Lecture #21

Carnegie Mellon University

Special Topics:

Self-Driving Database Management Systems

Autonomous Systems IV

@Ying_Jiang // 15-799 // Spring 2022

LAST CLASS

openGauss

→ Learned Optimizer, Learned Advisor, Model Validator…

→ (Partly) adopted by real customers in China

Verdict: "Database Learning"

→ Enhances the Approximate Query Processing engine

→ Learns from past query answers to get smarter

TODAY'S CLASS

A Unified Transferable Model for ML-Enhanced DBMS

Solving DBMS tasks using machine learning approaches.
Learned cardinality estimator,
Learned cost estimator,
Learned index, etc.

Unified: one model for multiple DBMS tasks.
Transferable: one model transferable across various datasets.

Source: Ziniu Wu

TODAY'S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts

TODAY'S AGENDA

Overview

Multi-tasking meta-learning framework

Query Optimizer: A Case Study

Experiments

Parting Thoughts

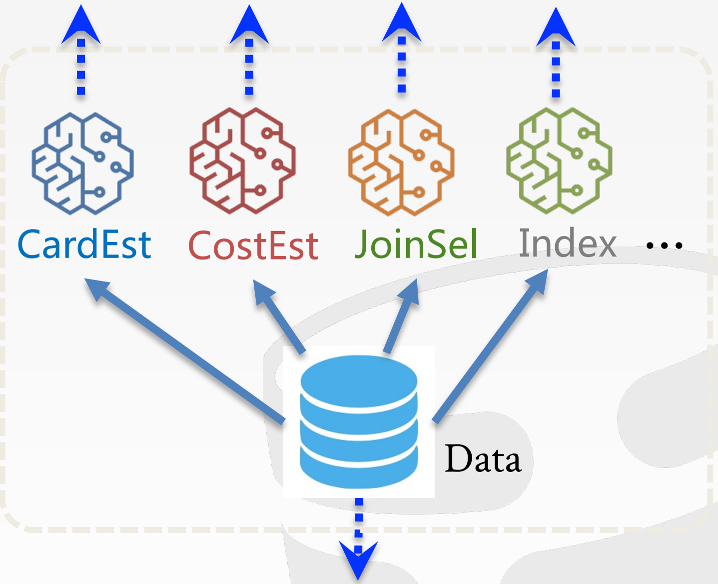
EXISTING ARCHITECTURES

One model for one task / one dataset (one DB)

Training from scratch is expensive
→ Requires huge amount of executed queries
→ Cold-start: not transferrable across DBs

Ignore relations between tasks
→ Redundant learning of common knowledge
→ Neither efficient nor effective

Each ML model serves a specific task

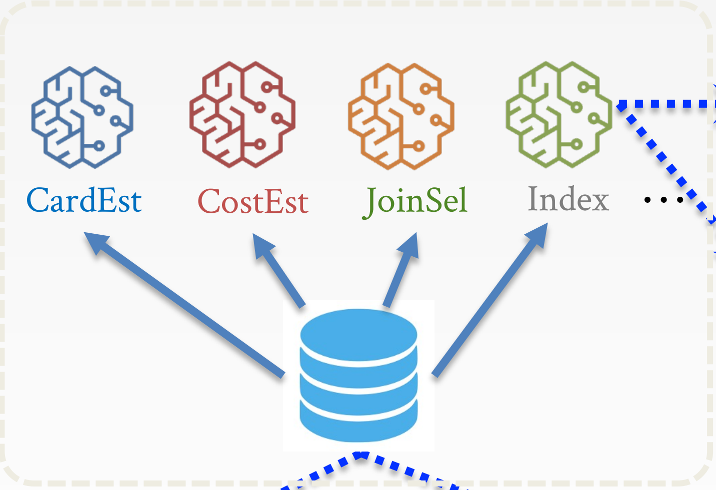


Each ML model trained from scratch on each new dataset

Source: Ziniu Wu

MOTIVATION: TRANSFERABILITY?

Learned knowledge decomposition: data and task



Task specific knowledge: method to solve a specific task, e.g. join order selection rules

Task shared knowledge: data and query representations used in all tasks

Data-agnostic meta-knowledge: common rules in DBs e.g., expert experiences and multi-table join rules

Data-specific knowledge: e.g., data distributions and join schema

Source: Ziniu Wu

OVERVIEW

Distill, share and reuse.

- → Identify and classify transferable and non-transferable knowledge
- → MTMLF, the mighty “multi-tasking meta-learning framework”
  - Multitask training procedure: Train different tasks simultaneously
  - Meta-learning paradigm: Pretrain + finetune to adapt quickly to new DB
- → Case study: a concrete model in Query Optimization
- → Experiments
- → Future research directions
  - More tasks incorporated!
  - Pre-train + finetune paradigm motivates federated learning

