AutoLoss: Automated Loss Function Search in Recommendations

Xiangyu Zhao, Haochen Liu, Wenqi Fan

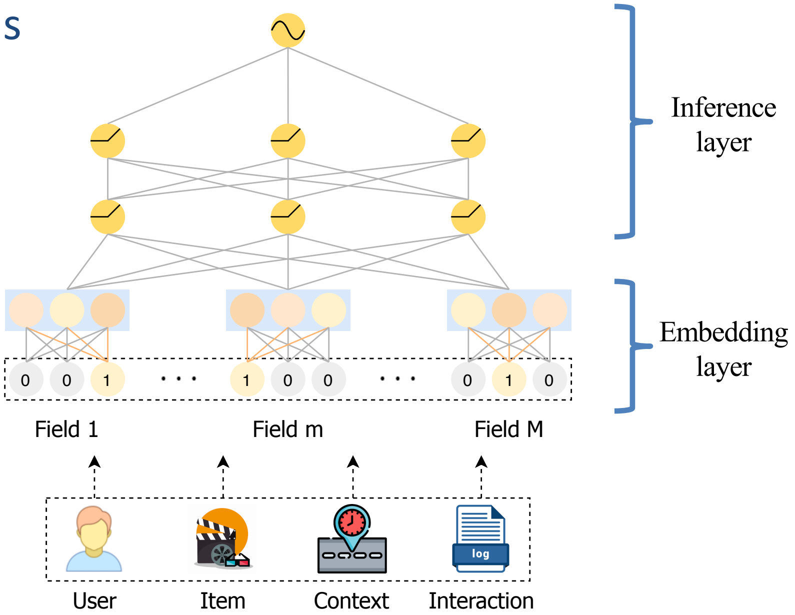
Hui Liu, Jiliang Tang, Chong Wang

1: City University of Hong Kong, 2: Michigan State University

3: The Hong Kong Polytechnic University, 4: Bytedance

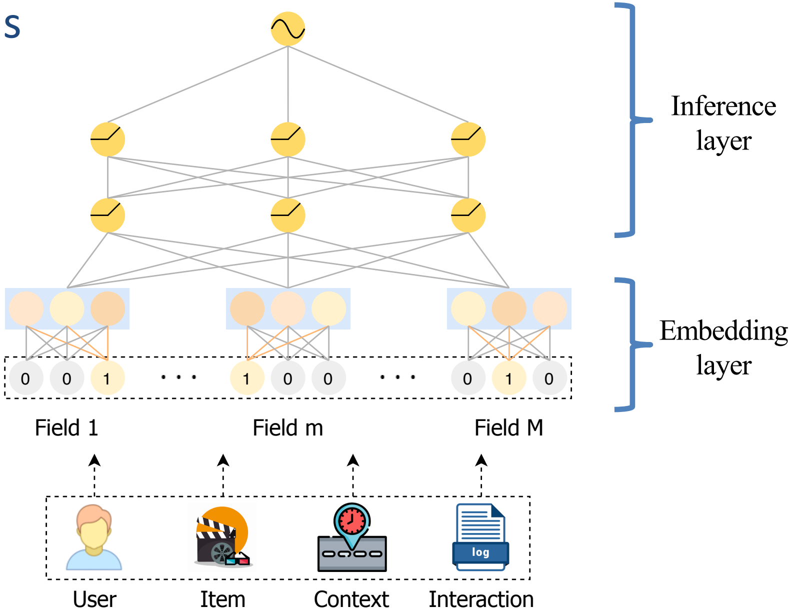
Deep Recommender Systems

- Advantages
  - Feature representations of users and items
  - Non-linear relationships between users and items
- Typical architecture
  - Embedding layer
  - Inference layer



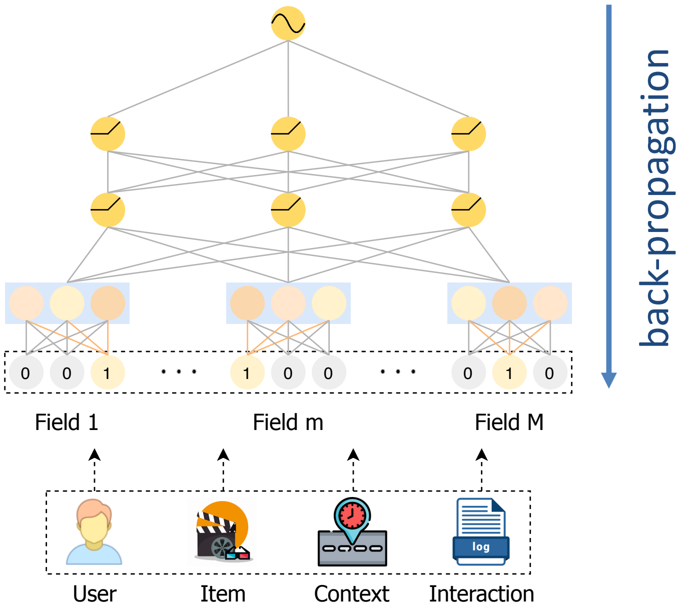
Deep Recommender Systems

- Advantages
  - Feature representations of users and items
  - Non-linear relationships between users and items
- Typical architecture
  - Embedding layer
  - Inference layer
- Well-designed loss functions
  - Item rating prediction (regression)
  - CTR prediction (binary classification)



