Supplementary Material: Making Paper Reviewing Robust to Bid Manipulation Attacks

A. Dataset Construction

In this section, we describe how we subsampled data from the Semantic Scholar Open Research Corpus (S2ORC) (Ammar et al., 2018), extracted reviewer/paper features such as subject area and TPMS, and simulated bids using citation. Our data is publicly released[7] for reproducibility and to facilitate future research.

A.1. Conference Simulation

The goal of our dataset is to simulate a NeurIPS-like conference environment, where the organizers assign reviewers to papers based on expertise and interest. We first retrieve the collection of 6956 papers from S2ORC that are published in ML/AI/CV/NLP venues between the years 2014-2015, which includes the following conferences: AAAI, AISTATS, ACL, COLT, CVPR, ECCV, EMNLP, ICCV, ICLR, ICML, IJCAI, NeurIPS, and UAI. We believe the diversity of subject areas represented by the above conferences is an accurate reflection of typical ML/AL conferences in recent years. We will refer to this collection of papers as the *corpus*.

Subject areas. Most conferences require authors to indicate primary and secondary subject areas for their submitted papers. However, the S2ORC only contains a *field of study* attribute for most of the retrieved papers in the corpus, which is often the broad category of *computer science*. To identify the suitable fine-grained subjects for each paper, we adopt an unsupervised learning approach of clustering the papers by relatedness and treating each discovered cluster as a subject area.

Similarity is defined in terms of co-citations – a common signal used in information retrieval for discovering related documents (Dean & Henzinger, 1999). For a paper p, let $N(p)$ denote the union of in-citations and out-citations for p. The similarity between two papers p, q is defined as

$$\sigma(p,q) = \frac{|N(p) \cap N(q)|}{\sqrt{|N(p)|} \cdot \sqrt{|N(q)|}}, \tag{S1}$$

which is the cosine similarity in document retrieval. We perform agglomerative clustering using average linkage[8] to reduce the set of papers to 1000 clusters. After removing small cluster (less than 5 papers), we obtain 368 clusters to serve as subject areas. Table S1 shows a few sample clusters along with papers contained in the cluster. Most of the discovered clusters are highly coherent with members sharing keywords in their titles despite the definition of similarity depending *entirely* on co-citations.

To populate the list of subject areas for a given paper p, we first compute its subject relatedness to a cluster C by:

$$\sigma(p,C) = \frac{1}{|C|} \sum_{q \in C} \sigma(p,q). \tag{S2}$$

Given the set of clusters representing subject areas, we identify the top-5 clusters according to $\sigma(p,C)$ to be the list of subject areas for the paper p, denoted subj(p).

Reviewers. The S2ORC dataset contains entries of authors along with their list of published papers. We utilize this information to simulate reviewers by collecting the set of authors who has cited at least one paper from the corpus. The total number of retrieved authors is 234,598. Because the vast majority of retrieved authors are very loosely related to the field of ML/AI, they would not be suitable reviewer candidates for a real ML/AI conference. Therefore, we retain only authors who have cited at least 15 papers from the corpus to serve as reviewers. We also remove authors who cited more than 50 papers

[7] https://drive.google.com/drive/folders/1khI9kaPy_8F0GtAzwR-48Jc3rsQmBhfe?usp=sharing
[8] https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

Subject Area	Papers
Multi-task learning	Encoding Tree Sparsity in Multi-Task Learning: A Probabilistic Framework Multi-Task Learning and Algorithmic Stability Exploiting Task-Feature Co-Clusters in Multi-Task Learning Efficient Output Kernel Learning for Multiple Tasks Learning Multiple Tasks with Multilinear Relationship Networks *Etc.*
Video segmentation	Efficient Video Segmentation Using Parametric Graph Partitioning Video Segmentation with Just a Few Strokes Co-localization in Real-World Images Semantic Single Video Segmentation with Robust Graph Representation PatchCut: Data-driven object segmentation via local shape transfer *Etc.*
Topic modeling	On Conceptual Labeling of a Bag of Words Topic Modeling with Document Relative Similarities Divide-and-Conquer Learning by Anchoring a Conical Hull Spectral Methods for Supervised Topic Models Model Selection for Topic Models via Spectral Decomposition *Etc.*
Feature selection	Embedded Unsupervised Feature Selection Feature Selection at the Discrete Limit Bayes Optimal Feature Selection for Supervised Learning with General Performance Measures Reconsidering Mutual Information Based Feature Selection: A Statistical Significance View Unsupervised Simultaneous Orthogonal basis Clustering Feature Selection *Etc.*

