Image Contextual Bandits: A Visual Transformer Approach

Ryan W.
University of Maryland, College Park

Abstract

Most contextual bandit algorithms are designed for traditional feature-based contexts and lack native image understanding. We propose ViTUCB, a bandit framework that bridges this gap by leveraging Vision Transformers with Low-Rank Adaptation for visual-aware decision making. Unlike the only existing image bandit CNNUCB, which relies on local convolutional features, our approach captures global visual context through transformer attention, enabling better image-based decision making. Experiments show that the proposed algorithm outperforms other UCB-based bandit algorithms on real-world image data sets, suggesting ViTUCB as an effective solution for visual content selection in recommendation systems.

1 Introduction

Online recommendation systems face a fundamental challenge: balancing the exploration of new content with the exploitation of known preferences. Contextual multi-armed bandits (MAB) provide an elegant framework for addressing this exploration-exploitation dilemma, with applications ranging from news article recommendation to online advertising. However, the majority of bandit algorithms operate on handcrafted feature vectors and lack native understanding of visual content—a critical limitation in today's visually-dominated digital landscape.

Contextual (MAB) is a sequential decision-making problem under uncertainty, combining the exploration-exploitation tradeoff of traditional MAB with contextual information. In each round t, the learner observes a set of arms (actions) represented by context vectors, selects an arm, and receives a reward. The goal is to maximize expected cumulative reward over T rounds.

Early bandit algorithms assumed linear reward functions, where the expected reward is linear to context vectors. However, the linear assumption is too restrictive in a real-world setting, making the algorithms impractical. More recently, neural bandit algorithms like NeuralUCB and its Thompson Sampling Variants [13, 14] have emerged, using deep neural networks to aid decision making. However, these methods primarily operate on structured feature vectors and lack native support for visual content.

The emergence of image-based platforms and visual advertising has created an urgent need for bandit algorithms that can directly process and understand visual information. The only existing image-aware bandit algorithm, CNNUCB [3], uses convolutional networks for decision making. While CNNUCB has shown promises in its performance, they suffer from inherent limitations of CNNs: capturing global image context and long-range dependencies. Vision Transformers (ViTs) offer a compelling alternative, with their self-attention mechanisms enabling superior modeling of complex visual patterns and compositional elements that are crucial for effective recommendation.

In this paper, we introduce ViTUCB, a novel contextual bandit algorithm that combines the representational power of Vision Transformers with the parameter efficiency of Low-Rank Adaptation (LoRA). Our approach not only addresses the visual understanding gap in bandit algorithms but also provides theoretical guarantees through a near-optimal regret bound of $\tilde{O}(\sqrt{T})$. By efficiently fine-tuning pre-trained ViTs using LoRA, we achieve state-of-the-art performance while maintaining computational feasibility for real-world deployment.

Our contributions are as follows: (1) We propose the first Vision Transformer-based bandit algorithm for visual recommendation, overcoming the limitations of CNN-based approaches; (2) We conducted extensive experiments, demonstrating empirically that ViTUCB outperforms existing methods, such as CNNUCB, across real-world image environments.

2 Related Work

The simplest case of contextual bandits is the linear case, where there exists $\theta*$ such that $\mathbb{E}(r_{t,a}) = \theta^{*\intercal} x_{t,a}$. One of the

first works is [8], proposing LinUCB with regret bound of $\tilde{O}(\sqrt{KdT})$. With later works like [1, 9] confirming or improving upon the regret bound. To handle nonlinear rewards, kernel-based methods [12] and generalized linear models [5] were developed. A Thompson Sampling variant of LinUCB was proposed by [2], also achieving a bound of $\tilde{O}(\sqrt{T})$.

Neural network's representation power has been leveraged to tackle the nonlinear case. [14] proposed Neural UCB, where they used a fully-connected network to estimate the rewards, achieving a regret bound of $\tilde{O}(\sqrt{T})$. They also utilized gradient features of the network to construct an upper confidence bound. [13] brings the neural network approach to Thompson Sampling, achieving the same regret bound.

However, fully-connected networks don't inherently capture visual patterns from an image. This led to the development of CNN-UCB, where they replace the fully-connected network in Neural UCB with a convolutional neural network (CNN). To adapt the regret bound to the CNN architecture, they perform regret analysis leveraging convolutional neural tangent kernel combined with ridge regression. Their analysis shows that although they also achieve a regret bound of $\tilde{O}(\sqrt{T})$, their bound is slightly better than that of Neural UCB.

