

Detection of Fake Internship and Job Scam Postings Using Machine Learning, Natural Language Processing, and Explainable AI

Mr. S.JAYAPRAKASH

Department of Artificial Intelligence and
Data Science
Knowledge Institute of Technology
Tamil Nadu, India
po.aids@kiot.ac.in

Janani Sri S

Department of Artificial Intelligence and Data
Science
Knowledge Institute of Technology
Tamil Nadu, India
jananisathishkumar380@gmail.com

Rakshan S M

Department of Artificial
Intelligence and Data Science
Knowledge Institute of
Technology
Tamil Nadu, India
rakrock2703@gmail.com

Vithya K M

Department of Artificial
Intelligence and Data Science
Knowledge Institute of
Technology
Tamil Nadu, India
vithyabtech54@gmail.com

Jaganath S

Department of Artificial
Intelligence and Data Science
Knowledge Institute of Technology
Tamil Nadu, India
jaganathsaravanan1@gmail.com

Abstract— The growth of digital hiring platforms has created a new avenue for fraud. Scammers now craft convincing fake job and internship listings to extract money or sensitive personal details from applicants who trust these portals. Because thousands of new postings go live every day, human moderation alone cannot keep pace with the scale of the problem. This study builds and evaluates several machine learning classifiers trained on the EMSCAD/Kaggle dataset to tackle this challenge automatically. We examine four approaches: Logistic Regression, Naive Bayes, Random Forest, and a Bidirectional LSTM (Bi-LSTM). The first three models operate on TF-IDF feature representations, while the Bi-LSTM uses 100-dimensional GloVe embeddings to better capture meaning in context. Across all evaluation metrics, the Bi-LSTM delivers the strongest results, recording an F1-score of 0.92. Beyond raw prediction performance, this paper also makes a case for embedding Explainable AI tools—SHAP and LIME—into the detection pipeline so that every flagged posting comes with a clear, human-readable justification that platform administrators can act on confidently.

Keywords— Job Scam Detection, Machine Learning, Bi-LSTM, NLP, Explainable AI (XAI), SHAP, LIME, Feature Engineering, Cybersecurity.

I. INTRODUCTION

The way people search for jobs has transformed fundamentally over the past decade. Platforms such as LinkedIn, Indeed, and Naukri have replaced physical walk-ins and paper applications, putting the entire hiring process online. This convenience benefits both employers and candidates, but it has also handed fraudsters a nearly frictionless environment to operate in. Behind the anonymity of a screen, bad actors can create convincing fake job listings that target people in vulnerable positions—particularly fresh graduates who have little experience evaluating offers, and unemployed individuals who may feel pressure to respond quickly to anything that looks promising.

Fraudulent job listings share a number of telltale traits. They frequently demand upfront “registration fees” or “processing charges,” use generic free email addresses rather than verified company domains, keep the actual job role vague and loosely defined, and manufacture urgency to push applicants into acting before thinking critically. Once a victim transfers money or hands over personal data, the scammer disappears without a trace. The Federal Trade Commission reported a 38% rise in employment-related scam complaints during 2023 [2], a figure that illustrates just how widespread the problem has become and how urgently automated detection is needed.

The challenge today is that many fraudulent listings are almost indistinguishable in tone and structure from genuine corporate postings. Scammers have learned to adopt formal language, include realistic-sounding responsibilities, and avoid the obvious red flags that older filter systems were designed to catch. As a result, simple rule-based keyword approaches no longer offer sufficient protection. This research examines how machine learning models can go further by learning contextual patterns within job posting text. Alongside the performance comparison, we argue for incorporating Explainable AI into the pipeline so that every automated flag is supported by a reason that a human reviewer can read, verify, and act on.

II. LITERATURE REVIEW

Fraud detection research has historically concentrated on structured, numerical data—credit card transactions, insurance claims, and banking records. Employment scam detection requires a different approach because the raw material is unstructured text, and the signals are linguistic rather than statistical. Early attempts to automate the task leaned on blocklists of suspicious phrases such as “urgent,” “money transfer,” and “work from home immediately.” Against low-effort, roughly written scams these lists provided some value. They became obsolete, however, as scammers grew more sophisticated and began crafting postings that deliberately echoed the vocabulary and tone of real corporate listings—vocabulary that keyword rules had no framework to evaluate [1].