Table S1. Sample subject areas and paper titles of cluster members.

from the corpus, since these reviewers represent senior researchers that would typically serve as area chairs. The number of remaining reviewers is $5,914$.

Most conferences also solicit self-reported subject areas from reviewers. We simulate this attribute by leveraging the clusters discovered through co-citation. For each subject area C, we count the number of times C appeared in $\text{subj}(p)$ for each of the papers p that the reviewer r has cited. The 5 most frequently appearing clusters (ties are broken randomly) serve as the reviewer's subject areas, denoted $\text{subj}(r)$.

TPMS score. The TPMS score (Charlin & Zemel, 2013) is computed by measuring the similarity between a reviewer's profile – represented by a set of papers that the reviewer uploads – and a target paper. We simulate this score using the language model-based approach from the original TPMS paper, which we detail below for completeness. For a reviewer r, let A_r denote the bag-of-words representation for the set of papers that the reviewer has authored. More specifically, we collect the abstracts of the papers that r has authored, remove all stop words, and pool the remaining words together into A_r as a multi-set. Similarly, let A_p denote the bag-of-words representation for the abstract of a paper p. The simulated TPMS is computed as:

$$\text{TPMS}_{r,p} = \sum_{w \in A_p} \log f_{rw}, \tag{S3}$$

where f_{rw} is the Dirichlet-smoothed normalized frequency of the word w in A_r. Let D denote the bag-of-words representation for the entire corpus of (abstracts of) papers, and let $D(w)$ (resp. $A_r(w)$) denote the occurrences of w in the corpus (resp. A_r). Then

$$f_{rw} := \left(\frac{|A_r|}{|A_r| + \beta}\right) \frac{|A_r(w)|}{|A_r|} + \left(\frac{\beta}{|A_r| + \beta}\right) \frac{|D(w)|}{|D|},$$

where β is a smoothing factor. We set $\beta = 1000$ in our experiment. The obtained scores are normalized per paper between 0 and 1.

A.2. Simulating Bids

The most challenging aspect of our simulation is the bids. At first, it may seem natural to simulate bids using citations, since it is a proxy of interest and can be easily obtained from the S2ORC dataset. However, we have observed that bids are heavily skewed towards a few very influential papers, while the distribution of bids is much more uniform across all papers. To overcome this issue, we instead model a reviewer's bidding behavior based on the following assumptions:

1. A reviewer will only bid on papers from subject areas that he/she is familiar with.
2. Given two papers from the same subject area, a reviewer favors bidding on a paper whose title/abstract is a better match with the reviewer's profile.

We define several scores that reflect the above aspects and combine them to obtain the final bids. In practice, reviewers will often also rely on TPMS to sort the papers to bid on. However, since our simulated TPMS depends entirely on the abstract, we omit TPMS in our bidding model. Nevertheless, we have observed empirically that TPMS is highly correlated with the bids that we obtain.

Subject score. We leverage citation to reflect the degree of interest in the subject of a paper. Let $\text{icf}(q)$ denote the *inverse citation frequency* (ICF) of a paper q in the corpus:

$$\text{icf}(q) = \log \frac{\text{\# total in-citations in the corpus}}{\text{\# in-citations for } q}.$$

The purpose of the ICF is to down-weight commonly cited papers to avoid overcrowding of bids. Denote by $C^*(q)$ the top cluster that q belongs to according to Eq. (S2). The *subject score* for a paper p is defined as:

$$\text{subject-score}_{r,p} = \sum_{q: r \text{ cites } q} \frac{\text{icf}(q)}{|C^*(q)|} \mathbb{1}\{p \in C^*(q)\}. \tag{S4}$$

