
Materialization Utilization Rate for different ratio of Cached Features URL

Sampling	Ratio of Cached Features	
	m = 0.2	m = 0.6
Uniform	0.52	0.91
Window-based	0.58	1.0
Time-based	0.68	0.97

Materialization Utilization Rate:
Ratio of preprocessed features that skipped the materialization step

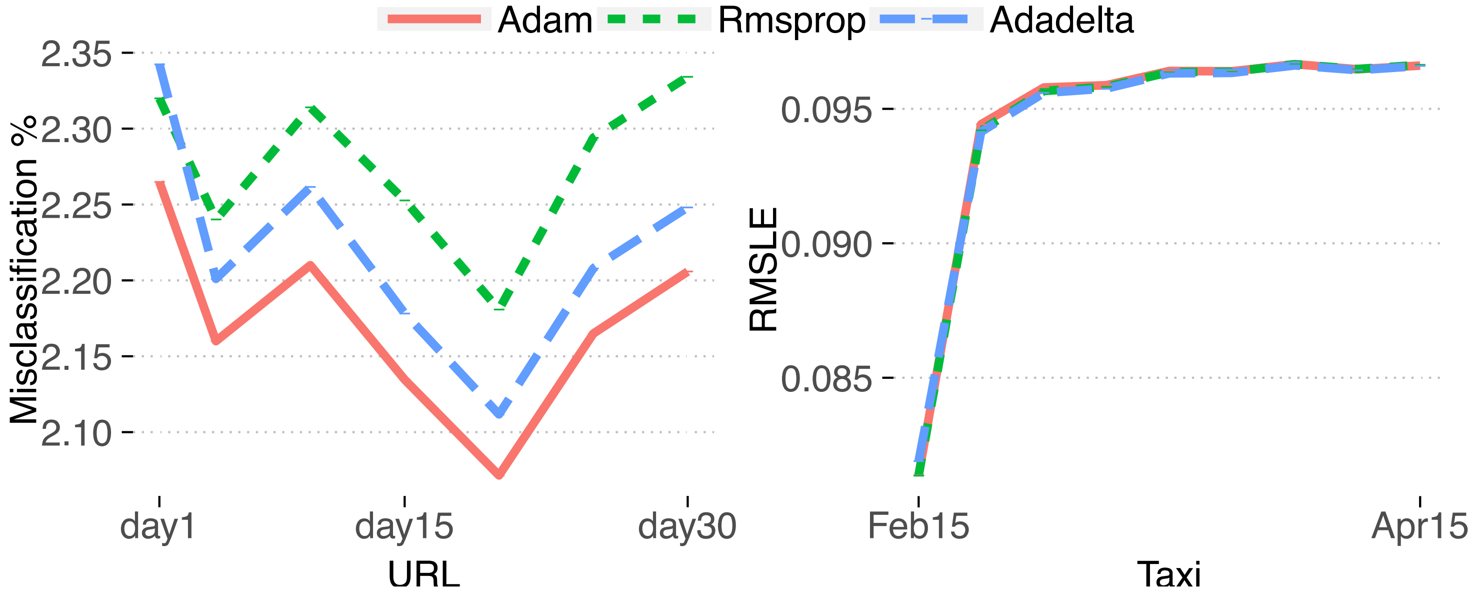


Time vs Quality (Taxi)



