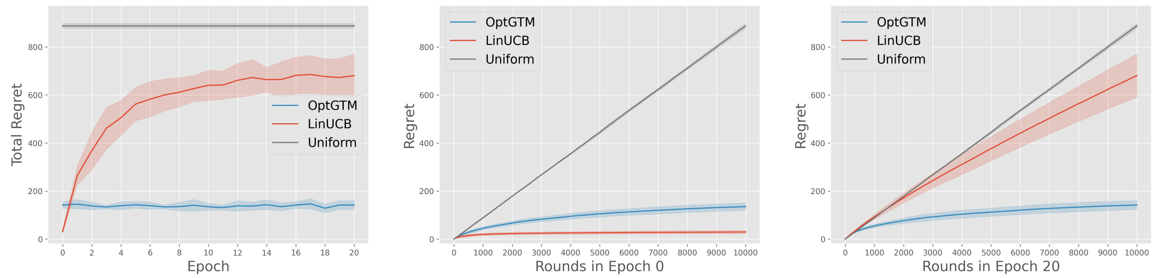

(a) Total strategic regret R_T as the arms adapt their strategies to the deployed algorithm over the course of 20 epochs.

(b) Epoch 0 (Truthful Arms): Regret as a function of t *before* the arms have interacted with the deployed algorithm.

(c) Epoch 20 (Strategic Arms): Regret as a function of t *after* the arms have interacted with the deployed algorithm.

Figure 1: Comparison of the strategic regret of OptGTM and LinUCB. The strategic arms adapt their strategies gradually over the course of 20 epochs. OptGTM performs similarly across all epochs, whereas LinUCB performs increasingly worse as the arms adapt to the algorithm (Figure 1a). Figure 1b and 1c provide a closer look at the regret of the algorithms across the T rounds in the initial epoch, where the arms are truthful, and the final epoch after the arms have adapted to the algorithms.

6 Experiments: Simulating Strategic Context Manipulation

We here experimentally analyze the efficacy of OptGTM when the arms strategically manipulate their contexts in response to our learning algorithm. We compare the performance of OptGTM with the traditional LinUCB algorithm [1, 7], which—as shown in Proposition 3.3—implicitly incentivizes the arms to manipulate their contexts and, as a result, is expected to suffer large regret when the arms are strategic.

Contrary to the assumption of arms playing in NE, we here model the strategic arm behavior by letting the arms update their strategy (i.e., what contexts to report) based on past interactions with the algorithms. More precisely, we assume that the strategic arms interact with the deployed algorithm (i.e., OptGTM or LinUCB) over the course of 20 epochs, with each epoch consisting of $T = 10k$ rounds. At the end of each epoch, every arm then updates its strategy using gradient ascent w.r.t. its utility. Importantly, this approach requires no prior knowledge from the arms, as they learn entirely through sequential interaction. This does not necessarily lead to equilibrium strategies, but instead serves as a way to study the performance and the implied incentivizes of OptGTM and LinUCB under a natural model of strategic gaming behavior.

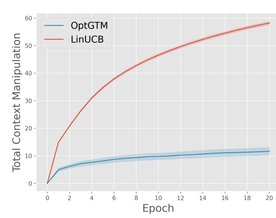

Figure 2: Context manipulation $\sum_{t,i} \|x^*_{t,i} - x_{t,i}\|_2$.

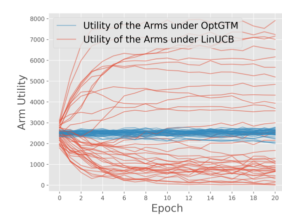

Figure 3: Utility of the arms for each of the 10 runs.

Experimental Setup. We associate each arm with a true feature vector $y^*_i \in \mathbb{R}^{d_1}$ (e.g., product features) and randomly sample a sequence of user vectors $c_t \in \mathbb{R}^{d_2}$ (i.e., customer features). We assume that every arm can alter its feature vector y^*_i by reporting some other vector y_i, but cannot alter the user contexts c_t. We use a feature mapping $\varphi(c_t, y_i) = x_{t,i}$ to map the reported features $y_i \in \mathbb{R}^{d_1}$ and the user features $c_t \in \mathbb{R}^{d_2}$ to an arm-specific context $x_{t,i} \in \mathbb{R}^d$ that the algorithm observes. At the end of every epoch, each arm then performs an approximated gradient step on y_i w.r.t. its utility, i.e., the number of times it is selected. We let $K = 5$ and $d = d_1 = d_2 = 5$ and average the results over 10 runs. More details and results can be found in Appendix B.

Results. In Figure 1a, we observe that OptGTM performs similarly well across all epochs, which suggests that OptGTM successfully discourages the emergence of harmful gaming behavior. In contrast, as the arms adapt their strategies (i.e., what features to report), LinUCB suffers increasingly more regret and almost performs as badly as uniform sampling in the final epoch. In epoch 0, when the all arms are truthful, i.e., are non-strategic, LinUCB performs better than OptGTM (Figure 1b). This is expected as OptGTM suffers additional regret due to maintaining independent estimates of θ^* for each arm (as a mechanism to incentivize truthfulness). However, OptGTM significantly outperforms LinUCB as the arms strategically adapt, which is most prominent in the final epoch (Figure 1c). Interestingly, as already suggested in Section 5, OptGTM cannot prevent manipulation in the feature space (see Figure 2). However, OptGTM does manage to bound the effect of the manipulation on the regret (Figure 1a) and, most importantly, the effect on the utility of the arms as well(Figure 3). As a result, the arms are discouraged from gaming their contexts heavily and the context manipulation has only a minor effect on the actions taken by OptGTM.