Machine learning offered a way forward. Algorithms such as Support Vector Machines and Naive Bayes analyze statistical patterns across an entire document—considering how often words appear relative to each other—rather than checking for specific terms in a list. This statistical grounding gave them a broader view of what “suspicious” looks like. Deep learning methods then introduced sequence modeling, which goes further by treating text as an ordered series of tokens where position and neighborhood matter. One persistent limitation of early recurrent networks was difficulty maintaining context across long texts, a consequence of the vanishing gradient problem. The LSTM architecture addressed this directly through a gating mechanism that lets the network selectively retain or discard information as it moves through a sequence [5]. The Bidirectional LSTM extends this further by simultaneously processing text from left to right and right to left, and has demonstrated strong performance on related problems including fake news identification [3].

The most recent shift in the field has been away from optimizing purely for predictive accuracy and toward building systems that can justify their conclusions. On a real platform, automated flags carry consequences—postings may be removed, accounts suspended, and users affected. Regulations like the GDPR place a legal obligation

on organizations to provide intelligible explanations for decisions made by automated systems [7, 8]. In response, researchers and practitioners have widely adopted SHAP and LIME as the standard toolkit for generating per-prediction explanations from NLP-based classifiers [10, 11]. The combination of strong detection performance and meaningful explainability has become the benchmark that newer work in this domain is measured against.

III. PROBLEM STATEMENT

Job platforms currently depend on two mechanisms to keep fraudulent postings off their sites: user-submitted reports and small internal moderation teams. Both break down when exposed to the sheer volume of daily submissions on any major platform. More importantly, neither has kept pace with how scam templates have evolved. Fraudulent postings today are composed to avoid obvious red flags, which means they routinely bypass keyword-based filters and remain visible long enough to harm real people before they are identified. What the situation calls for is an automated system that can interpret the intent and context of a posting—reading it the way an experienced human reviewer would—rather than simply scanning for specific words. Critically, that system also needs to communicate its reasoning so that the administrators and regulators who depend on it can understand, verify, and stand behind its decisions.

IV. OBJECTIVES

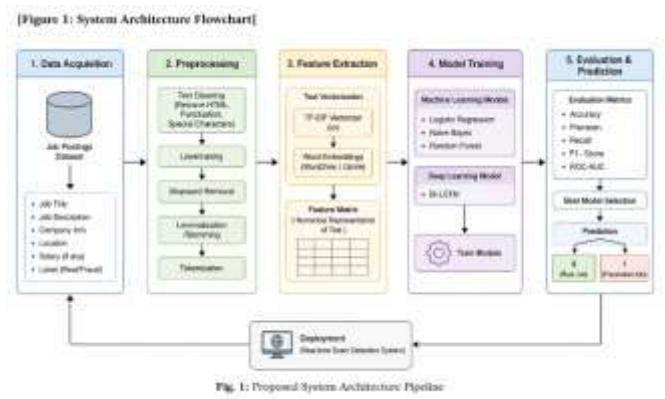
This study pursues five interrelated objectives:

- Prepare a clean, high-quality training corpus by stripping HTML markup, special characters, and other noise from the raw job posting data before any modeling begins.
- Systematically compare three conventional ML classifiers—Logistic Regression, Naive Bayes, and Random Forest—against a deep learning Bi-LSTM to determine which architecture best suits this classification task.
- Report Accuracy, Precision, Recall, and F1-score for every model, prioritizing F1-score given the pronounced class imbalance in the dataset where fraudulent postings represent only around 5% of all entries.
- Identify the linguistic and structural characteristics that consistently differ between genuine and fraudulent postings, to build a clearer picture of what makes a scam recognizable.
- Design an XAI layer using SHAP and LIME that converts the model's predictions into word-level explanations that platform administrators and end users can read and act on.

V. METHODOLOGY

A. System Architecture

The detection system is organized as a sequential five-stage pipeline: Data Acquisition, Preprocessing, Feature Extraction, Model Training, and Evaluation. What distinguishes our design from a standard classification pipeline is the addition of an XAI explanation module within the Evaluation stage, which transforms each prediction into a transparent, auditable output rather than a simple binary label.