TODAY'S AGENDA

Overview

Multi-tasking meta-learning framework

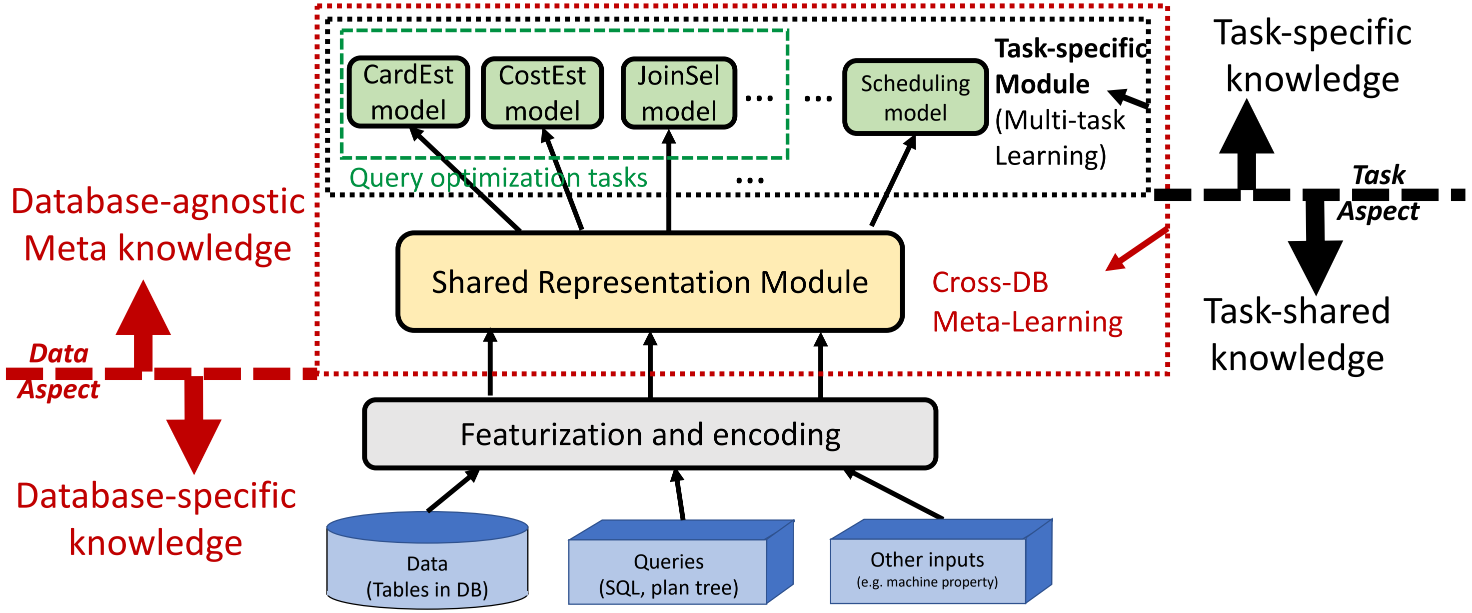
Query Optimizer: A Case Study

Experiments

Parting Thoughts

A UNIFIED MODEL

Detailed model architecture



Source: Ziniu Wu

AN ENVISIONED DB SERVICE

Service provider

→ Train on multiple anonymized users' DBs

→ Provide users the shared representation and task-specific modules

→ Periodically collect anonymized information from users to update the model as service upgrade

Users

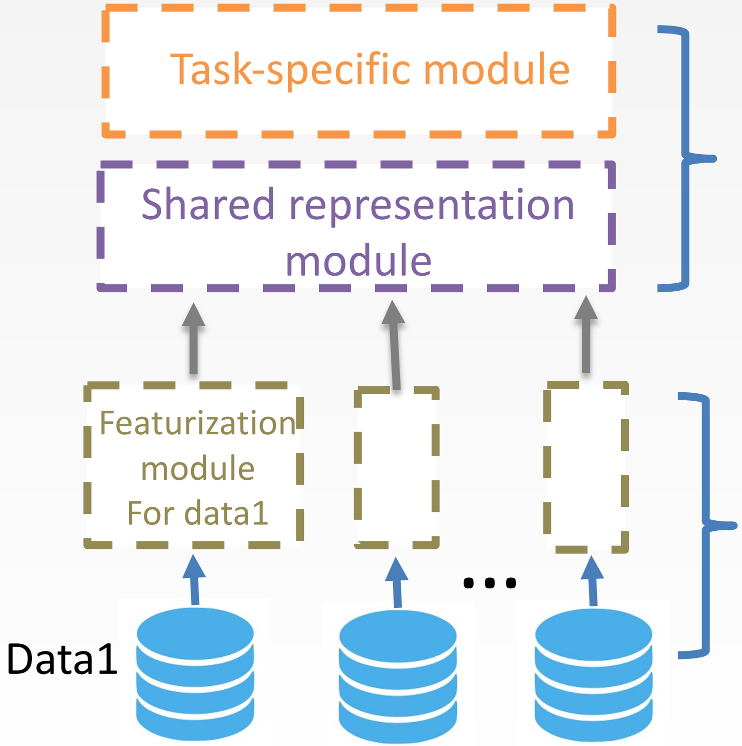
→ Train featurization module on data, plug in to the provided modules

→ Execute a small number of queries to fine-tune the entire model

→ Optionally provide anonymized information to the service provider

Source: Ziniu Wu

WORKFLOW



Service provider: provide pre-trained modules, fixed during training for various datasets

Users: train featurization module for their own DB

Source: Ziniu Wu

UNIFIED MODEL: ADVANTAGES

Architecture

→ More efficient without redundant learning

→ More effective task modeling

→ Transferability

DB service

→ Recyclable computation: pre-trained large models provided as a fundamental tool

→ Continuous optimization: see more (anonymized) data from users

→ Robust against data update and workload shift: user update F module

Source: Ziniu Wu

TODAY'S AGENDA

Overview

Multi-tasking meta-learning framework

Query Optimizer: A Case Study

Experiments

Parting Thoughts

QO: A CASE STUDY

Jointly learn multiple tasks in Query Optimization together

Technical contribution on join order selection task:

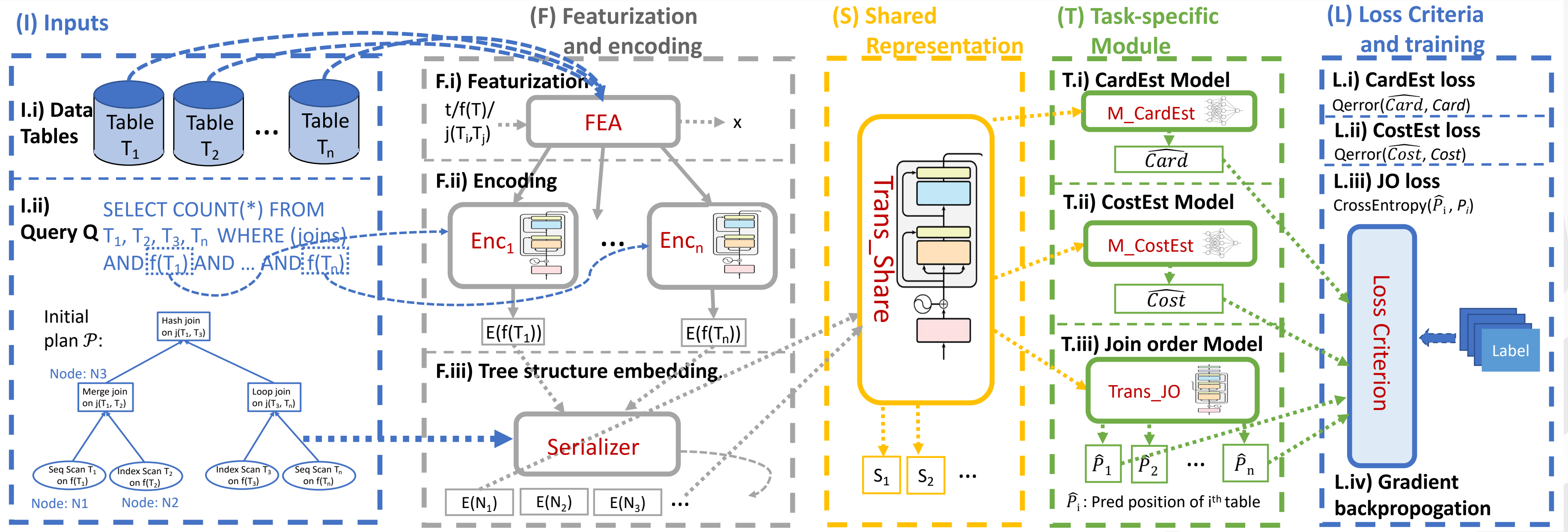
→ Details in [Wu et al. 2021]

