EE-NET: EXPLOITATION-EXPLORATION NEURAL NETWORKS IN CONTEXTUAL BANDITS

Yikun Ban, Yuchen Yan, Arindam Banerjee, Jingrui He
University of Illinois at Urbana-Champaign
{yikunb2, yucheny5, arindamb, jingrui}@illinois.edu

ABSTRACT

In this paper, we propose a novel neural exploration strategy in contextual bandits, EE-Net, distinct from the standard UCB-based and TS-based approaches. Contextual multi-armed bandits have been studied for decades with various applications. To solve the exploitation-exploration tradeoff in bandits, there are three main techniques: epsilon-greedy, Thompson Sampling (TS), and Upper Confidence Bound (UCB). In recent literature, linear contextual bandits have adopted ridge regression to estimate the reward function and combine it with TS or UCB strategies for exploration. However, this line of works explicitly assumes the reward is based on a linear function of arm vectors, which may not be true in real-world datasets. To overcome this challenge, a series of neural bandit algorithms have been proposed, where a neural network is used to learn the underlying reward function and TS or UCB are adapted for exploration. Instead of calculating a large-deviation based statistical bound for exploration like previous methods, we propose "EE-Net", a novel neural-based exploration strategy. In addition to using a neural network (Exploitation network) to learn the reward function, EE-Net uses another neural network (Exploration network) to adaptively learn potential gains compared to the currently estimated reward for exploration. Then, a decision-maker is constructed to combine the outputs from the Exploitation and Exploration networks. We prove that EE-Net can achieve $\mathcal{O}(\sqrt{T \log T})$ regret and show that EE-Net outperforms existing linear and neural contextual bandit baselines on real-world datasets.

1 INTRODUCTION

The stochastic contextual multi-armed bandit (MAB) (Lattimore and Szepesvári, 2020) has been studied for decades in machine learning community to solve sequential decision making, with applications in online advertising (Li et al., 2010), personal recommendation (Wu et al., 2016; Ban and He, 2021b), etc. In the standard contextual bandit setting, a set of n arms are presented to a learner in each round, where each arm is represented by a context vector. Then by certain strategy, the learner selects and plays one arm, receiving a reward. The goal of this problem is to maximize the cumulative rewards of T rounds.

MAB algorithms have principled approaches to address the trade-off between Exploitation and Exploration (EE), as the collected data from past rounds should be exploited to get good rewards but also under-explored arms need to be explored with the hope of getting even better rewards. The most widely-used approaches for EE trade-off can be classified into three main techniques: Epsilon-greedy (Langford and Zhang, 2008), Thompson Sampling (TS) (Thompson, 1933), and Upper Confidence Bound (UCB) (Auer, 2002; Ban and He, 2020).

Linear bandits (Li et al., 2010; Dani et al., 2008; Abbasi-Yadkori et al., 2011), where the reward is assumed to be a linear function with respect to arm vectors, have been well studied and succeeded both empirically and theoretically. Given an arm, ridge regression is usually adapted to estimate its reward based on collected data from past rounds. UCB-based algorithms (Li et al., 2010; Chu et al., 2011; Wu et al., 2016; Ban and He, 2021b) calculate an upper bound for the confidence ellipsoid of estimated reward and determine the arm according to the sum of estimated reward and UCB. TS-based algorithms (Agrawal and Goyal, 2013; Abeille and Lazaric, 2017) formulate each arm as a posterior distribution where mean is the estimated reward and choose the one with the maximal

Figure 1: Left figure: Structure of EE-Net. In the right figure, Case 1: "Upward" exploration should be made when the learner underestimates the reward; Case 2: "Downward" exploration should be chosen when the learner overestimates the reward. EE-Net has the ability to adaptively make exploration according to different cases. In contrast, UCB-based strategy will always make upward exploration, and TS-based strategy will randomly choose upward or downward exploration.

sampled reward. However, the linear assumption regarding the reward may not be true in real-world applications (Valko et al., 2013).

To learn non-linear reward functions, recent works have utilized deep neural networks to learn the underlying reward functions, thanks to its powerful representation ability. Considering the past selected arms and received rewards as training samples, a neural network f_1 is built for exploitation. Zhou et al. (2020) computes a gradient-based upper confidence bound with respect to f_1 and uses UCB strategy to select arms. Zhang et al. (2021) formulates each arm as a normal distribution where the mean is f_1 and deviation is calculated based on gradient of f_1, and then uses the TS strategy to choose arms. Both Zhou et al. (2020) and Zhang et al. (2021) achieve the near-optimal regret bound of $O(\sqrt{T}\log T)$.

In this paper, we propose a novel neural exploration strategy, named "EE-Net". Similar to other neural bandits, EE-Net has another exploitation network f_1 to estimate rewards for each arm. The crucial difference from existing works is that EE-Net has an exploration network f_2 to predict the potential gain for each arm compared to current reward estimate. The input to the exploration network is the gradient of f_1 and the ground-truth is residual difference between the true received reward and the estimated reward from f_1. The strategy is inspired by recent advances in the neural UCB strategies (Zhou et al., 2020; Ban et al., 2021). Finally, a decision-maker f_3 is constructed to select arms. f_3 has two modes: linear or nonlinear. In linear mode, f_3 is a linear combination of f_1 and f_2, inspired by the UCB strategy. In the nonlinear mode, f_3 is formulated as a neural network with input $(f_1; f_2)$ and the goal is to learn the probability of being an optimal arm for each arm. Figure 1 depicts the workflow of EE-Net and its advantages for exploration compared to UCB or TS-based methods (see more details in Appendix D). To sum up, the contributions of this paper can be summarized as follows:

1. We propose a novel neural exploration strategy, EE-Net, where another neural network is assigned to learn the potential gain compared to the current reward estimate.
2. Under standard assumptions of over-parameterized neural networks, we prove that EE-Net can achieve the regret upper bound $O(\sqrt{T\log T})$, which improves a multiplicative factor of $\sqrt{\log T}$ and is independent of either input or effective dimension, compared to existing state-of-the-art neural bandit algorithms.
3. We conduct extensive experiments on four real-world datasets, showing that EE-Net outperforms baselines including linear and neural versions of ϵ-greedy, TS, and UCB.