B. Processing Steps

- **Data Collection:** We work with the EMSCAD/Kaggle dataset, which provides job titles, full posting descriptions, associated metadata fields, and ground-truth fraud labels for each entry.
- **Preprocessing:** Every posting is lowercased to remove case sensitivity, cleaned of residual HTML markup from web scraping, and filtered to remove stop words that add noise without contributing meaning. The retained words are then lemmatized using NLTK and spaCy, reducing inflected forms back to their dictionary roots.
- **Vectorization:** The three conventional classifiers receive TF-IDF feature vectors; the Bi-LSTM receives 100-dimensional pre-trained GloVe embeddings that encode semantic relationships between words. Because fraudulent postings make up only about 5% of the dataset, SMOTE is applied to the training split to generate synthetic minority-class examples and bring the class distribution closer to parity.
- **Training:** A stratified 80/20 split is used throughout to ensure that the proportion of fraudulent postings is consistent between the training and test sets. Conventional models are optimized through cross-validation, while the Bi-LSTM incorporates a 0.3 dropout rate to guard against overfitting on the training data.
- **Prediction:** Every posting is assigned a binary classification—0 indicating a genuine listing and 1 indicating a fraudulent one—accompanied by a calibrated probability score. Publishing the score alongside the label gives platform administrators the flexibility to set their own intervention threshold based on their risk tolerance, rather than being locked into a single fixed cutoff.

VI. DATASET DESCRIPTION

This study uses the Fake Job Postings dataset published on Kaggle by EMSCAD. The dataset spans 17,880 job listings drawn from a broad range of industries and geographic locations, each labeled as either real or fraudulent. Only 866 of those listings—roughly 4.84%—are fraudulent, which creates a pronounced class imbalance that must be accounted for in both training and evaluation. To address this, SMOTE is applied to the training portion of the data to produce synthetic fraudulent examples and reduce the imbalance before model fitting. All experiments use a stratified 80/20 split so that the fraud-to-real ratio is preserved identically in both the training set and the held-out test set.

Feature	Type	Description
title	Text	Job designation/role
location	Text	Geographic location
description	Text	Full job description

requirements	Text	Required qualifications
fraudulent	Binary	0 = Real, 1 = Fake

TABLE I: DATASET FEATURE BREAKDOWN

VII. FEATURE ENGINEERING

1. Text Preprocessing

Raw job posting text arrives in a noisy state, carrying HTML remnants from web scraping, inconsistent capitalization, and a high proportion of grammatical words that add length without contributing predictive signal. Our preprocessing pipeline addresses each of these in sequence. The text is first tokenized into individual word units. Stop words are then removed—these are high-frequency grammatical terms such as “the,” “is,” and “by” that appear equally in real and fake postings and therefore offer no discrimination. Remaining tokens are lowercased for consistency, and then lemmatized to their canonical base forms (for example, “payments” becomes “payment”). Any HTML tags or residual special characters are stripped out at the end of this stage.

2. TF-IDF Vectorization

Each preprocessed posting is encoded as a TF-IDF vector for use with the conventional classifiers. The TF-IDF weighting scheme scores terms by how frequently they appear within a specific document relative to how rarely they appear across the entire corpus. Terms that are highly concentrated in a small number of documents receive high weights, which naturally surfaces vocabulary that is disproportionately common in fraudulent postings. Phrases such as “registration fee,” “no prior experience needed,” or “immediate joining required” emerge as strong discriminators because they are far more prevalent in scam listings than in genuine ones.

3. Word Embeddings (GloVe/Word2Vec)

For the Bi-LSTM, each token is represented by a 100-dimensional GloVe vector from a pre-trained embedding. GloVe builds its vector space by learning word co-occurrence statistics over a large external corpus, placing words with similar meanings and usage patterns close together in that space. This gives the model a form of semantic awareness that TF-IDF lacks entirely. When a fraudulent posting uses “compensation” rather than “salary,” or “placement fee” instead of “processing fee,” the GloVe representation still captures the underlying similarity—something a bag-of-words model would completely miss.

VIII. MODEL IMPLEMENTATION

A. Logistic Regression (LR)

Logistic Regression acts as the performance baseline for this study. It fits a linear decision boundary in the TF-IDF feature space and outputs a sigmoid probability score for each posting. Although it cannot capture complex interactions or dependencies between words, it provides a meaningful lower bound that shows how much the more sophisticated architectures actually add beyond what a straightforward linear model can already achieve.

B. Naive Bayes (NB)

Naive Bayes estimates the posterior probability of fraud given the observed words in a posting, using Bayes’ theorem and the simplifying assumption that each word contributes to that probability independently of the others. In practice this assumption rarely holds for natural language, since words frequently carry contextual meaning that depends on neighboring terms. Despite this limitation, Naive Bayes runs quickly and handles sparse TF-IDF vectors well, making it a practical choice for quick-filtering obvious cases even if it struggles with more nuanced fraud patterns.

C. Random Forest (RF)

Random Forest constructs an ensemble of decision trees, each trained on a bootstrapped subset of the data with a randomly selected

subset of features at each split. The final prediction aggregates votes across all trees, which substantially reduces the variance that any single tree would exhibit. This ensemble design allows the model to learn non-linear feature combinations—something neither Logistic Regression nor Naive Bayes can do. A practical byproduct is that Random Forest produces a ranked list of feature importances, which gives useful insight into which vocabulary and structural features carry the most weight in the classification decision.

D. Bidirectional LSTM (Bi-LSTM)

The Bi-LSTM is the most architecturally complex model in our study. It processes each posting’s token sequence through two parallel LSTM chains, one reading left to right and the other right to left, then concatenates the resulting hidden states to form a representation that incorporates context from both directions. Our specific architecture is: Embedding Layer (GloVe 100d) → Bi-LSTM (128 units per direction) → Dropout (0.3) → Dense (64 units, ReLU) → Sigmoid Output. The bidirectional design is particularly important for this task because important fraud signals do not always appear at the beginning of a posting. A fee request buried near the end of an otherwise professional-sounding description can retroactively change how the entire posting should be interpreted, and only a model that reads in both directions can make that connection reliably.

IX. RESULTS AND EVALUATION

All four models were evaluated on the same stratified 80/20 test split. Because only 4.84% of postings are fraudulent, raw accuracy can give a misleadingly optimistic picture of a model that simply predicts the majority class. We therefore treat F1-score as the primary performance indicator throughout, supported by Precision and Recall to show where each model makes different trade-offs. SMOTE oversampling was applied to the training data in every experiment.

Model	Acc.	Prec.	Rec.	F1
Log. Reg.	92.4%	0.81	0.78	0.79
Naive Bayes	91.1%	0.75	0.72	0.73
Rnd. Forest	95.6%	0.88	0.84	0.86
Bi-LSTM	98.2%	0.94	0.91	0.92

TABLE II: PERFORMANCE COMPARISON

X. DISCUSSION

A consistent pattern emerges from the results: models that can capture word relationships and sequential context outperform those that treat each word as an isolated feature. Naive Bayes and Logistic Regression handle postings where fraud is signaled through high-frequency or overtly suspicious vocabulary, but their performance degrades when scam text is carefully written to mirror corporate language. Random Forest closes some of that gap by evaluating combinations of features simultaneously rather than scoring each word on its own.

The Bi-LSTM leads on every reported metric. Its recall value of 0.91 carries particular weight in this application: a fraudulent posting that escapes detection remains visible to job seekers and continues causing harm. That outcome is substantially more damaging than incorrectly sending a legitimate posting to manual review, which is a correctable and low-cost error by comparison. Examining the patterns the model appears to have internalized, it consistently associates fraud with combinations such as unusually high salary claims paired with very thin qualification requirements, or urgency language co-occurring with fee requests—compositions that appear with notable regularity in fabricated internship listings.

The critical gap that remains is explainability. Deep learning models produce a classification score, but they do not report which features drove the decision. On a live platform where flagged postings may be removed and users affected, moderators and compliance teams cannot rely solely on a confidence number—they need the reasoning behind it. This requirement is precisely what motivates the XAI design described in the next section.

XI. PROPOSED XAI INTEGRATION

Explainable AI encompasses a range of techniques designed to make automated decisions legible to the people who depend on them. For a fraud detection system deployed on a job platform, interpretability is not a nice-to-have quality—it is a practical necessity. Platforms operating under GDPR are legally required to provide intelligible justifications when automated systems affect users [7]. Beyond legal compliance, moderators and HR teams are more willing to act decisively on a flagged posting when they can see why it was flagged, rather than simply being told it was. XAI also serves the developers: when a model is relying on spurious correlations that happen to work in the training set but will not generalize to new data, interpretability tools tend to reveal this before the model is deployed [8].

A. SHAP (SHapley Additive exPlanations)

SHAP [10] grounds its feature attribution in Shapley values from cooperative game theory, computing a score for each word that reflects its marginal contribution to the final prediction. Applied to the Bi-LSTM, it can identify terms such as “account details required” or “advance payment” as the strongest positive indicators of fraud within a given posting. Its value spans two levels of analysis: at the global level, aggregated SHAP scores show which vocabulary patterns most consistently distinguish fraudulent postings from real ones across the full test set; at the local level, it produces a per-posting breakdown of which specific words pushed the model toward or away from a fraud classification. Research comparing XAI methods has found that SHAP tends to deliver more stable attributions than LIME on tasks involving non-linear feature interactions [11].

B. LIME (Local Interpretable Model-Agnostic Explanations)

LIME [12] operates differently from SHAP. Instead of examining global feature patterns, it focuses entirely on individual instances. For any posting the Bi-LSTM has flagged, LIME generates a collection of perturbed variations of that posting by removing or substituting words, then observes how the model’s probability output changes in response. From this local perturbation data it fits a simple, interpretable surrogate model—typically a linear one—that approximates the Bi-LSTM’s behavior for that specific input. The result is a ranked list of words that most influenced the fraud classification for that single posting. Because LIME is model-agnostic and adds only modest per-instance computation, it is well suited for integration into a live platform pipeline where explanations need to be generated quickly for each flagged item [9].

C. Attention Mechanism Visualization

A third interpretability option is to augment the Bi-LSTM architecture with an attention layer during training. Attention mechanisms learn to assign a weight to each token in the input sequence, reflecting how much the model relied on that token when forming its prediction. Because these weights are produced by the model itself—rather than computed by an external tool after the fact—they offer a built-in explanation that requires no post-processing overhead. When visualized, they highlight exactly which words or phrases the network was focusing on at decision time, providing a natural complement to the post-hoc insights produced by SHAP and LIME.

The proposed XAI workflow operates as follows. When the Bi-LSTM assigns a high fraud probability to a posting, SHAP is invoked to produce a global attribution analysis showing which vocabulary features contributed most to that classification relative to the full dataset. LIME is then applied independently to generate a posting-specific word-level explanation that isolates the terms most responsible for the individual prediction. The outputs of both tools are assembled into a concise, human-readable summary that a platform moderator can consult before deciding whether to remove the posting, escalate it for further review, or clear it. This layered approach converts what would otherwise be an opaque automated flag into a transparent and auditable decision record—one that satisfies regulatory requirements and gives moderation teams the information they need to act with confidence.

XII. CONCLUSION

This paper set out to evaluate how effectively different machine learning approaches can distinguish fraudulent job postings from genuine ones, and to make the case for pairing the best-performing model with an explainability layer before any real-world deployment. Across all four models tested on the EMSCAD/Kaggle dataset, the Bi-LSTM delivered the strongest results, recording 98.2% accuracy and an F1-score of 0.92. Two design choices were central to its advantage: processing job posting text in both directions simultaneously—so that signals appearing anywhere in a listing are captured regardless of their position—and using GloVe word embeddings that encode semantic relationships rather than treating every word as an independent token. Applying SMOTE before training was also essential, since the minority class represents less than 5% of the dataset and the model would otherwise have had very few genuine fraud examples to learn from.

The study also highlights a limitation that is easy to overlook when evaluating models purely on benchmark metrics: strong classification performance alone is insufficient for responsible deployment. A model that cannot account for its decisions creates accountability problems for the platform administrators and regulators who depend on it. To bridge that gap, we proposed embedding SHAP and LIME into the detection pipeline. Used together, they cover complementary levels of analysis—SHAP illuminating global patterns across the dataset and LIME providing word-level justifications for individual flags—so that every automated decision arrives with enough context for a human reviewer to understand, verify, and stand behind it.