1. Propose a new encoding method for a join order
2. Design a novel decoding method to ensure the legitimacy of output
3. New join order loss function to empower end-to-end training

Source: Ziniu Wu

QO: A CASE STUDY

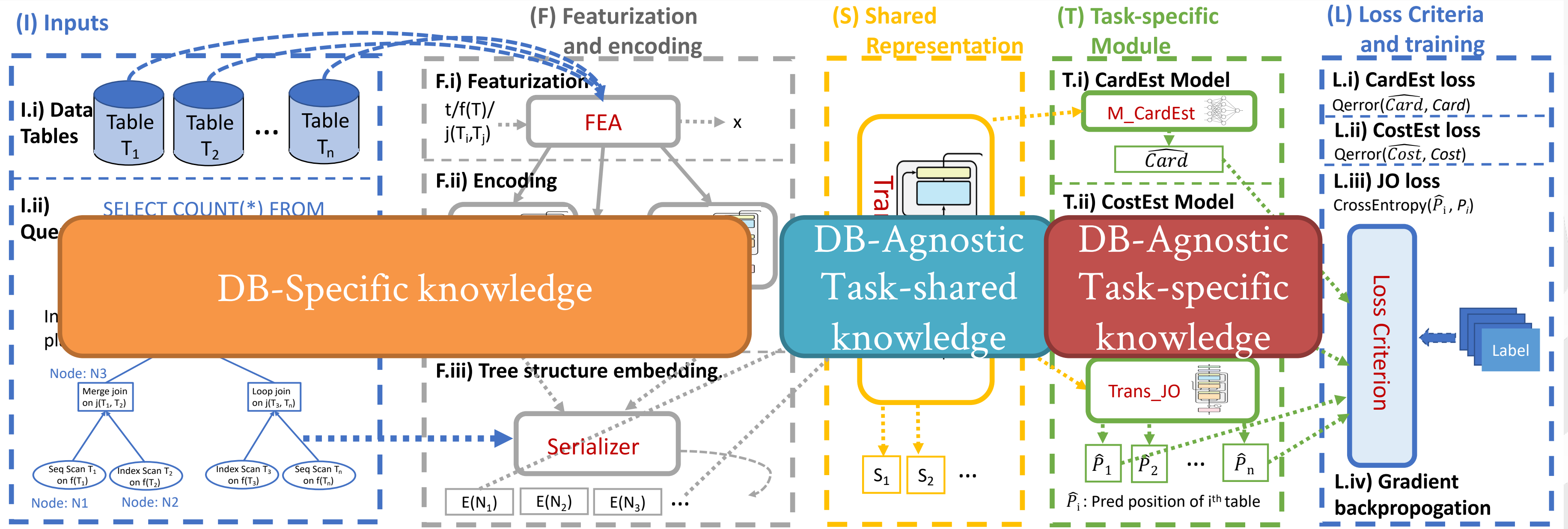
Jointly learn multiple tasks in QO together



Source: Ziniu Wu

QO: A CASE STUDY

Jointly learn multiple tasks in QO together



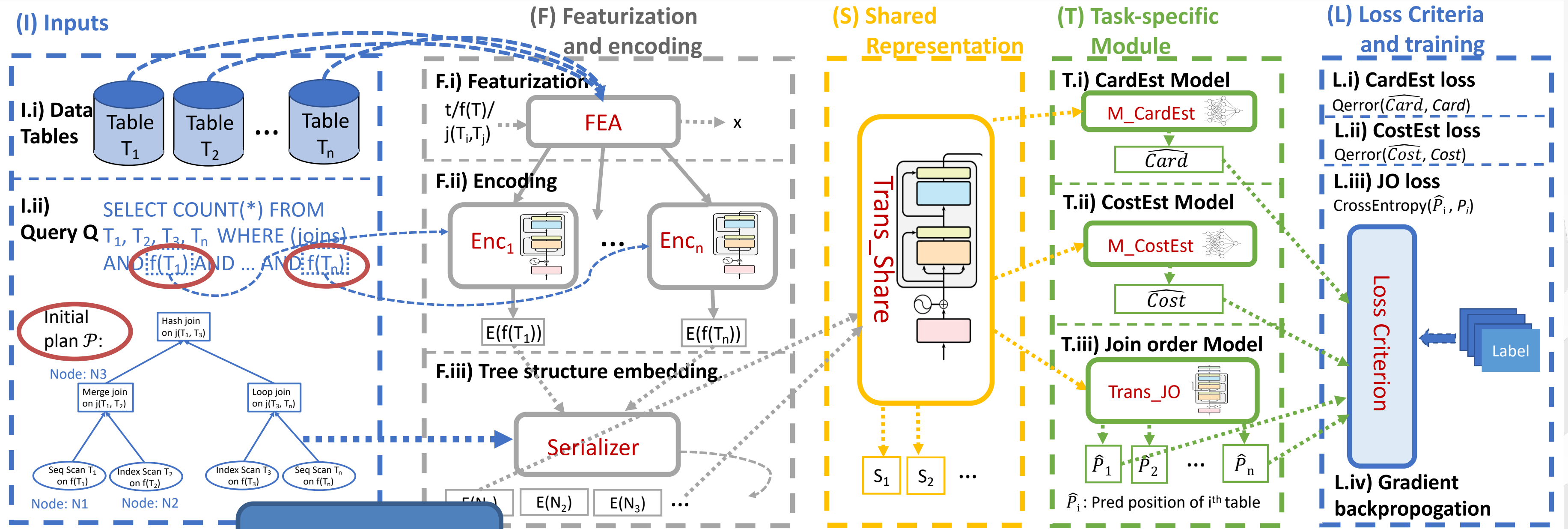
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