Next, we discuss the problem definition in Sec.3, elaborate on the proposed EE-Net in Sec.4, and present our theoretical analysis in Sec.5. In the end, we provide the empirical evaluation (Sec.6) and conclusion.

2 RELATED WORK

Constrained Contextual bandits The common constrain placed on the reward function is the linear assumption, usually calculated by ridge regression (Li et al., 2010; Abbasi-Yadkori et al., 2011; Valko et al., 2013; Dani et al., 2008). The linear UCB-based bandit algorithms (Abbasi-Yadkori et al., 2011; Li et al., 2016) and the linear Thompson Sampling (Agrawal and Goyal, 2013; Abeille and Lazaric, 2017) can achieve successful performance and the near-optimal regret bound $\tilde{O}(d\sqrt{T})$. To break the linear assumption, Filippi et al. (2010) generalizes the reward function to a composition of linear and non-linear functions and then adopt a UCB-based algorithm to deal with it; Bubeck et al. (2011) imposes the Lipschitz property on reward metric space and constructs a hierarchical optimistic optimization to make selections; Valko et al. (2013) embeds the reward function into Reproducing Kernel Hilbert Space and proposes the kernelized TS/UCB bandit algorithms.

Neural Bandits. To learn non-linear reward functions, deep neural networks have been adapted to bandits with various variants. Riquelme et al. (2018); Lu and Van Roy (2017) build L-layer DNN to learn the arm embeddings and apply Thompson Sampling on the last layer for exploration. Zhou et al. (2020) first introduces a provable neural-based contextual bandit algorithm with a UCB exploration strategy and then Zhang et al. (2021) extends the neural network to Thompson sampling framework. Their regret analysis is built on recent advances on the convergence theory in over-parameterized neural networks(Du et al., 2019; Allen-Zhu et al., 2019) and utilizes Neural Tangent Kernel (Jacot et al., 2018; Arora et al., 2019) to construct connections with linear contextual bandits (Abbasi-Yadkori et al., 2011). Ban and He (2021a) further adopts convolutional neural networks with UCB exploration aiming for visual-aware applications. Xu et al. (2020) performs UCB-based exploration on the last layer of neural networks to reduce the computational cost brought by gradient-based UCB. Different from the above existing works, EE-Net keeps the powerful representation ability of neural networks to learn reward function and first assigns another neural network to determine exploration.

3 PROBLEM DEFINITION

We consider the standard contextual multi-armed bandit with the known number of rounds T (Zhou et al., 2020; Zhang et al., 2021). In each round $t \in [T]$, where the sequence $[T] = [1, 2, \cdots, T]$, the learner is presented with n arms, $X_t = \{x_{t,1}, \cdots, x_{t,n}\}$, in which each arm is represented by a feature vector $x_{t,i} \in \mathbb{R}^d$ for each $i \in [n]$. After playing one arm $x_{t,i}$, its reward $r_{t,i}$ is assumed to be generated by the function:

$$r_{t,i} = h(x_{t,i}) + \eta_{t,i}, \tag{3.1}$$

where the unknown reward function $h(x_{t,i})$ can be either linear or non-linear and the noise $\eta_{t,i}$ is drawn from certain distribution with expectation $\mathbb{E}[\eta_{t,i}] = 0$. Following many existing works (Zhou et al., 2020; Ban et al., 2021; Zhang et al., 2021), we consider bounded rewards $r_{t,i} \in [a, b]$. For the brevity, we denote the selected arm in round t by x_t and the reward received from it by r_t. The pseudo regret of T rounds is defined as:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} (r_t^* - r_t)\right], \tag{3.2}$$

where $\mathbb{E}[r_t^* \mid X_t] = \max_{i \in [n]} h(x_{t,i})$ is the maximal expected reward in the round t. The goal of this problem is to minimize R_T by certain selection strategy.

Notation. We denote by $\{x_i\}_{i=1}^t$ the sequence $(x_1, \cdots, x_t)$. We use $\|v\|_2$ to denote the Euclidean norm for a vector v, and $\|W\|_2$ and $\|W\|_F$ to denote the spectral and Frobenius norm for a matrix W. We use $\langle \cdot, \cdot \rangle$ to denote the standard inner product between two vectors or two matrices. We may use $\nabla_{\frac{1}{t}} f_1(x_{t,i})$ or $\nabla_{\frac{1}{t}} f_1$ to represent the gradient $\nabla_{\frac{1}{t}} f_1(x_{t,i}; \frac{1}{t})$ for brevity. We use $\{x, r\}_{t=1}$ to represent the collected data up to round t.

4 PROPOSED METHOD: EE-NET

EE-Net is composed of three components. The first component is exploitation network $f_1(\cdot; \frac{1}{t})$, which focuses on learning the unknown reward function h based on the data collected in past rounds.

Table 1: Structure of EE-Net (Round t)

Input	Network	Label
$\{x_{t,i}\}_{i=1}^{K_t}$	$f_1(\cdot;\theta_1)$ (Exploitation)	$\{r_{t,i}\}_{i=1}^{K_t}$
$\{\nabla_{\theta_1} f_1(x_{t,i};\theta_{t-1}^1)\}_{i=1}^{K_t}$	$f_2(\cdot;\theta_2)$ (Exploration)	$\{r_{t,i} - f_1(x_{t,i};\theta_{t-1}^1)\}_{i=1}^{K_t}$
$\{(f_1(x_{t,i};\theta_{t-1}^1), f_2(\nabla_{\theta_1} f_1;\theta_{t-1}^2))\}_{i=1}^{K_t}$	$f_3(\cdot;\theta_3)$ (Decision-maker with non-linear function)	$\{p_{t,i}\}_{i=1}^{K_t}$

The second component is the exploration network $f_2(\cdot;\theta_2)$, which focuses on characterizing the level of exploration needed for each arm in the present round. The third component is the decision-maker f_3, which focuses on suitably combining the outputs of the exploitation and exploration networks leading to the arm selection.