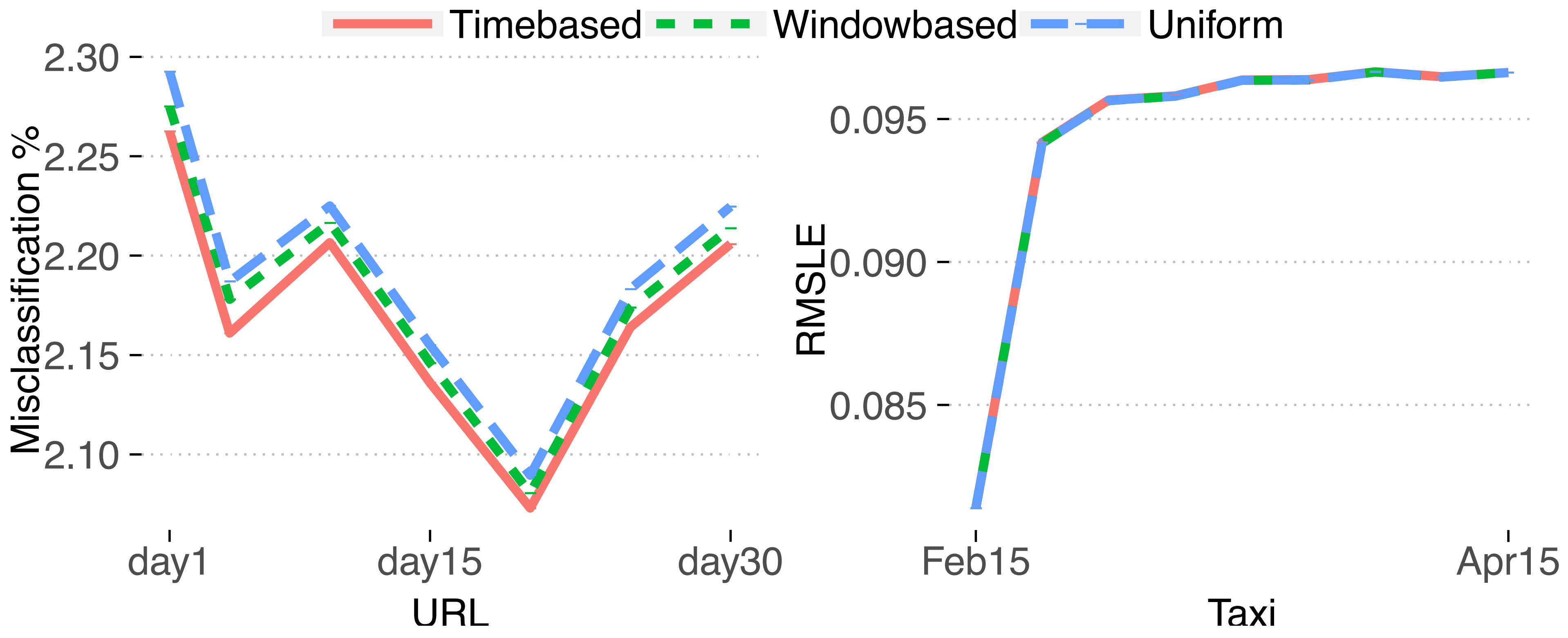
Adaptation	URL			Taxi		
	r = 1E-2	r = 1E-3	r = 1E-4	r = 1E-2	r = 1E-3	r = 1E-4
Adam	0.030	**0.026**	0.035	0.09553	0.09551	**0.09551**
RMSProp	0.030	**0.027**	0.034	0.09552	0.09552	**0.09550**
Adadelta	0.029	**0.028**	0.034	**0.09609**	0.09610	0.09619