Existing Optimization Methods

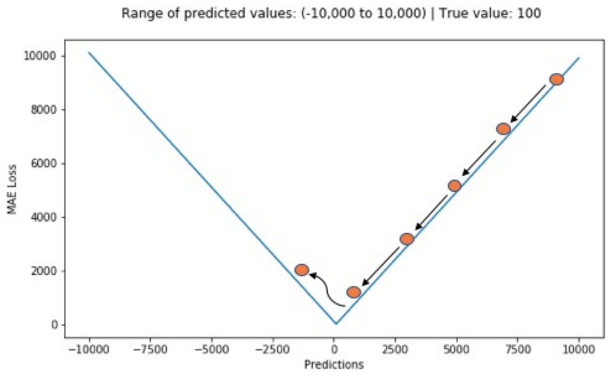
- Predefined and fixed loss functions
  - E.g., MAE or MSE loss for regression tasks



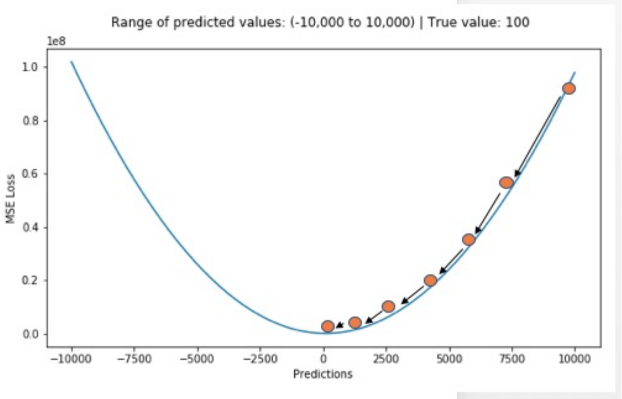
Existing Optimization Methods

- Predefined and fixed loss functions
  - E.g., MAE or MSE loss for regression tasks

- The gradients generated from a given loss function are optimal?



MAE



MSE

Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

[1] Panoptic feature pyramid networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
[2] Upsnet: A unified panoptic segmentation network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

- Disadvantages
  - Exhaustively or manually searching for loss weights → Costly in computation and time

[1] Panoptic feature pyramid networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
[2] Upsnet: A unified panoptic segmentation network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

- Disadvantages
  - Exhaustively or manually searching for loss weights → Costly in computation and time
  - Unified and static loss weights → Overlooking the different convergence behaviors

[1] Panoptic feature pyramid networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
[2] Upsnet: A unified panoptic segmentation network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

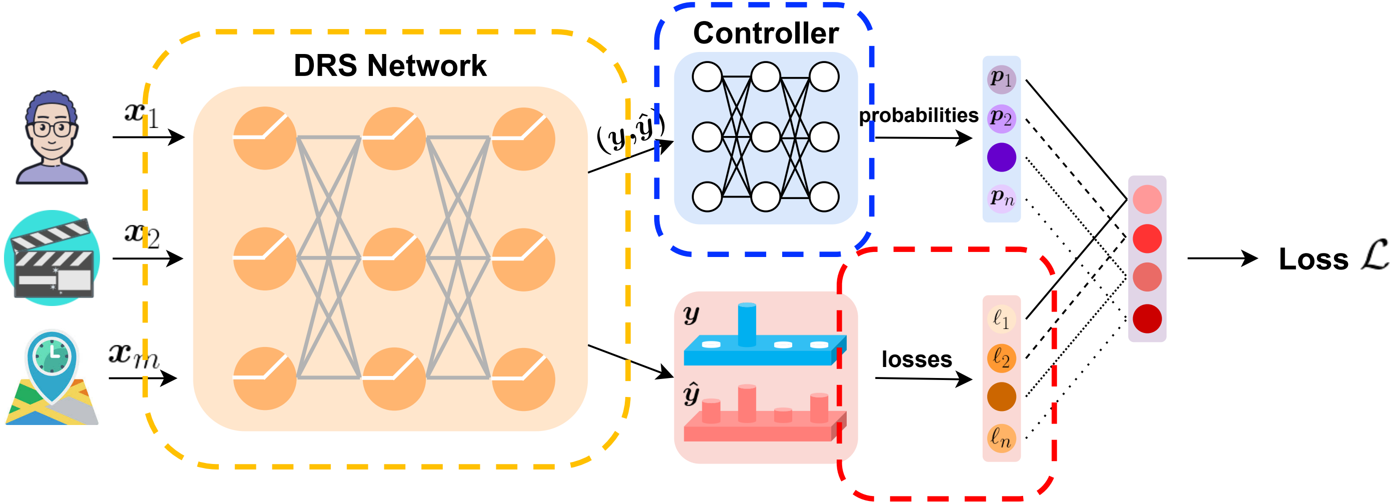
Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