In other words, for each paper q that r cites, we merge all papers from the same subject area of q, represented by $C^*(q)$, into the reviewer's pool. Each paper in $C^*(q)$ is weighted by the reciprocal of the cluster size and the ICF of q, and the subject score is the resulting sum after accumulating over all papers q that the reviewer cites. Note that every paper within the same subject cluster has the *exact same* subject score, which is non-zero only if the reviewer has bid on a paper within this subject area. This property reflects the assumption that a reviewer is only interested in papers from familiar subject areas, and is indifferent to different papers in the same subject absent of title/abstract information. To avoid overcrowding by frequently cited papers, we set $\text{subject-score}_{r,p} = 0$ for any paper p that received over 1000 citations.

Title/abstract score. To measure the degree of title/abstract similarity between a reviewer and a paper, we compute the inner product between the TF-IDF vectors of the reviewer's and paper's title/abstract. Let $\text{idf}(w)$ denote the *inverse document frequency* of a word w. For each reviewer r, let $\text{tf-idf}(r)$ denote the vector, indexed by words, such that $\text{tf-idf}(r)_w = (|A_r(w)|/|A_r|) \cdot \text{idf}(w)$ for each word w. Similarly, we can define the TF-IDF vector for a paper p, and the *abstract score* between a pair (r, p) is given by the inner product:

$$\text{abstract-score}_{r,p} = \text{tf-idf}(r) \cdot \text{tf-idf}(p). \tag{S5}$$

We can define the *title score* in an analogous manner based on the bag-of-words representation of titles instead of abstracts.

Bidding. We simulate bids by combining the subject/title/abstract scores as follows. First, we define a *total score*:

$$\text{total-score}_{r,p} = (\text{title-score}_{r,p} + \text{abstract-score}_{r,p}) \cdot \text{subject-score}_{r,p}, \tag{S6}$$

which reflects the assumptions we made about a reviewer's bidding behavior, *i.e.*, a higher total score reflects a higher reviewer interest in the paper. The total score gives us a ranking of papers in the corpus, denoted by $\text{rank}_r(p)$, for each paper p. To obtain the positive bids, we randomly retain high-ranked papers with a decaying probability:

$$\Pr(r \text{ bids on } p) = 1/(1 + \exp(\alpha \cdot (\text{rank}_r(p) - \mu)),$$