1) Exploitation Net. The exploitation net f_1 is a neural network which learns the mapping from arms to rewards. In round t, denote the network by $f_1(\cdot;\theta_{t-1}^1)$, where the superscript of θ_{t-1}^1 is the index of network and the subscript represents the round where the parameters finished the last update. Given an arm $x_{t,i}; i \in [n]$, $f_1(x_{t,i};\theta_{t-1}^1)$ is considered the "exploitation score" for $x_{t,i}$. By some criterion, after playing arm x_t, we receive a reward r_t. Therefore, we can conduct gradient descent to update θ_{t-1}^1 based on the collected training samples $\{x_i; r_i\}_{i=1}^t$ and denote the updated parameters by θ_t^1.

2) Exploration Net. Our exploration strategy is inspired by existing UCB-based neural bandits (Zhou et al., 2020; Ban et al., 2021). Based on the Lemma 5.2 in (Ban et al., 2021), given an arm $x_{t,i}$, with probability at least $1-\delta$, we have the following UCB form:

$$|h(x_{t,i}) - f_1(x_{t,i};\theta_{t-1}^1)| \le \gamma(\nabla_{\theta_1} f_1(x_{t,i};\theta_{t-1}^1)); \tag{4.1}$$

where h is defined in Eq. (3.1) and γ is an upper confidence bound represented by a function with respect to the gradient $\nabla_{\theta_{t-1}^1} f_1$ (see more details and discussions in Appendix D). Then we have the following definition.

Definition 4.1. In round t, given an arm $x_{t,i}$, we define $h(x_{t,i}) - f_1(x_{t,i};\theta_{t-1}^1)$ as the "expected potential gain" for $x_{t,i}$ and $r_{t,i} - f_1(x_{t,i};\theta_{t-1}^1)$ as the "potential gain" for $x_{t,i}$.

Let $y_{t,i} = r_{t,i} - f_1(x_{t,i};\theta_{t-1}^1)$. When $y_{t,i} > 0$, the arm $x_{t,i}$ has positive potential gain compared to the estimated reward $f_1(x_{t,i};\theta_{t-1}^1)$. A large positive $y_{t,i}$ makes the arm more suitable for exploration, whereas a small (or negative) $y_{t,i}$ makes the arm unsuitable for exploration. Recall that traditional approaches such as UCB intend to estimate such potential gain using standard tools, e.g., Markov inequality, Hoeffding bounds, etc., from large deviation bounds.

Instead of calculating a large-deviation based statistical bound for $y_{t,i}$, we use a neural network $f_2(\cdot;\theta_2)$ to represent it, where the input is $\nabla_{\theta_{t-1}^1} f_1(x_{t,i})$ and the ground truth is $r_{t,i} - f_1(x_{t,i};\theta_{t-1}^1)$. Adopting gradient $\nabla_{\theta_{t-1}^1} f_1(x_{t,i})$ as the input also is due to the fact that it incorporates two aspects of information: the feature of the arm and the discriminative information of f_1.

Moreover, in the upper bound of NeuralUCB or the variance of NeuralTS, there is a recursive term $A_{t-1} = I + \sum_{i=1}^{t-1} g_{\theta_1} f_1(x_i) g_{\theta_1} f_1(x_i)^\top$ which is a function of past gradients up to $t-1$ and incorporates relevant historical information. On the contrary, in EE-Net, the recursive term which depends on past gradients is θ_1 in the exploration network f_2 because we have conducted gradient descent for θ_{t-1}^2 based on $g_{\theta_1} f_1(x_i) g_{i=1}^{t-1}$. Therefore, this form θ_{t-1}^2 is similar to A_{t-1} in neuralUCB/TS, but EE-net does not (need to) make a specific assumption about the functional form of past gradients, and is also more memory-efficient.

To sum up, in round t, we consider $f_2(\nabla_{\theta_{t-1}^1} f_1(x_{t,i});\theta_{t-1}^2)$ as the "exploration score" of $x_{t,i}$, because it indicates the potential gain of $x_{t,i}$ compared to our current exploitation score $f_1(x_{t,i};\theta_{t-1}^1)$. Therefore, after receiving the reward r_t, we can use gradient descent to update θ_{t-1}^2 based on collected training samples $\{\nabla_{\theta_{t-1}^1} f_1(x_i); r_i - f_1(x_i;\theta_{t-1}^1)\}_{i=1}^t$. We also provide other two heuristic forms for f_2's ground-truth label: $|r_{t,i} - f_1(x_{t,i};\theta_{t-1}^1)|$ and $\mathrm{ReLU}(r_{t,i} - f_1(x_{t,i};\theta_{t-1}^1))$. We compare them in an ablation study in Appendix B.

Algorithm 1 EE-Net

Input: $f_1; f_2; f_3$, T (number of rounds), η_1 (learning rate for f_1), η_2 (learning rate for f_2), η_3 (learning rate for f_3), K_1 (number of iterations for f_1), K_2 (number of iterations for f_2), K_3 (number of iterations for f_3), $\Pi_{\mathcal{D}}$ (normalization operator)