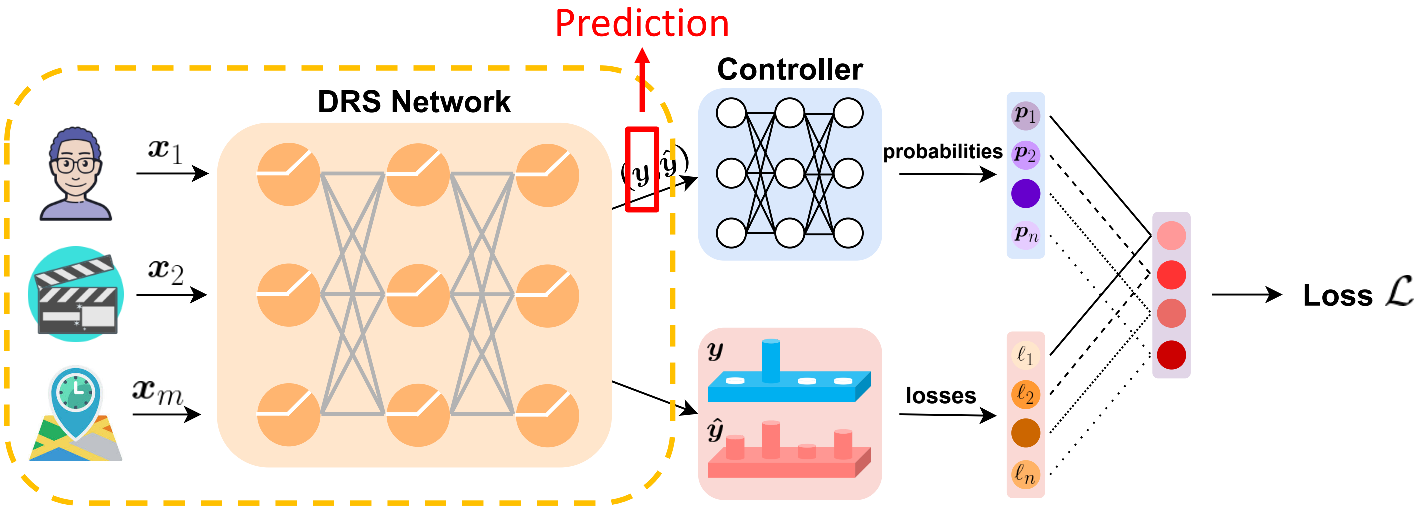
- Disadvantages
  - Exhaustively or manually searching for loss weights → Costly in computation and time
  - Unified and static loss weights → Overlooking the different convergence behaviors
  - Retraining loss weights is always desired → Bad generalizability and transferability

[1] Panoptic feature pyramid networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
[2] Upsnet: A unified panoptic segmentation network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

