ALF: Advertiser Large Foundation Model for Multi-Modal Advertiser Understanding

Santosh Rajagopalan*
Google
Mountain View, CA, USA
yrj@google.com

Jonathan Vronsky*
Google
Mountain View, CA, USA
jvronsky@google.com

Songbai Yan
Google
Mountain View, CA, USA
songbai@google.com

S. Alireza Golestaneh
Google
Mountain View, CA, USA
alirezag@google.com

Shubhra Chandra
Google
Mountain View, CA, USA
shubhrac@google.com

Min Zhou
Google
Mountain View, CA, USA
minzhouis@google.com

Abstract

We present ALF (Advertiser Large Foundation model), a multi-modal transformer architecture for understanding advertiser behavior and intent across text, image, video, and structured data modalities. Through contrastive learning and multi-task optimization, ALF creates unified advertiser representations that capture both content and behavioral patterns. Our model achieves state-of-the-art performance on critical tasks including fraud detection, policy violation identification, and advertiser similarity matching. In production deployment, ALF demonstrates significant real-world impact by delivering simultaneous gains in both precision and recall, for instance boosting recall by over 40 percentage points on one critical policy and increasing precision to 99.8% on another. The architecture's effectiveness stems from its novel combination of multi-modal transformations, inter-sample attention mechanism, spectrally normalized projections, and calibrated probabilistic outputs.

CCS Concepts

• **Computing methodologies → Machine learning**; • **Information systems → Online advertising**.

Keywords

Advertiser Understanding; Structured Data

ACM Reference Format:
Santosh Rajagopalan, Jonathan Vronsky, Songbai Yan, S. Alireza Golestaneh, Shubhra Chandra, and Min Zhou. 2026. ALF: Advertiser Large Foundation Model for Multi-Modal Advertiser Understanding. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD 2026), August 9–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3770854.3783927

1 Introduction

Online advertising platforms serve as the economic engine of the modern web. A core challenge in this ecosystem is to accurately and efficiently understand advertiser intent and behavior. This understanding is critical for several key applications, including matching users with ads and identifying fraud and policy violations. Addressing this challenge requires a holistic approach, processing diverse data types including structured account information (e.g., account age, billing details), multi-modal ad creative assets (text, images, videos), and landing page content. For example, an advertiser might have a recently created account, have text and image ads for a well-known large brand, and have had a credit card payment declined once. Although each element could exist innocently in isolation, the combination strongly suggests a fraudulent operation.

Developing such a system presents several key challenges:

- **Heterogeneous and high-dimensional data:** Advertiser data is inherently heterogeneous, encompassing structured (numerical, categorical, univalent, and multivalent) and unstructured (text, image, video) data. Each data type has different scales, statistical properties, and semantic interpretations. Furthermore, the sheer number of features can lead to a high-dimensional representation, creating computational and statistical challenges. Effectively integrating and modeling these disparate data types to create a unified representation, while preserving the informational richness of each, requires a sophisticated approach.
- **Unbounded sets of creative assets:** Advertisers can have a huge and highly variable number of creative assets. Malicious actors may even exploit this by obfuscating fraudulent assets within a large volume of innocent ones. The system must be able to process an unbounded and potentially adversarial set of assets, identifying malicious content even when hidden among numerous benign creatives, without being overwhelmed by the volume of data.
- **Real-world reliability and trustworthiness:** Predictions made by the system directly impact real businesses and users. Therefore, the system must be highly reliable and provide well-calibrated confidence scores to ensure that the model's outputs are interpretable and actionable. Producing calibrated probabilities reduces the overall system-wide operational complexity by allowing downstream components

[*] Both authors contributed equally to this research.

KDD 2026, Jeju Island, Republic of Korea.

ACM ISBN 979-8-4007-2258-5/2026/08
https://doi.org/10.1145/3770854.3783927

to use these probabilities to make decisions (for example, should we suspend the account, or ask for identification from this advertiser?) without such components needing to be re-tuned after every model upgrade.

These challenges are not adequately addressed by prior work. Domain-specific solutions, such as those for creative asset classification [12], are too narrow and ignore the rich interactions between different data types. On the other hand, recent transformer-based tabular models are not equipped to handle multi-modal inputs, while large multi-modal models are architecturally ill-suited for processing massive, tabular-heavy datasets and are not optimized for the kind of scalable, calibrated predictions required in production environments.

To bridge this critical gap, we present ALF (Advertiser Large Foundation model), a unified transformer architecture for real-world advertiser understanding. ALF creates a holistic understanding of advertisers by jointly modeling their structured data and multi-modal assets. Our primary contributions, which directly address the challenges of heterogeneity, scale, and reliability, are as follows:

- **An Efficient Holistic Architecture for Heterogeneous Data:** We propose an efficient, holistic architecture featuring a specialized transformer. Its unified encoding strategy performs an early fusion where structured data and multi-modal embeddings are mapped into a shared space, enabling a dual-attention mechanism to learn deep cross-modal interactions. This mechanism includes a scalable, projection-based inter-sample attention that overcomes the hidden dimension and input length scaling limitations of prior work [28], allowing the model to effectively learn from interactions across large advertiser batches.
- **Scalable Handling of Unbounded Creative Assets:** To efficiently manage a large and variable number of creatives, we propose to process pre-computed asset embeddings instead of raw bytes and uses a top-k selection mechanism. This ensures robust and scalable performance, particularly in adversarial scenarios.
- **Trustworthy Predictions for Real-World Deployment:** We integrate Spectrally-Normalized Neural Gaussian Process (SNGP) heads during fine-tuning. This produces well-calibrated probabilistic outputs, which are crucial for high-stakes decision-making and simplify integration with downstream production systems.

Our experiments show ALF significantly outperforms a heavily-tuned production baseline while also performing strongly on public benchmarks. In production, ALF delivers substantial and simultaneous gains in precision and recall, boosting recall by over 40 percentage points on one critical policy while increasing precision to 99.8% on another. This performance lift is driven by ALF's unique ability to integrate multi-modal creative content with tabular features, a task where traditional architectures falter.

The remainder of this paper is organized as follows. Section 2 discusses related work in multi-modal learning and advertiser understanding. Section 3 formulates the problem space. Section 4 describes our model architecture in detail. Section 5 presents our training methodology, while Section 6 provides comprehensive experimental results and analysis. Finally, Section 7 concludes with a discussion of future work.