Effect Learning Rate Adaption and Regularization Parameter on Initial Training



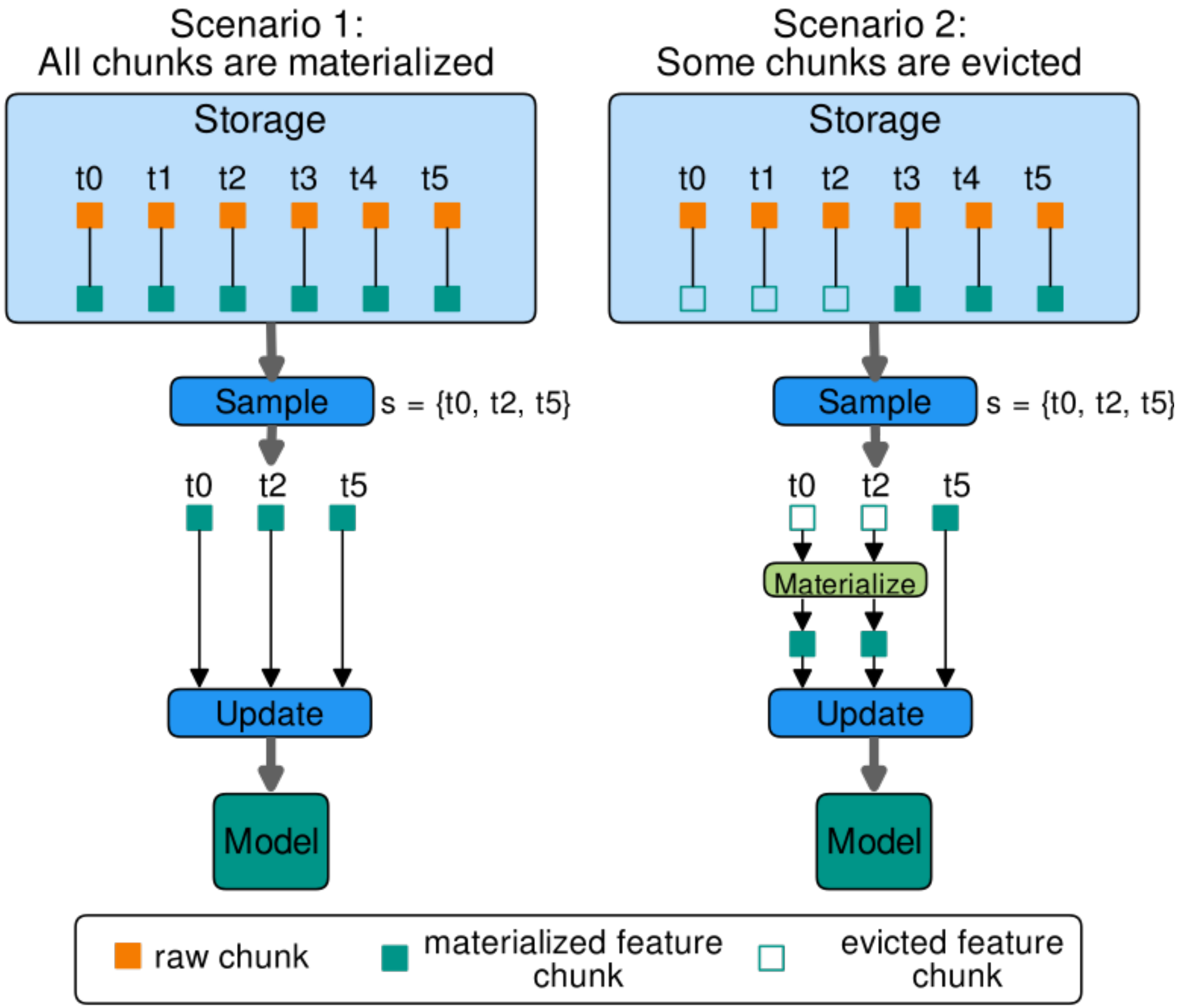
Effect of Learning Rate Adaption during the Deployment