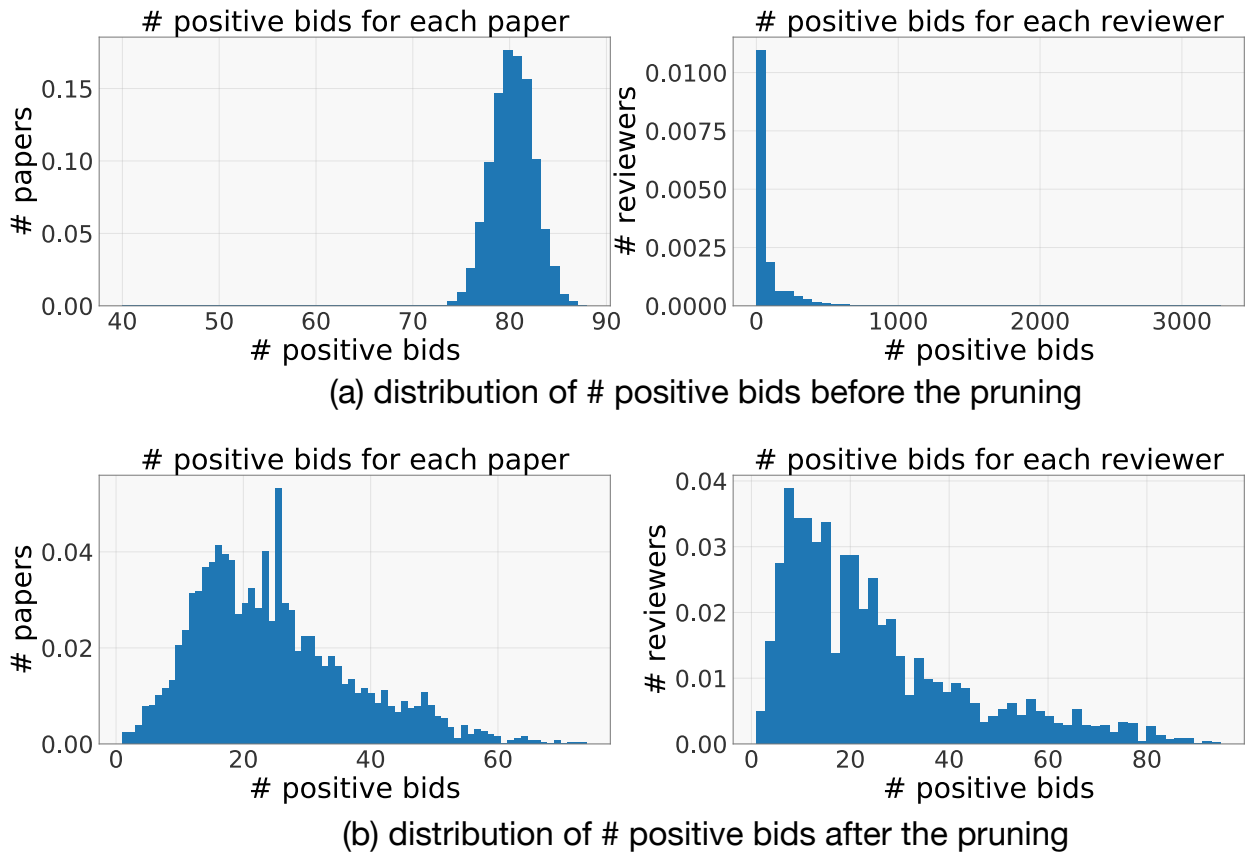

Figure S1. Distribution of the number of positive bids before and after subsampling.

where α and μ are hyperparameters that control the steepness of the drop in sampling probability for low-ranked papers, and the average number of papers that each reviewer bids on. We set $\alpha = 0.2$ and $\mu = 80$ in our experiment.

The quality of bids obtained from this sampling procedure is very reasonable. However, the majority of papers had very few bids (see Fig. S1(a)) – contrary to statistics observed in a real conference such as NeurIPS-2016 (see Figure 1 in (Shah et al., 2018)). To match the distribution of the number of bids per reviewer/paper to that of a real conference, we further subsample papers (resp. reviewers) to encourage selecting ones with more bids. The distribution of the number positive bids per reviewer/paper after subsampling is shown in Fig. S1(b). Our finalized conference dataset contains $m = 2483$ reviewers and $n = 2446$ submitted papers – a realistic balance of papers and reviewers for recent ML/AI conferences.

Finally, some conferences allow more fine-grained bids, such as *in a pinch*, *willing* and *eager* for conferences managed using CMT. To simulate *bid scores* that reflect the degree of interest, we quantize the total score of all positive bids into the discrete range $\{1, 2, 3\}$ based on the distribution of bid scores in a real conference: at a ratio of $8 : 53 : 39$ for the bids $1, 2$ and 3.

B. Features and Training

We provide details regarding feature extraction and model training in this section. To fully imitate a conference management environment, we extract relevant features from papers and reviewers that are obtainable in a realistic scenario, including: paper/reviewer subject area (5 areas for each), bag-of-words vector for paper title, and (simulated) TPMS. These features are further processed and concatenated as input to the linear regression model in Section 3.