2 Related Work

2.1 Multi-modal Learning

Recent work in multi-modal learning has demonstrated the effectiveness of transformer architectures in combining different data types. Models like CLIP [24] and DALL-E [25] have shown strong performance on joint image-text tasks through contrastive learning approaches. More recently, large generative multi-modal models, such as Flamingo [1], FLAVA [27], VLMo [3], OmniVL [31] introduced a unified foundation model for visual and language tasks, showing effective zero-shot transfer across multiple tasks. However, such large models are computationally expensive at our scale because they operate on limited number of raw multi-modal data (e.g., image pixels) rather than dense embeddings. Our approach, in contrast, leverages efficient, pre-computed embeddings as inputs to a specialized predictive architecture.

In the advertising domain, Hussain et al. [12] developed techniques for creative asset classification, while Rayavarapu et al. [26] investigated transformers for detecting bad quality ads. However, these efforts primarily focused on isolated modalities or specific tasks, unlike our holistic approach that integrates all available advertiser signals.

2.2 Transformers for Structured Data

Traditional approaches to structured data typically rely on gradient-boosted decision trees (GBDT) [5, 15]. However, recent work has shown promising results using transformers for tabular data. TabTransformer [11] demonstrated strong performance through categorical embedding learning, while SAINT [28] showed particular effectiveness through intersample attention mechanisms.

FT-Transformer [9] provided a comprehensive analysis of transformer architectures for tabular data, showing competitive performance against traditional approaches. Recent work by Yoon et al. [35] on self-supervised learning for tabular data has shown the importance of handling missing values and heterogeneous features. Work on Non-Parametric Transformers (NPT) [16], in addition to SAINT [28], shows the importance of cross-attention across data points in transformers for tabular data.

Our work builds on this line of research by adapting the transformer architecture with cross-attention to the specific challenges of advertiser modeling. Specifically, we extend the model's ability to handle both structured and unstructured multi-modal data of large volume and dimensionality, and introduce techniques for generating calibrated predictions, critical for real-world deployment.

A recent line of work has explored the application of large language models (LLMs) to numerical data using custom tokenization schemes (e.g., P10) [29]. Similar to the large multi-modal models mentioned previously, these methods suffer from significant cost and scaling issues.

2.3 Advertiser Understanding

Prior work on advertiser understanding has focused on specific aspects of the advertising ecosystem. Dave et al. [6] developed methods for click fraud detection using temporal patterns and behavioral analysis. Landing page and image categorization for display ads has been explored by Kae et al. [14].

Deep learning approaches to ad fraud detection have been investigated by Gopali et al. [8] using sequential models and Xu et al. [33] through graph-based methods. Policy violation detection in online advertising has been studied by Mittal et al. [20] using Foundation models and Qu et al. [23] through multi-task learning approaches.

Our work differs from these approaches by providing a unified model that jointly processes all advertiser-related signals, including structured data, creative assets, and behavioral patterns. This unified approach, purpose-built for Advertiser understanding, allows for better feature interaction and knowledge transfer across tasks.

3 Problem Formulation

In this paper, we consider an advertiser understanding task, where each example is a snapshot of an advertiser x drawn from a distribution D, consisting of:

- Structured features $s_x \in \mathbb{R}^d$ including categorical and numerical, and possibly multivalent, account attribute and historical performance metrics;
- Text content $T_x = \{t_1, ..., t_n\}$ including ad text, keywords, and landing page text;
- Image content $I_x = \{i_1, ..., i_m\}$ from ad creatives, landing pages (and landing page screenshots);
- Video content $V_x = \{v_1, ..., v_k\}$ from video ads.

We are given m tasks: for the t-th task, we assume that there is a ground truth label y^t drawn from an unknown distribution $D_t(Y \mid X = x)$. We are given a loss function l_t and we would like to find a model m that minimizes $\mathbb{E}_{D_t}[l_t(m(x), y^t)]$. In this paper, we focus on multiclass classification tasks, but our method can be easily generalized to broader tasks.

4 Model Architecture

The ALF architecture, shown in Figure 1, constructs a unified advertiser representation from heterogeneous, multi-modal data sources. It begins by projecting heterogeneous raw features – including structured numerical/categorical data (both univalent and multivalent), and unstructured text, image, and video—into a common embedding space using a novel encoding approach. This enables ALF to handle a variety of advertiser data in a scalable manner. A novel dual-attention transformer encoder, combining self-attention with scalable inter-sample attention, then processes these embeddings to produce a robust advertiser embedding. During pre-training, we apply FFN and projection layers over the advertiser embedding to train ALF via self-supervised reconstruction and contrastive learning tasks. For downstream prediction tasks in fine-tuning, ALF employs task-specific, Spectrally-Normalized Neural Gaussian Process (SNGP) heads. These SNGP heads provide calibrated probabilistic outputs, crucial for reliable, risk-aware decision-making in advertising. We explain each component in the following two sections.

4.1 Input Processing

We have a heterogeneous feature space with structured categorical and numerical features, and possibly unbounded number of advertiser ad texts, images, and videos. Common approaches to model such heterogeneous features and the interactions among them are either through gradient boosted decision trees [5, 15] or variants of factorized machines [32], but they often struggle with the scale and complexity of modern advertising data, or limit cross-modal interactions. In contrast, ALF leverages a Transformer architecture backbone. To harness this power, we introduce a novel encoding approach (depicted in Figure 2) that transforms all input modalities into a unified, shared embedding space. This scalable encoding is essential for the downstream dual-attention mechanism and forms the foundation of our coherent customer representation.

It is important to note that positional encoding is not used at this stage. The input to the transformer is treated as an unordered set of features, as the structured data and asset embeddings do not have an inherent sequential order. Any positional information relevant to sequence-based modalities like text or video is handled by the upstream models that generate the asset embeddings.

4.1.1 Structured Data Encoding. Following Somepalli et al. [28], we map each individual structured feature into d dimensional space. For numeric features, unlike [28] which uses multilayer perceptrons to map 1-dimensional scalars into d dimensional space, here we directly encode them using sinusoidal positional encoding. This choice offers a good balance of non-linear transformation capability and parameter efficiency, which is crucial for handling a large number of numerical features:

$$\mathrm{SE}(x, 2i) = \sin(x \cdot f_i \cdot \pi) \tag{1}$$

$$\mathrm{SE}(x, 2i+1) = \cos(x \cdot f_i \cdot \pi) \tag{2}$$

where f_i $(i = 1, \ldots, d)$ are learned frequency parameters. This reduces the number of parameters for encoding and helps scale up the number of features we could support, while our empirical evaluation shows comparable performance to MLPs.