Recently, advances in the Transformer architecture made it applicable to not just natural language processing tasks, but also computer vision [4]. The Transformer architecture adapted to computer vision is named Vision Transformer (ViT). Studies such as [10] show that ViTs outperform CNNs in many image classification tasks. Through the self-attention mechanism, ViT are able to capture long-range dependencies and global contextual information, something that CNNs struggle with. Motivated by the promises of ViTs, this paper proposes a UCB algorithm that utilizes the power of ViTs.

3 Problem Setting

We consider a stochastic contextual bandit problem where, at each round $t = 1, 2, \ldots, T$, the agent observes a set of K candidate contexts $\mathcal{X}_t = \{\boldsymbol{x}_{t,1}, \ldots, \boldsymbol{x}_{t,K}\}$ and must select an arm $a_t \in [K]$. Each arm corresponds to a context vector $\boldsymbol{x}_{t,a}$. In our case, each context $\boldsymbol{x}_{t,a} \in [0,255)^{3\times H\times W}$ corresponds to an image (or visual observation). After selecting an arm, the agent observes a reward $r_{t,a}$. The goal of the agent is to maximize cumulative reward (or equivalently, minimize cumulative regret).

Reward Model. Each arm a has an unknown reward

$$r_{t,a} = f^*(\boldsymbol{x}_{t,a}) + \xi_{t,a},$$

where $f^* : \mathcal{X} \to \mathbb{R}$ is the true but unknown reward function, and $\xi_{t,a}$ is sub-Gaussian noise with parameter σ.

Objective. Let $a_t^* = \arg\max_a f^*(\boldsymbol{x}_{t,a})$ denote the optimal action at time step t. The goal is to minimize cumulative expected regret:

$$\begin{aligned} R_T &= \mathbb{E}\left[\sum_{t=1}^{T}(r_{t,a_t^*} - r_{t,a_t})\right] \\ &= \sum_{t=1}^{T}\left(f^*(\boldsymbol{x}_{t,a_t^*}) - f^*(\boldsymbol{x}_{t,a_t})\right), \end{aligned}$$

4 ViT UCB

The backbone of the ViT-UCB algorithm is a Visual Transformer $f_{\text{ViT}}(\boldsymbol{x}; \theta_{\text{ViT}}, \theta_{\text{LoRA}})$ used to predict the reward of context x. The ViT is equipped with learnable LoRA adapters, denoted by θ_{LoRA}. This allows the ViT to be trained without tuning every parameter [6].

Algorithm 1 ViT-UCB with LoRA Parameters

Require: Vision Transformer $f_{\text{ViT}}(x; \theta_{\text{LoRA}})$, regularization $\lambda > 0$, exploration coefficient $\alpha > 0$

Initialize LoRA parameters θ^0_{LoRA} of the ViT
Initialize $A_0 = \lambda I$
for each round $t = 1, 2, \ldots, T$ **do**
  Observe candidate arms $\mathcal{X}_t = \{x_{t,1}, \ldots, x_{t,K}\}$
  for each arm $x_{t,i} \in \mathcal{X}_t$ **do**
    Compute ViT prediction: $\hat{r}_{t,i} = f_{\text{ViT}}(x_{t,i}; \theta^{t-1}_{\text{LoRA}})$
    Compute gradient: $g_{t,i} = \nabla_{\theta_{\text{LoRA}}} f_{\text{ViT}}(x_{t,i}; \theta^{t-1}_{\text{LoRA}})$
    Compute exploration bonus: $b_{t,i} = \alpha \| \frac{g_{t,i}}{\sqrt{d_{LoRA}}} \|_{A_{t-1}^{-1}}$
    Compute UCB: $U_{t,i} = \hat{r}_{t,i} + b_{t,i}$
  end for
  Select arm $a_t = \arg\max_i U_{t,i}$
  Observe reward r_t for arm a_t
  Update Gram matrix: $A_t = A_{t-1} + g_{t,a_t} g_{t,a_t}^\top$
  Update LoRA parameters: θ^t_{LoRA} with Gradient Descent on observed rewards $\{(x_{i,a_i}, r_i)\}_{i=1}^T$
end for

5 Theoretical Analysis of ViT-LoRA-UCB

In this section, we analyze the ViTUCB algorithm, ultimately leading up to our main theorem.