Initialize $\theta^1_0; \theta^2_0; \theta^3_0, b^1_0 = \theta^1_0, b^2_0 = \theta^2_0, b^3_0 = \theta^3_0$
for $t = 1; 2; \ldots; T$ do
  Observe n arms $\{x_{t;1}; \ldots; x_{t;n}\}$
  for each $i \in [n]$ do
    Compute $f_1(x_{t;i}; \theta^1_{t-1}); f_2(\nabla_{\theta^1_{t-1}} f_1(x_{t;i})); \theta^2_{t-1}); f_3((f_1; f_2); \theta^3_{t-1})$
  end for
  $x_t = \arg\max_{x_{t;i}; i \in [n]} f_3 \quad f_1(x_{t;i}; \theta^1_{t-1}); f_2(\nabla_{\theta^1_{t-1}} f_1(x_{t;i})); \theta^2_{t-1}); \theta^3_{t-1}$
  Play x_t and observe reward r_t
  $\theta^1_t, \theta^2_t, \theta^3_t = \textsc{GradientDescent}(\theta_0, \{x\}^t_{\tau=1}; \{r\}^t_{\tau=1})$
end for

procedure $\textsc{GradientDescent}(\theta_0, \{x\}^t_{\tau=1}; \{r\}^t_{\tau=1})$
  $L_1 = \frac{1}{2} \sum_{\tau=1}^{t} \quad f_1(x_\tau; \theta^1) \quad r_\tau)^2$
  $\theta^{1;(0)} = \theta^1_0$
  for $k \in \{1; \ldots; K_1\}$ do
    $\theta^{1;(k)} = \theta^{1;(k-1)} \quad \eta_1 \nabla_{\theta^{1;(k-1)}} L_1$
  end for
  $\theta^1_t = \theta^{1;(K_1)}$
  $L_2 = \frac{1}{2} \sum_{\tau=1}^{t} \quad f_2(\nabla_{\theta^1_{t-1}} f_1(x_\tau)); \theta^2) \quad (r_\tau \quad f_1(x_\tau; \theta^1_{t-1}))^2$
  $\theta^{2;(0)} = \theta^2_0$
  for $k \in \{1; \ldots; K_2\}$ do
    $\theta^{2;(k)} = \theta^{2;(k-1)} \quad \eta_2 \nabla_{\theta^{2;(k-1)}} L_2$
  end for
  $b^2_t = \theta^{2;(K_2)}$
  Determine label p_t
  $L_3 = \frac{1}{t} \sum_{i=1}^{t} \quad p_t \log f_3((f_1; f_2); \theta^3) + (1 \quad p_t) \log(1 \quad f_3((f_1; f_2); \theta^3))$
  $\theta^{3;(0)} = \theta^3_0$
  for $k \in \{1; \ldots; K_3\}$ do
    $\theta^{3;(k)} = \theta^{3;(k-1)} \quad \eta_3 \nabla_{\theta^{3;(k-1)}} L_3$
  end for
  $b^3_t = \theta^{3;(K_3)}$
  Randomly choose (θ^1_t, θ^2_t) uniformly from $\{(b^1_0; b^2_0); (b^1_1; b^2_1); \ldots; (b^1_t; b^2_t)\}$
  Randomly choose θ^3_t uniformly from $\{b^3_0; b^3_1; \ldots; b^3_t\}$
  Return $\theta^1_t, \theta^2_t, \theta^3_t$
end procedure

3) Decision-maker. In round t, given an arm $x_{t;i}; i \in [n]$, with the computed exploitation score $f_1(x_{t;i}; \theta^1_{t-1})$ and exploration score $f_2(\nabla_{\theta^1_{t-1}} f_1; \theta^2_{t-1})$, we use a function $f_3 \quad f_1; f_2; \theta^3$ to trade off between exploitation and exploration and compute the final score for $x_{t;i}$. The selection criterion is defined as

$$x_t = \arg \max_{x_{t;i}; i \in [n]} f_3 \quad f_1(x_{t;i}; \theta^1_{t-1}); f_2 \quad \nabla_{\theta^1_{t-1}} f_1(x_{t;i}); \theta^2_{t-1} \; ; \theta^3_{t-1} :$$

Note that f_3 can be either linear or non-linear functions. We provide the following two forms.

(1) Linear function: f_3 can be formulated as a linear function with respect to f_1 and f_2:

$$f_3(f_1; f_2; \theta^3) = w_1 f_1(x_{t;i}; \theta^1) + w_2 f_2(\nabla_{\theta^1} f_1; \theta^2)$$

where w_1, w_2 are two weights preset by the learner. When $w_1 = w_2 = 1$, f_3 can be thought of as UCB-type policy, where the estimated reward and potential gain in f_1 and f_2 are simply added together. In experiments, we report its empirical performance in ablation study (Appendix B).

(2) Non-linear function f_3 also can be formulated as a neural network to learn the mapping from (f_1, f_2) to the optimal arm. We transform the bandit problem into a binary classification problem. Given an arm $x_{t,i}$, we define $p_{t,i}$ as the probability of being the optimal arm for $x_{t,i}$ in round t. For brevity, we denote by p_t the probability of being the optimal arm for the selected arm x_t in round t. According to different reward distributions, we have different approaches to determine p_t.

1. Binary reward $\forall t \in [T]$, suppose r_t is a binary variable over $a, b (a < b)$, it is straightforward to set: $p_t = 1.0$ if $r_t = b$; $p_t = 0.0$, otherwise.
2. Continuous reward $\forall t \in [T]$, suppose r_t is a continuous variable over the range $[a, b]$, we provide two ways to determine p_t. (1) p_t can be directly set as $\frac{r_t - a}{b - a}$. (2) The learner can set a threshold $\tau (a < \tau < b)$. Then $p_t = 1.0$ if $r_t > \tau$; $p_t = 0.0$, otherwise.

Therefore, with the collected training samples $\{f_1(x_{1}; \theta_1^{1}); f_2(\nabla_{\theta_1^{1}} f_1(x_{1}; \theta_1^{1}); p_1)\}_{t=1}$ in round t, we can conduct gradient descent to update parameters θ_3 of f_3.

Table 1 details the working structure of EE-Net. Algorithm 1 depicts the work flow of EE-Net, where the input of f_2 is normalized, i.e., $(\nabla_{\theta_1^{1}} f_1(x_{t,i}))$. Algorithm 1 provides a version of gradient descent (GD) to update EE-Net, where drawing (θ_t^1, θ_t^2) uniformly from their stored historical parameters is for the sake of analysis. One can easily extend EE-Net to stochastic GD to update the parameters incrementally.