AutoLoss Framework

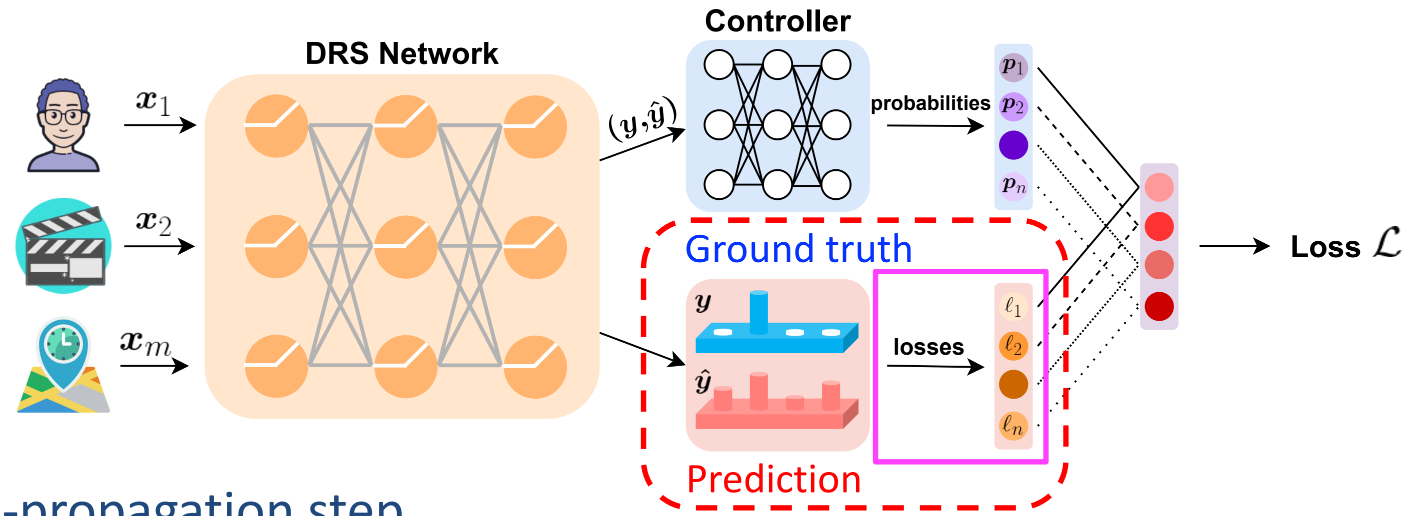


AutoLoss Framework



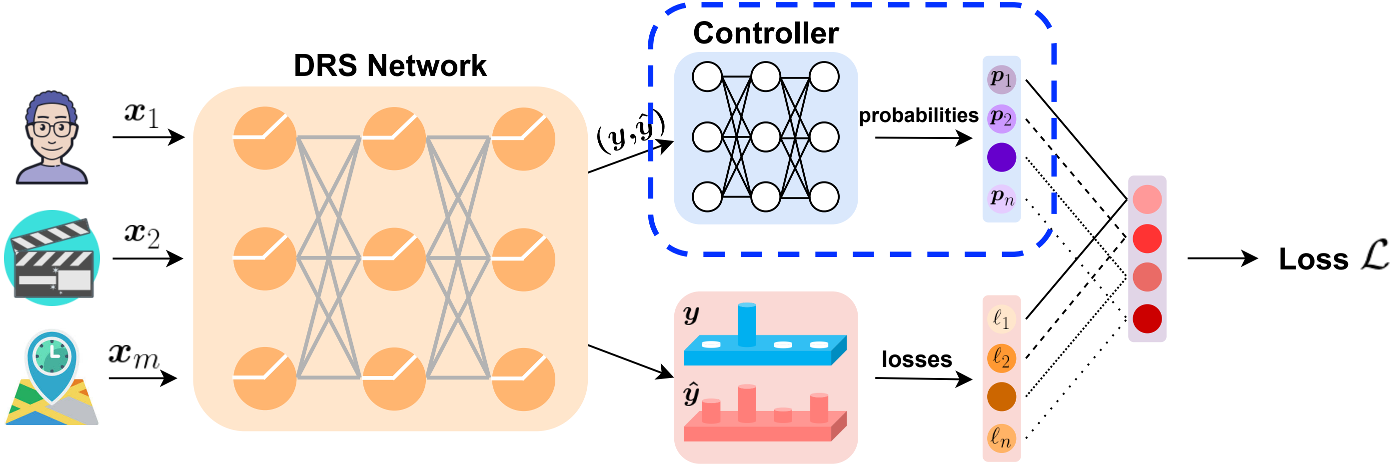
- Forward-propagation step
  - Generating predictions

AutoLoss Framework



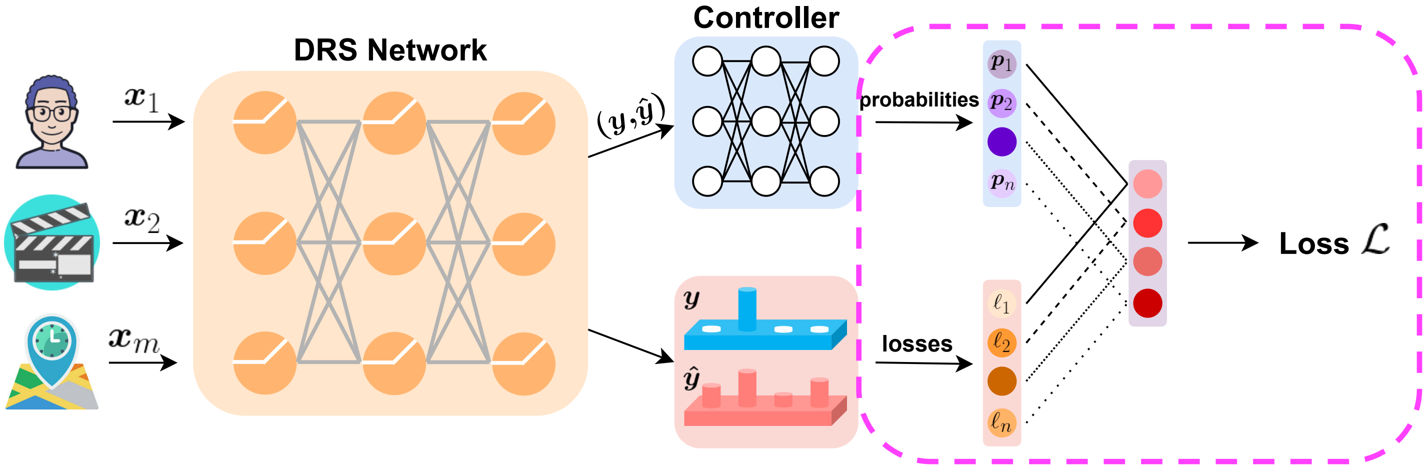
- Forward-propagation step
  - Generating predictions
  - Calculating candidate losses

AutoLoss Framework



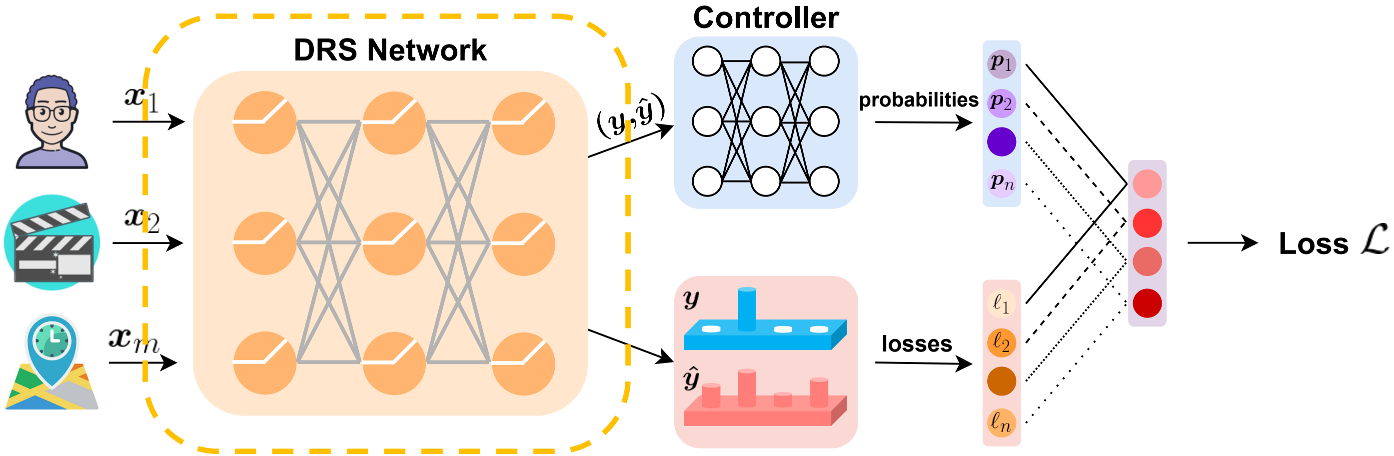
- Forward-propagation step
  - Generating predictions
  - Calculating candidate losses
  - Calculating probabilities

AutoLoss Framework



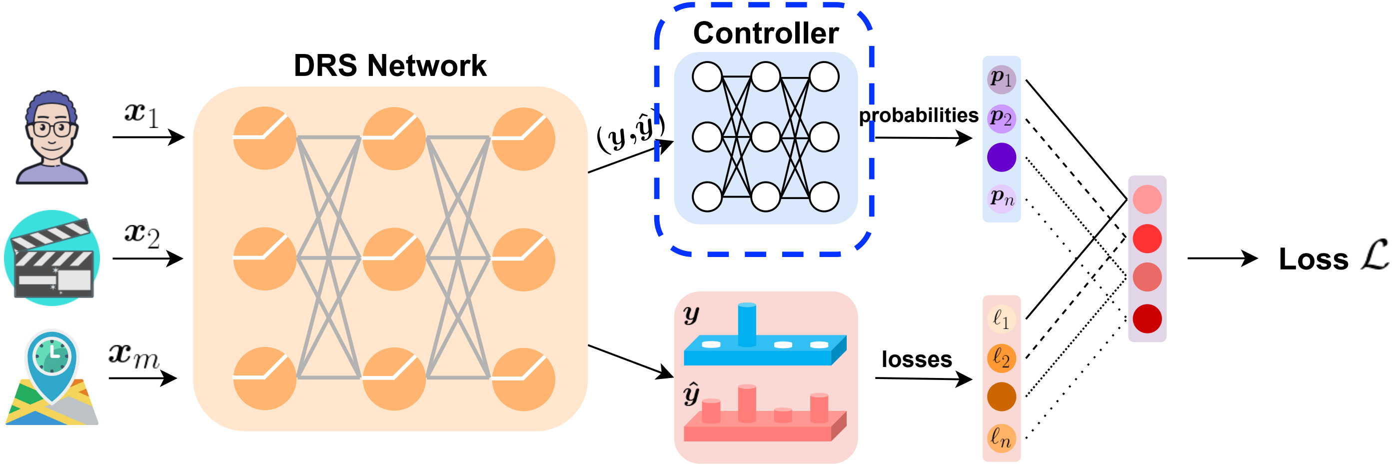
- Forward-propagation step
  - Generating predictions
  - Calculating candidate losses
  - Calculating probabilities
  - Calculating the overall loss

AutoLoss Framework



- Backward-propagation step
  - Updating the main DRS network parameters upon the training data examples

AutoLoss Framework



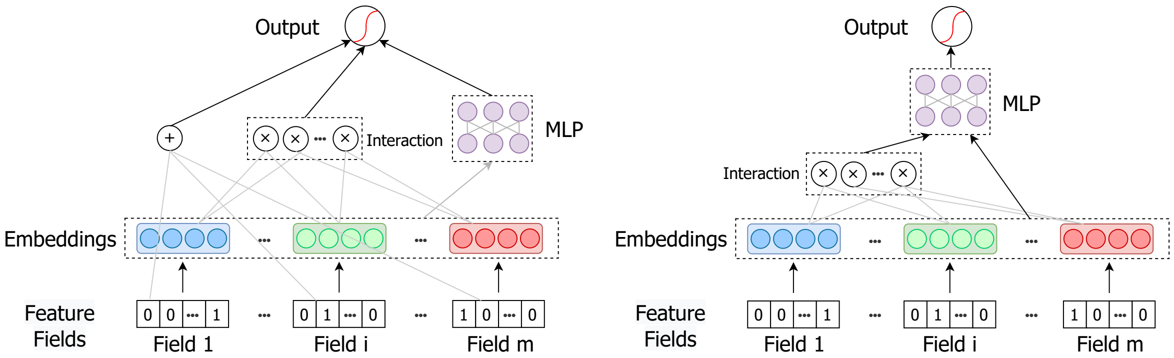
- Backward-propagation step
  - Updating the main DRS network parameters upon the training data examples
  - Optimizing the controller network parameters based on validation data examples

Experimental Settings

- Two recommendation datasets/tasks
  - Criteo: binary classification
  - ML-20m: multiclass classification

Data	Criteo	ML-20m
# Interactions	45,840,617	20,000,263
# Feature Fields	39	2
# Feature Values	1,086,810	165,771
# Behavior	click or not	rating 1~5

- Two deep recommendation models
  - DeepFM and IPNN



(a) DeepFM Architecture (b) IPNN Architecture

Overall Performance

Dataset	Model	Metric	Methods								
			Focal	KL	Hinge	CE	MeLU	BOHB	DARTS	SLF	AutoLoss
Criteo	DeepFM	AUC ↑	0.8046	0.8042	0.8049	0.8056	0.8063	0.8065	0.8067	0.8081	**0.8092***
		Logloss ↓	0.4466	0.4469	0.4463	0.4457	0.4436	0.4435	0.4433	0.4426	**0.4416***
Criteo	IPNN	AUC ↑	0.8077	0.8072	0.8079	0.8085	0.8090	0.8092	0.8093	0.8098	**0.8108***
		Logloss ↓	0.4435	0.4437	0.4432	0.4428	0.4423	0.4422	0.4423	0.4418	**0.4409***
ML-20m	DeepFM	AUC ↑	0.7681	0.7682	0.7685	0.7692	0.7695	0.7695	0.7696	0.7705	**0.7717***
		Logloss ↓	1.2320	1.2317	1.2316	1.2310	1.2307	1.2305	1.2305	1.2299	**1.2288***
ML-20m	IPNN	AUC ↑	0.7721	0.7722	0.7725	0.7733	0.7735	0.7734	0.7736	0.7745	**0.7756***
		Logloss ↓	1.2270	1.2269	1.2266	1.2260	1.2256	1.2257	1.2255	1.2249	**1.2236***

- Fixed loss function: Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks

Overall Performance

Dataset	Model	Metric	Methods								
			Focal	KL	Hinge	CE	MeLU	BOHB	DARTS	SLF	AutoLoss
Criteo	DeepFM	AUC ↑	0.8046	0.8042	0.8049	0.8056	0.8063	0.8065	0.8067	0.8081	**0.8092***
		Logloss ↓	0.4466	0.4469	0.4463	0.4457	0.4436	0.4435	0.4433	0.4426	**0.4416***
Criteo	IPNN	AUC ↑	0.8077	0.8072	0.8079	0.8085	0.8090	0.8092	0.8093	0.8098	**0.8108***
		Logloss ↓	0.4435	0.4437	0.4432	0.4428	0.4423	0.4422	0.4423	0.4418	**0.4409***
ML-20m	DeepFM	AUC ↑	0.7681	0.7682	0.7685	0.7692	0.7695	0.7695	0.7696	0.7705	**0.7717***
		Logloss ↓	1.2320	1.2317	1.2316	1.2310	1.2307	1.2305	1.2305	1.2299	**1.2288***
ML-20m	IPNN	AUC ↑	0.7721	0.7722	0.7725	0.7733	0.7735	0.7734	0.7736	0.7745	**0.7756***
		Logloss ↓	1.2270	1.2269	1.2266	1.2260	1.2256	1.2257	1.2255	1.2249	**1.2236***

- Fixed loss function: Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
- Fixed weights over loss functions: MeLU (meta-learning), BOHB and DARTS (AutoML)

Overall Performance

Dataset	Model	Metric	Methods								
			Focal	KL	Hinge	CE	MeLU	BOHB	DARTS	SLF	AutoLoss
Criteo	DeepFM	AUC ↑	0.8046	0.8042	0.8049	0.8056	0.8063	0.8065	0.8067	0.8081	**0.8092***
		Logloss ↓	0.4466	0.4469	0.4463	0.4457	0.4436	0.4435	0.4433	0.4426	**0.4416***
Criteo	IPNN	AUC ↑	0.8077	0.8072	0.8079	0.8085	0.8090	0.8092	0.8093	0.8098	**0.8108***
		Logloss ↓	0.4435	0.4437	0.4432	0.4428	0.4423	0.4422	0.4423	0.4418	**0.4409***
ML-20m	DeepFM	AUC ↑	0.7681	0.7682	0.7685	0.7692	0.7695	0.7695	0.7696	0.7705	**0.7717***
		Logloss ↓	1.2320	1.2317	1.2316	1.2310	1.2307	1.2305	1.2305	1.2299	**1.2288***
ML-20m	IPNN	AUC ↑	0.7721	0.7722	0.7725	0.7733	0.7735	0.7734	0.7736	0.7745	**0.7756***
		Logloss ↓	1.2270	1.2269	1.2266	1.2260	1.2256	1.2257	1.2255	1.2249	**1.2236***

- Fixed loss function: Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
- Fixed weights over loss functions: MeLU (meta-learning), BOHB and DARTS (AutoML)
- Data example-wise loss weights: SLF (stochastic loss function)

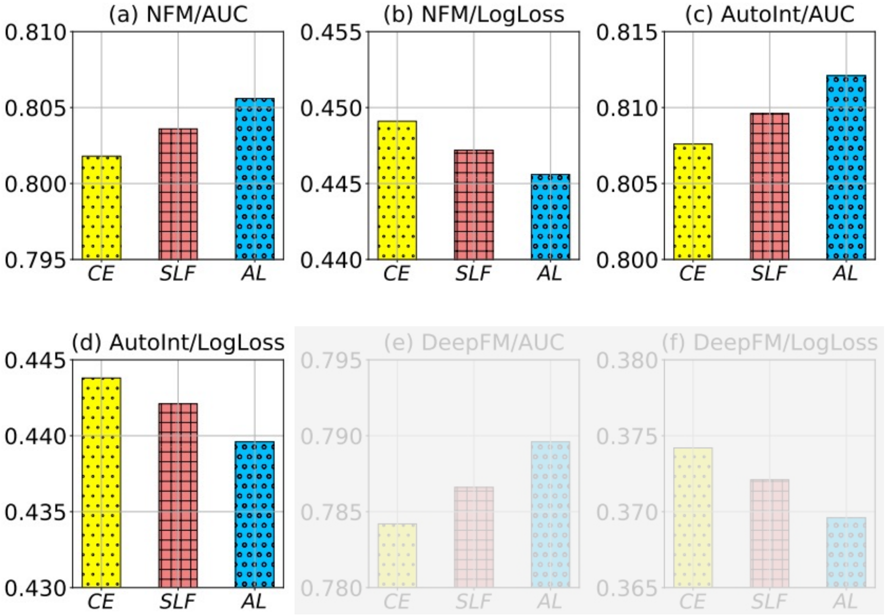
Overall Performance

Dataset	Model	Metric	Methods								
			Focal	KL	Hinge	CE	MeLU	BOHB	DARTS	SLF	AutoLoss
Criteo	DeepFM	AUC ↑	0.8046	0.8042	0.8049	0.8056	0.8063	0.8065	0.8067	0.8081	**0.8092***
		Logloss ↓	0.4466	0.4469	0.4463	0.4457	0.4436	0.4435	0.4433	0.4426	**0.4416***
Criteo	IPNN	AUC ↑	0.8077	0.8072	0.8079	0.8085	0.8090	0.8092	0.8093	0.8098	**0.8108***
		Logloss ↓	0.4435	0.4437	0.4432	0.4428	0.4423	0.4422	0.4423	0.4418	**0.4409***
ML-20m	DeepFM	AUC ↑	0.7681	0.7682	0.7685	0.7692	0.7695	0.7695	0.7696	0.7705	**0.7717***
		Logloss ↓	1.2320	1.2317	1.2316	1.2310	1.2307	1.2305	1.2305	1.2299	**1.2288***
ML-20m	IPNN	AUC ↑	0.7721	0.7722	0.7725	0.7733	0.7735	0.7734	0.7736	0.7745	**0.7756***
		Logloss ↓	1.2270	1.2269	1.2266	1.2260	1.2256	1.2257	1.2255	1.2249	**1.2236***

- Fixed loss function: Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
- Fixed weights over loss functions: MeLU (meta-learning), BOHB and DARTS (AutoML)
- Data example-wise loss weights: SLF (stochastic loss function)

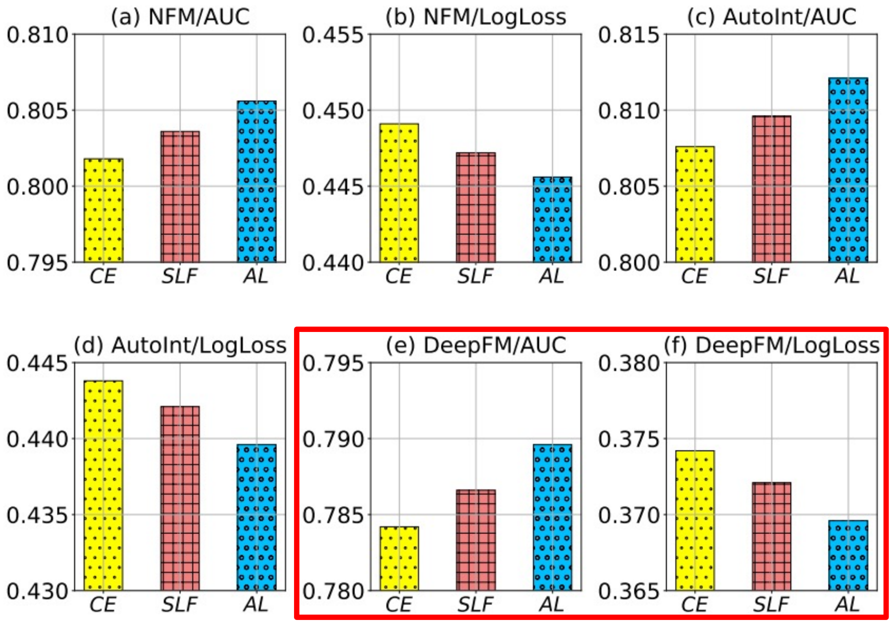
Transferability Study

- Transferability among DRS models
  - DeepFM → NFM and AutoInt
  - CE: cross-entropy loss
  - SLF: SLF controller from DeepFM
  - AL: AutoLoss controller from DeepFM
- Transferability among datasets
  - Criteo → Avazu
  - CE: cross-entropy loss on Avazu
  - SLF: SLF controller from Criteo
  - AL: AutoLoss controller from Criteo

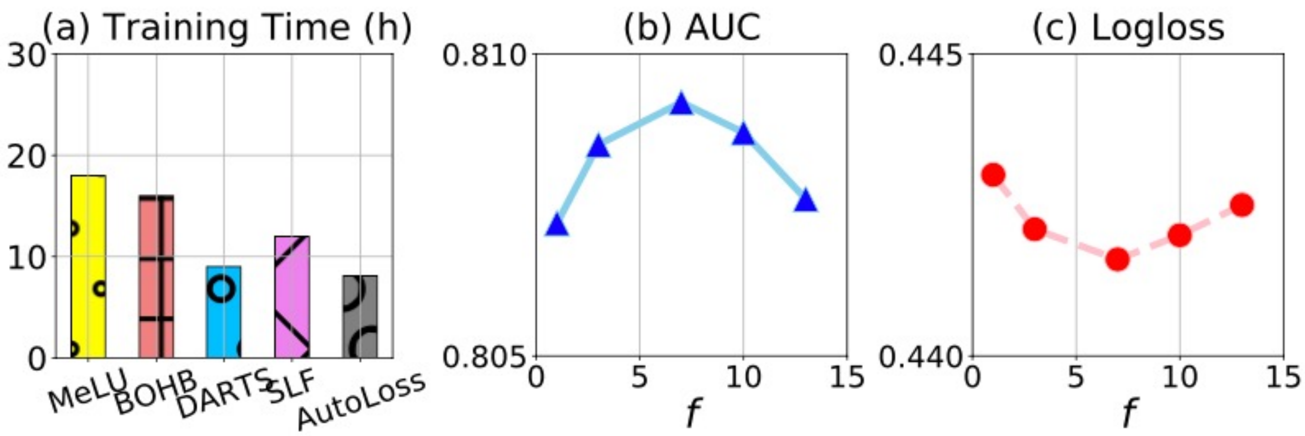


Transferability Study

- Transferability among DRS models
  - DeepFM → NFM and AutoInt
  - CE: cross-entropy loss
  - SLF: SLF controller from DeepFM
  - AL: AutoLoss controller from DeepFM
- Transferability among datasets
  - Criteo → Avazu
  - CE: cross-entropy loss on Avazu
  - SLF: SLF controller from Criteo
  - AL: AutoLoss controller from Criteo



Efficiency Study



- Fastest training speed
  - AutoLoss can generate the most appropriate gradients to update DRS, which increases the optimization efficiency
  - We update the controller once after every 7 times DRS is updated, which not only reduces the training time (~ 60%) with fewer computations, but also enhances the performance

Conclusion

- We propose an end-to-end framework, AutoLoss, which can automatically select the proper loss functions for training DRS frameworks
  - Better recommendation performance and training efficiency

- A novel controller network is developed to adaptively adjust the probabilities over multiple loss functions according to different data examples' convergence behaviors
  - Enhancing the model generalizability between different DRS frameworks and datasets