5.1 Setup

5.1.1 ViT-LoRA Model

We employ a Vision Transformer (ViT) model with frozen pre-trained backbone and trainable LoRA (Low-Rank Adaptation) adapters. Let:

- $\boldsymbol{\theta}_{\text{ViT}} \in \mathbb{R}^{d_{\text{ViT}}}$: Fixed pre-trained ViT parameters

- $\boldsymbol{\theta}_{\text{LoRA}} \in \mathbb{R}^{d_{\text{LoRA}}}$: Trainable LoRA parameters ($d_{\text{LoRA}} \ll d_{\text{ViT}}$)
- $f_{\text{ViT}}(\boldsymbol{x}; \boldsymbol{\theta}_{\text{ViT}}, \boldsymbol{\theta}_{\text{LoRA}})$: ViT output with LoRA adapters

The LoRA adaptation for a weight matrix $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ is:

$$\boldsymbol{W}' = \boldsymbol{W} + \boldsymbol{B}\boldsymbol{A}$$

where $\boldsymbol{A} \in \mathbb{R}^{r \times n}$, $\boldsymbol{B} \in \mathbb{R}^{m \times r}$ are low-rank matrices with rank $r \ll \min(m, n)$.

5.1.2 Neural Tangent Kernel (NTK) Features

We operate in the *Lazy Training* regime, where the network is linearized around its initialization $\boldsymbol{\theta}^0_{\text{LoRA}}$. We define the scaled gradient feature map $\boldsymbol{\phi}(\boldsymbol{x})$:

$$\boldsymbol{g}_{\text{ViT}}(\boldsymbol{x}) := \nabla_{\boldsymbol{\theta}_{\text{LoRA}}} f_{\text{ViT}}(\boldsymbol{x}; \boldsymbol{\theta}_{\text{ViT}}, \boldsymbol{\theta}_{\text{LoRA}})\Big|_{\boldsymbol{\theta}_{\text{LoRA}} = \boldsymbol{\theta}^0_{\text{LoRA}}} \tag{1}$$

$$\boldsymbol{\phi}(\boldsymbol{x}) := \frac{\boldsymbol{g}_{\text{ViT}}(\boldsymbol{x})}{\sqrt{d_{\text{LoRA}}}} \tag{2}$$

Here, $\sqrt{d_{\text{LoRA}}}$ is the standard NTK scaling factor for the width of the LoRA adapters.

Assumption 1 (Representability)**.** *We assume the reward function is well-approximated by the first-order tangent model in the LoRA subspace. In other words, there exists $\boldsymbol{\theta}^*_{LoRA} \in \mathbb{R}^{d_{LoRA}}$ such that for any $\boldsymbol{x}$ in the training set:*

$$f^*(\boldsymbol{x}) = f_{lin}(x) + \varepsilon(\boldsymbol{x})$$

where

$$|\varepsilon(\boldsymbol{x})| \leq \varepsilon_{\max}$$

With a overparameterized transformer and a LoRA subspace that is expressive enough, $\varepsilon_{\max}$ is expected to be small.

5.1.3 The Ridge Estimator

At round t, we have observed history $\mathcal{H}_t = \{(\boldsymbol{x}_\tau, r_\tau)\}_{\tau=1}^t$. We define the regularized design matrix (Gram matrix):

$$\boldsymbol{A}_t = \lambda \boldsymbol{I}_p + \sum_{\tau=1}^{t} \boldsymbol{\phi}(\boldsymbol{x}_\tau)\boldsymbol{\phi}(\boldsymbol{x}_\tau)^\top \tag{3}$$

The ridge regression estimator for the residual reward is:

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{A}_t^{-1} \sum_{\tau=1}^{t} \left(r_\tau - f_{\text{ViT}}(\boldsymbol{x}_\tau; \boldsymbol{\theta}_{\text{ViT}}, \boldsymbol{\theta}^0_{\text{LoRA}}) \right) \boldsymbol{\phi}(\boldsymbol{x}_\tau) \tag{4}$$

5.2 Error Decomposition

We aim to bound the error $|f^*(\boldsymbol{x}) - f_{\text{ViT}}(\boldsymbol{x}; \boldsymbol{\theta}^t_{\text{LoRA}})|$. Note that in Neural UCB, the "prediction" used for decision making is often the linearized prediction, but we analyze the error relative to the actual network parameter $\boldsymbol{\theta}^t_{\text{LoRA}}$ obtained via gradient descent.