Remark 4.1 (Network structure). The networks f_1, f_2, f_3 can be different structures according to different applications. For example, in the vision tasks, f_1 can be set up as convolutional layers (LeCun et al., 1995). For the exploration network f_2, the input $\nabla_{\theta_1^{1}} f_1$ may have exploding dimensions when the exploitation network f_1 becomes wide and deep, which may cause huge computation cost for f_2. To address this challenge, we can apply dimensionality reduction techniques to obtain low-dimensional vectors of $\nabla_{\theta_1^{1}} f_1$. In the experiments, we use Roweis and Saul (2000) to acquire a 10-dimensional vector for $\nabla_{\theta_1^{1}} f_1$ and achieve the best performance among all baselines. □

Remark 4.2 (Exploration direction). EE-Net has the ability to determine exploration direction. Given an arm $x_{t,i}$, when the estimation $f_1(x_{t,i})$ is lower than the expected reward $h(x_{t,i})$, the learner should make the "upward" exploration, i.e., increase the chance of $x_{t,i}$ being explored; When $f_1(x_{t,i})$ is higher than $h(x_{t,i})$, the learner should do the "downward" exploration, i.e., decrease the chance of $x_{t,i}$ being explored. EE-Net uses the neural network f_2 to learn $h(x_{t,i}) - f_1(x_{t,i})$ (which has positive and negative scores) and has the ability to determine the exploration direction. In contrast, NeuralUCB will always make "upward" exploration and NeuralTS will randomly choose between "upward" exploration and "downward" exploration (see selection criteria in Table 2 and more details in Appendix D). □

Remark 4.3 (Space complexity). NeuralUCB and NeuralTS have to maintain the gradient outer product matrix (e.g., $A_t = \sum_{=1}^{t} \nabla_{\theta_1^{1}} f_1(x_{}; \theta_1^{1}) \nabla_{\theta_1^{1}} f_1(x_{}; \theta_1^{1})^\top \in \mathbb{R}^{p \times p}$) and, for $\theta_1^{1} \in \mathbb{R}^{p}$, have a space complexity of $O(p^2)$ to store the outer product. On the contrary, EE-Net does not have this matrix and only regards $\nabla_{\theta_1^{1}} f_1$ as the input of f_2. Thus, EE-Net reduces the space complexity from $O(p^2)$ to $O(p)$. □

5 Regret Analysis

In this section, we provide the regret analysis of EE-Net when f_3 is set as the linear function $f_3 = f_1 + f_2$, which can be thought of as the UCB-type trade-off between exploitation and exploration. For the sake of simplicity, we conduct the regret analysis on some unknown but fixed data distribution $\mathcal{D}$. In each round t, n samples $\{(x_{t,1}; r_{t,1}), (x_{t,2}; r_{t,2}), \cdots, (x_{t,n}; r_{t,n})\}$ are drawn i.i.d. from $\mathcal{D}$. This is standard distribution assumption in over-parameterized neural networks (Cao and Gu, 2019). Then, for the analysis, we have the following assumption, which is a standard input assumption in neural bandits and over-parameterized neural networks(Zhou et al., 2020; Allen-Zhu et al., 2019).

Assumption 5.1(-Separability) For any $t \in [T], i \in [n], \|x_{t,i}\|_2 = 1$, and $r_{t,i} \in [0,1]$. Then, for every pair $x_{t,i}, x_{t',i'} \in [T], i' \in [k]$; and $(t,i) \neq (t',i')$, $\|x_{t,i} - x_{t',i'}\|_2 > $, and suppose there exists an operator such that $\|\ \|_2 = 1$ and $\|(\mathcal{O}_1 f_1(x_{t,i})) - (\mathcal{O}_1 f_1(x_{t',i'}))\|_2$

For example, the operator can be designed as $(\mathcal{O}_1 f_1(x_{t,i})) = (\frac{\mathcal{O}_1 f_1(x_{t,i})}{2\|\mathcal{O}_1 f_1(x_{t,i})\|_2}, \frac{x_{t,i}}{2})$. The analysis will focus on over-parameterized neural networks (Jacot et al., 2018; Du et al., 2019; Allen-Zhu et al., 2019). Given an input $x \in \mathbb{R}^d$, without loss of generality, we define the fully-connected network f with depth $L \geq 2$ and width m:

$$f(x;) = W_L (W_{L-1} (W_{L-2} \cdots (W_1 x))) \tag{5.1}$$

where is the ReLU activation function, $W_1 \in \mathbb{R}^{m \times d}$, $W_l \in \mathbb{R}^{m \times m}$, for $2 \leq l \leq L-1$, $W_L \in \mathbb{R}^{1 \times m}$, and $= [\mathrm{vec}(W_1)^\top, \mathrm{vec}(W_2)^\top, \ldots, \mathrm{vec}(W_L)^\top]^\top$.

Initialization. For any $l \in [L-1]$, each entry of W_l is drawn from the normal distribution $\mathcal{N}(0, \frac{2}{m})$ and W_L is drawn from the normal distribution $\mathcal{N}(0, \frac{1}{m})$. Note that EE-Net at most has three networks f_1, f_2, f_3. We define them following the definition of f for brevity, although they may have different depth or width. Then, we have the following theorem for EE-Net. Recall that $_1, _2$ are the learning rates for f_1, f_2; K_1 is the number of iterations of gradient descent for f_1 in each round; and K_2 is the number of iterations for f_2.