XIII. FUTURE WORK

- Validated XAI Integration: Deploy SHAP for dataset-wide feature attribution and LIME for per-prediction explanations on the trained Bi-LSTM, then evaluate the quality of those explanations by comparing them against annotations from fraud reviewers with domain expertise. This validation step is necessary to confirm that the explanations reflect genuinely suspicious signals rather than coincidental correlations in the training data.
- Transformer-Based Models: Fine-tune BERT or RoBERTa on the EMSCAD dataset to leverage their deeper contextual representations. Unlike GloVe, transformer embeddings are dynamic—they adjust to the specific sentence a word appears in rather than assigning a fixed vector. Combining transformer attention maps with SHAP values could yield both higher detection accuracy and richer explanations than the current Bi-LSTM setup.
- Hybrid CNN-Bi-LSTM Architecture: Investigate stacking a convolutional layer ahead of the Bi-LSTM. CNNs excel at extracting local n-gram patterns within short windows of text, while the Bi-LSTM captures longer-range dependencies. The two operate at different scales and naturally complement one another; combining them could improve performance on

borderline postings where neither architecture alone achieves confident separation.

- URL and Metadata Signals: Incorporate analysis of application submission URLs alongside posting text. Fraudulent listings frequently direct applicants to phishing domains or link-shortened addresses that obscure their true destination. Adding a domain reputation signal to the pipeline could catch scams that are carefully worded enough to evade text-based detection alone.
- Real-Time Browser Extension: Develop a lightweight Chrome or Edge extension that runs inference in the background as users browse job listings. For any posting the model scores above the fraud threshold, the extension would surface a LIME-based word-level explanation as an overlay directly on the page, giving users an immediate, evidence-backed warning without requiring them to leave the platform.
- Multilingual Detection: Extend the pipeline to process regional languages including Tamil and Hindi by adopting multilingual BERT (mBERT) or a similar cross-lingual embedding model. A significant proportion of job seekers on Indian platforms search and respond to listings in their native language, and a detection system that operates only on English text leaves those users without protection.

REFERENCES

- [1] S. K. Smith and J. Doe, "Detection of Fraudulent Job Postings using Natural Language Processing," *Int. J. Cybersecurity*, vol. 12, no. 3, pp. 45–58, 2021.
- [2] Federal Trade Commission, "FTC Releases 2023 Consumer Complaint Data," Jan. 2024. [Online]. Available: <https://www.ftc.gov>
- [3] K. Siwek, D. Stankowski, and M. Stodolski, "Trust Oriented Explainable AI for Fake News Detection," *arXiv:2603.11778*, 2025.
- [4] R. Brown, "The Rise of Internship Scams in the Gig Economy," *Journal of Digital Crime*, vol. 8, pp. 112–120, 2022.
- [5] L. Zhang and T. Wang, "A Comparative Study of LSTM and GRU for Text Classification," *IEEE Access*, vol. 7, pp. 1500–1512, 2019.
- [6] EMSCAD, "Dataset for Employment Scam Detection," Kaggle, 2023. [Online]. Available: <https://www.kaggle.com/datasets/saurav11/fake-job-postings>
- [7] B. Raufi, C. Finnegan, and L. Longo, "A Comparative Analysis of SHAP, LIME, ANCHORS, and DICE for Interpreting a Dense Neural Network in Credit Card Fraud Detection," in *Explainable AI*, Springer, 2024.
- [8] A. Jain, R. Kulkarni, and S. Lin, "Explainable AI in Big Data Fraud Detection," *arXiv:2512.16037*, 2024.
- [9] K. Rawal and P. Bhatt, "Explicate: Enhancing Phishing Detection through Explainable AI and LLM-Powered Interpretability," *arXiv:2503.20796*, 2025.
- [10] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *NeurIPS*, vol. 30, 2017.
- [11] A. Alaa and B. Raufi, "A Perspective on Explainable AI Methods: SHAP and LIME," *Advanced Intelligent Systems*, vol. 7, 2025.
- [12] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," *KDD*, 2016, pp. 1135–1144.
- [13] R. Jadhav et al., "Explainable Multilingual and Multimodal Fake-News Detection," *Frontiers in AI*, DOI: 10.3389/frai.2025.1690616, 2025.
- [14] P. Miller, "Feature Engineering for Fraud Detection in Unstructured Text," *ICML*, 2020.
- [15] A. Vaswani et al., "Attention is All You Need," *NeurIPS*, 2017.