DB-Specific knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



f() in the leaves

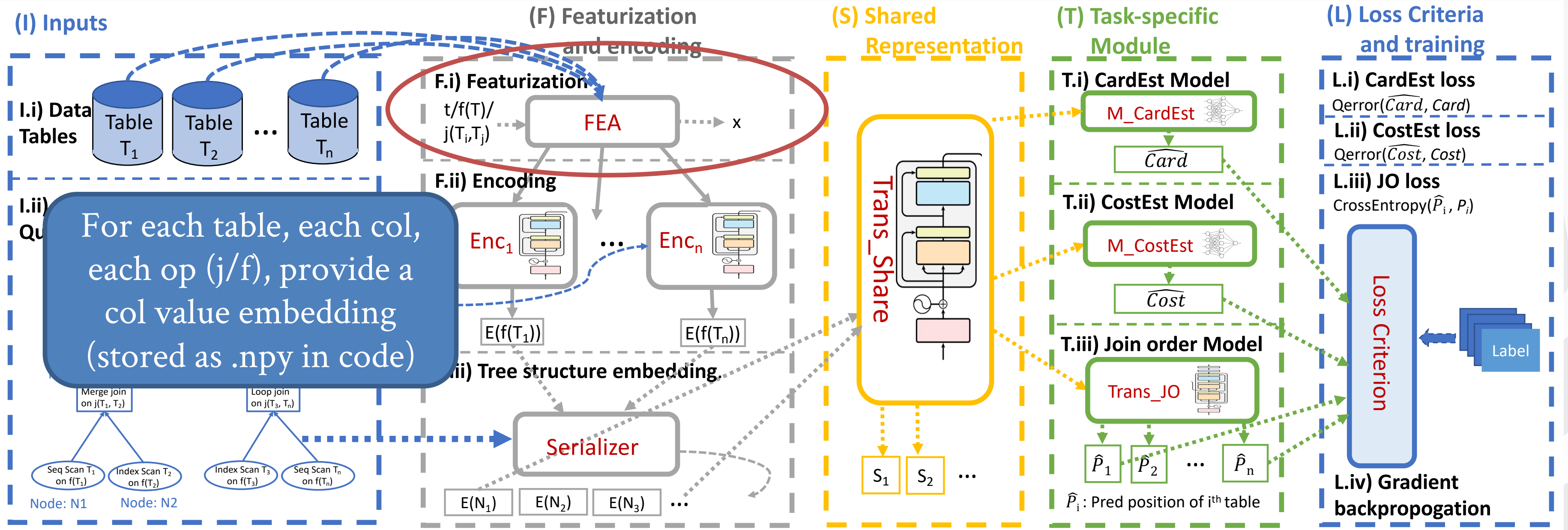
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

DB-Specific knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



For each table, each col, each op (j/f), provide a col value embedding (stored as .npy in code)

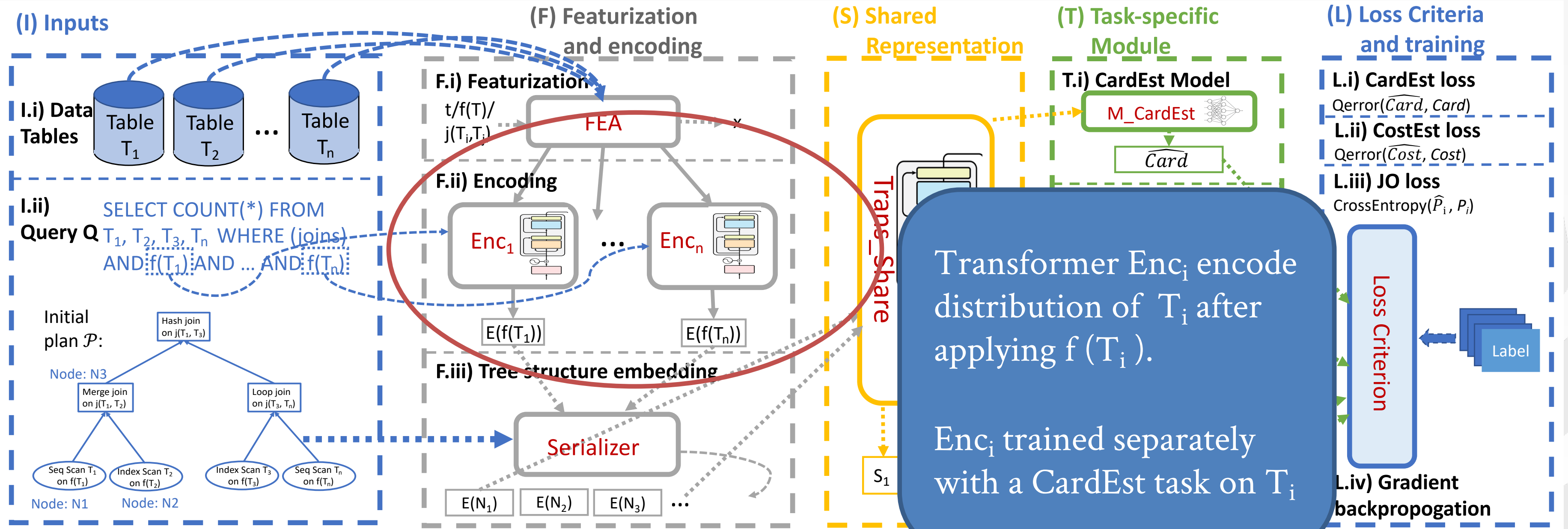
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

DB-Specific knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



Transformer Enc_i encode distribution of T_i after applying $f(T_i)$.