Theorem 1. Let f_1, f_2 follow the setting of f (Eq. (5.1)) with the same width m and depth L. Let L_1, L_2 be loss functions defined in Algorithm 1. Set $f_3 = f_1 + f_2$. For any $\in (0,1)$; $\in (0, O(\frac{1}{T})]$; $\in (0, O(\frac{1}{L})]$, suppose

$$\begin{aligned} m \quad & e \quad \mathrm{poly}(T, n, L, \ ^{-1}) \ \log(1/) \ e^{\sqrt{\log(Tn/)}}) \ ; \\ _1 = \ _2 &= \min\left(\frac{T^{-5}}{2 \ ^2 m}, \ \frac{}{\mathrm{poly}(T, n, L) \ m} \right) ; \\ K_1 = K_2 &= \frac{\mathrm{poly}(T, n, L)}{^2} \log \ ^{-1} : \end{aligned} \tag{5.2}$$

Then, with probability at least 1 over the initialization, the pseudo regret of EE-Net in T rounds satisfies

$$R_T \quad O(1) + (2\sqrt{T} \quad 1)3\sqrt{2}O(L) + O\left((2\sqrt{T} \quad 1) \ 2\log\frac{O(Tn)}{}\right) : \tag{5.3}$$

Comparison with existing works. Under the similar assumptions in over-parameterized neural networks, the regret bounds complexity of NeuralUCB (Zhou et al., 2020) and NeuralTS (Zhang et al., 2021) both are

$$R_T \quad O\left(\sqrt{\tilde{d}T\log T}\right) \quad O\left(\sqrt{\tilde{d}T\log T}\right) \ ; \text{ and } \tilde{d} = \frac{\log\det(I + H/)}{\log(1 + Tn/)}$$

where H is the neural tangent kernel matrix (NTK) (Jacot et al., 2018; Arora et al., 2019) and is the regularization parameter. Similarly, in linear contextual bandits, Abbasi-Yadkori et al. (2011) achieve $O(d\sqrt{T}\log T)$ and Li et al. (2017) achieve $O(\sqrt{dT}\log T)$.

Remark 5.1. Compared to NeuralUCB/TS, EE-Net roughly improves by a multiplicative factor of $\sqrt{\log T}$, because our proof of EE-Net is directly built on recent advances in convergence theory (Allen-Zhu et al., 2019) and generalization bound (Cao and Gu, 2019) of over-parameterized neural networks. Instead, the analysis for NeuralUCB/TS contains three parts of approximation error by calculating the distances between the expected reward and ridge regression, ridge regression and NTK, and NTK and network function. □

Remark 5.2. The regret bound of EE-Net does not have the effective dimension $\tilde{d}$ or input dimension d. $\tilde{d}$ or d may cause significant error, when the determinant of H is extremely large or $d > T$. □

The proof of Theorem 1 is in Appendix C and mainly based on the following generalization bound. The bound results from an online-to-batch conversion while using convergence guarantees of deep learning optimization.

Lemma 5.1. For any $\in (0;1)$; $\in (0;1)$; $\in (0; O(\frac{1}{L}))$, suppose m; $_1$; $_2$; K_1; K_2 satisfy the conditions in Eq. (5.2) and $(x_{;i}; r_{;i}) \sim D$; $\forall \in [t]; i \in [n]$. Let

$$x_t = \arg \max_{x_{t;i};i \in [n]} \left[f_2 \left(O\left(\tfrac{1}{t}\right) f_1(x_{t;i}; {}^{1}_{t\ 1}); {}^{2}_{t\ 1} \right) + f_1(x_{t;i}; {}^{1}_{t\ 1}) \right];$$

and r_t is the corresponding reward, given $(x_{t;i}; r_{t;i}); i \in [n]$. Then, with probability at least $(1 \quad)$ over the random of the initialization, it holds that

$$\begin{aligned} &\mathop{\mathbb{E}}_{(x_{t;i};r_{t;i});i \in [n]} \left[f_2 \left(O\left(\tfrac{1}{t}\right) f_1(x_{t;i}; {}^{1}_{t\ 1}); {}^{2}_{t\ 1} \right) r_t \quad f_1(x_t; {}^{1}_{t\ 1}) \; \middle| \; f x\ ; r\ g_{=1}^{t\ 1} \right] \\ &\qquad \sqrt{\frac{2}{t}} + O\left(\frac{3L}{\sqrt{2t}} \right) + (1 + 2\ ) \sqrt{\frac{2\log(O(tn=\ ))}{t}}; \end{aligned} \tag{5.4}$$

where the expectation is also taken over $(r^1_{t\ 1}; {}^{2}_{t\ 1})$ that are uniformly drawn from $(b^1; b^2)$; $\in [t \quad 1]$.

Remark 5.3. Lemma 5.1 provides a xed $O(\frac{1}{\sqrt{t}})$-rate generalization bound for exploitation-exploration networks $f_1; f_2$ in contrast with the relative bound w.r.t. the Neural Tangent Random Feature (NTRF) benchmark (Cao and Gu, 2019). We achieve this by working in the regression rather than classi cation setting and utilizing the convergence guarantees for square loss (Allen-Zhu et al., 2019). Note that the bound in Lemma 5.1 holds in the setting of bounded (possibly random) rewards $r \in [0; 1]$ instead of a xed function in the conventional classi cation setting.

6 EXPERIMENTS

In this section, we evaluate EE-Net on four real-world datasets comparing with strong state-of-the-art baselines. We rst present the setup of experiments, then show regret comparison and report ablation study. Codes are available[1] at

We use four real-world datasets: Mnist, Yelp, Movielens, and Disin, the details and settings of which are attached in Appendix A.

Figure 2: Regret comparison on Movielens and Yelp (mean of 10 runs with standard deviation (shadow)). With the same exploitation network f_1, EE-Net outperforms all baselines.

Baselines To comprehensively evaluate EE-Net, we choose 3 neural-based bandit algorithms, one linear and one kernelized bandit algorithms.

1. LinUCB (Li et al., 2010) explicitly assumes the reward is a linear function of arm vector and unknown user parameter and then applies the ridge regression and un upper con dence bound to determine selected arm.
2. KernelUCB (Valko et al., 2013) adopts a prede ned kernel matrix on the reward space combined with a UCB-based exploration strategy.
3. Neural-Epsilon adapts the epsilon-greedy exploration strategy on exploitation network f_1. I.e., with probability $1 \quad$, the arm is selected by $x_t = \arg\max_{i \in [n]} f_1(x_{t;i}; {}^{1})$ and with probability $$, the arm is chosen randomly.