Lemma 1 (Tri-term Decomposition)**.** *Define the linearized network function:*

$$f_{lin}(\boldsymbol{x}; \boldsymbol{\theta}) := f_{ViT}(\boldsymbol{x}; \boldsymbol{\theta}^0_{LoRA}) + \left\langle \boldsymbol{\phi}(\boldsymbol{x}), \sqrt{d_{LoRA}}(\boldsymbol{\theta} - \boldsymbol{\theta}^0_{LoRA}) \right\rangle$$

Let $\boldsymbol{\theta}^*$ be the optimal parameter in the parameter space that best approximates f^*. The error decomposes as:

$$\begin{aligned} |f^*(\boldsymbol{x}) - f_{ViT}(\boldsymbol{x}; \boldsymbol{\theta}^t_{LoRA})| \leq & \underbrace{|f_{lin}(\boldsymbol{x}; \boldsymbol{\theta}^*) - f_{lin}(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)|}_{\textit{(I) Estimation Variance}} \\ & + \underbrace{|f^*(\boldsymbol{x}) - f_{lin}(\boldsymbol{x}; \boldsymbol{\theta}^*)|}_{\textit{(II) Misrepresentation}} \\ & + \underbrace{|f_{lin}(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t) - f_{ViT}(\boldsymbol{x}; \boldsymbol{\theta}^t_{LoRA})|}_{\textit{(III) Nonlinearity \& Drift}} \end{aligned}$$

Proof. Apply triangle inequality: $|A - B| \leq |A - C| + |C - D| + |D - B|$. □

5.3 Bounding Term (I): Estimation Variance

This term represents the uncertainty in estimating the linear parameters due to limited data samples.

Lemma 2 (Self-Normalized Bound)**.** *Assume the noise ξ_t is R-sub-Gaussian and $\|\boldsymbol{\theta}^* - \boldsymbol{\theta}^0_{LoRA}\|_2 \leq S$. With probability at least $1 - \delta$:*

$$|f_{lin}(\boldsymbol{x}; \boldsymbol{\theta}^*) - f_{lin}(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)| \leq \alpha_t \|\boldsymbol{\phi}(\boldsymbol{x})\|_{\boldsymbol{A}_t^{-1}}$$

where $\alpha_t = R\sqrt{d_{LoRA} \log(1 + tL^2/\lambda d_{LoRA}) + 2\log(1/\delta)} + \sqrt{\lambda d_{LoRA}} S$.

Proof. Recall $f_{\text{lin}}(\boldsymbol{x}; \boldsymbol{\theta}) = \text{Const} + \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{w} \rangle$, where $\boldsymbol{w} = \sqrt{d_{\text{LoRA}}}(\boldsymbol{\theta} - \boldsymbol{\theta}^0_{\text{LoRA}})$. The difference is:

$$\Delta(\boldsymbol{x}) = \left\langle \boldsymbol{\phi}(\boldsymbol{x}), \sqrt{d_{\text{LoRA}}}(\boldsymbol{\theta}^* - \boldsymbol{\theta}^0_{\text{LoRA}}) - \sqrt{d_{\text{LoRA}}}(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^0_{\text{LoRA}}) \right\rangle$$

Let $\boldsymbol{w}^* = \sqrt{d_{\text{LoRA}}}(\boldsymbol{\theta}^* - \boldsymbol{\theta}^0_{\text{LoRA}})$ and $\hat{\boldsymbol{w}}_t = \sqrt{d_{\text{LoRA}}}(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^0_{\text{LoRA}})$. The quantity $\hat{\boldsymbol{w}}_t$ is exactly the Regularized Least Squares estimator for the true parameter $\boldsymbol{w}^*$ under features $\boldsymbol{\phi}(\boldsymbol{x})$. By the Theorem 2 of [1], the estimation error is bounded:

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_{\boldsymbol{A}_t} \leq \alpha_t$$

Applying Cauchy-Schwarz:

$$|\langle \boldsymbol{\phi}(\boldsymbol{x}), \hat{\boldsymbol{w}}_t - \boldsymbol{w}^* \rangle| \leq \|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_{\boldsymbol{A}_t} \|\boldsymbol{\phi}(\boldsymbol{x})\|_{\boldsymbol{A}_t^{-1}} \leq \alpha_t \|\boldsymbol{\phi}(\boldsymbol{x})\|_{\boldsymbol{A}_t^{-1}}$$

□

5.4 Bounding Term (III): Linearization & Drift

This term captures the error arising from the fact that the Neural Network is not actually linear, and that the parameters $\boldsymbol{\theta}^t_{\text{LoRA}}$ (found via Gradient Descent) may drift from the initialization $\boldsymbol{\theta}^0_{\text{LoRA}}$.

5.4.1 Smoothness of the ViT-LoRA Architecture

We must prove that the Hessian of the function is bounded.