Enc_i trained separately with a CardEst task on T_i

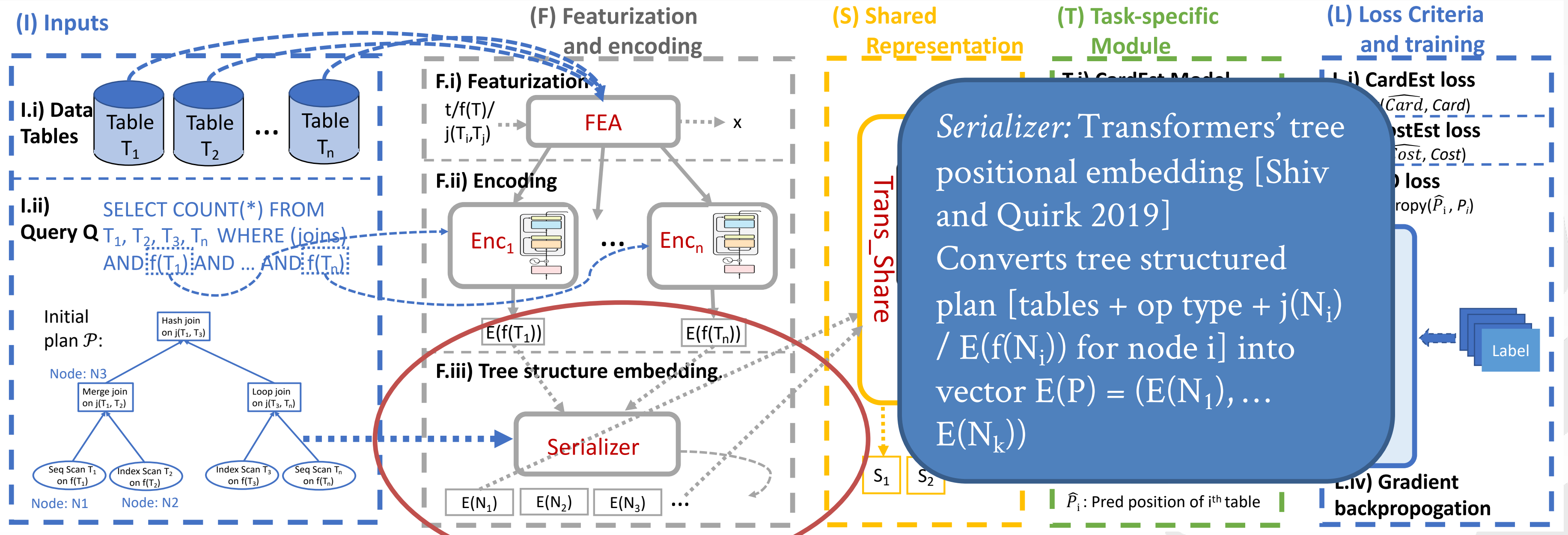
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED DBMS
CIDR 2022

DB-Specific knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



Serializer: Transformers' tree positional embedding [Shiv and Quirk 2019]
Converts tree structured plan [tables + op type + $j(N_i)$ / $E(f(N_i))$ for node i] into vector $E(P) = (E(N_1), \dots E(N_k))$

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

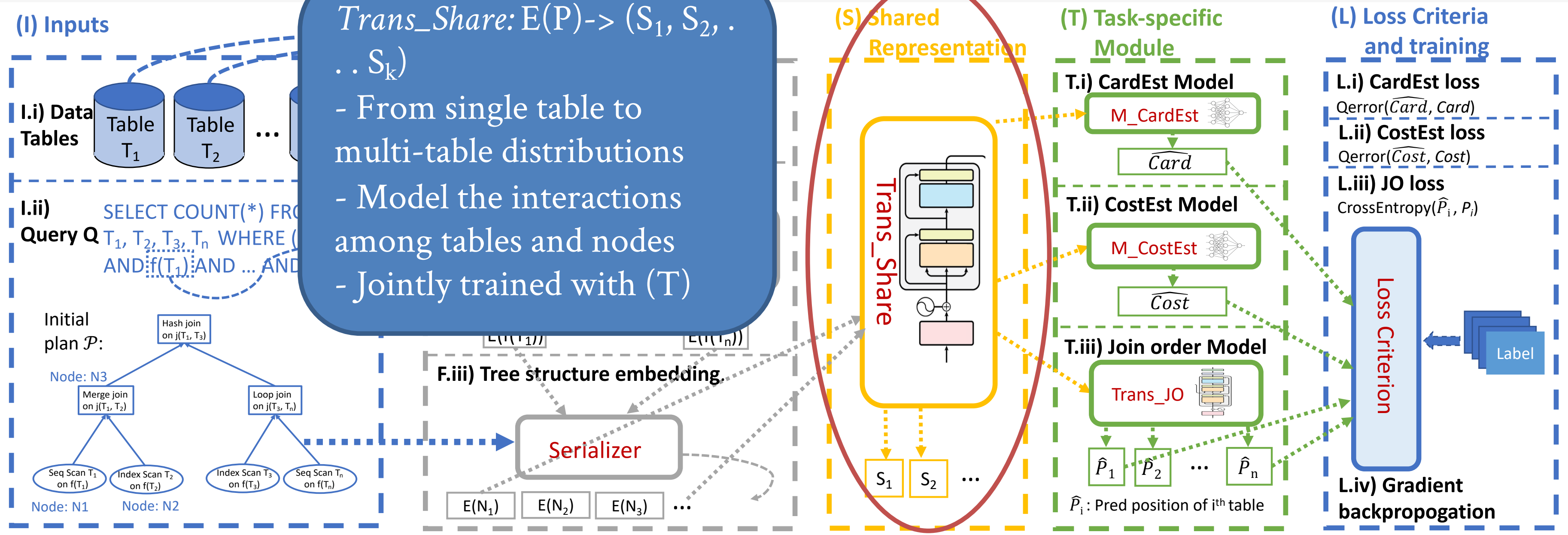
DB-Agnostic
Task-shared
knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together

Trans_Share: E(P)-> (S_1, S_2, . . . S_k)
- From single table to multi-table distributions
- Model the interactions among tables and nodes
- Jointly trained with (T)