[1] https://github.com/banyikun/EE-Net-ICLR-2022

Figure 3: Regret comparison on Mnist and Disin (mean of 10 runs with standard deviation (shadow)). With the same exploitation network f_1, EE-Net outperforms all baselines.

4. NeuralUCB (Zhou et al., 2020) uses the exploitation network f_1 to learn the reward function coming with an UCB-based exploration strategy.
5. NeuralTS (Zhang et al., 2021) adopts the exploitation network f_1 to learn the reward function coming with an Thompson Sampling exploration strategy.

Note that we do not report results of LinTS and KernelTS in experiments, because of the limited space in gures, but LinTS and KernelTS have been signi cantly outperformed by NeuralTS (Zhang et al., 2021).

Setup for EE-Net. To compare fairly, for all the neural-based methods including EE-Net, the exploitation network f_1 is built by a 2-layer fully-connected network with 100 width. For the exploration network f_2, we use a 2-layer fully-connected network with 100 width as well. For the decision maker f_3, by comprehensively evaluate both linear and nonlinear functions, we found that the most effective approach is combining them together, which we call "hybrid decision maker". In detail, for rounds $t \leq 500$, f_3 is set as $f_3 = f_2 + f_1$, and for $t > 500$, f_3 is set as a neural network with two 20-width fully-connected layers. Setting f_3 in this way is because the linear decision maker can maintain stable performance in each running (robustness) and the non-linear decision maker can further improve the performance (see details in Appendix B). The hybrid decision maker can combine these two advantages together. The con gurations of all methods are attached in Appendix A.

Results. Figure 2 and Figure 3 show the regret comparison on these four datasets. EE-Net consistently outperforms all baselines across all datasets. For LinUCB and KernelUCN, the simple linear reward function or prede ned kernel cannot properly formulate ground-truth reward function existed in real-world datasets. In particular, on Mnist and Disin datasets, the correlations between rewards and arm feature vectors are not linear or some simple mappings. Thus, LinUCB and KernelUCB barely exploit the past collected data samples and fail to select correct arms. For neural-based bandit algorithms, the exploration probability of Neural-Epsilon is xed and dif cult to be adjustable. Thus it is usually hard to make effective exploration. To make exploration, NeuralUCB statistically calculates a gradient-based upper con dence bound and NeuralTS draws each arm's predicted reward from a normal distribution where the standard deviation is computed by gradient. However, the con dence bound or standard deviation they calculated only consider the worst cases and thus may not be able represent the actual potential of each arm, and they cannot make "upward" and "downward" exploration properly. Instead, EE-Net uses a neural network f_2 to learn each arm's potential by neural network's powerful representation ability. Therefore, EE-Net can outperform these two state-of-the-art bandit algorithms. Note that NeuralUCB/TS does need two parameters to tune UCB/TS according to different scenarios while EE-Net only needs to set up a neural network and automatically learns it.

Ablation Study. In Appendix B, we conduct ablation study regarding the label function of f_2 and the different setting of f_3.

7 Conclusion

In this paper, we propose a novel exploration strategy, EE-Net. In addition to a neural network that exploits collected data in past rounds , EE-Net has another neural network to learn the potential gain compared to current estimation for exploration. Then, a decision maker is built to make selections to further trade off between exploitation and exploration. We demonstrate that EE-Net outperforms NeuralUCB and NeuralTS both theoretically and empirically, becoming the new state-of-the-art exploration policy.

Acknowledgements:We are grateful to Shiliang Zuo and Yunzhe Qi for the valuable discussions in the revisions of EE-Net. This research work is supported by National Science Foundation under Awards No. IIS-1947203, IIS-2002540, IIS-2137468, IIS-1908104, OAC-1934634, and DBI-2021898, and a grant from C3.ai. The views and conclusions are those of the authors and should not be interpreted as representing the of cial policies of the funding agencies or the government.

REFERENCES

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In Advances in Neural Information Processing Systems, pages 2312–2320, 2011.

M. Abeille and A. Lazaric. Linear thompson sampling revisited. Arti cial Intelligence and Statistics, pages 176–184. PMLR, 2017.

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In International Conference on Machine Learning, pages 127–135. PMLR, 2013.

H. Ahmed, I. Traore, and S. Saad. Detecting opinion spams and fake news using text classi cation. Security and Privacy, 1(1):e9, 2018.

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In International Conference on Machine Learning, pages 242–252. PMLR, 2019.

S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an in nitely wide neural net. In Advances in Neural Information Processing Systems, pages 8141–8150, 2019.

P. Auer. Using con dence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research, 3(Nov):397–422, 2002.

Y. Ban and J. He. Generic outlier detection in multi-armed bandit. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 913–923, 2020.

Y. Ban and J. He. Convolutional neural bandit: Provable algorithm for visual-aware advertising. arXiv preprint arXiv:2107.07438, 2021a.

Y. Ban and J. He. Local clustering in contextual multi-armed bandits. Proceedings of the Web Conference 2021, pages 2335–2346, 2021b.

Y. Ban, J. He, and C. B. Cook. Multi-facet contextual bandits: A neural network perspective. In The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021, pages 35–45, 2021.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-armed bandits. Journal of Machine Learning Research, 12(5), 2011.

Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. Advances in Neural Information Processing Systems, 32:10836–10846, 2019.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. Advances in neural information processing systems, 14, 2001.

E. Chlebus. An approximate formula for a partial sum of the divergent p-series. Applied Mathematics Letters, 22(5):732–737, 2009.

W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. In Proceedings of the Fourteenth International Conference on Arti cial Intelligence and Statistics, pages 208–214, 2011.

V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. 2008.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent nds global minima of deep neural networks. In International Conference on Machine Learning, pages 1675–1685. PMLR, 2019.

S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári. Parametric bandits: The generalized linear case. In Advances in Neural Information Processing Systems, pages 586–594, 2010.

D. Fu and J. He. SDG: A simplified and dynamic graph neural network. In SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pages 2273–2277. ACM, 2021.

F. M. Harper and J. A. Konstan. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4):1–19, 2015.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In Advances in neural information processing systems, pages 8571–8580, 2018.