For categorical features, we use learnable embeddings $E_c \in \mathbb{R}^{|V| \times d}$ where $|V|$ is the vocabulary size. Embeddings for multi-label features are combined through summation:

$$e_{\text{cat}} = \sum_{i \in C} E_c[i] \tag{3}$$

4.1.2 Multi-modal Encoders. Text, image and video contents are converted to embeddings using transformer-based encoders. Here we adopt multi-lingual LaBSE [7] for texts and GRAPH-RISE [13] for images and videos, but we expect any encoder trained to maximize the entropy of the generated embeddings [4, 17] will work equivalently.

$$e_{\text{text}} = \mathrm{LaBSE}(t) \in \mathbb{R}^{d_t} \tag{4}$$

$$e_{\text{image}} = \mathrm{ImageEncoder}(i) \in \mathbb{R}^{d_i} \tag{5}$$

$$e_{\text{video}} = \mathrm{VideoEncoder}(v) \in \mathbb{R}^{d_v} \tag{6}$$

These embeddings are then mapped to d-dimensional space (same as the encoding for structured data) via multilayer perceptrons.

4.1.3 Selecting Advertiser Assets and Landing Pages. An advertiser can have a potentially unbounded number of assets, such as images, videos, and landing pages. Including all of them would increase the cost of both training and inference. Instead, we select the top-k

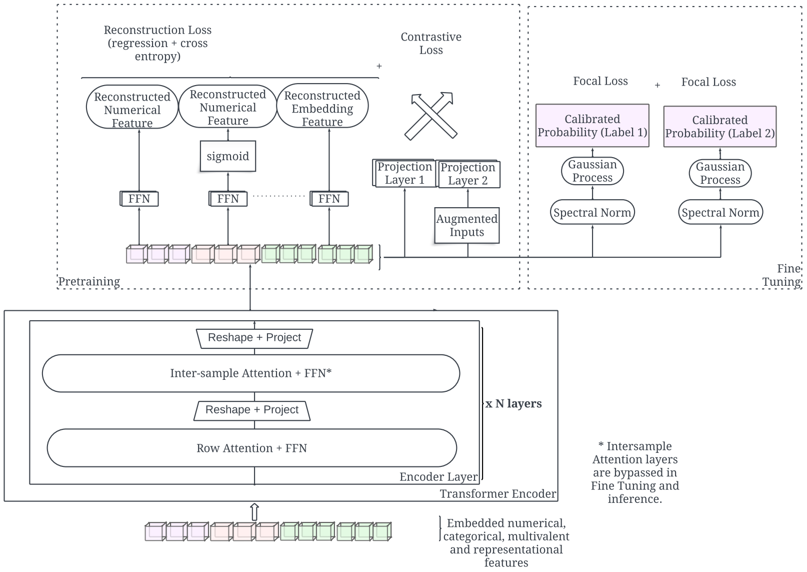
Reconstruction Loss (regression + cross entropy)
+
Contrastive Loss
Reconstructed Numerical Feature
Reconstructed Numerical Feature
Reconstructed Embedding Feature
sigmoid
FFN
FFN
FFN
Projection Layer 1
Projection Layer 2
Augmented Inputs
Focal Loss
+
Focal Loss
Calibrated Probability (Label 1)
Calibrated Probability (Label 2)
Gaussian Process
Gaussian Process
Spectral Norm
Spectral Norm
Pretraining
Fine Tuning
Reshape + Project
Inter-sample Attention + FFN*
Reshape + Project
Row Attention + FFN
x N layers
Encoder Layer
Transformer Encoder
* Intersample Attention layers are bypassed in Fine Tuning and inference.
Embedded numerical, categorical, multivalent and representational features

Figure 1: ALF model architecture showing the multi-modal encoders, dual attention mechanisms, and output heads.

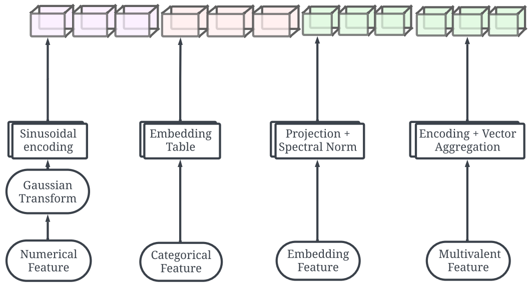

Figure 2: ALF input processing for each feature type.

most representative assets of an advertiser. The top-k selection method employed depends on the specific application and could include selecting cluster centroids and outliers, or selecting based on recency, user engagement, and content type. These could be supplemented with a random selection of assets as well to give a comprehensive understanding of the advertiser and their intent.

4.2 Feature Selection

The number of structured features for an advertiser can be very high, encompassing a wide range of metadata, attributes, and derived metrics. To ensure that our model focuses on the most relevant signals and to maintain computational efficiency, we employ a preliminary feature selection step before the main training pipeline. This step uses a backward elimination approach, where a Random Forest model trained with the Yggdrasil Decision Forests (YDF) library [10][1] iteratively prunes the least important features based on their importance scores.

This feature selection process is not part of the core ALF training pipeline but is a crucial pre-processing step that allows us to work with a more manageable and informative set of structured features. The final selected features are then used as input to the main ALF model, where they are combined with the multi-modal embeddings for the dual-attention transformer.

4.3 Scalable Dual Attention Transformer Architecture

ALF uses a modified transformer architecture with dual attention mechanisms. Define the attention function:

$$\text{Attn}(X; W_Q, W_K, W_V) = \text{softmax}\left(\frac{(XW_Q)(XW_K)^T}{\sqrt{d}}\right)(XW_V) \quad (7)$$

Recall that in standard row attention, for a batch of examples $X \in \mathbb{R}^{B\times N\times d}$ of B samples, we directly apply the attention function on X with three trainable matrices $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$. As a result, $Q, K, V = \text{Attn}(Q, K, V)$ are all tensors in $X \in \mathbb{R}^{B\times N\times d}$.