A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

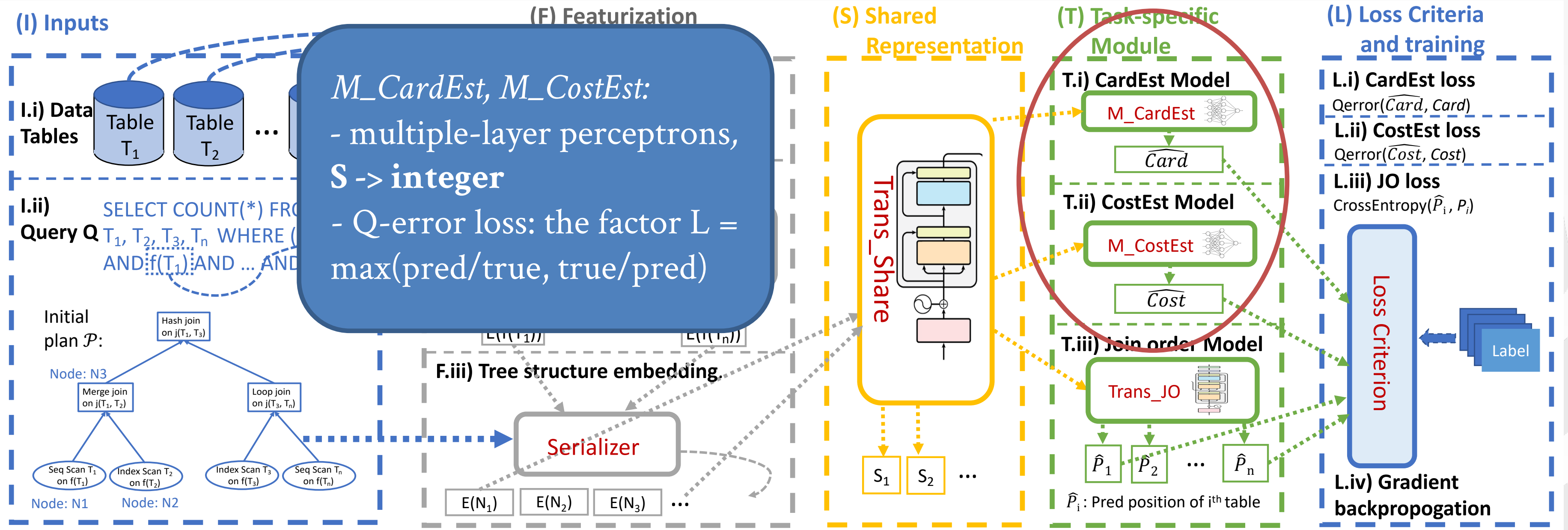
DB-Agnostic Task-specific knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together

M_CardEst, M_CostEst:
- multiple-layer perceptrons, **S -> integer**
- Q-error loss: the factor L = max(pred/true, true/pred)



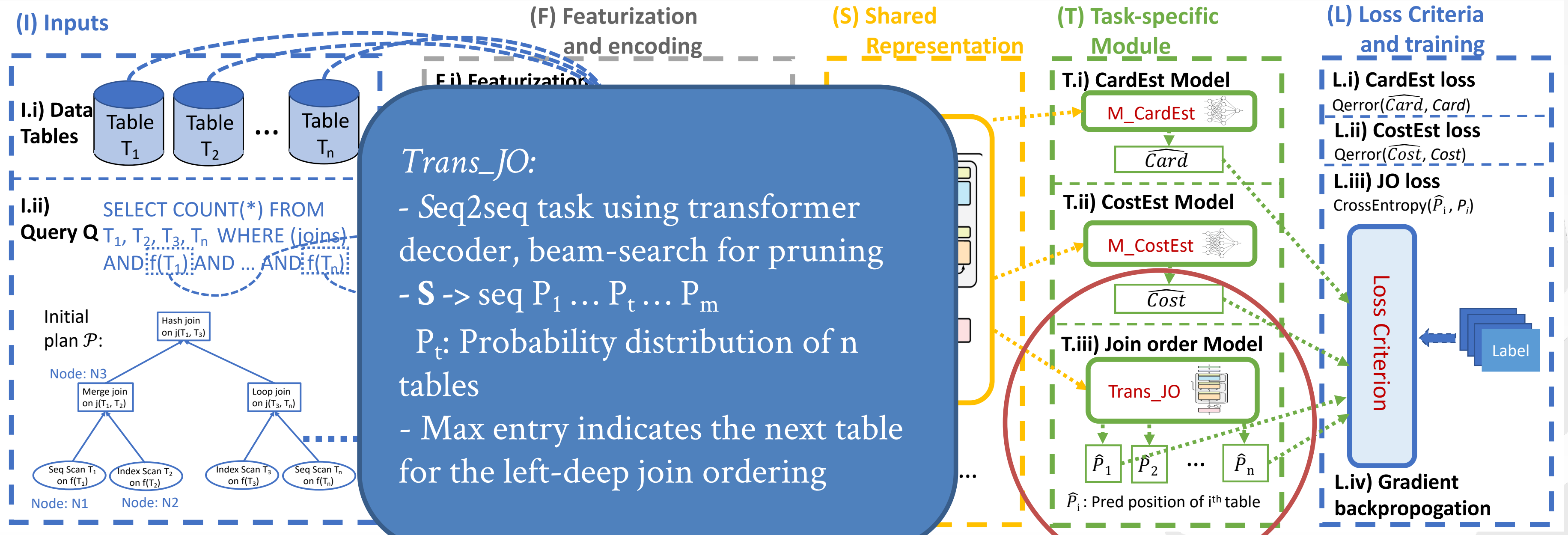
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

DB-Agnostic
Task-specific
knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



Trans_JO:

- *Seq2seq task using transformer decoder, beam-search for pruning*
- **S** -> seq $P_1 \ldots P_t \ldots P_m$

  P_t: Probability distribution of n tables
- Max entry indicates the next table for the left-deep join ordering

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED DBMS
CIDR 2022

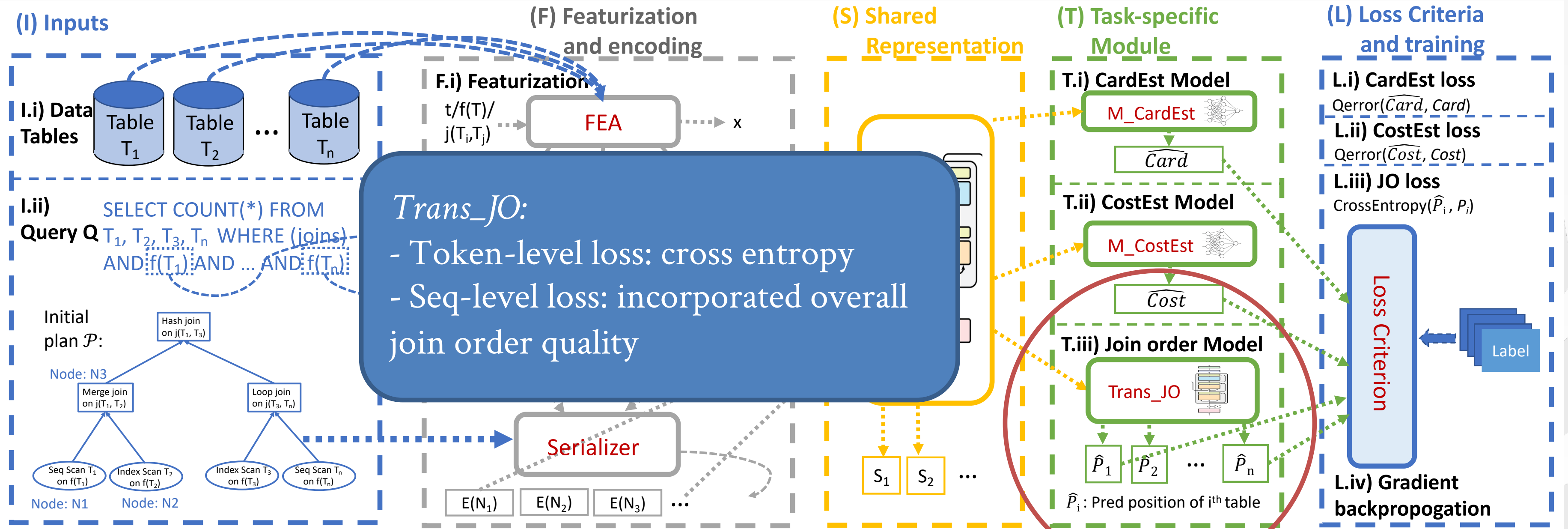
DB-Agnostic
Task-specific
knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together

Trans_JO:
- Token-level loss: cross entropy
- Seq-level loss: incorporated overall join order quality



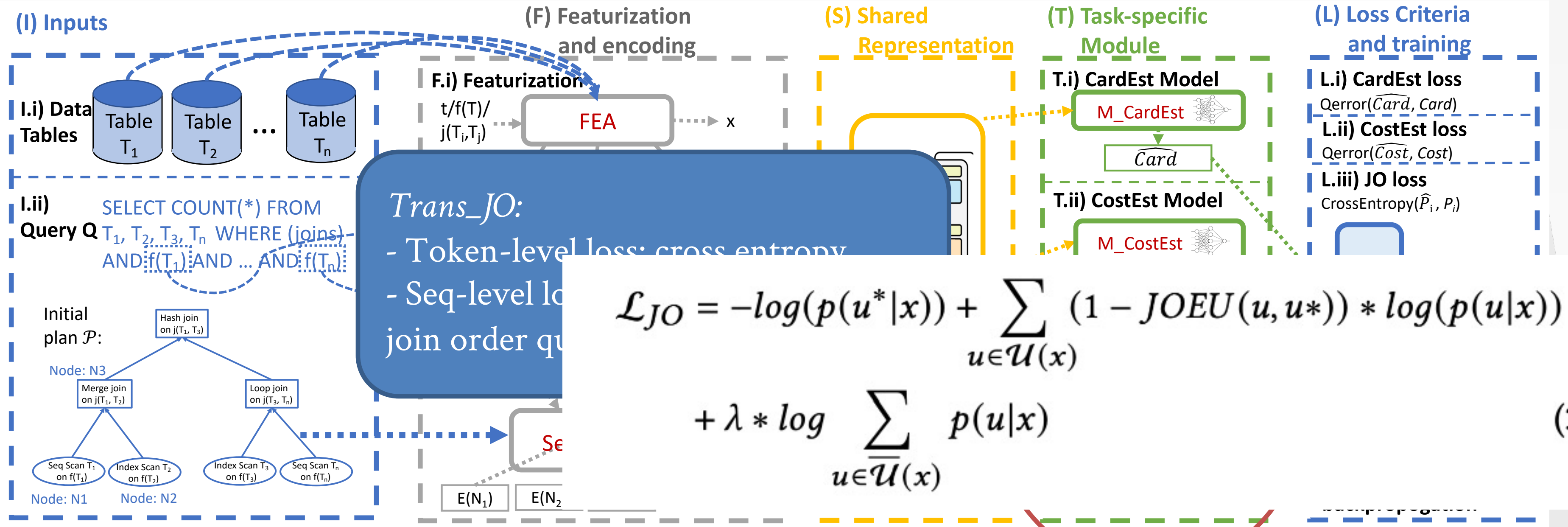
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

DB-Agnostic
Task-specific
knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



Trans_JO:
- Token-level loss: cross entropy
- Seq-level lo
join order qu

$$\mathcal{L}_{JO} = -log(p(u^*|x)) + \sum_{u \in \mathcal{U}(x)} (1 - JOEU(u, u*)) * log(p(u|x))$$

$$+ \lambda * log \sum_{u \in \overline{\mathcal{U}}(x)} p(u|x) \quad (3$$

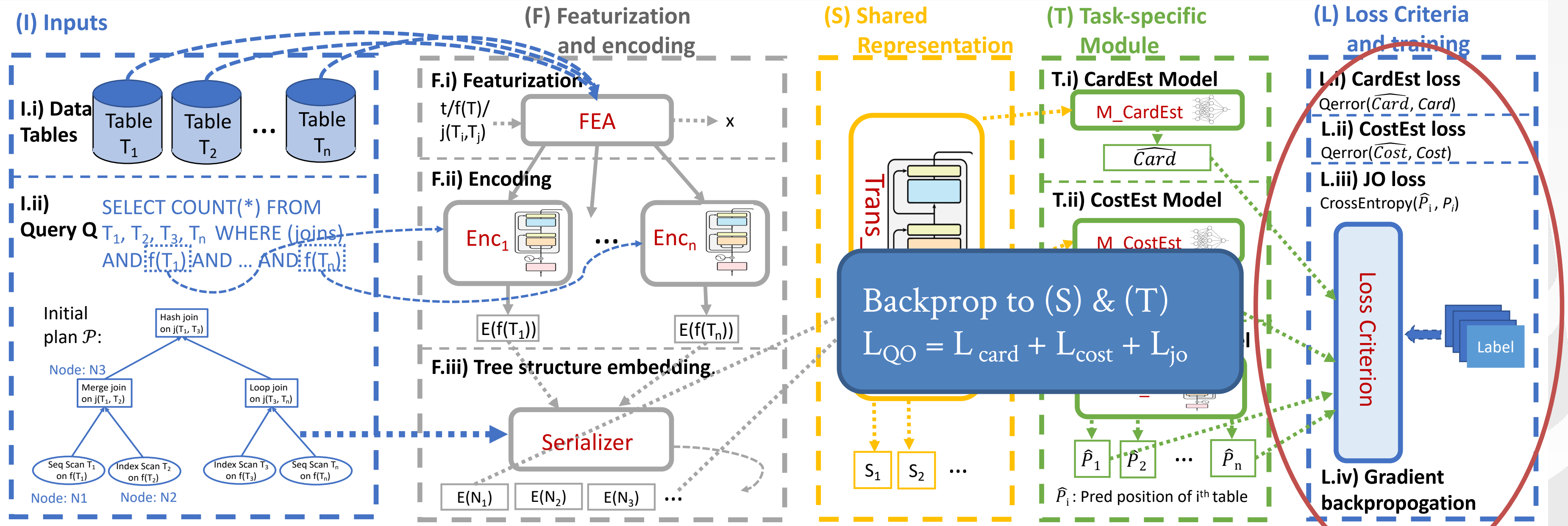
A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

DB-Agnostic
Task-specific
knowledge

QO: A CASE STUDY

Notation:
k nodes
m timesteps / tables touched
n total tables

Jointly learn multiple tasks in QO together



Backprop to (S) & (T)

$$L_{QO} = L_{card} + L_{cost} + L_{jo}$$

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED DBMS
CIDR 2022

TODAY'S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts

QO: Experiments

Experimental objective:

1. Show effectiveness on multi-task training.
2. Show the transferability to unseen DBs.

Experimental setups:

- Evaluation of #1 & #2: single dataset
  IMDB single dataset, Join order benchmark (21 tables, 113 queries)
- Evaluation of #3:
  Artificially generated 11 DBs with workloads using various distribution and table join schemas

Source: Ziniu Wu

EFFECTIVENESS ON A SINGLE DATASET

Accurate performance

- Q-Error = max(Predict/True, True/Predict)
- Better than Tree-LSTM
- Comparison with other state-of-the-art models left as future work

Method	Cardinality Q-Error			Cost Q-Error		
	median	max	mean	median	max	mean
PostgreSQL	184	670,000	10,416	4.90	4,920	105
Tree-LSTM	8.78	696.29	36.83	4.00	290.35	15.01
Ours	**4.48**	**614.45**	**28.69**	**2.10**	**37.54**	**4.20**

Source: Ziniu Wu

EFFECTIVENESS ON A SINGLE DATASET

Multi-task learning > separate learning

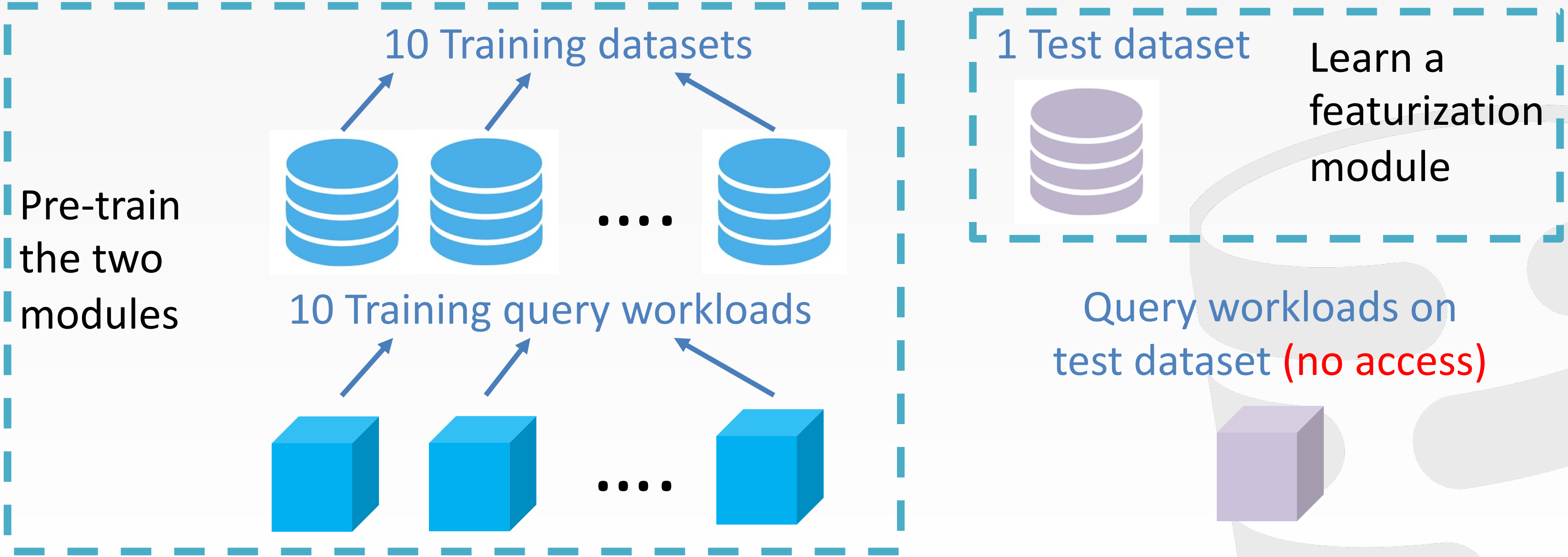
- 3.5x speed-up over PostgreSQL baseline
- Comparison with the state-of-the-art models left as future work

Method	JoinSel	
	Execution Time (min)	Improvement Ratio
PostgreSQL	1143.2	---
Optimal	209.1	81.7%
Multi-task	318.3	72.2%
One-task	450.4	60.6%

Source: Ziniu Wu

TRANSFERABILITY ACROSS DATASETS

Train the S and T modules on 10 datasets, plug in the F module from the test dataset without access to its query workload.



Source: Ziniu Wu

TRANSFERABILITY ACROSS DATASETS

Single: Trained from scratch on only the test dataset with access to the query workload on test dataset.

1 Test dataset

Learn a featurization module

Query workloads on test dataset (with access)

Train the two modules from scratch

Method	Execution Time (min)	Improvement Ratio
PostgreSQL	393.3	---
Transferable	234.1	40.6%
Single	219.5	44.3%

Source: Ziniu Wu

CONCLUSION AND FUTURE WORK

A vision for a new form of ML-enhanced DBMS

→ More efficient and effective for multiple DB tasks

→ Pre-trained model transferable to new datasets

Future Work

→ Explore multi-task learning on other DBMS tasks

→ Design details and implement the new DB service

Source: Ziniu Wu

TODAY'S AGENDA

Overview

Solving Constrained Optimization

Boosting Tuning Process with Meta-Learning

Evaluation

Parting Thoughts

PARTING THOUGHTS

Revolutionary: classifying knowledge of DB

- Splitting stages of the unified model in a conceptually reasonable way, and show its effectiveness

Transferability between multi-tasks & multi-DB

Practical?

- A single gigantic model and massive data
  - One model for all?
  - Benefit coming from a large model
- Frequency to update components?

NEXT CLASS

Autonomous Systems V