Table S2 lists all the extracted features and their dimensions. Paper title (PT) is the vectorized count of words appearing in

Features	paper titles (PT)	paper subject area (PS)	reviewer subject area (RS)
# of Dimensions	930	368	368
Features	intersected subject area (IS)	TPMS vector (TV)	RS⊗PS
# of Dimensions	368	12	135424
Features	RS⊗PT	IS⊗PT	IS⊗TV
# of Dimensions	342240	342240	4410

Table S2. Extracted features and their dimensionalities. See the text for details.

		k=1	**k=2**	**k=3**	**k=4**	**k=5**	**k=6**	**k=7**	**k=8**	**k=9**	**k=10**
AP@k per reviewer	**train**	0.41	0.41	0.40	0.39	0.38	0.38	0.37	0.37	0.36	0.35
	test	0.38	0.41	0.39	0.38	0.38	0.37	0.36	0.36	0.35	0.34
AP@k per paper	**train**	0.55	0.53	0.51	0.50	0.49	0.47	0.46	0.45	0.43	0.42
	test	0.58	0.55	0.52	0.51	0.48	0.47	0.45	0.44	0.43	0.41

Table S3. Average precision@k per reviewer/paper for the trained linear regressor.

the paper's title, while paper subject area (PS), reviewer subject area (RS) and intersected subject area (IS) are categorical features represented using binary vectors. The first dimension for the TPMS vector (TV) is the TPMS score for the reviewer-paper pair. We also quantize the raw TPMS into 11 bins and use the bin index as well as the quantized scores, which results in the remaining 11 dimensions for the TPMS vector.

RS⊗PS, RS⊗PT, IS⊗PT and IS⊗TV are additional quadratic features that capture the interaction between feature pairs. The introduction of these quadratic features results in a very high-dimensional, albeit extremely sparse feature vector, and hence many dimensions could be collapsed without a significant impact to performance. We apply feature hashing (Weinberger et al., 2009) to the quadratic features at a hash ratio of 0.01, which reduces the total feature dimensionality to $d = 10,288$.

Model performance. To validate our linear regression model and the selected features, we test the average precision at k (AP@k) for the trained model on a train-test split. Table S3 shows the AP@k per reviewer (P@k for finding papers relevant to a reviewer, averaged across all reviewers) and the AP@k per paper for the linear regressor. It is evident that both metrics are at an acceptable level for real world deployment, and the train-test gap is minimal, indicating that the model is able to generalize well beyond observed bids.

We also perform a qualitative evaluation of the end-to-end assignment process using the relevance scoring model. We select six representative (honest) reviewers from our dataset – Kavita Bala[9], Ryan P. Adam[10], Peter Stone[11], Yejin Choi[12], Emma Brunskill[13] and Elad Hazan[14] – representing distinct areas of interest in ML/AI. Table S4 shows the assigned papers for the selected reviewers, which appear to perfectly match the area of expertise for the respective reviewers. Many of the assigned papers have a bid score of 0 despite being very relevant for the reviewer, which shows that the scoring model is able to discover missing bids and improve the overall assignment quality.

C. Additional Experiment on Detection

In Section 5 we evaluated our defense against attacks that succeeded in securing the target paper assignment. However, in doing so, it is possible that malicious reviewers that did not succeed initially will inadvertent become high-ranked *after* other reviewers are removed from the candidate set. Therefore, it may be necessary to detect *all* attack instances in the candidate

[9] https://scholar.google.com/citations?user=Rh16nsIAAAAJ
[10] https://scholar.google.com/citations?user=grQ_GBgAAAAJ
[11] https://scholar.google.com/citations?user=qnwjcfAAAAAJ
[12] https://scholar.google.com/citations?user=vhP-tlcAAAAJ
[13] https://scholar.google.com/citations?user=HaN8b2YAAAAJ
[14] https://scholar.google.com/citations?user=LnhCGNMAAAAJ