Inspired by [28], we apply inter-sample attention within a batch to incorporate information about the distribution for each individual feature, which would be helpful if a feature is missing or noisy for one sample. We propose here a more scalable version of inter sample attention (ISA) to address issues we observed going to higher dimensions, features and batch sizes.

After the initial attention and FFN layers, we reshape and project the Nd-dimensional embeddings to a lower dimension d' using a learnable projection matrix $P \in \mathbb{R}^{Nd\times d'}$ to improve computational efficiency:

$$X_{\text{reshaped}} = \text{reshape}(X, (1, B, Nd)) \quad (8)$$

$$X_{\text{proj}} = X_{\text{reshaped}}P \quad (9)$$

[1] https://github.com/google/yggdrasil-decision-forests

where $X_{\text{proj}} \in \mathbb{R}^{1\times B\times d'}$. This projection reduces the computational complexity of both the subsequent attention operation from $O(B^2Nd)$ to $O(B^2d')$ and the FFN layer from $O(BNd^2)$ to $O(Bd'^2)$, making it more scalable for larger batch sizes and feature dimensions.

Next, we apply the attention mechanism on the projected embedding X_{proj}. Note that here the second dimension of X_{proj} is of size B, so the attention here is inter-sample within the training batch to supplement the embedding vector with information from other examples in the batch for missing or noisy features.

$$A_{\text{IS}} = \text{Attn}(X_{\text{proj}}, W'_Q, W'_K, W'_V) \tag{10}$$

The projected representations then pass the FFN layer while maintaining the reduced dimensionality d'. After the FFN, we project back to the original dimension using a learnable matrix $P_{\text{restore}} \in \mathbb{R}^{d'\times Nd}$:

$$X_{\text{restored}} = \text{FFN}(A_{\text{IS}})P_{\text{restore}} \tag{11}$$

$$X_{\text{final}} = \text{reshape}(X_{\text{restored}}, (B, N, d)) \tag{12}$$

This projection-based architecture substantially improves computational efficiency while preserving the model's ability to capture inter-sample relationships.

As shown in Figure 1, we stack N such layers as our Dual Attention Transformer Encoder.

Note that inter-sample attention layers are bypassed in the fine tuning and inference stage to avoid cross-advertiser information leakage.

4.4 Spectral Normalization

All projection layers use spectral normalization to keep Lipschitz constants bounded:

$$W_{\text{SN}} = \frac{W}{\sigma(W)} \tag{13}$$

where $\sigma(W)$ is the spectral radius (largest singular value) of matrix W, computed efficiently using power iteration [21]. This normalization ensures that the Lipschitz constant of each layer remains bounded, which:

- Stabilizes training by preventing extreme variations in gradients
- Makes the embedding space more isotropic by avoiding dimension collapse
- Improves robustness of the learned representations against input perturbations

The power iteration method approximates $\sigma(W)$ using:

$$u_{t+1} = \frac{W^\top u_t}{\|W^\top u_t\|_2} \tag{14}$$

$$v_{t+1} = \frac{Wu_{t+1}}{\|Wu_{t+1}\|_2} \tag{15}$$

$$\sigma(W) \approx u_{t+1}^\top W v_{t+1} \tag{16}$$

where u_t, v_t are the left and right singular vectors corresponding to the largest singular value.

5 Training

The model is trained in two stages: a pre-training stage with self-supervised representation learning and a fine-tuning stage for supervised applications.

5.1 Pre-training

The model is pre-trained on over 100 million advertiser snapshots to learn a robust representation via a contrastive learning task and a reconstruction task after data augmentation.

Specifically, we employ two augmentation strategies:

- CutMix[36]: Combines different samples in input space
  $$x' = M \odot x_i + (1 - M) \odot x_j$$
  where x_i and x_j are two samples in the input space, and M is a binary mask drawn from a Bernoulli distribution $B(0.2)$.
- MixUp[37]: Interpolates samples in latent space
  $$h' = \alpha h_i + (1 - \alpha)h_j \tag{17}$$
  where h_i and h_j are two embedding vectors for two samples after the feature encoder but before the transformer encoder, and α is a weight parameter that can be tuned.

After data augmentation, we train the model to minimize a linear combination of two losses:

- Reconstruction losses: For augmented input $\tilde{x}$, we use several reconstruction objectives for each of the feature type:
  $$\mathcal{L}_{\text{num}} = \|f_{\text{num}}(g(\tilde{x})) - x_{\text{num}}\|^2 \tag{18}$$
  $$\mathcal{L}_{\text{ce}} = -\sum_i x_i \log(f_{\text{ce}}(g(\tilde{x}))_i) \tag{19}$$
  $$\mathcal{L}_{\text{mcat}} = -\sum_i \sum_j x_{ij} \log(f_{\text{mcat}}(g(\tilde{x}))_{ij}) \tag{20}$$
  $$\mathcal{L}_{\text{emb}} = \|f_{\text{emb}}(g(\tilde{x})) - x_{\text{emb}}\|^2 \tag{21}$$
  $$\mathcal{L}_{\text{memb}} = \sum_i \|f_{\text{memb}}(g(\tilde{x}))_i - x_{\text{memb},i}\|^2 \tag{22}$$
  where f_{num}, f_{ce}, f_{mcat}, f_{emb}, f_{memb} and g are the corresponding decoder and encoder functions, $\mathcal{L}_{\text{num}}$ is the MSE Loss on the reconstruction of numerical features, $\mathcal{L}_{\text{ce}}$ is the Cross-Entropy Loss for categorical feature reconstruction, $\mathcal{L}_{\text{mcat}}$ is the multivalent Categorical Loss for multi-label scenarios, $\mathcal{L}_{\text{emb}}$ is the MSE loss for pre-trained embeddings, and $\mathcal{L}_{\text{memb}}$ is the MSE loss for multivalent embeddings where each feature can have multiple embedding vectors. These reconstruction losses, in conjunction with the encoder g and their respective decoders (f_{num}, f_{ce}, etc.), ensure that the encoder is information-preserving by accurately reconstructing the various input feature types.
- Contrastive loss: We use InfoNCE loss between original and augmented samples:
  $$\mathcal{L}_{\text{con}} = -\log \frac{\exp(s(x, x^+)/\tau)}{\sum_{x^-} \exp(s(x, x^-)/\tau)} \tag{23}$$
  where x is the original input, x^+ is the augmented input of x, x^- is the augmentation of another input drawn within the batch, $s(\cdot,\cdot)$ is cosine similarity and τ is a temperature parameter.

The total loss is then computed as:

$$\mathcal{L}_{\text{total}} = \alpha_{\text{num}}\mathcal{L}_{\text{num}} + \alpha_{\text{ce}}\mathcal{L}_{\text{ce}} + \alpha_{\text{mcat}}\mathcal{L}_{\text{mcat}} + \alpha_{\text{con}}\mathcal{L}_{\text{con}} + \alpha_{\text{emb}}\mathcal{L}_{\text{emb}} + \alpha_{\text{memb}}\mathcal{L}_{\text{memb}} \tag{24}$$

where α_i are the corresponding scaling factors for each loss term.

5.2 Fine-tuning

After pre-training, we fine-tune the model for specific downstream tasks using a Spectral-normalized Neural Gaussian Process (SNGP) layer [19][2]. SNGP combines spectral normalization with a random feature approximation of a Gaussian process, enabling both calibrated uncertainty estimation and robust predictions far from the training distribution.

For each task t, we use focal loss [18] to handle class imbalance:

$$\mathcal{L}_{\text{focal},t} = -\sum_{i=1}^{N}(1-p_i)^{\gamma} y_i \log(p_i) \tag{25}$$

where p_i is the model's predicted probability for the correct class, y_i is the ground truth label, and γ is the focusing parameter that reduces the relative loss for well-classified examples.

A potential concern with focal loss is its tendency to degrade model calibration. In our framework, this is mitigated by the SNGP head, which is the primary component responsible for calibration. SNGP yields reliable uncertainty estimates, making it particularly effective for out-of-distribution detection [19]. We find the two methods to be complementary: SNGP ensures that model outputs are well-calibrated, while focal loss improves accuracy on in-distribution data by addressing severe class imbalance. This combination allows us to achieve both high accuracy on our imbalanced dataset and reliable calibration[34].

The total fine-tuning loss is then computed as sum across all tasks:

$$\mathcal{L}_{\text{finetune}} = \sum_{t} \mathcal{L}_{\text{focal},t} \tag{26}$$

5.3 Inference

During inference, the inter-sample attention layers are bypassed. This ensures that predictions for a single advertiser are self-contained and can be made without requiring a batch of other advertisers. This architectural choice allows for efficient, on-demand evaluation of individual advertisers in a production environment. The additional runtime overhead of ALF compared to simpler models is primarily due to the transformer backbone itself, as the conversion of creative assets into embeddings occurs asynchronously and does not impact online inference latency.

6 Experiments

6.1 Evaluation with Public Datasets

We validate ALF against state-of-the-art competitors using a suite of public benchmark datasets. However, since public data combining structured and multi-modal inputs is not readily available, these benchmarks primarily serve to demonstrate baseline competitiveness. ALF's core strengths—specifically its ability to integrate multi-modal signals at scale—are comprehensively evaluated on proprietary production data in the next subsection.

Dataset Name	Datapoints	Features	Positive Class%
albert	425240	79	50
dota2games	92650	117	52.7
adult	34190	25	85.4
blastchar	7043	20	26.5
1995 income	32561	14	24.1

Table 1: Benchmark datasets. All datasets are binary classification tasks. Positive Class% is the fraction of data points that belongs to the positive class.

Dataset Name	albert	dota2games	adult	blastchar	1995_income
MLP	0.740 ± 0.001	0.631 ± 0.002	0.725 ± 0.010	0.839 ± 0.010	0.905 ± 0.003
Sparse MLP	0.741 ± 0.001	0.633 ± 0.004	0.740 ± 0.007	0.842 ± 0.015	0.904 ± 0.004
TabTransformer	0.757 ± 0.002	0.633 ± 0.002	0.737 ± 0.009	0.835 ± 0.014	0.906 ± 0.003
TabNet	0.705 ± 0.005	0.529 ± 0.025	0.663 ± 0.016	0.816 ± 0.014	0.875 ± 0.006
VIB	0.737 ± 0.001	0.628 ± 0.003	0.733 ± 0.009	0.842 ± 0.012	0.904 ± 0.003
Logistic Regression	0.726 ± 0.001	**0.634 ± 0.003**	0.721 ± 0.010	0.844 ± 0.010	0.899 ± 0.002
GBDT	0.763 ± 0.001	0.621 ± 0.004	**0.756 ± 0.011**	0.847 ± 0.016	0.906 ± 0.002
ALF	**0.773 ± 0.007**	0.621 ± 0.005	0.733 ± 0.005	**0.848 ± 0.012**	**0.911 ± 0.002**

Table 2: AUC score for models on the benchmark datasets. Values are the mean over 5 cross-validation splits, plus or minus the standard deviation. Larger values mean better result.

Datasets. We selected a set of public datasets that are standard benchmarks in the field of tabular deep learning. We considered only datasets with large number of examples and/or features. The details of these datasets are summarized in Table 1, with their public source links provided in Appendix Table 2.

Baselines. We consider classical models for tabular data, including GBDT, Logistic Regression, MLP, Sparse MLP [22], and recently proposed TabTransformer [11] and the Variational Information Bottleneck model [2] as baselines. To ensure a fair and direct comparison, the performance results for all baselines are cited from [11], reflecting the outcomes of their original hyperparameter optimization. We adopt the same methodology for ALF, i.e., using 5-fold cross validation splits to train and reporting mean and standard deviation of AUCROC over the 5 splits.

Results. As shown in Table 2, ALF outperforms baseline models on three of the five datasets and is competitive on the other two. This advantage was most evident on the large-scale Albert dataset (425,000+ samples, 78 features), where ALF's performance was significantly superior to all other models.

6.2 Evaluation with Production Data

We deployed ALF to the Google Ads Safety system to identify fraudulent advertisers who violate various ads policies. Here, we report online model metrics as well as offline analysis to demonstrate the effectiveness of our proposed method.

6.2.1 Methodology.

Data. The Google Ads datasets consist of daily snapshots of advertisers, capturing a comprehensive view of their attributes and activities at a specific point in time. Each snapshot contains structured data, text content from ads and landing pages, images,

[2] https://keras.io/examples/keras_recipes/uncertainty_modeling_with_sngp/

and available video content. We use daily snapshots to mitigate the noise from the continuous evolution of advertiser data. The data is split chronologically into training, validation, and test sets:

- Training Set: Over 100 million advertiser snapshots.
- Validation Set: Over 10 million snapshots, used for hyperparameter tuning.
- Test Set: Over 10 million snapshots from a future timeframe, ensuring a realistic evaluation of the model's predictive performance on unseen data.

Baselines. The baseline model considered is our previous production model, which is the result of an extensive neural architecture search and hyperparameter tuning. The architectures considered include DNNs, ensembles, GBDTs, and logistic regression with feature cross exploration. Notably, the standard architectures within this search space do not support pre-training, focal loss, or calibration via SNGP heads. For each model, the best-performing architecture and hyperparameters are selected automatically via Vizier [30] to ensure optimal performance.

We omit encoder-based models, such as FLAVA [27], TabTransformer [11], FT-Transformer[9], and SAINT [28], from baselines, since they are not designed to handle the combination of complex structured data, multivalent features, and the scale required for our application. As shown in Table 4, adapting our dataset to these models would require feature restrictions that would invalidate a fair comparison. We also exclude Large multi-modal generative models (e.g., Flamingo [1], VLMo [3]) from baselines. They are cost-prohibitive at our scale because they operate on raw data (e.g., image pixels) instead of embeddings.

Hyperparameters. We used Vizier [30] for hyperparameter optimization. A selection of the most interesting hyperparameters is presented in Table 6 in Appendix. With these settings, the total number of trainable parameters for our model is about 322M.

For individual ablation experiments, due to computational constraints, hyperparameters were not re-tuned. However, we observed that the model's performance was not highly sensitive to minor variations in hyperparameters.

Reporting. Due to the significant computational cost of training, we were unable to perform multiple runs for formal significance testing. However, all reported metrics were computed on a test set of over 10 million examples, which provides a high degree of statistical power for our comparisons.

6.2.2 *Production Metrics.* We successfully deployed our model to a production abuse detection system, where it has led to a significant reduction in policy violations across a range of abuse categories. The system's goal is a binary classification task: identifying advertisers engaged in user harm or financial abuse.

For three key financial abuse policies designated as Policy A, Policy B, and Policy C, as shown in Table 3, our model operates at similar or higher recall values while achieving significantly higher precision compared to the baseline production model.

Our latency as measured in production when running on CPUs is 29ms, which is higher than the 8ms observed for the baseline production model. Our model is also larger in parameter count and occupies more CPU memory than the baseline production model. However, both of these are within the acceptable bounds for the production workflow this model was deployed in. Running ALF on TPUs/GPUs should lower the latency for latency-sensitive applications.

	Policy A		**Policy B**		**Policy C**	
Metric	**Baseline**	**ALF**	**Baseline**	**ALF**	**Baseline**	**ALF**
Precision	95.3%	**99.8%**	80%	**88%**	99.2%	**99.5%**
Recall	85.7%	**92.4%**	35%	**37%**	21.0%	**64.0%**

Table 3: Performance metrics of ALF vs Baseline across 3 policies in production environment.

	Auprc	Auroc
ALF (proposed)	0.46	0.93
ALF without embedding (proposed)	0.42	0.91
Production	0.41	0.91
Production without embedding	0.45	0.74
Logistic Regression	0.43	0.91
Random Forest	0.40	0.90
GBDT	0.41	0.91
Deep Neural Network	0.38	0.87
Regression without embedding	0.34	0.87
Random Forest without embedding	0.42	0.91
GBDT without embedding	0.45	0.91
Deep Neural Network without embedding	0.35	0.88

Table 4: Performance comparison of our proposed model against commonly used methods on Policy C data.

6.2.3 *Offline Analysis.* We perform a deep-dive offline analysis on Policy C to benchmark our proposed model against several common tabular architectures. These architectures, shown in Table 1, are the candidates in our extensive neural architecture search and hyperparameter tuning process that resulted in the production baseline model. The evaluation was conducted with two distinct feature sets: one with multi-modal embedding features and one without them. As detailed in Table 4, our proposed model achieves superior AUPRC and AUROC performance compared to all other approaches. The results further reveal that a significant portion of this performance gain is directly attributable to the embedding features, highlighting their critical role in the model's success.

In addition, we evaluate our model on the Advertiser Understanding task, which involves using the embeddings generated by our model to measure the trustworthiness of Advertisers. As shown in Figure 3, the outputs of our model are meaningfully separated, supporting our claim that the embeddings from our model improve performance in tasks focused on calculating advertiser trustworthiness and clustering fraudulent advertisers.

6.2.4 *Ablation Studies.* In Table 5, we present ablation experiments to evaluate the contribution of each component in our proposed method, comparing their effects on a subset of our dataset. The table presents AUPRC performance for financial fraud (e.g., stolen credit

	Financial Fraud	User Harm	Average
Structured Features Only	0.8979	0.6262	0.7620
Content Features Only	0.4505	0.3080	0.3792
No Spectral Norm	0.8982	0.6260	0.7621
No Inter-Sample Attention	0.9075	0.6570	0.7822
No Contrastive Loss	0.9088	0.6568	0.7828
No Reconstruction Loss	0.9080	0.6581	0.7830
Our Proposed Model	0.9102	0.6672	0.7887

Table 5: Ablation experiments on the effects of different components of our proposed model, evaluated using the AUPR metric.

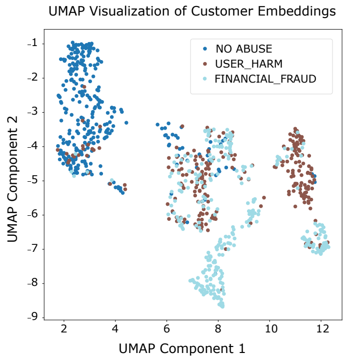

Figure 3: UMAP visualization of advertiser embeddings, colored by advertiser intent.

cards) and user harm (e.g., phishing websites), with an average in the third column. We observe that the contribution of content features is task-dependent: while structured data provides a strong baseline for financial fraud, content features are critical for user harm as expected, providing a relative AUPRC lift of nearly 7% (0.6262 to 0.6672). Furthermore, spectrally normalized projections, calibrated probabilistic outputs, Inter-Sample Attention, Contrastive Loss, and Reconstruction Loss are identified as crucial components, each significantly contributing to the model's overall performance.

We intentionally omit ablations on the choice of 'K' and the specific criteria for top-K asset selection. Revealing these details could provide adversaries with insights into our enforcement strategies, potentially enabling them to circumvent our detection systems.

