Transformer-based Tabular Modeling and Transfer Learning Applications

2025.12.

조용수 (jys1537@korea.ac.kr)

발표자 소개

❖ **조용수 (Yongsu Jo)**
- 고려대학교 산업경영공학과 석사과정(2024.03 ~)
- Data Mining & Quality Analytics Labs. (김성범 교수님)

❖ **관심 연구 분야**
- Supervised Learning
- Tabular Data

❖ **E-Mail**
- jys1537@korea.ac.kr

Seminar @ DMQA

종료

Transformer-based Bayesian Inference for Tabular Data

Transformer-based Bayesian Inference fo

발표자: 조용수

2025년 7월 11일

오전 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

Tabular Data Generation
: Practical Challenges & Foundational Approaches

Tabular Data Generation: Practical Challe

발표자: 윤지현

2025년 5월 23일

오전 9시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

Advancing Tabular Data Analysis

Advancing Tabular Data Analysis

발표자: 조용수

2024년 12월 6일

오전 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

Diffusion Models for Tabular Data

Diffusion Models for Tabular Data

발표자: 윤지현

2024년 10월 18일

오전 9시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

What is Next for Tabular Data?
Exploring Advances in Self-Supervised Learning

What is Next for Tabular Data? Exploring /

발표자: 채고은

2024년 4월 5일

오후 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

elf/Semi-Supervised Learning for
Tabular Data

2022.10.14

Byeongeun Ko

Self/Semi-Supervised Learning for Tabul

발표자: 고병은

2022년 10월 14일

오후 1시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

Revisiting Deep Learning Models for Tabular Data

❖ **NeurIPS 2021 게재, 1367회 인용**

Revisiting Deep Learning Models for Tabular Data

Yury Gorishniy[*†‡] **Ivan Rubachev**[†♣] **Valentin Khrulkov**[†] **Artem Babenko**[†♣]

† Yandex, Russia
‡ Moscow Institute of Physics and Technology, Russia
♣ National Research University Higher School of Economics, Russia

**“Tabular Data에 대해 최신 DL 기반 모델이 기존 GBDT 모델을 실제로 능가하는지 확인하고
이를 통해 Tabular Data의 DL 모델의 현재 수준 확인 및 새로운 Baseline 제시 "**

Motivate

Benchmark for CV?

Benchmark for NLP?

Benchmark for Tabular?

????

Motivate

Tabular Data

Best Performance

TabNet

NODE

AutoInt

그래서 뭐가 좋은건데?

모델 만들었는 데.. 뭐랑 비교해야해요?
진짜 GBDT 보다 좋은 게 맞아요?

Motivate

Tabular Data

Best Performance

TabNet

NODE

AutoInt

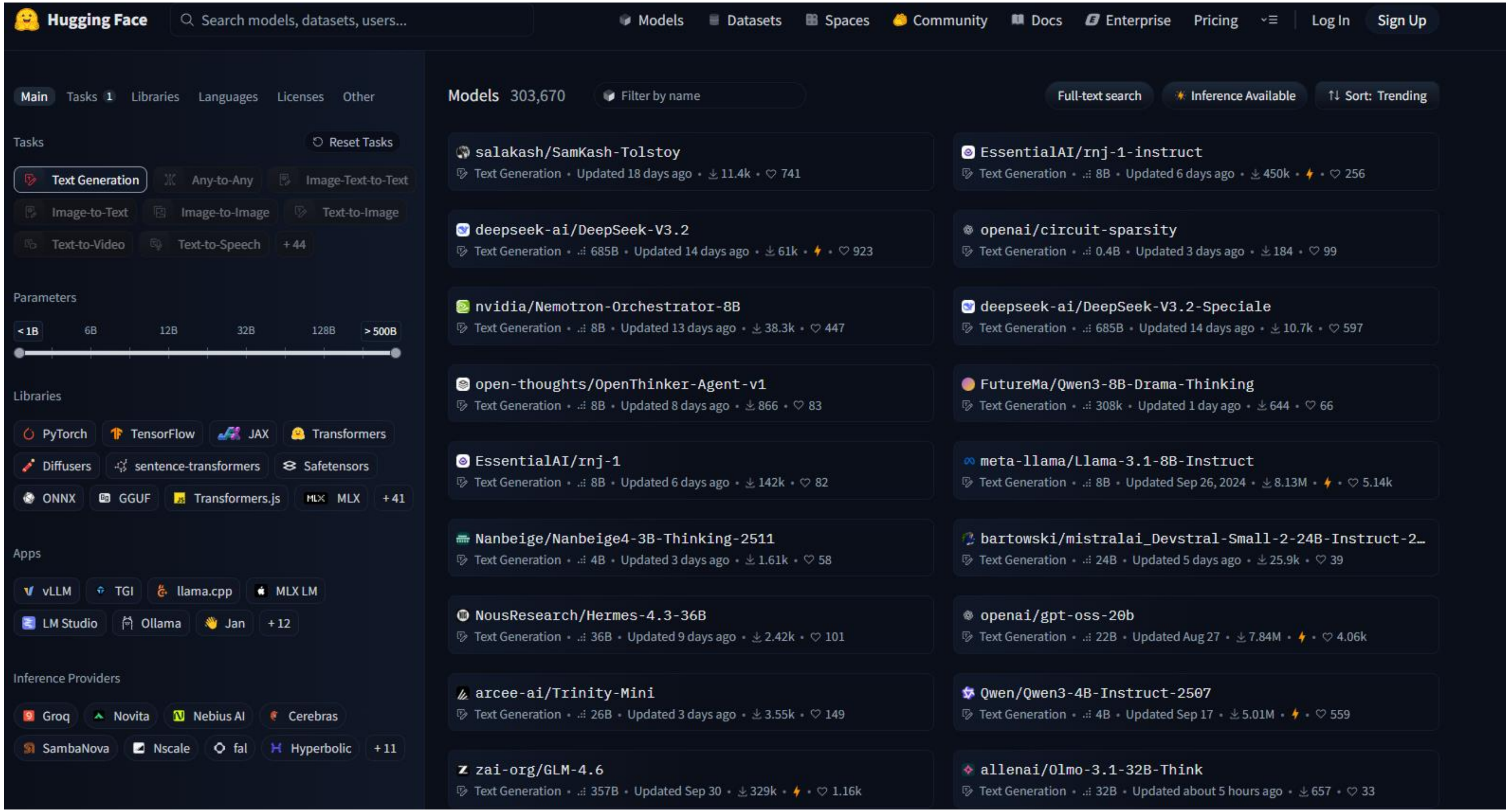
그래서 뭐가 좋은건데?

모델 만들었는 데.. 뭐랑 비교해야해요?

진짜 GBDT 보다 좋은 게 맞아요?

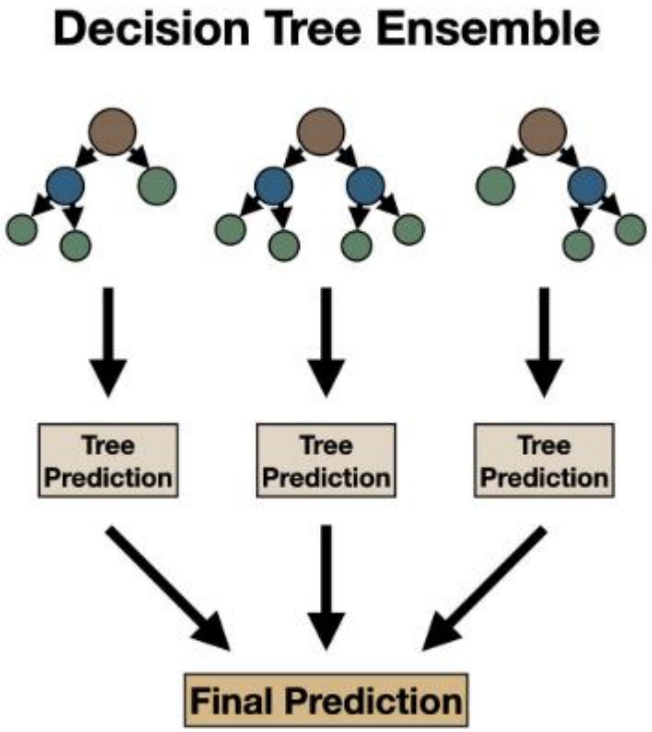
1. DL 기반의 모델들과 GBDT 모델과의 엄밀한 성능 비교
2. Tabular Data 연구에서 충분히 좋은 성능의 Baseline 모델 제시

State of the arts - CV / NLP



The “Shallow” state of the art

❖ **Tabular Data에서의 일반적인 선택은 여전히 GBDT**



미분 불가
→ Gradient 를 활용한 최적화 불가능
→ End to End 로 활용할 수 없음

The "Shallow" state of the art

❖ Tree 모델을 미분 가능하게 하면 End to End로 활용할 수 있지 않을까?

1. Differentiable trees

The Tree Ensemble Layer: Differentiability meets Conditional Computation

Hussein Hazimeh [1] Natalia Ponomareva [2] Petros Mol [2] Zhenyu Tan [3] Rahul Mazumder [1]

Deep Neural Decision Forests

Peter Kontschieder[1] Madalina Fiterau[*,2] Antonio Criminisi[1] Samuel Rota Bulò[1,3]

Microsoft Research[1] Cambridge, UK Carnegie Mellon University[2] Pittsburgh, PA Fondazione Bruno Kessler[3] Trento, Italy

NEURAL OBLIVIOUS DECISION ENSEMBLES FOR DEEP LEARNING ON TABULAR DATA

Sergei Popov
Yandex
sapopov@yandex-team.ru

Stanislav Morozov
Yandex
Lomonosov Moscow State University
stanis-morozov@yandex.ru

Artem Babenko
Yandex
National Research University
Higher School of Economics
artem.babenko@phystech.edu

Tree 내부의 결정 함수를 Smoothing 하여 Tree 전체와 Routing을 미분 가능하게 해보자

The “Shallow” state of the art

❖ 최신 Architecture인 Attention 구조를 사용하면 정형데이터에도 효과적이지 않을까?

1. Differentiable trees   **2. Attention-based Models**

AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks

Weiping Song*
Department of Computer Science,
School of EECS, Peking University
weiping.song@pku.edu.cn

Chence Shi
Department of Computer Science,
School of EECS, Peking University
chenceshi@pku.edu.cn

Zhiping Xiao
Department of Computer Science,
University of California, Los Angeles
patriciaxiao@g.ucla.edu

Zhijian Duan, Yewen Xu
Department of Computer Science,
School of EECS, Peking University
{zjduan,xuyewen}@pku.edu.cn

Ming Zhang†
Department of Computer Science,
School of EECS, Peking University
mzhang_cs@pku.edu.cn

Jian Tang†
Mila-Quebec AI Institute,
HEC Montreal & CIFAR AI Chair
jian.tang@hec.ca

TabTransformer: Tabular Data Modeling Using Contextual Embeddings

Xin Huang,[1] Ashish Khetan,[1] Milan Cvitkovic[2] Zohar Karnin[1]
[1] Amazon AWS
[2] PostEra
xinxh@amazon.com, khetan@amazon.com, mwcvitkovic@gmail.com, zkarnin@amazon.com

TabNet: Attentive Interpretable Tabular Learning

Sercan Ö. Arık, Tomas Pfister
Google Cloud AI
Sunnyvale, CA
soarik@google.com, tpfister@google.com

다양한 도메인에서 Attention 기반 모델이 좋으니 정형데이터에도 활용해보자

The “Shallow” state of the art

❖ Tree 구조의 장점이 피쳐간 상호작용의 활용이니 DL 구조에서도 사용하면 성능이 올라가지 않을까?

1. Differentiable trees | 2. Attention-based Models | 3. Explicit modeling of multiplicative interactions

Latent Cross: Making Use of Context in Recurrent Recommender Systems

Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li*, Vince Gatto, Ed H. Chi
Google, Inc.
Mountain View, California
{alexbeutel, pcovington, sagarj, canxu, vgatto, edchi}@google.com, vena900620@gmail.com

Deep & Cross Network for Ad Click Predictions

Ruoxi Wang
Stanford University
Stanford, CA
ruoxi@stanford.edu

Bin Fu
Google Inc.
New York, NY
binfu@google.com

Gang Fu
Google Inc.
New York, NY
thomasfu@google.com

Mingliang Wang
Google Inc.
New York, NY
mlwang@google.com

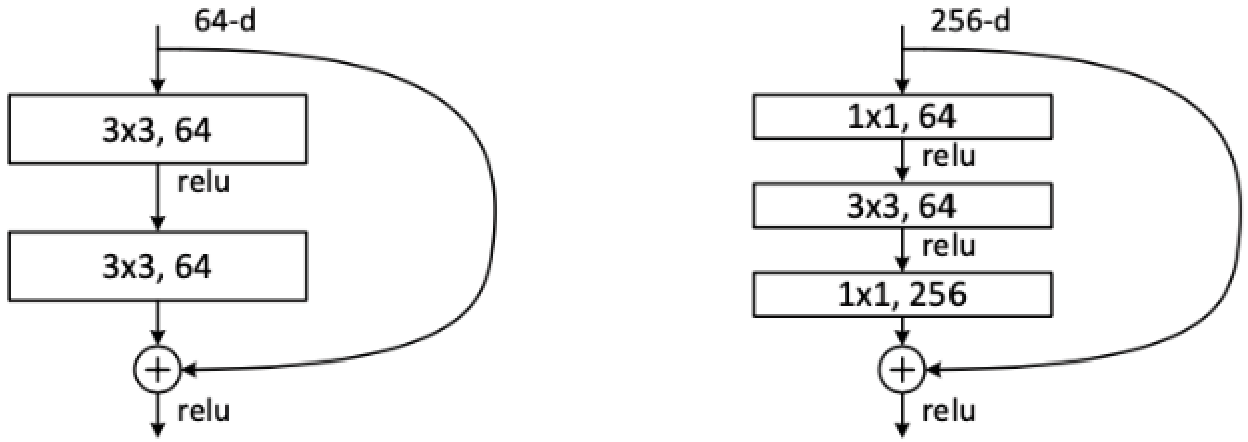
ARE NEURAL RANKERS STILL OUTPERFORMED BY GRADIENT BOOSTED DECISION TREES?

Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, Marc Najork
Google Research
{zhenqin,lyyanle,hlz,yitay,ramakumar,xuanhui,bemike,najork}@google.com

Tree 모델처럼 Feature 간 조합을 DL 구조 안에 직접 통합해 보자

Proposed baseline – ResNet like

ResNet



ResNet-like

$$\begin{aligned}
\texttt{ResNet}(x) &= \texttt{Prediction}\left(\texttt{ResNetBlock}\left(\ldots\left(\texttt{ResNetBlock}\left(\texttt{Linear}(x)\right)\right)\right)\right) \\
\texttt{ResNetBlock}(x) &= x + \texttt{Dropout}(\texttt{Linear}(\texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(\texttt{BatchNorm}(x)))))) \\
\texttt{Prediction}(x) &= \texttt{Linear}\left(\texttt{ReLU}\left(\texttt{BatchNorm}\left(x\right)\right)\right)
\end{aligned}$$

FT-Transformer

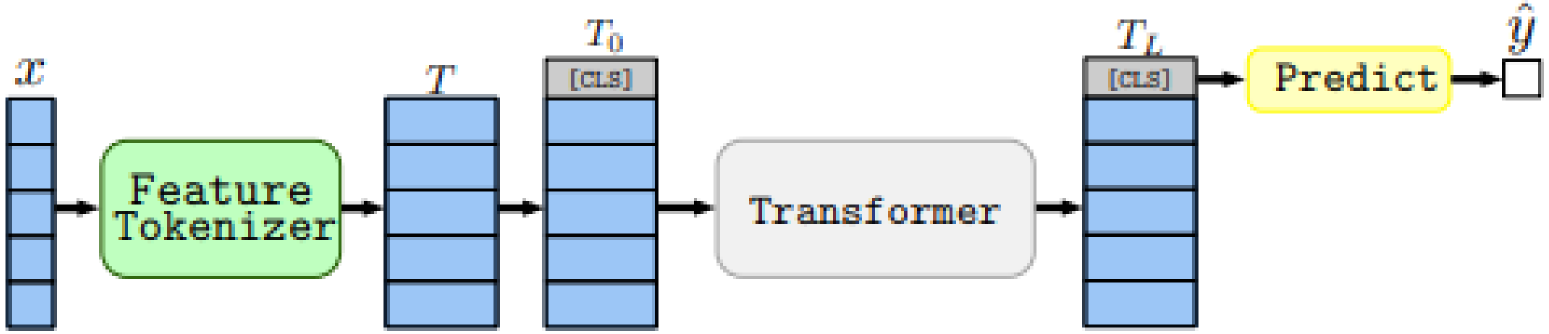


Figure 1: The FT-Transformer architecture. Firstly, Feature Tokenizer transforms features to embeddings. The embeddings are then processed by the Transformer module and the final representation of the [CLS] token is used for prediction.

FT-Transformer

❖ **FT-Transformer (Feature tokenizer + Transformer)**

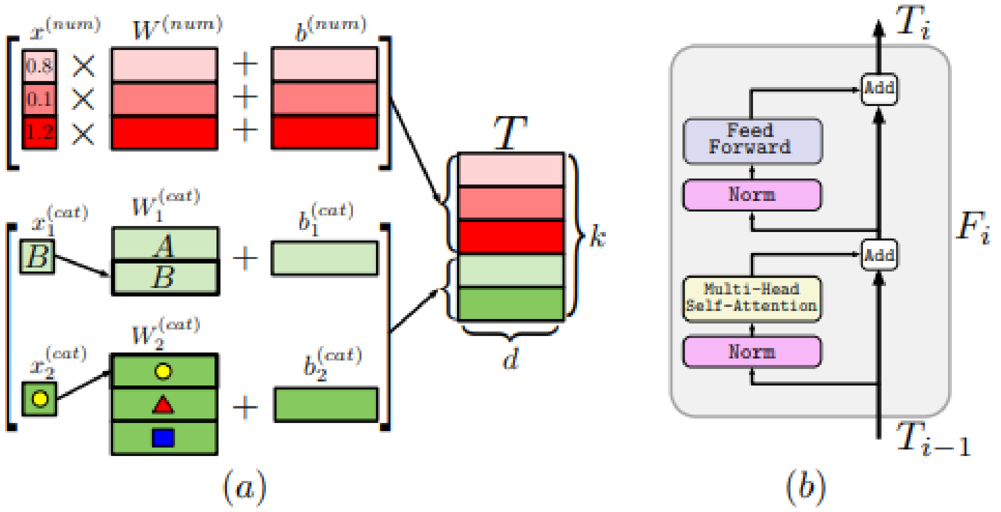


Figure 2: (a) Feature Tokenizer; in the example, there are three numerical and two categorical features; (b) One Transformer layer.

$$T_j = b_j + f_j(x_j) \in \mathbb{R}^d \qquad f_j : \mathbb{X}_j \to \mathbb{R}^d.$$

Feature Tokenizer (Numerical)

$$\boxed{T_j^{(num)} = b_j^{(num)} + x_j^{(num)} \cdot W_j^{(num)} \qquad \in \mathbb{R}^d,}$$

$$T_j^{(cat)} = b_j^{(cat)} + e_j^T W_j^{(cat)} \qquad \in \mathbb{R}^d,$$

$$T = \texttt{stack}\left[T_1^{(num)}, \ldots, T_{k^{(num)}}^{(num)}, T_1^{(cat)}, \ldots, T_{k^{(cat)}}^{(cat)}\right] \in \mathbb{R}^{k \times d}.$$

30
20.5
32

→ $d - dim$

0.1	**-0.2**	**0.3**

(Learnable)

$+ \; b$ →

3+b	-6+b	9+b
2.05+b	-4.1+b	6.15+b
3.2+b	-6.4+b	9.6+b

Feature Tokenizer (Categorical)

$$T_j^{(num)} = b_j^{(num)} + x_j^{(num)} \cdot W_j^{(num)} \quad \in \mathbb{R}^d,$$

$$\boxed{T_j^{(cat)} = b_j^{(cat)} + e_j^T W_j^{(cat)} \quad \in \mathbb{R}^d,}$$

$$T = \mathtt{stack}\left[T_1^{(num)},\ \ldots,\ T_{k^{(num)}}^{(num)},\ T_1^{(cat)},\ \ldots,\ T_{k^{(cat)}}^{(cat)}\right] \in \mathbb{R}^{k \times d}.$$

사과
바나나
고양이

Label encoding

0
1
2

Lookup table

0	0.1 -0.2 0.3
1	0.2 -0.1 0
2	0.4 0.5 0.6

사과	0.1 -0.2 0.3
바나나	0.2 -0.1 0
고양이	0.4 0.5 0.6

Lookup table에 저장된 Vector
(Learnable)

FT-Transformer

❖ FT-Transformer (Feature tokenizer + Transformer)

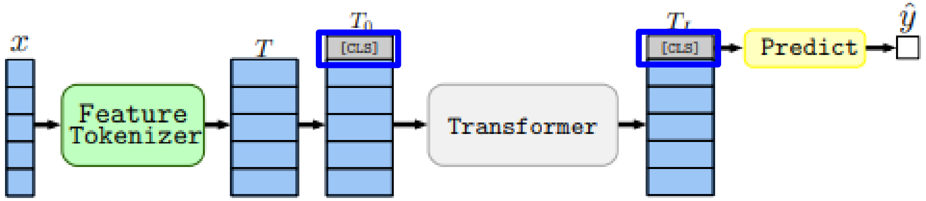


Figure 1: The FT-Transformer architecture. Firstly, Feature Tokenizer transforms features to embeddings. The embeddings are then processed by the Transformer module and the final representation of the [CLS] token is used for prediction.

최종 Task는 CLS 토큰 활용

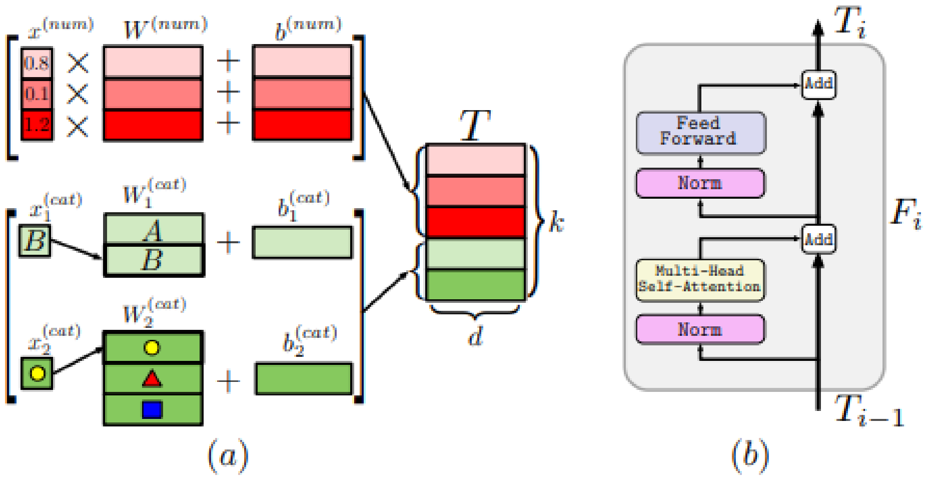


Figure 2: (a) Feature Tokenizer; in the example, there are three numerical and two categorical features; (b) One Transformer layer.

FT-Transformer

❖ FT-Transformer (Feature tokenizer + Transformer)

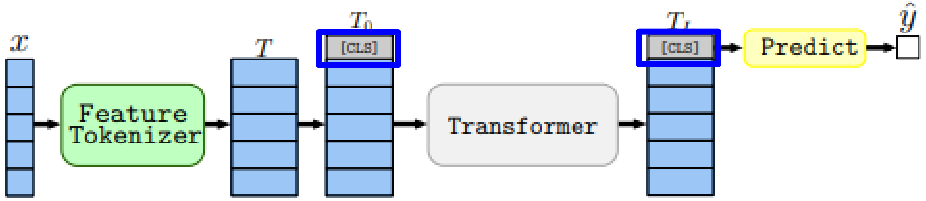


Figure 1: The FT-Transformer architecture. Firstly, Feature Tokenizer transforms features to embeddings. The embeddings are then processed by the Transformer module and the final representation of the [CLS] token is used for prediction.

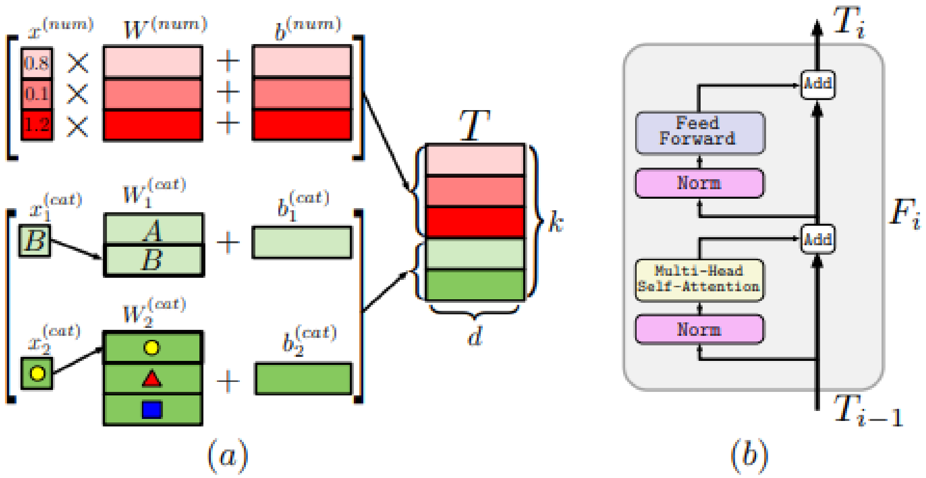


Figure 2: (a) Feature Tokenizer; in the example, there are three numerical and two categorical features; (b) One Transformer layer.

	키	몸무게	나이
A	100	30	10
B	150	50	20
C	160	60	23

=

	키	몸무게	나이
C	160	60	23
A	100	30	10
B	150	50	20

So What?

Positional encoding 없음

Experiment

❖ Comparing DL models

Table 2: Results for DL models. The metric values averaged over 15 random seeds are reported. See supplementary for standard deviations. For each dataset, top results are in **bold**. "Top" means "the gap between this result and the result with the best score is not statistically significant". For each dataset, ranks are calculated by sorting the reported scores; the "rank" column reports the average rank across all datasets. Notation: FT-T ~ FT-Transformer, ↓ ~ RMSE, ↑ ~ accuracy

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓	rank (std)
TabNet	0.510	0.850	0.378	0.723	0.719	0.954	0.8896	8.909	0.957	0.823	0.751	7.5 (2.0)
SNN	0.493	0.854	0.373	0.719	0.722	0.954	0.8975	8.895	0.961	0.761	0.751	6.4 (1.4)
AutoInt	0.474	**0.859**	0.372	0.721	0.725	0.945	0.8949	8.882	0.934	0.768	0.750	5.7 (2.3)
GrowNet	0.487	**0.857**	–	–	0.722	–	0.8970	8.827	–	0.765	0.751	5.7 (2.2)
MLP	0.499	0.852	0.383	0.719	0.723	0.954	0.8977	8.853	0.962	0.757	0.747	4.8 (1.9)
DCN2	0.484	0.853	0.385	0.716	0.723	0.955	0.8977	8.890	0.965	0.757	0.749	4.7 (2.0)
NODE	0.464	**0.858**	0.359	0.727	0.726	0.918	0.8958	**8.784**	0.958	**0.753**	**0.745**	3.9 (2.8)
ResNet	0.486	0.854	**0.396**	0.728	0.727	**0.963**	0.8969	8.846	0.964	0.757	0.748	3.3 (1.8)
FT-T	**0.459**	**0.859**	0.391	**0.732**	**0.729**	0.960	**0.8982**	8.855	**0.970**	0.756	0.746	1.8 (1.2)

Experiment

❖ Comparing DL models

Table 2: Results for DL models. The metric values averaged over 15 random seeds are reported. See supplementary for standard deviations. For each dataset, top results are in **bold**. "Top" means "the gap between this result and the result with the best score is not statistically significant". For each dataset, ranks are calculated by sorting the reported scores; the "rank" column reports the average rank across all datasets. Notation: FT-T ~ FT-Transformer, ↓ ~ RMSE, ↑ ~ accuracy

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓	rank (std)
TabNet	0.510	0.850	0.378	0.723	0.719	0.954	0.8896	8.909	0.957	0.823	0.751	7.5 (2.0)
SNN	0.493	0.854	0.373	0.719	0.722	0.954	0.8975	8.895	0.961	0.761	0.751	6.4 (1.4)
AutoInt	0.474	**0.859**	0.372	0.721	0.725	0.945	0.8949	8.882	0.934	0.768	0.750	5.7 (2.3)
GrowNet	0.487	**0.857**	–	–	0.722	–	0.8970	8.827	–	0.765	0.751	5.7 (2.2)
MLP	0.499	0.852	0.383	0.719	0.723	0.954	0.8977	8.853	0.962	0.757	0.747	4.8 (1.9)
DCN2	0.484	0.853	0.385	0.716	0.723	0.955	0.8977	8.890	0.965	0.757	0.749	4.7 (2.0)
NODE	0.464	**0.858**	0.359	0.727	0.726	0.918	0.8958	**8.784**	0.958	**0.753**	**0.745**	3.9 (2.8)
ResNet	0.486	0.854	**0.396**	0.728	0.727	**0.963**	0.8969	8.846	0.964	0.757	0.748	3.3 (1.8)
FT-T	**0.459**	**0.859**	0.391	**0.732**	**0.729**	0.960	**0.8982**	8.855	**0.970**	0.756	0.746	1.8 (1.2)

단순 구조 but, 훌륭한 검증 기준

Experiment

❖ Comparing DL models

Table 2: Results for DL models. The metric values averaged over 15 random seeds are reported. See supplementary for standard deviations. For each dataset, top results are in **bold**. “Top” means “the gap between this result and the result with the best score is not statistically significant”. For each dataset, ranks are calculated by sorting the reported scores; the “rank” column reports the average rank across all datasets. Notation: FT-T ~ FT-Transformer, ↓ ~ RMSE, ↑ ~ accuracy

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓	rank (std)
TabNet	0.510	0.850	0.378	0.723	0.719	0.954	0.8896	8.909	0.957	0.823	0.751	7.5 (2.0)
SNN	0.493	0.854	0.373	0.719	0.722	0.954	0.8975	8.895	0.961	0.761	0.751	6.4 (1.4)
AutoInt	0.474	**0.859**	0.372	0.721	0.725	0.945	0.8949	8.882	0.934	0.768	0.750	5.7 (2.3)
GrowNet	0.487	**0.857**	–	–	0.722	–	0.8970	8.827	–	0.765	0.751	5.7 (2.2)
MLP	0.499	0.852	0.383	0.719	0.723	0.954	0.8977	8.853	0.962	0.757	0.747	4.8 (1.9)
DCN2	0.484	0.853	0.385	0.716	0.723	0.955	0.8977	8.890	0.965	0.757	0.749	4.7 (2.0)
NODE	0.464	**0.858**	0.359	0.727	0.726	0.918	0.8958	**8.784**	0.958	**0.753**	**0.745**	3.9 (2.8)
ResNet	0.486	0.854	**0.396**	0.728	0.727	**0.963**	0.8969	8.846	0.964	0.757	0.748	3.3 (1.8)
FT-T	**0.459**	**0.859**	0.391	**0.732**	**0.729**	0.960	**0.8982**	8.855	**0.970**	0.756	0.746	1.8 (1.2)

단순 구조 but, 훌륭한 검증 기준

효과적인 Baseline

Experiment

❖ Comparing DL models

Table 2: Results for DL models. The metric values averaged over 15 random seeds are reported. See supplementary for standard deviations. For each dataset, top results are in **bold**. "Top" means "the gap between this result and the result with the best score is not statistically significant". For each dataset, ranks are calculated by sorting the reported scores; the "rank" column reports the average rank across all datasets. Notation: FT-T ~ FT-Transformer, ↓ ~ RMSE, ↑ ~ accuracy

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓	rank (std)
TabNet	0.510	0.850	0.378	0.723	0.719	0.954	0.8896	8.909	0.957	0.823	0.751	7.5 (2.0)
SNN	0.493	0.854	0.373	0.719	0.722	0.954	0.8975	8.895	0.961	0.761	0.751	6.4 (1.4)
AutoInt	0.474	**0.859**	0.372	0.721	0.725	0.945	0.8949	8.882	0.934	0.768	0.750	5.7 (2.3)
GrowNet	0.487	**0.857**	–	–	0.722	–	0.8970	8.827	–	0.765	0.751	5.7 (2.2)
MLP	0.499	0.852	0.383	0.719	0.723	0.954	0.8977	8.853	0.962	0.757	0.747	4.8 (1.9)
DCN2	0.484	0.853	0.385	0.716	0.723	0.955	0.8977	8.890	0.965	0.757	0.749	4.7 (2.0)
NODE	0.464	**0.858**	0.359	0.727	0.726	0.918	0.8958	**8.784**	0.958	**0.753**	**0.745**	3.9 (2.8)
ResNet	0.486	0.854	**0.396**	0.728	0.727	**0.963**	0.8969	8.846	0.964	0.757	0.748	3.3 (1.8)
FT-T	**0.459**	**0.859**	0.391	**0.732**	**0.729**	0.960	**0.8982**	8.855	**0.970**	0.756	0.746	1.8 (1.2)

단순 구조 but, 훌륭한 검증 기준

효과적인 Baseline

대부분의 Task 에서 최고성능

Experiment

❖ **Comparing DL models**

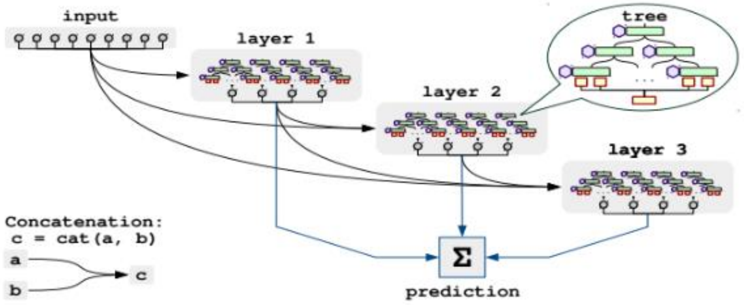


Figure 2: The NODE architecture, consisting of densely connected NODE layers. Each layer contains several trees whose outputs are concatenated and serve as input for the subsequent layer. The final prediction is obtained by averaging the outputs of all trees from all the layers.

Structure of NODE

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
NODE	0.461	0.860	0.361	0.730	0.727	0.921	0.8970	**8.716**	0.965	0.750	0.744
ResNet	0.478	0.857	0.398	0.734	0.731	0.966	0.8976	8.770	0.967	0.751	0.745
FT-Transformer	**0.448**	**0.860**	**0.398**	**0.739**	**0.731**	**0.967**	**0.8984**	8.751	**0.973**	**0.747**	**0.743**

FT Transformer / ResNet-like에서 Ensemble 적용결과

Experiment

❖ **Comparing DL models and GBDT**

Table 4: Results for ensembles of GBDT and the main DL models. For each model-dataset pair, the metric value averaged over three ensembles is reported. See supplementary for standard deviations. Notation follows Table 3.

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
Default hyperparameters											
XGBoost	0.462	**0.874**	0.348	0.711	0.717	0.924	0.8799	9.192	0.964	0.761	0.751
CatBoost	**0.428**	0.873	0.386	0.724	0.728	0.948	0.8893	8.885	0.910	0.749	0.744
FT-Transformer	0.454	0.860	**0.395**	**0.734**	**0.731**	**0.966**	**0.8969**	**8.727**	**0.973**	**0.747**	**0.742**
Tuned hyperparameters											
XGBoost	0.431	0.872	0.377	0.724	0.728	–	0.8861	8.819	0.969	**0.732**	0.742
CatBoost	**0.423**	**0.874**	0.388	0.727	0.729	–	0.8898	8.837	0.968	0.740	**0.741**
ResNet	0.478	0.857	0.398	0.734	0.731	0.966	0.8976	8.770	0.967	0.751	0.745
FT-Transformer	0.448	0.860	**0.398**	**0.739**	**0.731**	**0.967**	**0.8984**	**8.751**	**0.973**	0.747	0.743

Experiment

❖ **Comparing DL models and GBDT**

Table 4: Results for ensembles of GBDT and the main DL models. For each model-dataset pair, the metric value averaged over three ensembles is reported. See supplementary for standard deviations. Notation follows Table 3.

	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
Default hyperparameters											
XGBoost	0.462	**0.874**	0.348	0.711	0.717	0.924	0.8799	9.192	0.964	0.761	0.751
CatBoost	**0.428**	0.873	0.386	0.724	0.728	0.948	0.8893	8.885	0.910	0.749	0.744
FT-Transformer	0.454	0.860	**0.395**	**0.734**	**0.731**	**0.966**	**0.8969**	**8.727**	**0.973**	**0.747**	**0.742**
Tuned hyperparameters											
XGBoost	0.431	0.872	0.377	0.724	0.728	–	0.8861	8.819	0.969	**0.732**	0.742
CatBoost	**0.423**	**0.874**	0.388	0.727	0.729	–	0.8898	8.837	0.968	0.740	**0.741**
ResNet	0.478	0.857	0.398	0.734	0.731	0.966	0.8976	8.770	0.967	0.751	0.745
FT-Transformer	0.448	0.860	**0.398**	**0.739**	**0.731**	**0.967**	**0.8984**	**8.751**	**0.973**	0.747	0.743

Experiment

❖ Comparing DL models and GBDT

Table 4: Results for ensembles of GBDT and the main DL models. For each model-dataset pair, the metric value averaged over three ensembles is reported. See supplementary for standard deviations. Notation follows Table 3.

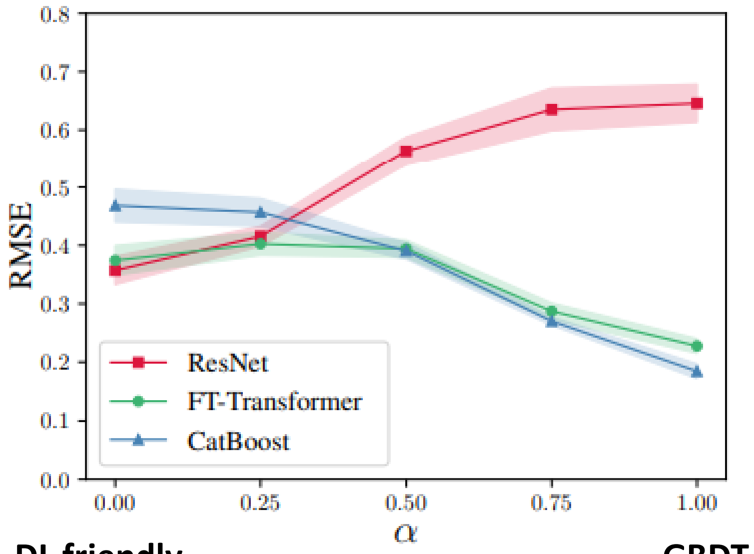
	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
Default hyperparameters											
XGBoost	0.462	**0.874**	0.348	0.711	0.717	0.924	0.8799	9.192	0.964	0.761	0.751
CatBoost	**0.428**	0.873	0.386	0.724	0.728	0.948	0.8893	8.885	0.910	0.749	0.744
FT-Transformer	0.454	0.860	**0.395**	**0.734**	**0.731**	**0.966**	**0.8969**	**8.727**	**0.973**	**0.747**	**0.742**
Tuned hyperparameters											
XGBoost	0.431	0.872	0.377	0.724	0.728	–	0.8861	8.819	0.969	**0.732**	0.742
CatBoost	**0.423**	**0.874**	0.388	0.727	0.729	–	0.8898	8.837	0.968	0.740	**0.741**
ResNet	0.478	0.857	0.398	0.734	0.731	0.966	0.8976	8.770	0.967	0.751	0.745
FT-Transformer	0.448	0.860	**0.398**	**0.739**	**0.731**	**0.967**	**0.8984**	**8.751**	**0.973**	0.747	0.743

GBDT가 유리한 Task에서 ResNET 대비 FT-Transformer가 성능이 좋다
→ 데이터 특성에 따른 차이가 작다

Experiment

❖ **When FT-Transformer is better than ResNet?**

$$x \sim \mathcal{N}(0, I_k), \qquad y = \alpha \cdot f_{GBDT}(x) + (1 - \alpha) \cdot f_{DL}(x).$$



DL friendly **GBDT friendly**

Figure 3: Test RMSE averaged over five seeds (shadows represent std. dev.). One α corresponds to one task; each task has the same set of train, validation and test features, but different targets.

Xtab : Cross-table Pretraining for Tabular Transformers

❖ ICML 2023 게재, 119회 인용

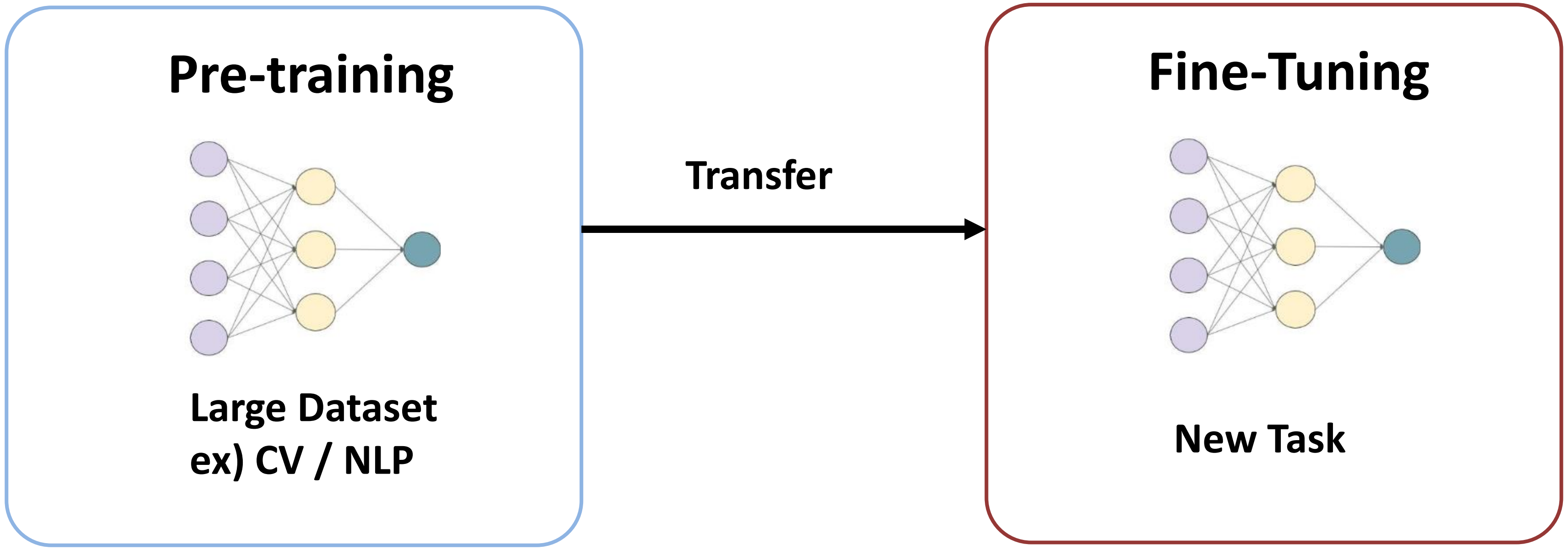
XTab: Cross-table Pretraining for Tabular Transformers

Bingzhao Zhu [1 2 *] Xingjian Shi [3 †] Nick Erickson [4] Mu Li [3 †] George Karypis [4] Mahsa Shoaran [1]

Motivation

❖ 정형 데이터 (Tabular Data)의 한계

✓ 대규모 사전 학습 모델



예시 1) 방대한 양의 사진으로 사물 식별법을 배운 모델을 가져와 공장의 불량품 판독에 이용
예시 2) 인터넷의 방대한 텍스트를 학습한 모델을 가져와 법률 문서 요약에 이용

Motivation

❖ **정형 데이터 (Tabular Data)의 한계**

✓ Tabular Data의 구조적 이질성 (Structural heterogeneity)

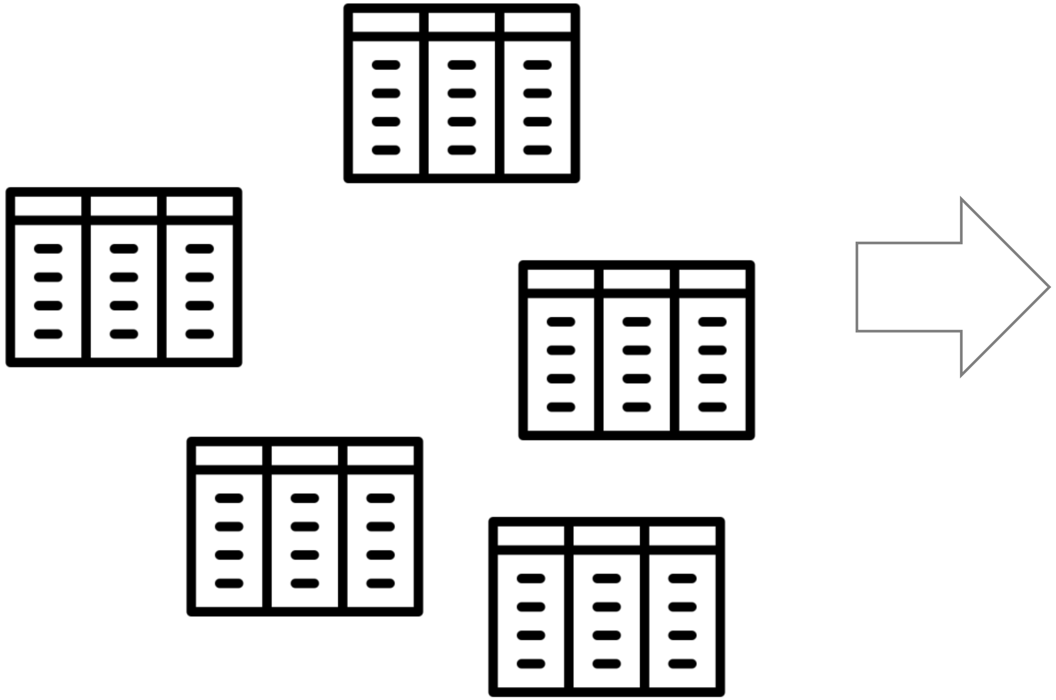
이름	나이	생존여부
John	34	O
June	22	X
Bob	34	X
Harry	35	X
Tom	50	O

넓이	위치	주택가격	건축시기
85	서울	28억	2015
118	경기	12억	2022
59	전라	3억	2007
74	경상	5억	1999
85	제주	6억	2001

Column 개수도 다르고, Column이 의미하는 바도 다름
→ 지식의 전이(Transfer)가 불가능에 가까움

Purpose

❖ 가설 : Column의 이름 / 개수는 다르지만, 데이터 간의 상호작용, 분포의 처리 방식에는 공통점이 있지 않을까?



Column 간의 상호작용 파악하는 방법

A 테이블(부동산): ‘집 크기’ 와 ‘가격’의 관계.

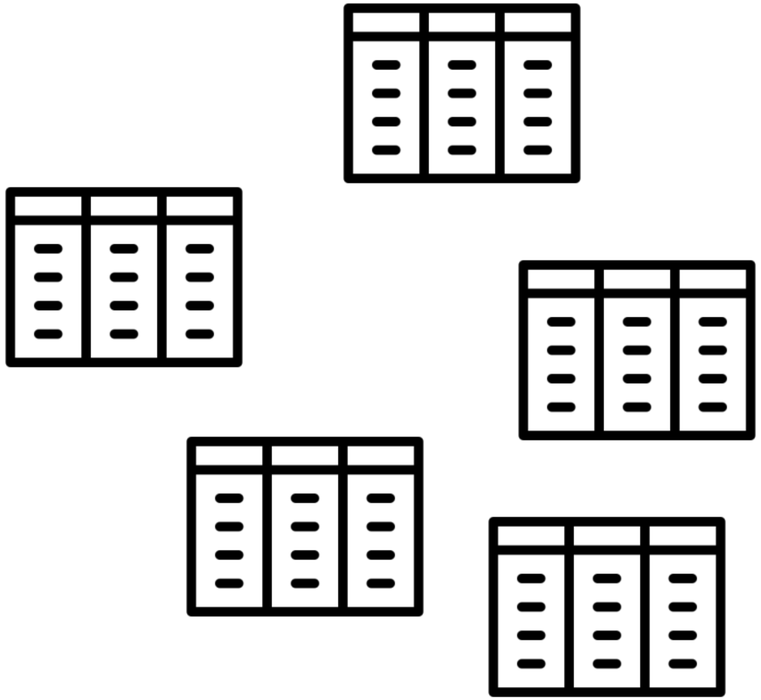
B 테이블(의료): ‘BMI’와 ‘당뇨 위험’의 관계.

→ 집 크기 / BMI 가 아닌

<u>입력된 A가 변할 때 B가 어떻게 반응하는지 수학적 상호작용 규칙</u>

Purpose

❖ 가설 : Column의 이름 / 개수는 다르지만, 데이터 간의 상호작용, 분포의 처리 방식에는 공통점이 있지 않을까?



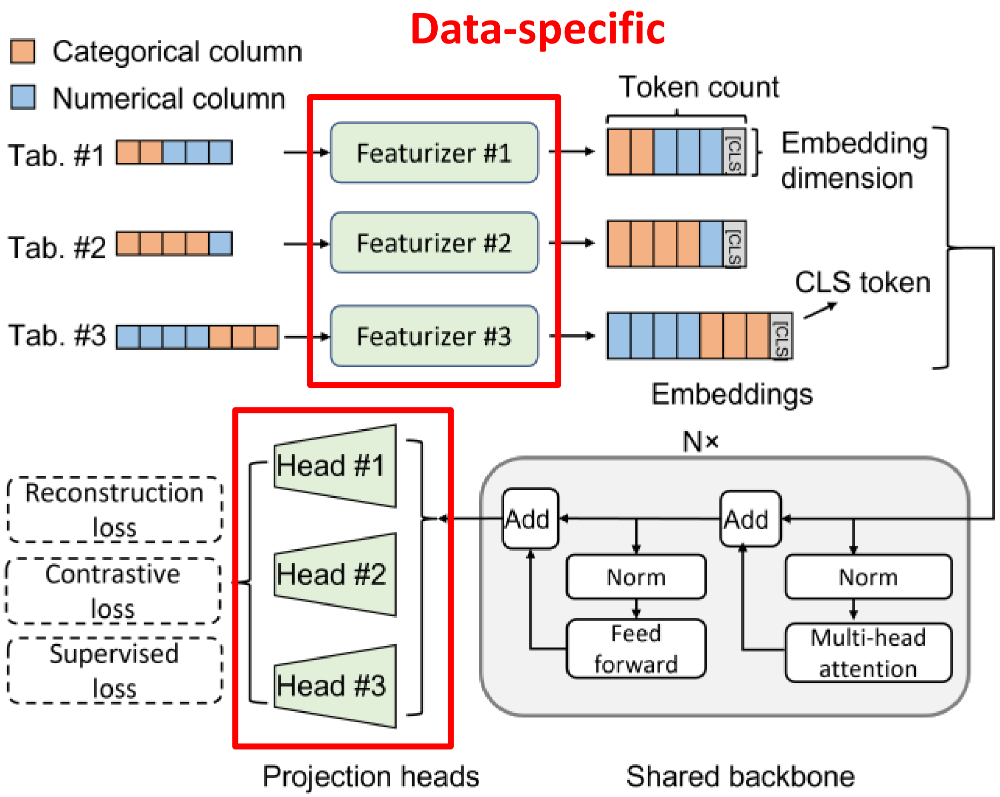
Column 간의 상호작용 파악하는 방법

A 테이블(부동산): '집 크기' 와 '가격'의 관계.
B 테이블(의료): 'BMI'와 '당뇨 위험'의 관계.
→ 집 크기 / BMI 가 아닌
<u>입력된 A가 변할 때 B가 어떻게 반응하는지 수학적 상호작용 규칙</u>

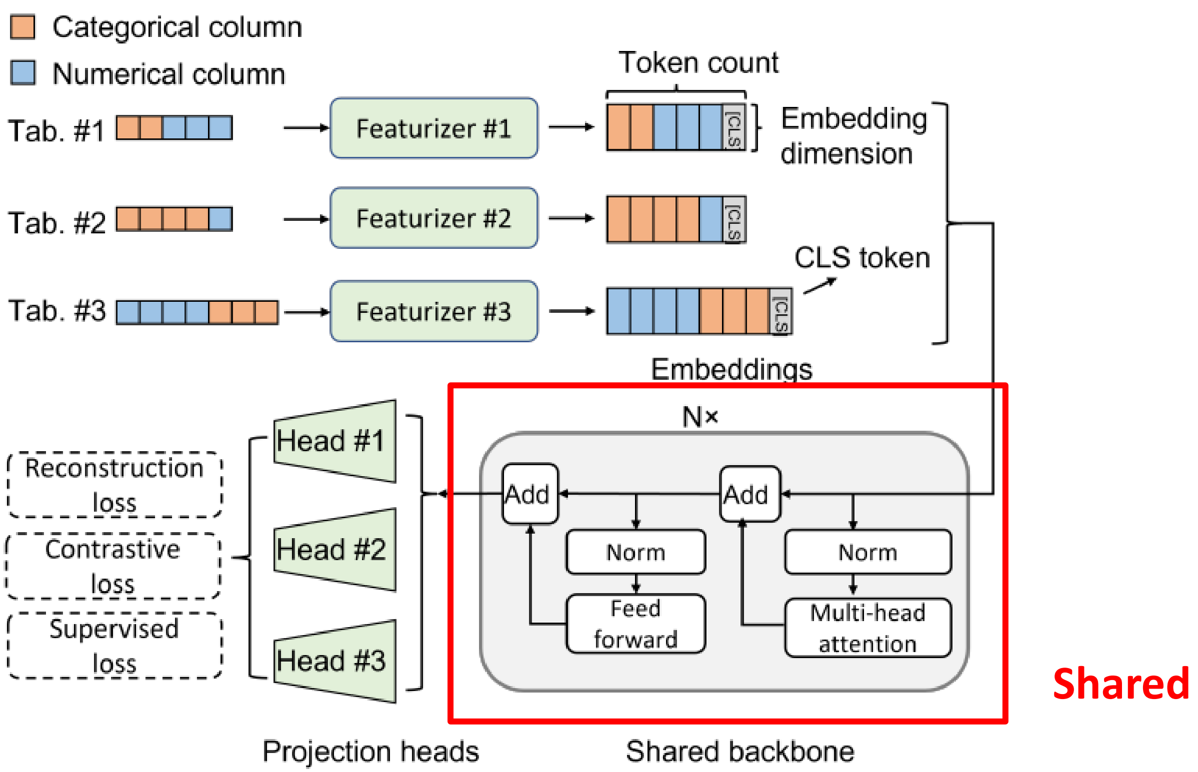
최적의 초기 가중치 분포

서로 다른 Tabular 라도 모델의 초기 가중치를 어떤 테이블에서도
잘 학습할 수 있는 상태로 만드는 것
→ 다양한 테이블을 아우르는 최적의 초기 가중치 학습

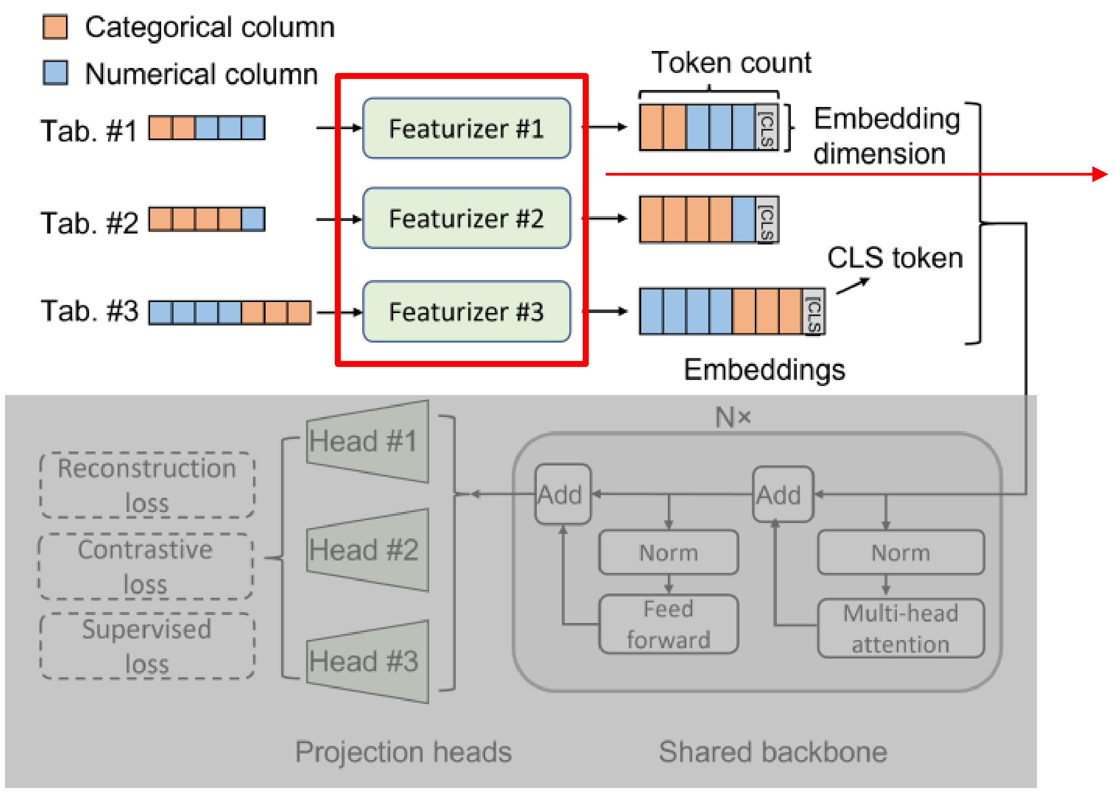
Model Structure

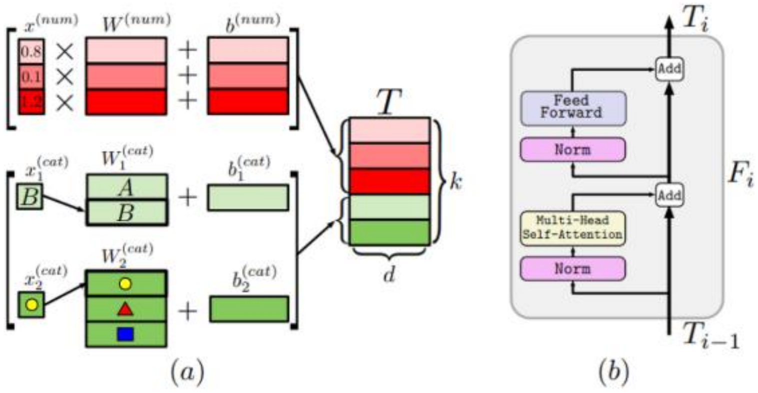


Model Structure

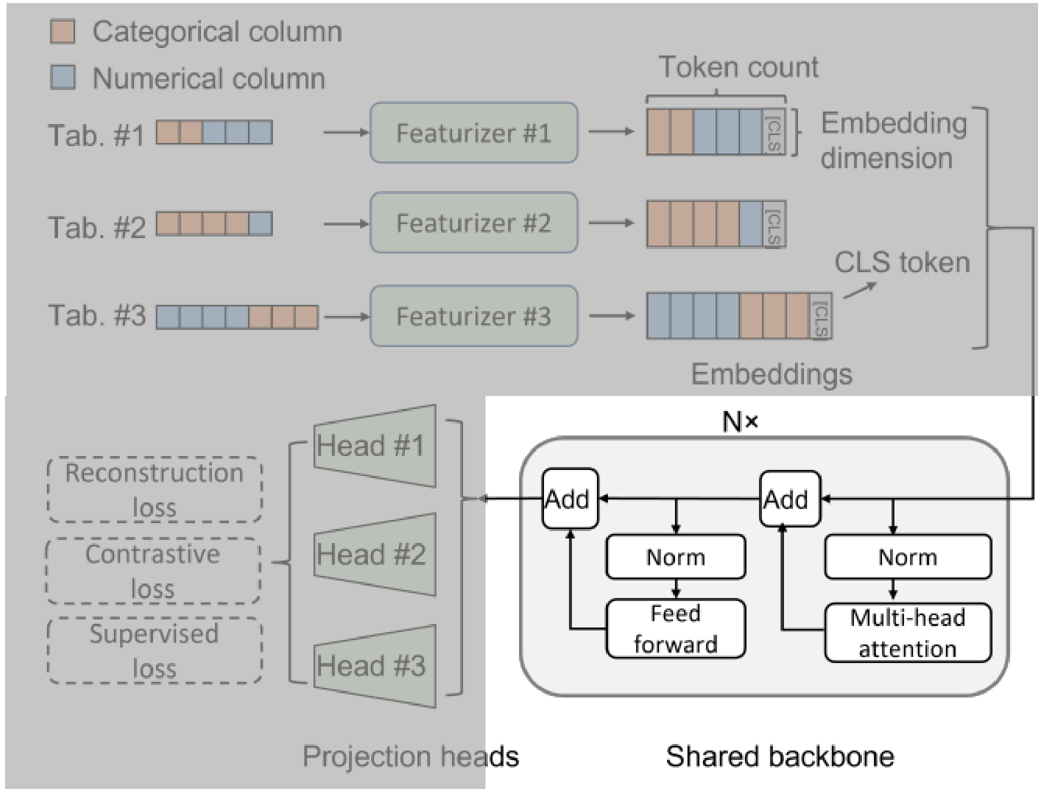


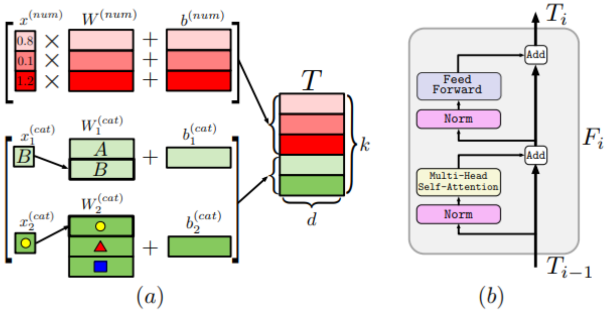
Model Structure



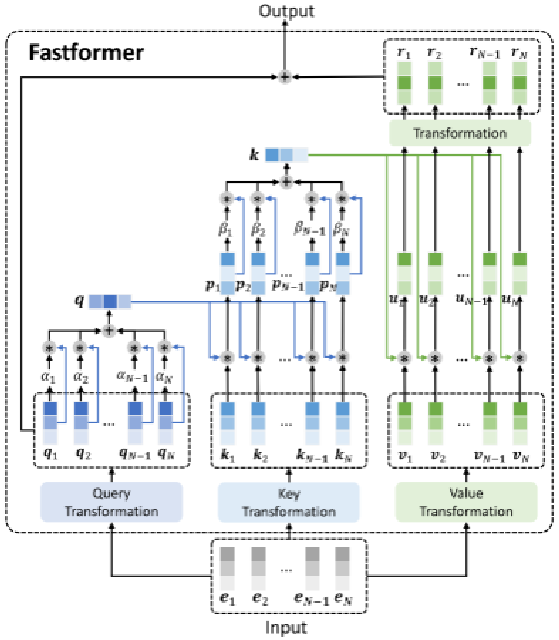


Model Structure

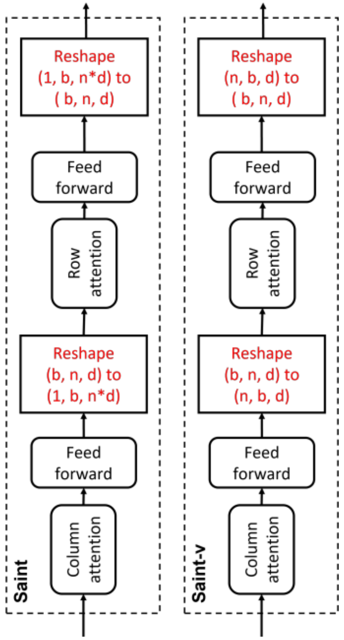




FT transformer

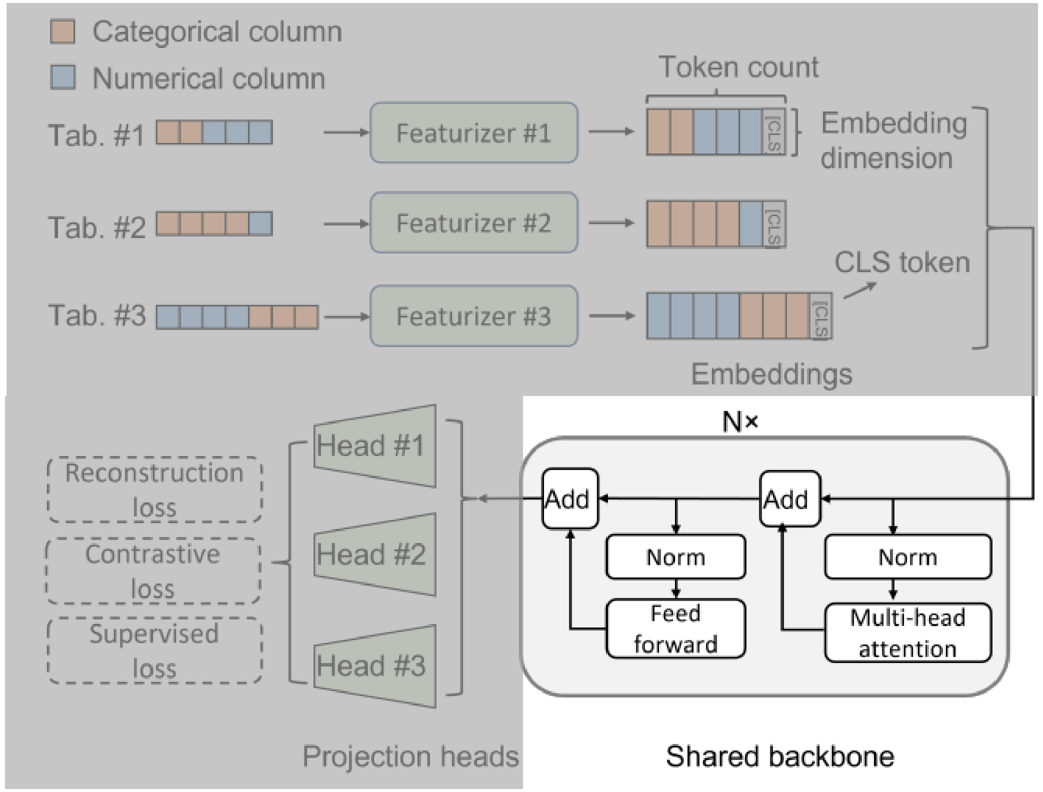


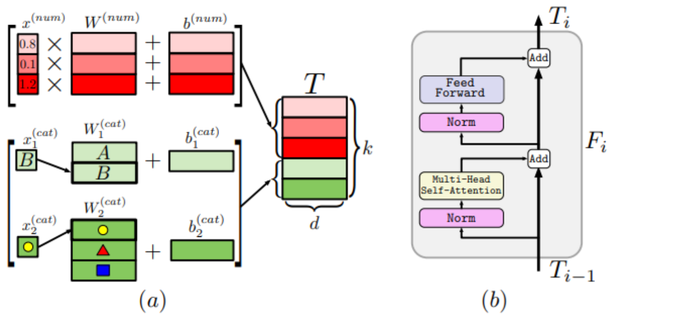
Fastformer



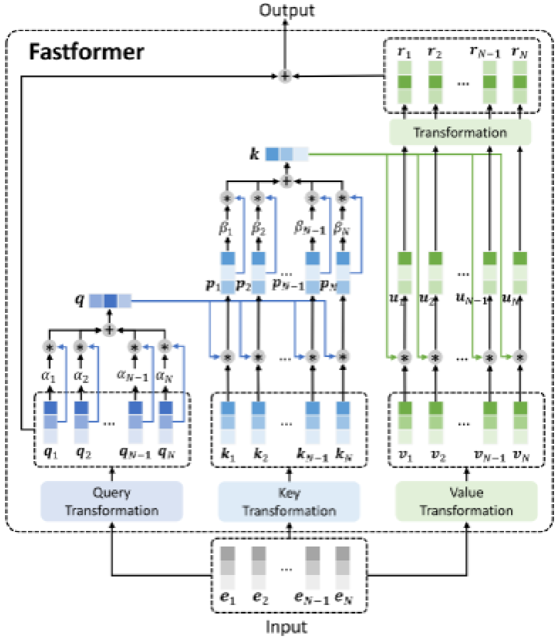
SAINT-v

Model Structure

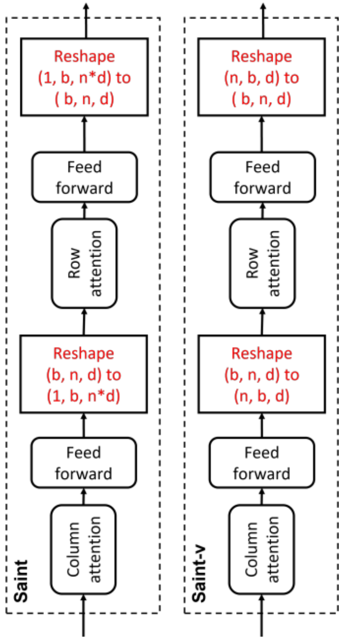




FT transformer

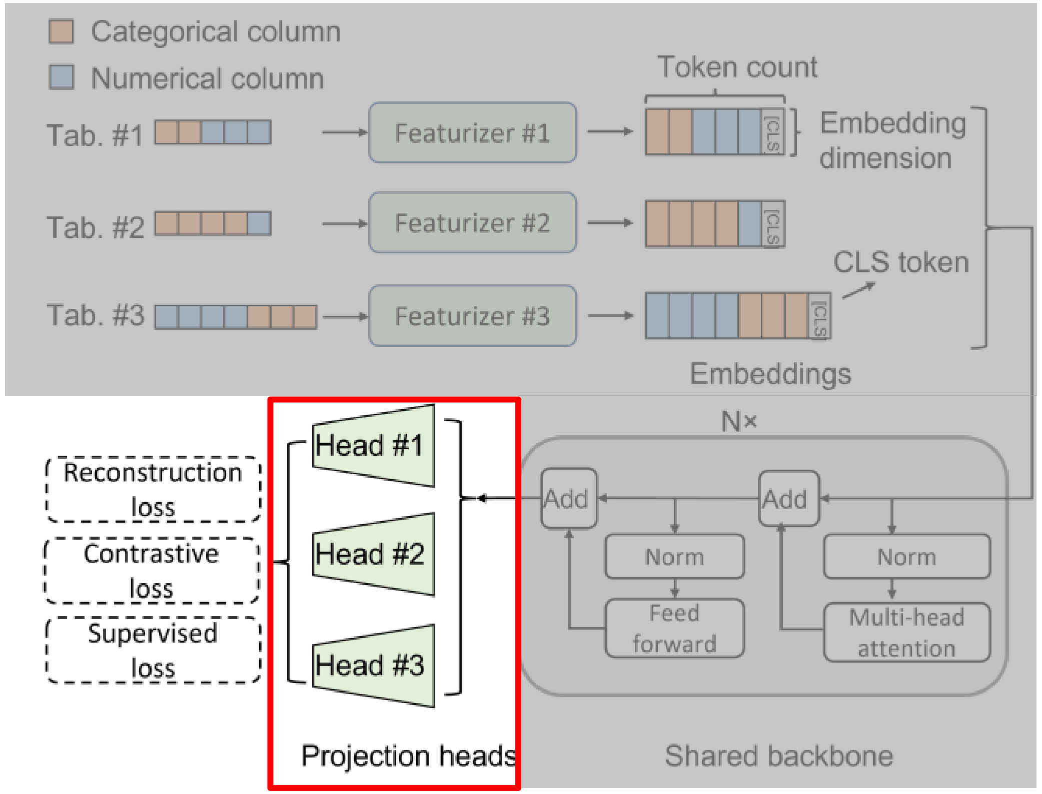


Fastformer



SAINT-v

Model Structure



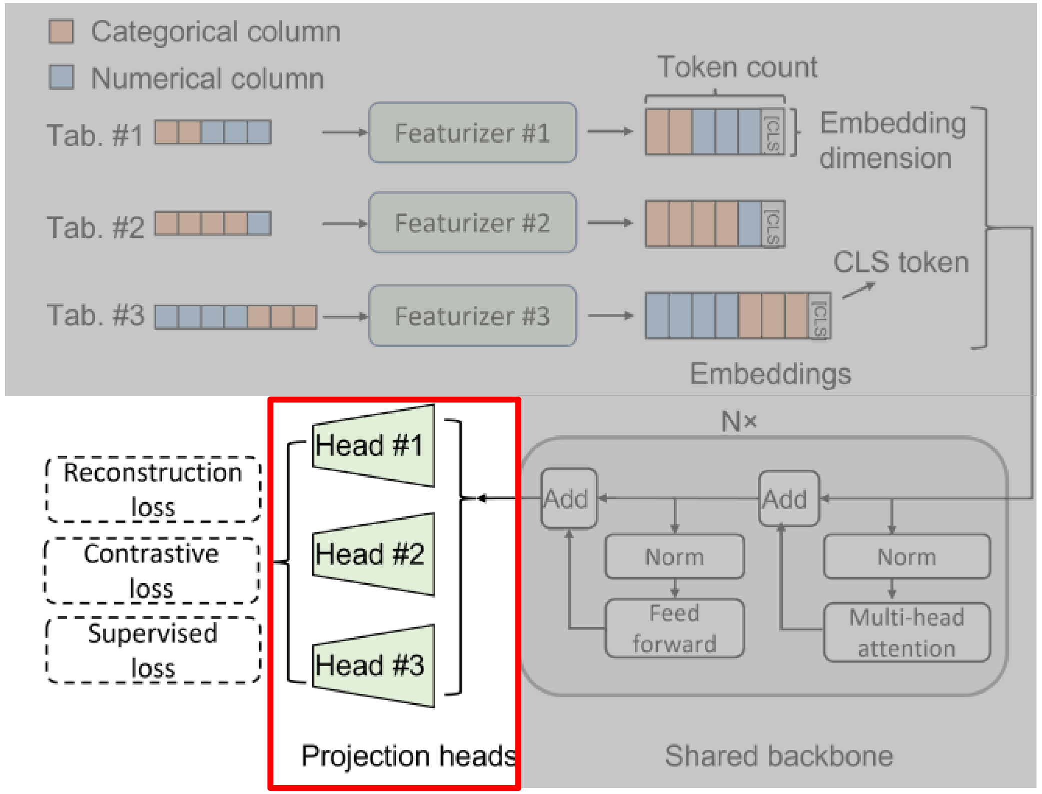
2*(192dim + ReLU)

Reconstruction Head

Contrastive Head

Supervised Head

Model Structure



2*(192dim + ReLU)

Reconstruction Head

Contrastive Head

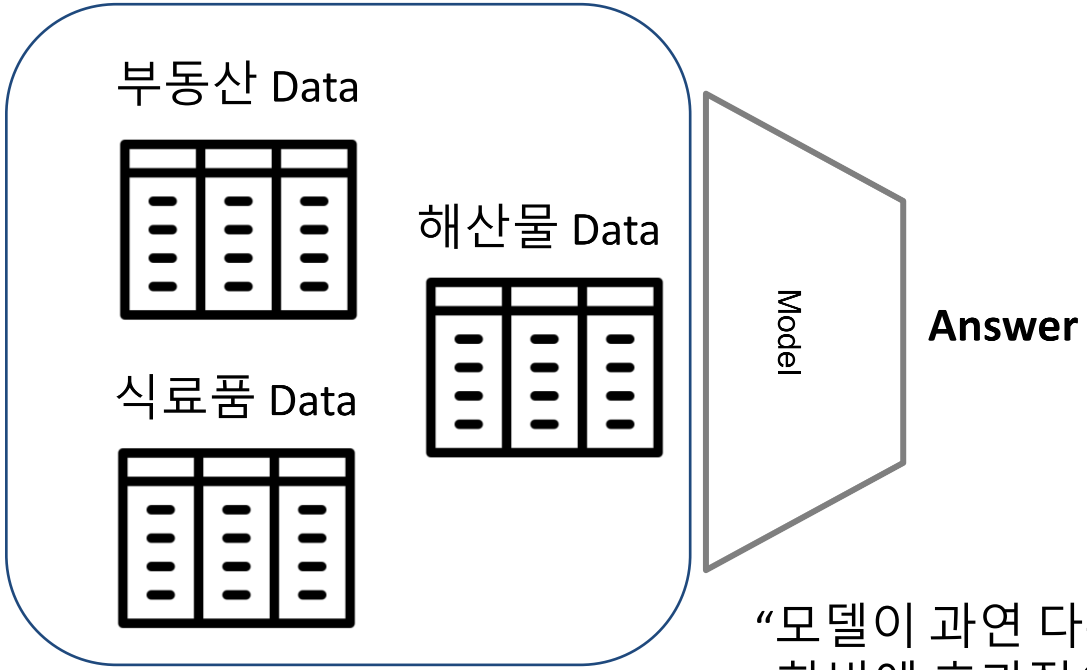
Supervised Head

구조는 OK,
어떻게 Table 별로 효율적으로 학습할까?

Model Structure

❖ 논문에서의 Federated Learning 구조

1. 테이블 구조의 이질성 해결 (Structural Heterogeneity)

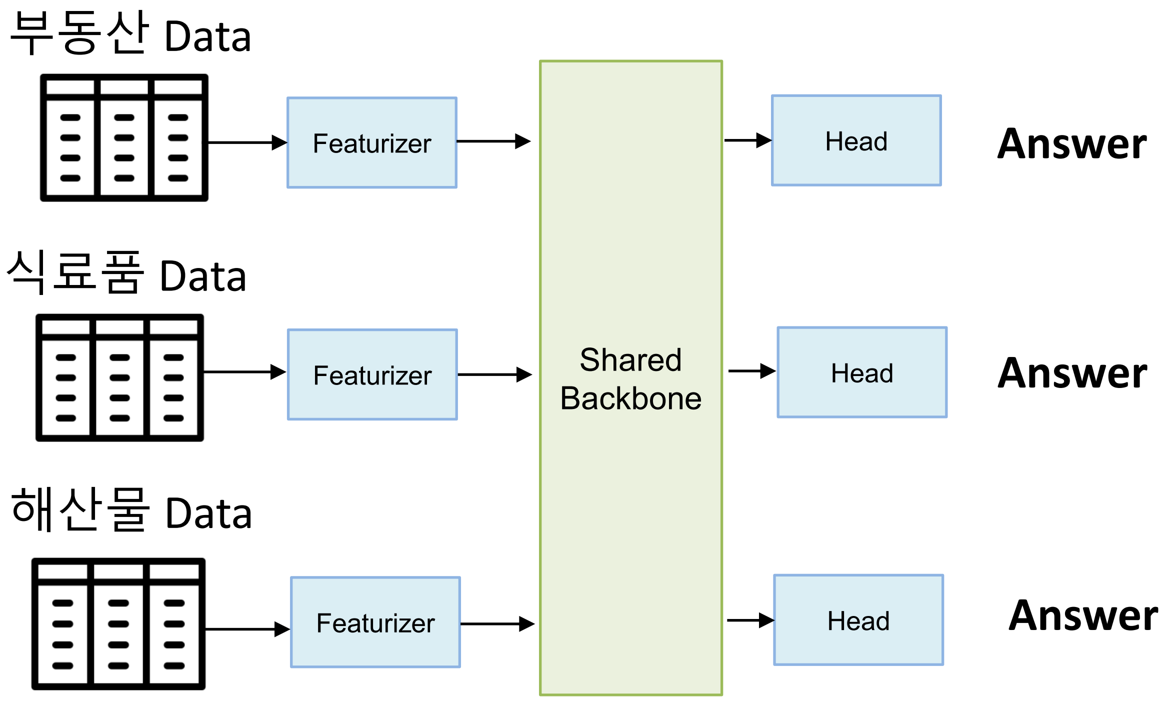


"모델이 과연 다른 종류의 데이터셋을 한번에 효과적으로 학습할 수 있나?"

Model Structure

❖ 논문에서의 Federated Learning 구조

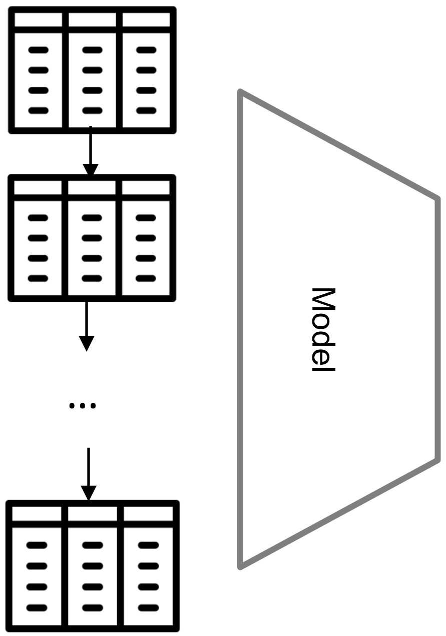
1. 테이블 구조의 이질성 해결 (Structural Heterogeneity)



"테이블마다 다른 의미를 가지는 데이터의 학습문제를 연합학습 구조로 해결 "

❖ 논문에서의 Federated Learning 구조

2. 대규모 학습의 확장성 (Scalability)

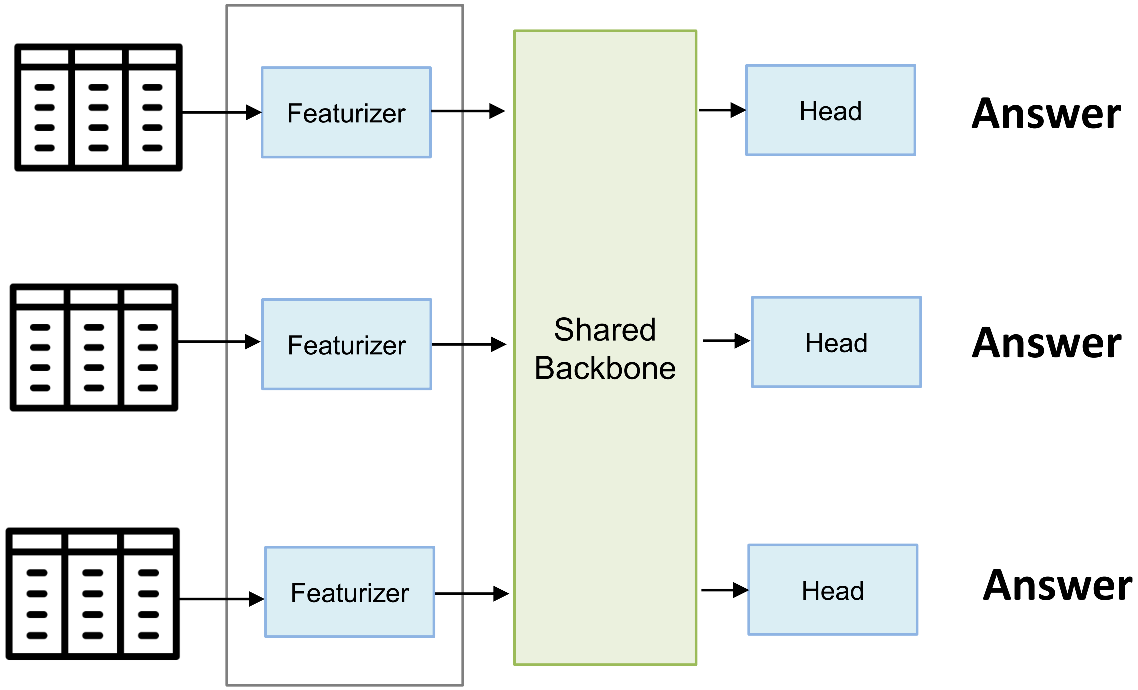


Answer

"Table1, Table2, 다음.. Table 283"
순차적 학습하기에 너무 오랜 시간 소요됨

❖ 논문에서의 Federated Learning 구조

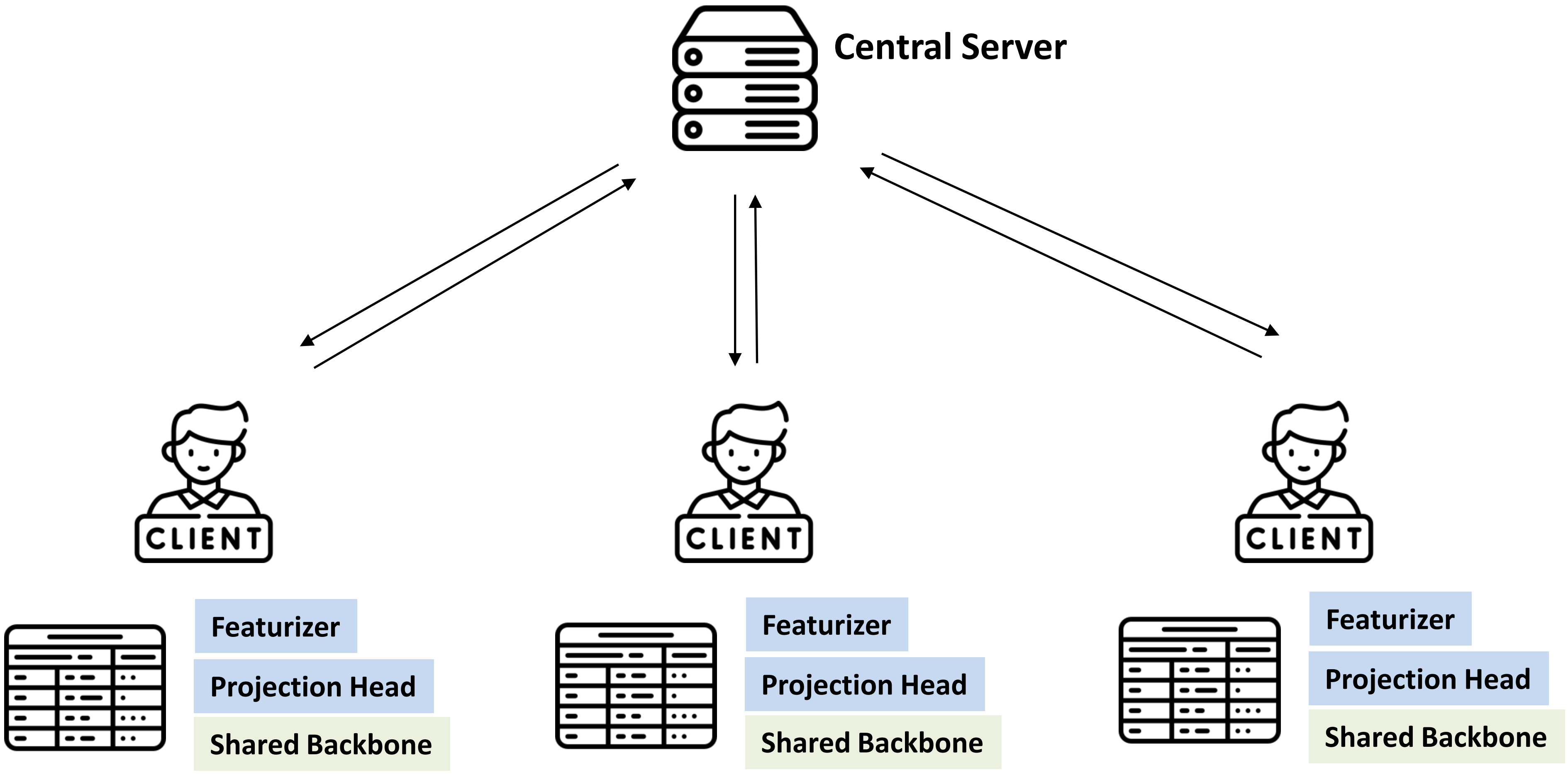
2. 대규모 학습의 확장성 (Scalability)



"여러 GPU(cluster)에서 병렬로 학습하며 중앙 서버에서는 Gardient만 합치므로 효율적으로 대규모 학습 가능 "

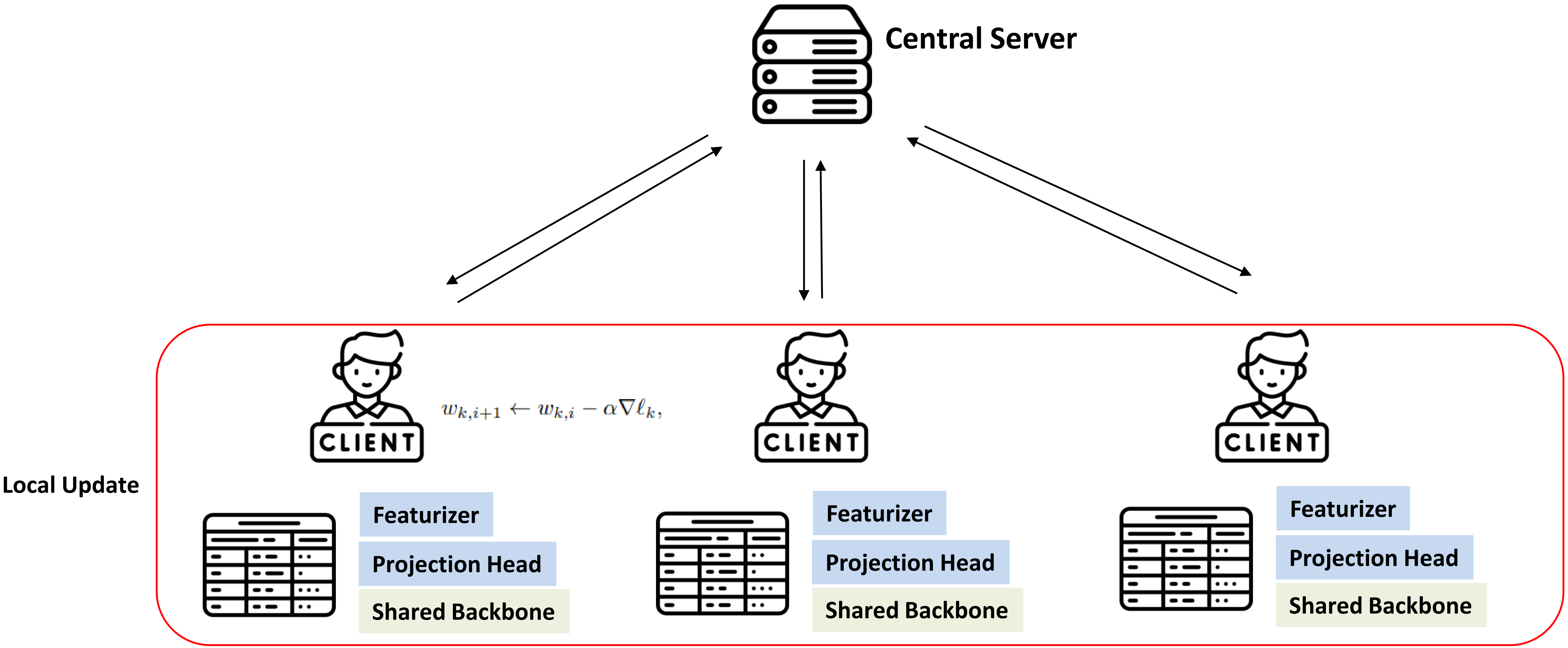
Model Structure

❖ Federated Learning



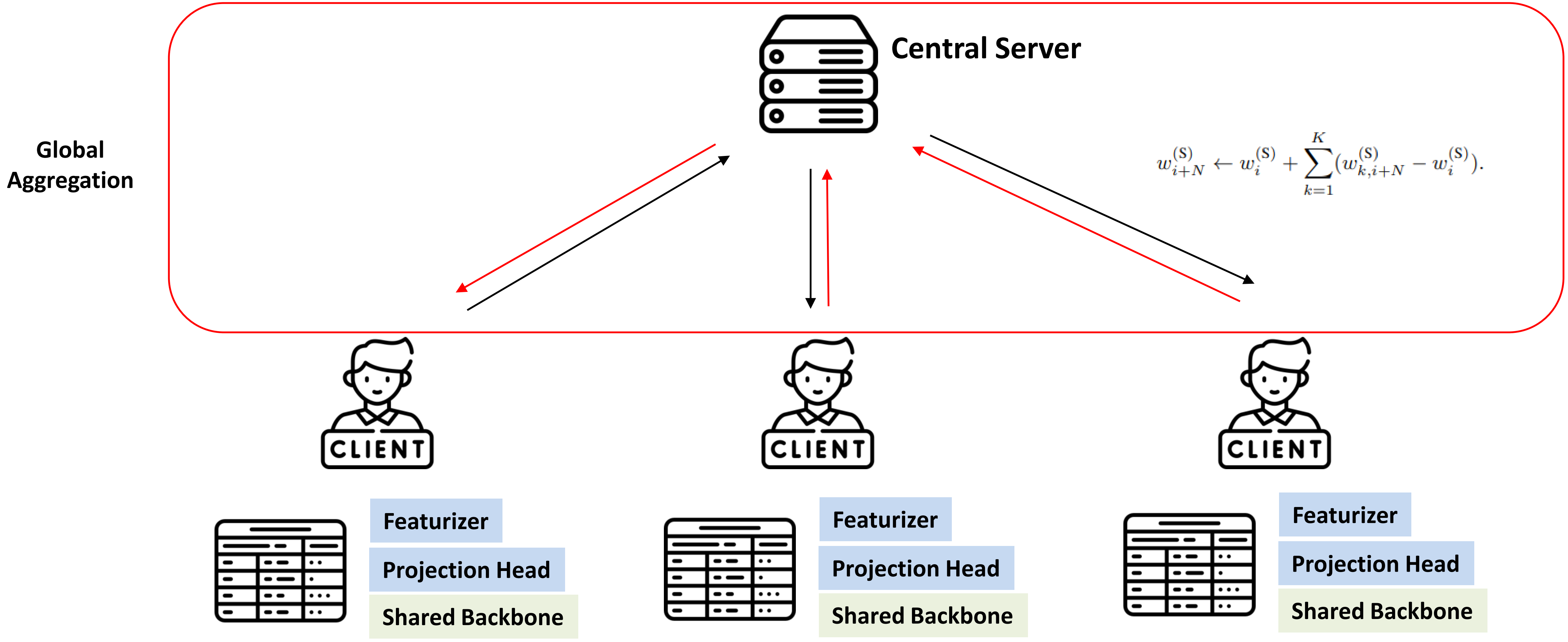
Model Structure

❖ **Federated Learning**



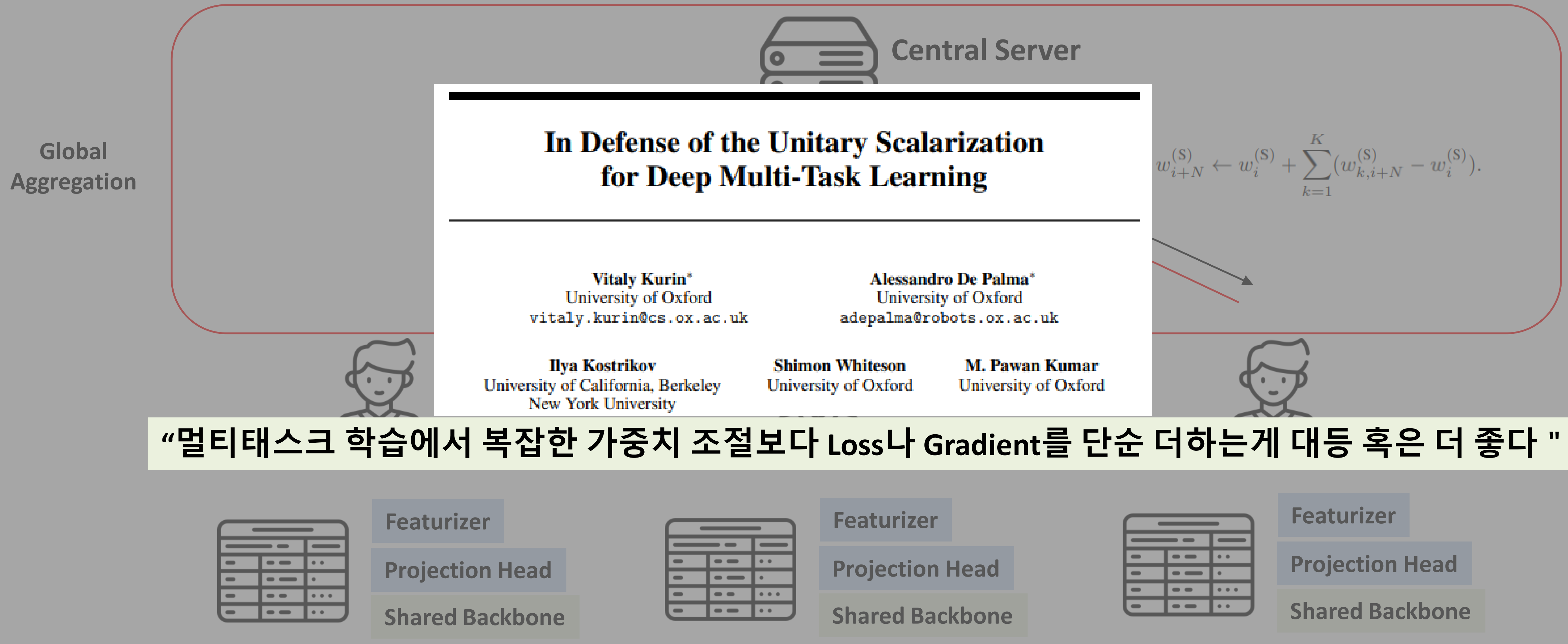
Model Structure

❖ **Federated Learning**



Model Structure

❖ Federated Learning



In Defense of the Unitary Scalarization for Deep Multi-Task Learning

Vitaly Kurin* University of Oxford vitaly.kurin@cs.ox.ac.uk

Alessandro De Palma* University of Oxford adepalma@robots.ox.ac.uk

Ilya Kostrikov University of California, Berkeley New York University

Shimon Whiteson University of Oxford

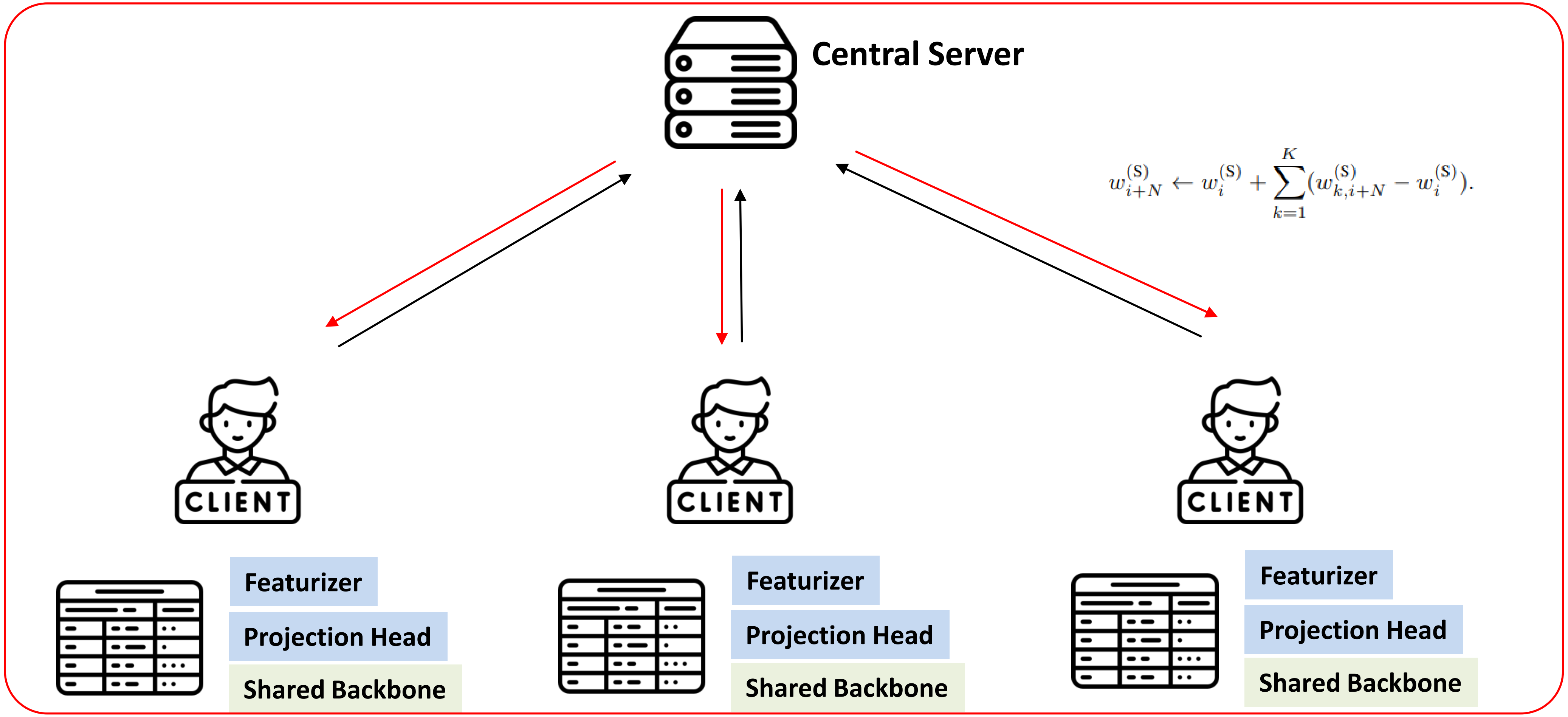
M. Pawan Kumar University of Oxford

"멀티태스크 학습에서 복잡한 가중치 조절보다 Loss나 Gradient를 단순 더하는게 대등 혹은 더 좋다 "

Model Structure

❖ **Federated Learning**

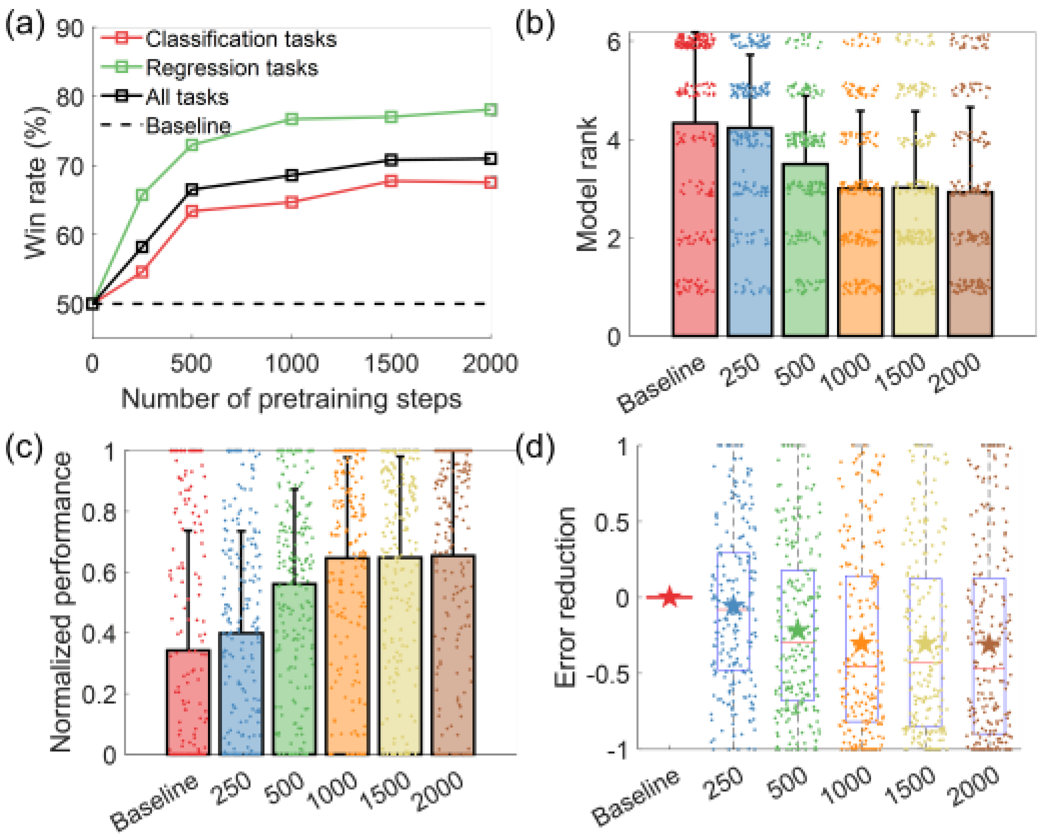
Broadcast



$$w_{i+N}^{(S)} \leftarrow w_i^{(S)} + \sum_{k=1}^{K}(w_{k,i+N}^{(S)} - w_i^{(S)}).$$

Experiment

❖ Cross Table 환경의 사전학습은 Downstream task 성능을 향상시키는가?



(a) 다양한 Downstream task 에서의 Win rate
→ 분류 / 회귀 task 모두에서 더 많은 사전학습이 유리

(b) 사전학습 count에 따른 모델 성능 순위

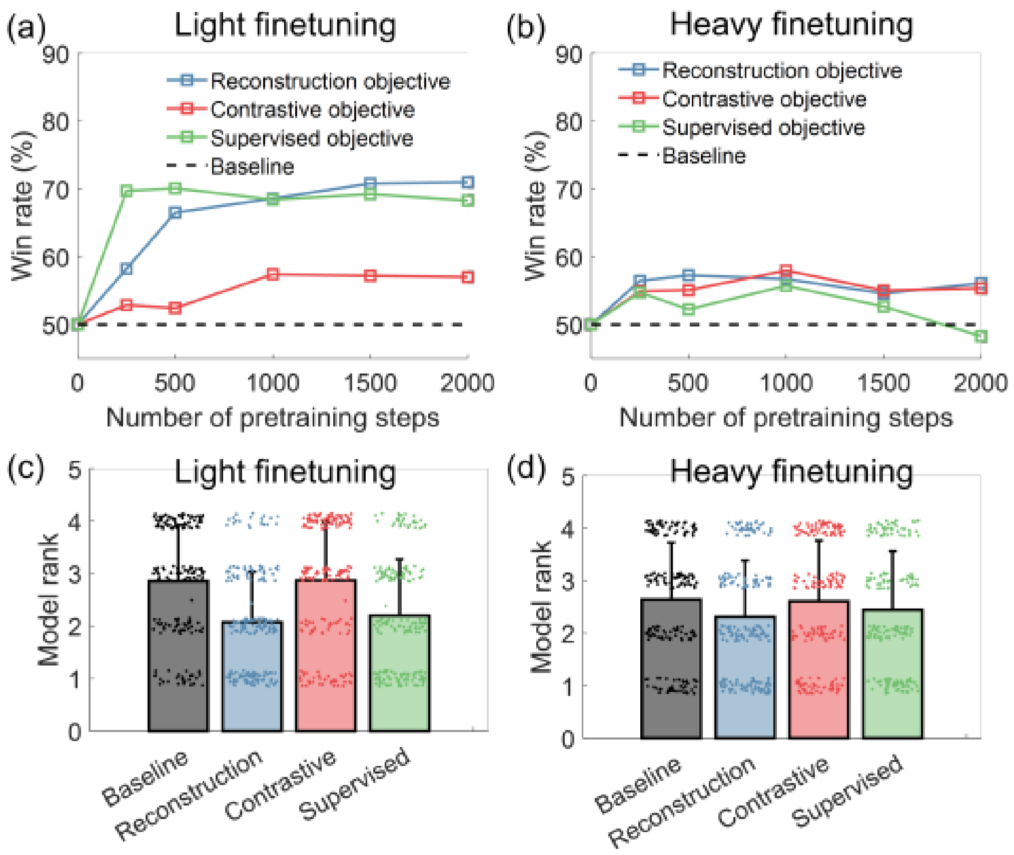
(c) 사전학습 count에 따른 정규화된 예측성능
→ 가장 성능 낮은 모델 0점, 높은 모델 1점으로 정규화

(d) 사전학습 count에 따른 오류 감소율
→ 베이스라인보다 낮은 오류를 가진 모델

사전 학습된 FT-Transformer는 무작위 초기화(random initialization)에 비해 평균적으로 더 높은 정규화된 성능과 감소된 오류를 가짐

Experiment

❖ 어떤 Head를 사용했을 때, Fine tuning 을 어떻게 하면 Downstream task 성능이 가장 좋을까?



1. Reconstruction Loss를 활용한 Head 사용
2. 가벼운 파인튜닝

Experiment

❖ 그래서 결국 GBDT보다 좋은가?

	Methods	Time (s)	Rank
Default hyperparameter	RF	66.8†	7.14 ± 3.81
	XGBoost	43.1†	5.06 ± 3.08
	LightGBM	23.9†	5.23 ± 3.25
	CatBoost	**322.8**†	**2.98** ± **2.66**
	FastAI	89.6	7.24 ± 3.44
	NN	188.8	7.40 ± 3.43
	TransTab-sl*	539.7	11.04 ± 2.75
	TransTab-cl*	312.0	10.79 ± 3.00
	FTT-l	189.2	10.19 ± 2.43
	XTab-l	189.8	9.21 ± 2.57
	FTT-h	532.5	7.29 ± 2.20
	XTab-h	506.3	6.93 ± 2.09
	FTT-best	810.9	4.94 ± 2.25
	XTab-best	**755.9**	**4.39** ± **2.36**
HPO	RF	1084.4†	5.00 ± 2.40
	XGBoost	862.3†	3.69 ± 2.45
	LightGBM	285.0†	4.40 ± 1.93
	CatBoost	**1529.3**†	**3.25** ± **2.10**
	FastAI	549.7	5.24 ± 2.38
	NN	1163.5	5.32 ± 2.20
	FTT	2221.1	4.58 ± 2.08
	XTab	**2335.3**	**4.51** ± **2.00**

† CPU training time.
* Only evaluated on classification tasks.

Conclusion

❖ **Revisiting Deep Learning Models for Tabular Data**

- Transformer의 변형인 FT Transformer 구조를 제안하였으며 대부분 Task에서 다른 DL 방법론보다 우위적인 성능 확인
- GBDT와 비교시에 여전히 일부 Task에서는 GBDT 계열의 모델이 우위
- ResNet Like / FT-Transformer 모델이 이후 정형 데이터의 훌륭한 Baseline이 될 수 있음

❖ Xtab : Cross-table Pretraining for tabular transformers

- Tabular 데이터의 이질성(Heterogeneity) 로 인한 전이학습의 어려움을 구조적 개선을 통해 해결
- 개인정보 보호 수단의 연합학습을 대규모 테이블의 사전학습에 활용하도록 재해석
- Tabular Foundation 모델의 가능성 제안

Thank you