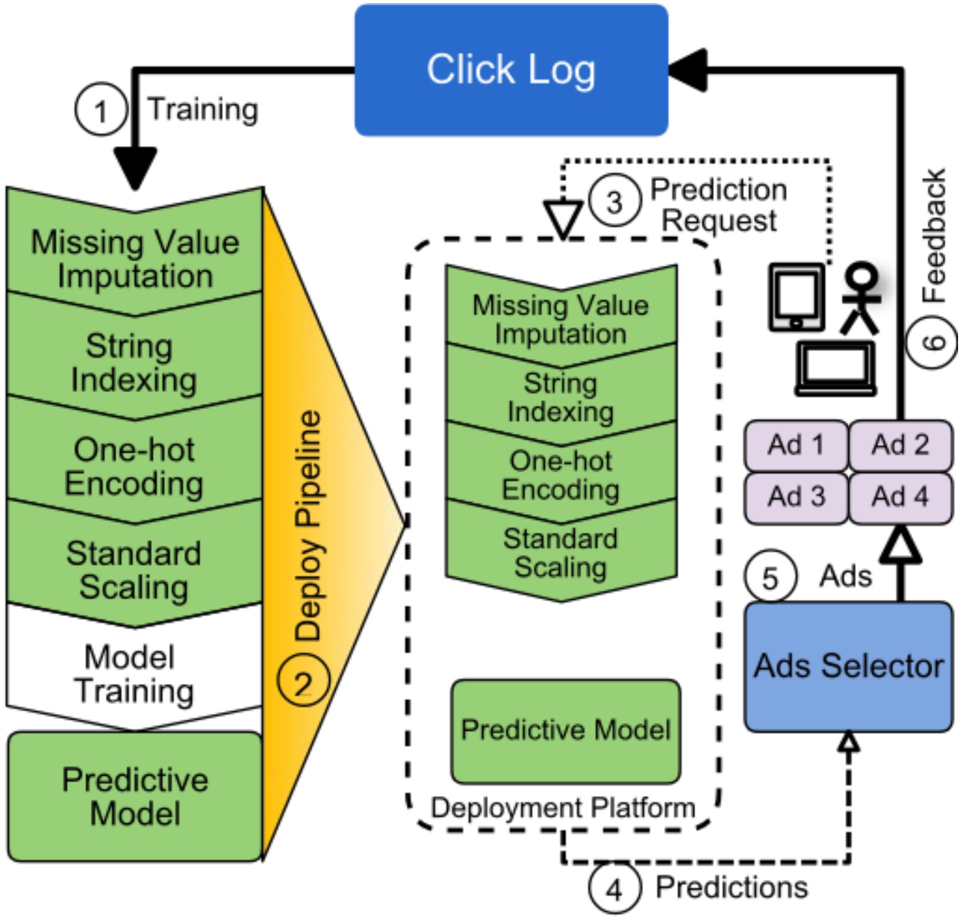
Effect of Sampling on model quality

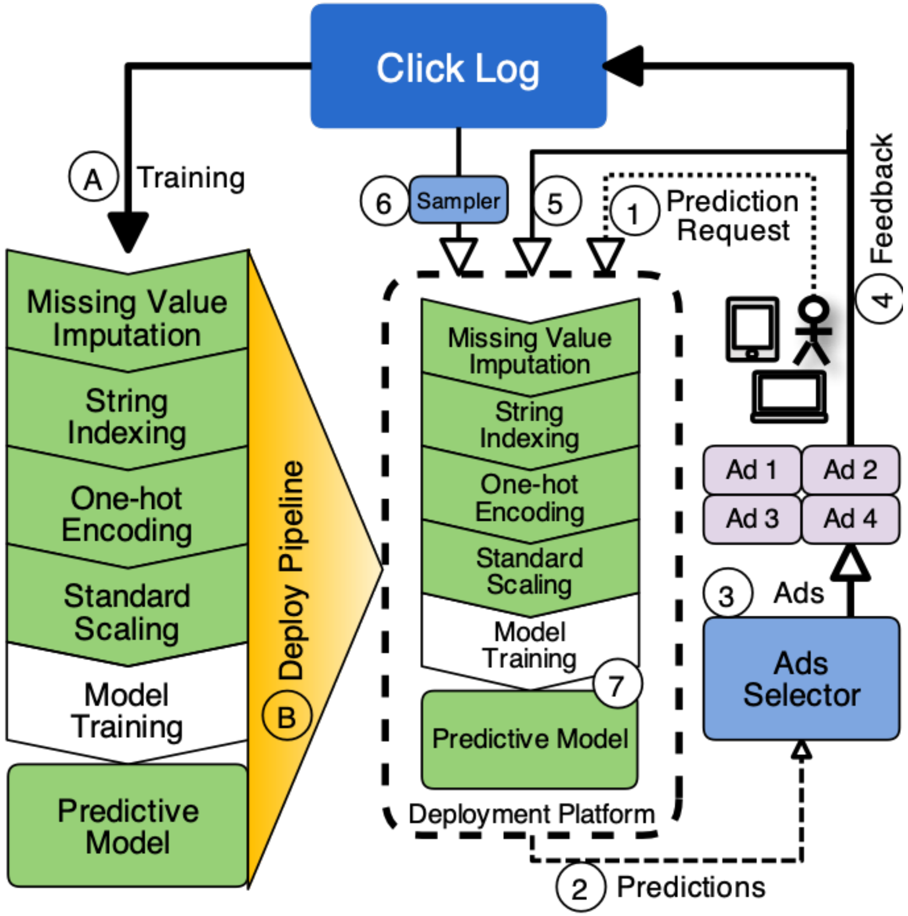


Effect of Sampling Method on the Error Rate

Materialization Process




Click Log
1 Training
Missing Value Imputation
String Indexing
One-hot Encoding
Standard Scaling
Model Training
Predictive Model
Deploy Pipeline
2
3 Prediction Request
Missing Value Imputation
String Indexing
One-hot Encoding
Standard Scaling
Predictive Model
Deployment Platform
4 Predictions
6 Feedback
Ad 1
Ad 2
Ad 3
Ad 4
5 Ads
Ads Selector


Click Log
A Training
Missing Value Imputation
String Indexing
One-hot Encoding
Standard Scaling
Model Training
Predictive Model
Deploy Pipeline
B
6 Sampler
5
1 Prediction Request
Missing Value Imputation
String Indexing
One-hot Encoding
Standard Scaling
Model Training
7
Predictive Model
Deployment Platform
2 Predictions
4 Feedback
Ad 1
Ad 2
Ad 3
Ad 4
3 Ads
Ads Selector