6.2.5 Embedding Quality Analysis. Figure 3 shows the learned embeddings clearly clustered by advertiser intent in our final model. In contrast, Figure 4 shows the embeddings from a lower-quality model when we remove the inter-sample attention, where we can see that the Advertiser Intents are not well-clustered or separated. We observe a noticeable difference especially when it comes to separating user harm policies from financial fraud.

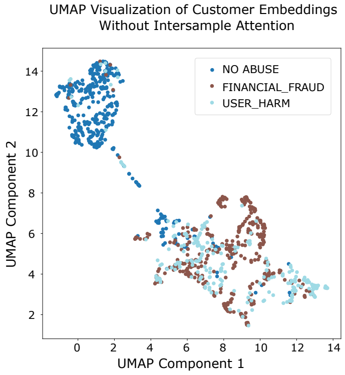

Figure 4: UMAP visualization of advertiser embeddings *without* intersample attention, colored by advertiser intent.

7 Conclusion

This paper addresses a critical gap between two lines of research: large multi-modal models, which are often impractical for predictive tasks on structured data due to their prohibitive cost at scale, and tabular models that cannot natively handle multi-modal signals. We bridge this gap through ALF, a framework that demonstrates the successful real-world deployment of a state-of-the-art transformer architecture for large-scale advertiser understanding. ALF is designed to directly solve the key industrial challenges of heterogeneous data, unbounded creative assets, and the need for real-world reliability. To tackle heterogeneous data, it uses an early-fusion, dual-attention mechanism; the inter-sample attention component is made computationally feasible at our scale through a scalable, projection-based approach we introduced, overcoming key limitations of prior work. For unbounded assets, it employs an efficient top-k selection on embeddings. To ensure reliability, it integrates SNGP heads for well-calibrated predictions.

The effectiveness of this approach was validated against an exceptionally strong production baseline, itself the result of an extensive search across various architectures and hyperparameters, including DNNs, ensembles, GBDTs, and logistic regression with feature cross exploration. While ALF's latency is higher due to its larger model size, it remains well within the acceptable range for our production environment and can be further optimized using hardware accelerators. Experiments show ALF significantly outperforms the baseline on key risk detection tasks, a performance lift driven by its unique ability to holistically model content embeddings, which simpler architectures struggled to leverage. This trade-off is justified by its successful deployment, where ALF serves millions of requests daily.

Future work could explore several promising directions. The current model could be extended to incorporate temporal dynamics to better detect evolving abuse patterns. Additionally, investigating scaling properties and simplifying the architecture to reduce complexity offers a valuable avenue for analyzing model size-performance trade-offs and broadening applicability. While our top-k approach for handling unbounded assets works well in practice, a theoretical analysis of its information loss could guide further improvements. Finally, the success of our unified architecture suggests its potential for a broader range of advertiser-related tasks, including creative optimization and audience modeling. In conclusion, ALF not only provides an effective solution for multi-modal entity modeling but also serves as a valuable case study for deploying advanced transformer architectures in complex, high-stakes industrial systems.

8 Acknowledgements

We extend our gratitude to Anish Das Sarma for funding and providing initial feedback for this effort. We thank Benoît Raybaud and Nihar Khedekar for leading the team that made this work possible. Special thanks to Dongjin Kwon, Alborz Mazloomian, Jossue Loubet, and Mike Martin for their tireless efforts in feature processing and for successfully bringing the model into production. We also thank Shengwei Xu for his insightful discussions and review of the manuscript. Finally, we acknowledge the invaluable support from Jun Gao's and Dake He's teams in bridging the gap between modeling and real-world business impact.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. arXiv:2204.14198 [cs.CV] https://arxiv.org/abs/2204.14198

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410* (2016).

[3] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. 2022. VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts. arXiv:2111.02358 [cs.CV] https://arxiv.org/abs/2111.02358

[4] Deep Chakraborty, Yann LeCun, Tim G. J. Rudner, and Erik Learned-Miller. 2024. Improving Pre-Trained Self-Supervised Embeddings Through Effective Entropy Maximization. *arXiv preprint arXiv:2411.15931* (2024). https://arxiv.org/abs/2411.15931 19 pages including appendix, 5 figures.

[5] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *CoRR* abs/1603.02754 (2016). arXiv:1603.02754 http://arxiv.org/abs/1603.02754

[6] Vacha Dave, Saikat Guha, and Yin Zhang. 2012. Measuring and Fingerprinting Click-Spam in Ad Networks. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. ACM, New York, NY, USA, 175–186. doi:10.1145/2342356.2342394

[7] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT Sentence Embedding. *arXiv preprint arXiv:2007.01852* (2020). https://arxiv.org/abs/2007.01852 To be presented at ACL 2022.

[8] Saroj Gopali, Akbar S. Namin, Faranak Abri, and Keith S. Jones. 2024. The Performance of Sequential Deep Learning Models in Detecting Phishing Websites Using Contextual Features of URLs. arXiv:2404.09802 [cs.CR] https://arxiv.org/abs/2404.09802

[9] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2023. Revisiting Deep Learning Models for Tabular Data. arXiv:2106.11959 [cs.LG] https://arxiv.org/abs/2106.11959

[10] Mathieu Guillame-Bert, Sebastian Bruch, Richard Stotz, and Jan Pfeifer. 2022. Yggdrasil Decision Forests: A Fast and Extensible Decision Forests Library. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022). https://api.semanticscholar.org/CorpusID:254274955

[11] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. 2020. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. arXiv:2012.06678 [cs.LG] https://arxiv.org/abs/2012.06678

[12] Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, et al. 2017. Automatic understanding of image and video advertisements. In *CVPR*. 1705–1715.

[13] Da-Cheng Juan, Chun-Ta Lu, Zhen Li, Futang Peng, Aleksei Timofeev, YiTing Chen, Yaxi Gao, Tom Duerig, Andrew Tomkins, and Sujith Ravi. 2018. Graph-RISE: Graph-Regularized Image Semantic Embedding. In *Proceedings of Woodstock '18: ACM Symposium on Neural Gaze Detection* (Woodstock, NY). ACM, New York, NY, USA, Article 4, 9 pages. doi:10.1145/1122445.1122456

[14] Andrew Kae, Kin Kan, Vijay K. Narayanan, and Dragomir Yankov. 2011. Categorization of display ads using image and landing page features. In *Proceedings of the Third Workshop on Large Scale Data Mining: Theory and Applications (LDMTA '11)*. ACM, 1–8. doi:10.1145/2002945.2002946

[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[16] Jannik Kossen, Neil Band, Clare Lyle, Aidan N. Gomez, Tom Rainforth, and Yarin Gal. 2022. Self-Attention Between Datapoints: Going Beyond Individual Input-Output Pairs in Deep Learning. arXiv:2106.02584 [cs.LG] https://arxiv.org/abs/2106.02584

[17] Yuxin Liang, Rui Cao, Jie Zheng, Jie Ren, and Ling Gao. 2021. Learning to Remove: Towards Isotropic Pre-trained BERT Embedding. *arXiv preprint arXiv:2104.05274* (2021). https://arxiv.org/abs/2104.05274 Accepted by ICANN 2021.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.

[19] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems* 33 (2020), 7498–7512.

[20] Sid Mittal, Vineet Gupta, Frederick Liu, and Mukund Sundararajan. 2023. Using Foundation Models to Detect Policy Violations with Minimal Supervision. arXiv:2306.06234 [cs.CL] https://arxiv.org/abs/2306.06234

[21] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*.

[22] Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *Advances in neural information processing systems* 32 (2019).

[23] Bo Qu, Zhurong Wang, Minghao Gu, Daisuke Yagi, Yang Zhao, Yinan Shan, and Frank Zahradnik. 2024. Multi-task CNN Behavioral Embedding Model For Transaction Fraud Detection. arXiv:2411.19457 [cs.LG] https://arxiv.org/abs/2411.19457

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*. 8748–8763.

[25] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, et al. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*. 8821–8831. https://proceedings.mlr.press/v139/ramesh21a.html

[26] Vijaya Teja Rayavarapu, Bharath Bhat, Myra Nam, Vikas Bahirwani, and Shobha Diwakar. 2022. Multimodal Transformers for Detecting Bad Quality Ads on YouTube. In *Proceedings of The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2022 (AdKDD '22)*. ACM, New York, NY, USA, 6. doi:XXXXXXX.XXXXXXX

[27] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. FLAVA: A Foundational Language And Vision Alignment Model . *Proceedings of CVPR* (2022), 15617–15629. https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01519

[28] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. 2021. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. *arXiv preprint arXiv:2106.01342* (2021). https://arxiv.org/abs/2106.01342 Accessed: 2025-01-02.

[29] Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. 2025. OmniPred: Language Models as Universal Regressors. arXiv:2402.14547 [cs.LG] https://arxiv.org/abs/2402.14547

[30] Xingyou Song, Qiuyi Zhang, Chansoo Lee, Emily Fertig, Tzu-Kuo Huang, Lior Belenki, Greg Kochanski, Setareh Ariafar, Srinivas Vasudevan, Sagi Perel, et al. 2024. The vizier gaussian process bandit algorithm. *arXiv preprint arXiv:2408.11527* (2024).

[31] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. 2022. OmniVL:One Foundation Model for Image-Language and Video-Language Tasks. arXiv:2209.07526 [cs.CV] https://arxiv.org/abs/2209.07526

[32] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*. 1785–1797.

[33] Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. 2024. Revisiting Graph-Based Fraud Detection in Sight of Heterophily and Spectrum. arXiv:2312.06441 [cs.LG] https://arxiv.org/abs/2312.06441

[34] Tong Ye, Zhitao Li, Jianzong Wang, Ning Cheng, and Jing Xiao. 2023. Efficient Uncertainty Estimation with Gaussian Process for Reliable Dialog Response Retrieval. arXiv:2303.08599 [cs.CL] https://arxiv.org/abs/2303.08599

[35] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. 2020. VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 11033–11043. https://proceedings.neurips.cc/paper_files/paper/2020/file/7d97667a3e056acab9aaf653807b4a03-Paper.pdf

[36] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *CoRR* abs/1905.04899 (2019). arXiv:1905.04899 http://arxiv.org/abs/1905.04899

[37] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2017. mixup: Beyond Empirical Risk Minimization. *CoRR* abs/1710.09412 (2017). arXiv:1710.09412 http://arxiv.org/abs/1710.09412

A Responsible AI Considerations

The development and deployment of ALF were guided by responsible AI principles to mitigate risks and ensure fair, transparent operation.

Mitigating Misuse: The model is used exclusively within Google's internal systems for advertiser risk assessment and is not publicly released. This controlled environment minimizes the potential for external misuse.

Privacy: All advertiser data is processed after being stripped of Personally Identifying Information (PII), ensuring that the model does not rely on sensitive user data.

Human Oversight and Accountability: The model's predictions are not used in isolation. They serve as a signal within a larger system that includes robust human review and appeals processes. False positives are carefully monitored, and advertisers have channels to appeal decisions, ensuring accountability and fairness.

Combating Adversarial Behavior: A key challenge in this domain is the presence of adversaries who actively try to conceal fraudulent or malicious assets. ALF's architecture is designed to enhance the effectiveness of our existing asset aggregation and selection techniques. By processing a holistic representation of an advertiser, the model can better identify subtle inconsistencies and suspicious patterns that might be missed when analyzing assets in isolation. While specific details of our anti-abuse strategies are omitted to prevent them from being circumvented, the model's design plays a crucial role in our resilience to adversarial attacks.

B Additional Details for Experiments

We present links to datasets and hyperparameters of our model in the tables below.

Hyperparameter	**Value**
Batch size	32768
Dimension of the created embedding	32
Number of attention heads in the transformer	8
Number of layers in the transformer	6
Dimensionality of the intermediate layer in the transformer	512
Activation	gelu
Optimizer	AdamW
Learning rate schedule	cosine decay
Initial learning rate (pretrain and finetune)	5e-05
Learning rate cosine decay alpha (pretrain and finetune)	0.1
Learning rate decay steps factor (pretrain and finetune)	0.95
Learning rate warm up target (pretrain)	10
Learning rate warm up target (finetune)	2

Table 6: Best hyperparameter values for production data

Dataset Name	URL	
albert	http://automl.chalearn.org/data	
dota2games	https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results	
adult	http://automl.chalearn.org/data	
blastchar	https://www.kaggle.com/blastchar/telco-customer-churn	
1995 income	https://www.kaggle.com/lodetomasi1995/income-classification	

Table 7: Benchmark Dataset Links.