J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In Advances in neural information processing systems, pages 817–824, 2008.

T. Lattimore and C. Szepesvári. Bandit algorithms. Cambridge University Press, 2020.

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995, 1995.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web, pages 661–670, 2010.

L. Li, Y. Lu, and D. Zhou. Provably optimal algorithms for generalized linear contextual bandits. In International Conference on Machine Learning, pages 2071–2080. PMLR, 2017.

S. Li, A. Karatzoglou, and C. Gentile. Collaborative filtering bandits. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 539–548, 2016.

X. Lu and B. Van Roy. Ensemble sampling. arXiv preprint arXiv:1705.07347, 2017.

C. Riquelme, G. Tucker, and J. Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. arXiv preprint arXiv:1802.09127, 2018.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. science, 290(5500):2323–2326, 2000.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3/4):285–294, 1933.

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. arXiv preprint arXiv:1309.6869, 2013.

Q. Wu, H. Wang, Q. Gu, and H. Wang. Contextual bandits in a collaborative environment. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 529–538, 2016.

P. Xu, Z. Wen, H. Zhao, and Q. Gu. Neural contextual bandits with deep representation and shallow exploration. arXiv preprint arXiv:2012.01780, 2020.

W. Zhang, D. Zhou, L. Li, and Q. Gu. Neural thompson sampling. In International Conference on Learning Representations, 2021.

D. Zhou, L. Li, and Q. Gu. Neural contextual bandits with ucb-based exploration. In International Conference on Machine Learning, pages 11492–11502. PMLR, 2020.

A DATASETS AND SETUP

MNIST dataset. MNIST is a well-known image dataset (LeCun et al., 1998) for the 10-class classification problem. Following the evaluation setting of existing works (Valko et al., 2013; Zhou et al., 2020; Zhang et al., 2021), we transform this classification problem into bandit problem. Consider an image $x \in R^d$, we aim to classify it from 10 classes. First, in each round, the image x is transformed into 10 arms and presented to the learner, represented by vectors in sequence $x_1 = (x; 0; \dots; 0); x_2 = (0; x; \dots; 0); \dots; x_{10} = (0; 0; \dots; x) \in R^{10d}$. The reward is defined as 1 if the index of selected arm matches the index of x's ground-truth class; Otherwise, the reward is 0.

Yelp[2] **and Movielens (Harper and Konstan, 2015) datasets.** Yelp is a dataset released in the Yelp dataset challenge, which consists of 4.7 million rating entries for 1.57×10^5 restaurants by 1.18 million users. MovieLens is a dataset consisting of 25 million ratings between 1.6×10^5 users and 6×10^4 movies. We build the rating matrix by choosing the top 2000 users and top 10000 restaurants(movies) and use singular-value decomposition (SVD) to extract a 10 dimension feature vector for each user and restaurant(movie). In these two datasets, the bandit algorithm is to choose the restaurants(movies) with bad ratings. We generate the reward by using the restaurant(movie)'s gained stars scored by the users. In each rating record, if the user scores a restaurant(movie) less than 2 stars (5 stars totally), its reward is 1; Otherwise, its reward is 0. In each round, we set 10 arms as follows: we randomly choose one with reward 1 and randomly pick the other 9 restaurants(movies) with 0 rewards; then, the representation of each arm is the concatenation of corresponding user feature vector and restaurant(movie) feature vector.

Disin (Ahmed et al., 2018) dataset. Disin is a fake news dataset on kaggle[3] including 12600 fake news articles and 12600 truthful news articles, where each article is represented by the text. To transform the text into vectors, we use the approach (Fu and He, 2021) to represent each article by a 300-dimension vector. Similarly, we form a 10-arm pool in each round, where 9 real news and 1 fake news are randomly selected. If the fake news is selected, the reward is 1; Otherwise, the reward is 0.

Configurations. For LinUCB, following (Li et al., 2010), we do a grid search for the exploration constant α over $(0.01; 0.1; 1)$ which is to tune the scale of UCB. For KernelUCB (Valko et al., 2013), we use the radial basis function kernel and stop adding contexts after 1000 rounds, following (Valko et al., 2013; Zhou et al., 2020). For the regularization parameter λ and exploration parameter α in KernelUCB, we do the grid search for λ over $(0.1; 1; 10)$ and for α over $(0.01; 0.1; 1)$. For NeuralUCB and NeuralTS, following setting of (Zhou et al., 2020; Zhang et al., 2021), we use the exploiation network f_1 and conduct the grid search for the exploration parameter ν over $(0.001; 0.01; 0.1; 1)$ and for the regularization parameter λ over $(0.01; 0.1; 1)$. For NeuralEpsilon, we use the same neural network f_1 and do the grid search for the exploration probability ϵ over $(0.01; 0.1; 0.2)$. For the neural bandits NeuralUCB/TS, following their setting, as they have expensive computation cost to store and compute the whole gradient matrix, we use a diagonal matrix to make approximation. For all neural networks, we conduct the grid search for learning rate η over $(0.01; 0.001; 0.0005; 0.0001)$. For all grid-searched parameters, we choose the best of them for the comparison and report the averaged results of 10 runs for all methods.

Create exploration samples for f_2. When the selected arm is not optimal in a round, the optimal arm must exist among the remaining arms, and thus the exploration consideration should be added to the remaining arms. Based on this fact, we create additional samples for the exploration network f_2 in practice. For example, in setting of binary reward, e.g., 0 or 1 reward, if the received reward $r_t = 0$ while selecting x_t, we add new train samples for f_2, $(x_{t;i}; c_r)$ for each $i \in [i] \setminus x_{t;i} \neq x_t$, where $c_r \in (0; 1)$ usually is a small constant. This measure can further improve the performance of EE-Net in our experiments.

B ABLATION STUDY

In this section, we conduct ablation study regarding the label function for exploration network f_2 and seting of decision maker f_3 on two representative datasets Movielens and Mnist.

[2] https://www.yelp.com/dataset

[3] https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset