A General Form of Selection Tensor Decomposition

In this section, we further extend the selection tensor decomposition in Section 3.3.1 from a special case, where $t = 2$, to a more general case, where $2 \leq t \leq n$. The interaction selection tensor $\mathbf{A}_v^t$ for t-th order features is also semi-positive and sysmmtric. By extending the Takagi factorization [2], we have:

$$\mathbf{A}_v^t = \mathbf{\Sigma} \times_1 \mathcal{U} \times_2 \mathcal{U} \times_3 \cdots \times_t \mathcal{U}, \tag{13}$$

where $\mathbf{\Sigma}$ is a $\underbrace{d^{'} \times \cdots \times d^{'}}_{\text{t times}}$ diagonal tensor, $\times_t$ denotes the t-mode matrix multiplication [4], $\mathcal{U} \in R^{m \times d^{'}}$ and $d^{'} < m$. Similar to the special case where $t = 2$, we adopt multi-mode tensor factorization [4] to replace $\mathcal{U}$ as an output of a neural network, denoted as:

$$\mathcal{U} \approx f_{\hat{\theta}}(\hat{\mathbf{E}}), \tag{14}$$

where $f_{\hat{\theta}} : \mathbb{R}^{m \times \hat{d}} \rightarrow \mathbb{R}^{m \times d^{'}}$, $\hat{d} \ll d^{'}$ is a neural network with parameter $\hat{\theta}$ and $\hat{\mathbf{E}} \in \mathbb{R}^{m \times \hat{d}}$ is an additional embedding table for generating feature interaction selection tensor. The element of architecture metric $\mathbf{A}^t_{v(k_{i_1}, \cdots, k_{i_t})}$ can be calculated given the following equation:

$$\mathbf{A}^t_{v(k_{i_1}, \cdots, k_{i_t})} = \mathbf{\Sigma} \times_1 f_{\hat{\theta}}(\hat{\mathbf{E}}_{k_{i_1},:}) \times_2 \cdots \times_t f^T_{\hat{\theta}}(\hat{\mathbf{E}}_{k_{i_t},:}). \tag{15}$$

The original value-grained selection tensor $\mathbf{A}_v^t$ consists of $\mathcal{O}(m^t)$ elements. The trainable elements is reduced to $\mathcal{O}(md^{'})$ after the Takagi factorization [2] and to $\mathcal{O}(\hat{d}(m + d^{'}))$ after the multi-mode tensor factorization [4].

B Experiment Setup

B.1 Dataset and Preprocessing

We conduct our experiments on two public real-world benchmark datasets. The statistics of all datasets are given in Table 3. We describe all these datasets and the pre-processing steps below.

Table 3: Dataset Statistics

Dataset	#samples	#field	#value	pos ratio
Criteo	4.6×10^7	39	6.8×10^6	0.2562
Avazu	4.0×10^7	24	4.4×10^6	0.1698
KDD12	1.5×10^8	11	6.0×10^6	0.0445

Note: *#samples* refers to the total samples in the dataset, *#field* refers to the number of feature fields for original features, *#value* refers to the number of feature values for original features, *pos ratio* refers to the positive ratio.

Criteo dataset consists of ad click data over a week. It consists of 26 categorical feature fields and 13 numerical feature fields. Following the best practice [26], we discretize each numeric value x to $\lfloor \log^2(x) \rfloor$, if $x > 2$; $x = 1$ otherwise. We replace infrequent categorical features with a default "OOV" (i.e. out-of-vocabulary) token, with *min_count*=2.

Avazu dataset contains 10 days of click logs. It has 24 fields with categorical features. Following the best practice [26], we remove the *instance_id* field and transform the *timestamp* field into three new fields: *hour*, *weekday* and *is_weekend*. We replace infrequent categorical features with the "OOV" token, with *min_count*=2.

KDD12 dataset contains training instances derived from search session logs. It has 11 categorical fields, and the click field is the number of times the user clicks the ad. We replace infrequent features with an "OOV" token, with *min_count*=10.

B.2 Parameter Setup

To ensure the reproducibility of experimental results, here we further introduce the implementation setting in details. We implement our methods using PyTorch. We adopt the Adam optimizer with a mini-batch size of 4096. We set the embedding sizes to 16 in all the models. We set the predictor as an MLP model with [1024, 512, 256] for all methods. All the hyper-parameters are tuned on the validation set with a learning rate from [1e-3, 3e-4, 1e-4, 3e-5, 1e-5] and weight decay from [1e-4, 3e-5, 1e-5, 3e-6, 1e-6]. We also tune the learning ratio for the feature interaction selection parameters from [1e-4, 3e-5, 1e-5, 3e-6, 1e-6] and while weight decay from [1e-4, 3e-5, 1e-5, 3e-6, 1e-6, 0]. The initialization parameters for the retraining stage is selected from the best-performed model parameters and randomly initialized ones.

B.3 Hardware Platform

All experiments are conducted on a Linux server with one Nvidia-Tesla V100-PCIe-32GB GPU, 128GB main memory and 8 Intel(R) Xeon(R) Gold 6140 CPU cores.

C Ablation Study

C.1 Feature Interaction Operation

In this section, we conduct an ablation study on the feature interaction operation, comparing the performance of the default setting, which uses the *inner product*, with the *outer product* operation. We evaluate these operations on *OptFeature* and its two variants: *OptFeature-f* and *OptFeature-v*. The results are summarized in Table 4.

Table 4: Performance Comparison over Different Feature Interaction Operation.

Dataset		Criteo		Avazu		KDD12	
Category	Model	AUC	Logloss	AUC	Logloss	AUC	Logloss
inner product	OptFeature-f	0.8115	0.4404	0.7920	0.3744	0.7978	0.1530
	OptFeature-v	0.8116	0.4403	0.7920	0.3742	0.7981	0.1529
	OptFeature	0.8116	0.4402	0.7925	0.3741	0.7982	0.1529
outer product	OptFeature-f	0.8114	0.4404	0.7896	0.3760	0.7957	0.1535
	OptFeature-v	0.8113	0.4405	0.7902	0.3752	0.7961	0.1533
	OptFeature	0.8115	0.4403	0.7899	0.3753	0.7961	0.1533

From the table, we observe that the *inner product* operation outperforms the *outer product* operation. This performance gap is particularly significant on the Avazu and KDD12 datasets, while it is relatively insignificant on the Criteo dataset. The drop in performance with the *outer product* operation is likely due to the introduction of a significantly larger number of inputs into the final predictor. This makes it more challenging for the predictor to effectively balance the information from raw inputs and feature interactions.

C.2 Dimension Selection

In this section, we perform an ablation study on the feature interaction selection dimension $\hat{d}$. We compare the AUC performance with the corresponding dimension $\hat{d}$ and present the results in Figure 5. From the figure, we can observe that as the dimension $\hat{d}$ increases, the AUC performance remains relatively consistent over the Criteo dataset. This suggests that it is relatively easy to distinguish value-level selection on the Criteo dataset. However, on the Avazu and KDD12 datasets, the AUC performance improves as the selection dimension $\hat{d}$ increases. This indicates that distinguishing informative values is comparatively more challenging on these two datasets.

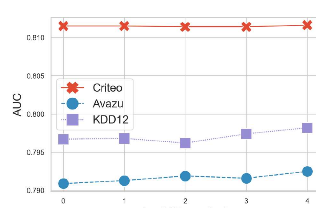
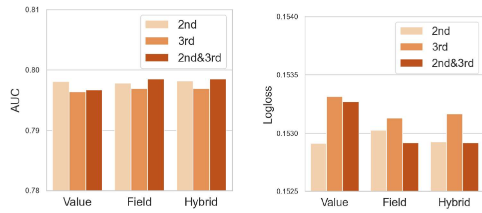


Figure 5: Ablation over feature interaction selection dimension on OptFeature.

C.3 Higher-Order Feature Interactions

In this section, we investigate the influence of higher-order feature interactions over the final results on the KDD12 dataset. We compare the default setting where only considering second-order interactions with two other settings: (i) only third-order interactions and (ii) both second and third-order interactions. We visualize the result in Figure 6.

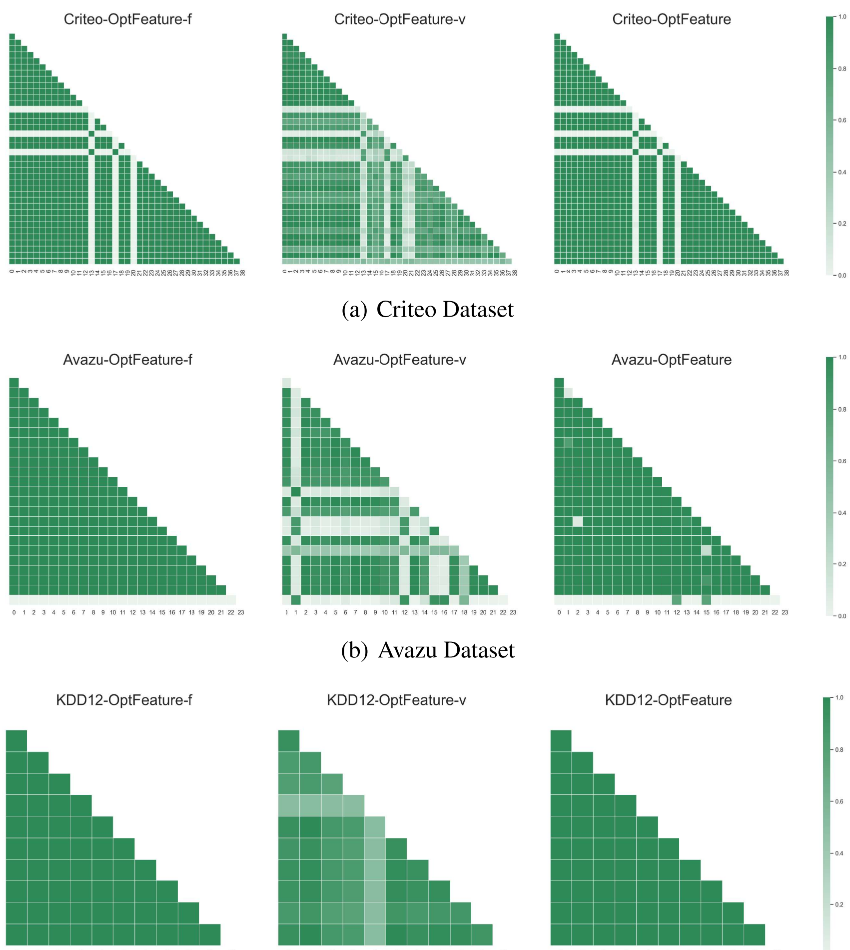


(a) AUC (b) Logloss

Figure 6: Performance Comparison over Different Feature Interaction Orders.

From the figure, we can draw the following observations. First, only considering third-order interactions leads to the worst performance. This aligns with the common understanding that second-order interactions are typically considered the most informative in deep sparse prediction [13]. Second, for field-level selection, the performance improves when both second and third-order interactions are incorporated into the model. This finding is consistent with previous studies [10, 6], as the inclusion of additional interactions introduces extra information that enhances the performance. In contrast, for value-level selection, the performance tends to decrease when both second and third-order interactions are included. This could be attributed to the fact that value-level selection operates at a finer-grained level and might be more challenging to optimize directly. Finally, OptFeature constantly outperforms its two variants over all settings. This indicates the feasibility and effectiveness of hybrid-grained selection, which combines both field-level and value-level interactions.

C.4 Selection Visualization



(a) Criteo Dataset

(b) Avazu Dataset

(c) KDD12 Dataset

Figure 7: Visualization of the Feature Interaction Selection Results.

In this section, we present the visualization of the interaction selection results for OptFeature and its two variants in Figure 7. OptFeature-f performs a binary selection for each interaction field, allowing for easy visualization through a heatmap representation where one indicates keep and zero indicates drop. On the other hand, OptFeature-v and OptFeature involve value-level interaction selection. Hence, we visualize them by setting each element as the percentage of being selected over the training set. The detailed equation for calculating the value for interaction field (i, j) is shown in Equation 16.

$$\mathbf{P}_{(i,j)} = \frac{\#\text{Samples keeping interaction field (i, j)}}{\#\text{Training Samples}} \tag{16}$$

From the visualization, we can observe that OptFeature acts as a hybrid approach, exhibiting a combination of both field-level and value-level interactions. Interestingly, we note significant differences between certain interaction fields in the KDD12 and Avazu datasets. OptFeature-f retains all of its interactions, while OptFeature-v only keeps a proportion of the value-level interactions. This observation further emphasizes the importance of exploring interactions at a finer-grained level.

D Broader Impact

Successfully identifying informative feature interactions could be a double-edged sword. One the one hand, by proving that introducing noisy features into model could harm the performance, feature interaction selection could be used as supporting evidences in preventing certain business, such as advertisement recommendation, from over-collecting users' information, thereby protecting user privacy. On the other hand, these tools, if acquired by malicious people, can be used for filtering out potential victims, such as individuals susceptible to email fraud. As researchers, it is crucial for us to remain vigilant and ensure that our work is directed towards noble causes and societal benefits.