Reviewer	Assigned Papers	Bid Scores
Kavita Bala	1. Learning Lightness from Human Judgement on Relative Reflectance	3
	2. Simulating Makeup through Physics-Based Manipulation of Intrinsic Image Layers	3
	3. Learning Ordinal Relationships for Mid-Level Vision	3
	4. Automatically Discovering Local Visual Material Attributes	0
	5. Recognize Complex Events from Static Images by Fusing Deep Channels	0
	6. Learning a Discriminative Model for the Perception of Realism in Composite Images	0
Ryan P. Adams	1. Stochastic Variational Inference for Hidden Markov Models	3
	2. Parallel Markov Chain Monte Carlo for Pitman-Yor Mixture Models	0
	3. Celeste: Variational Inference for a Generative Model of Astronomical Images	0
	4. Measuring Sample Quality with Stein'S Method	0
	5. Parallelizing MCMC with Random Partition Trees	0
	6. Hamiltonian ABC	0
Peter Stone	1. Qualitative Planning with Quantitative Constraints for Online Learning of Robotic Behaviours	3
	2. An Automated Measure of MDP Similarity for Transfer in Reinforcement Learning	3
	3. On Convergence and Optimality of Best-Response Learning with Policy Types in Multiagent Systems	3
	4. A Framework for Task Planning in Heterogeneous Multi Robot Systems Based on Robot Capabilities	3
	5. A Strategy-Aware Technique for Learning Behaviors from Discrete Human Feedback	0
	6. Stick-Breaking Policy Learning in Dec-Pomdps	0
Yejin Choi	1. Don'T Just Listen, Use Your Imagination: Leveraging Visual Common Sense for Non-Visual Tasks	3
	2. Segment-Phrase Table for Semantic Segmentation, Visual Entailment and Paraphrasing	0
	3. Refer-To-As Relations as Semantic Knowledge	0
Emma Brunskill	1. Policy Evaluation Using the Ω-Return	3
	2. Towards More Practical Reinforcement Learning	3
	3. High Confidence Policy Improvement	3
	4. Sample Efficient Reinforcement Learning With Gaussian Processes	3
	5. Policy Tree: Adaptive Representation for Policy Gradient	3
	6. Abstraction Selection in Model-Based Reinforcement Learning	0
Elad Hazan	1. Online Linear Optimization via Smoothing	3
	2. Online Learning for Adversaries with Memory: Price of Past Mistakes	0
	3. Hierarchies of Relaxations for Online Prediction Problems with Evolving Constraints	0
	4. Hard-Margin Active Linear Regression	0
	5. Online Gradient Boosting	0
	6. Robust Multi-Objective Learning With Mentor Feedback	0

Table S4. Assigned papers for six representative reviewers.

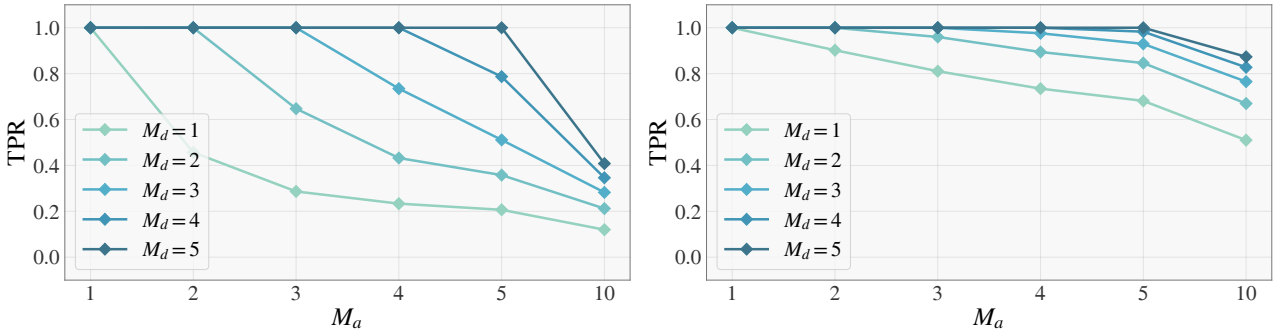

Figure S2. TPR for detecting *colluding white-box attacks* (left) and *colluding black-box attacks* (right) that succeed in achieving top-50 rank.

set rather than ones that were successfully assigned.

Fig. S2 shows the detection TPR for all attackers that were initially ranked below $K = 50$ but managed to move into the candidate set after the attack. Since this attacker pool includes many that obtained a relatively low rank, detection TPR is much higher than that of Fig. 3 and Fig. 5. For instance, for $M_d = 5$, even when the colluding party is significantly larger

at $M_a = 10$, detection remains viable with a TPR of more than 40% against *colluding white-box attack*. This experiment shows that our detection mechanism is unlikely to inadvertently increase the success rate of failed attacks.