Lemma 3 (Bounded ViT Hessian)**.** *Consider bounded input (as with the case of images) $X \in \mathbb{R}^{n\times d}$ such that $\|X\| \le B$ for some $B < \infty$*

$$\left\|\nabla^2_{\boldsymbol{\theta}} f_{ViT}(\boldsymbol{x};\boldsymbol{\theta})\right\|_2 \le H_{ViT} < \infty$$

Proof. Decompose the Hessian as in [11]:

$$H = H_O + H_F,$$

where H_O is the outer-product (Gauss-Newton) term and H_F is the functional Hessian term.

1. Outer-product term (H_O): According to [11], for a self-attention block,

$$\|H_O\| = O(\|X\|^6).$$

Since the input X is bounded, $\|X\| \le B$, it follows that $\|H_O\| \le C_1 < \infty$ for some constant C_1.

2. Functional Hessian term (H_F): Similarly, [11] shows $\|H_F\| = O(\|X\|^5)$, so $\|H_F\| \le C_2 < \infty$ for bounded X.

3. GELU activation: GELU is smooth (C^∞) with bounded first and second derivatives. Therefore, the chain rule contributions from activations do not increase the Hessian beyond a constant factor.

4. LayerNorm: LayerNorm normalizes each patch embedding to bounded variance and has bounded first and second derivatives with respect to its input. Hence, it prevents large amplification of the Hessian.

Combining the above, the spectral norm of the layer Hessian is bounded:

$$\|H\| \le \|H_O\| + \|H_F\| \le C_1 + C_2 < \infty.$$

Extending this argument to all layers of a ViT, the full Hessian remains bounded by a finite constant C_{ViT} that depends on network width, depth, and parameters. □

5.4.2 Taylor Expansion Bound

Lemma 4 (Linearization Remainder)**.**

$$|f_{lin}(\boldsymbol{x};\hat{\boldsymbol{\theta}}_t) - f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}^t_{LoRA})| \le C_{opt} + \frac{H_{ViT}}{2}\|\boldsymbol{\theta}^t_{LoRA} - \boldsymbol{\theta}^0_{LoRA}\|_2^2$$

Proof. We split the difference:

$$\begin{aligned}|f_{\text{lin}}(\boldsymbol{x};\hat{\boldsymbol{\theta}}_t) - f_{\text{ViT}}(\boldsymbol{x};\boldsymbol{\theta}^t_{\text{LoRA}})| &\le |f_{\text{lin}}(\boldsymbol{x};\hat{\boldsymbol{\theta}}_t) - f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta}^t_{\text{LoRA}})| \\ &\quad + |f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta}^t_{\text{LoRA}}) - f_{\text{ViT}}(\boldsymbol{x};\boldsymbol{\theta}^t_{\text{LoRA}})| \\ &= \left\langle \boldsymbol{\phi}(\boldsymbol{x}), \sqrt{d_{\text{LoRA}}}(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^t_{\text{LoRA}})\right\rangle \\ &\quad + |R_2(\boldsymbol{\theta}^t_{\text{LoRA}})|\end{aligned}$$

1. The first term is the Optimization Error C_{opt}: the discrepancy between the Ridge Regression solution $\hat{\boldsymbol{\theta}}_t$ and the Gradient Descent solution $\boldsymbol{\theta}^t_{\text{LoRA}}$. In the limit of infinite width/rank, these trajectories coincide. For finite cases, this is bounded.
2. The second term is the Taylor Remainder. By Taylor's theorem with Lagrange remainder:

$$R_2(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\xi})(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$$

Using Lemma 3, this is bounded by $\frac{H_{\text{ViT}}}{2}\|\boldsymbol{\theta}^t_{\text{LoRA}} - \boldsymbol{\theta}^0_{\text{LoRA}}\|_2^2$. □

5.5 Main Theorem

These results lead to the main theorem of this paper. We provide a bound for the difference between expected reward and the reward estimate from ViT. Term I of the bound is equivalent to the exploration bonus used in Algorithm 1, which justifies the choice of exploration bonus.

Theorem 1 (Error Bound for LoRA-Adapted ViT)**.** *Let $f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}^t_{LoRA})$ denote the output of a ViT with LoRA parameters $\boldsymbol{\theta}^t_{LoRA}$ at time t, and let $f^*(\boldsymbol{x})$ be the target function. Define the feature vector*

$$\boldsymbol{\phi}(\boldsymbol{x}) := \frac{\boldsymbol{g}_{ViT}(\boldsymbol{x})}{\sqrt{d_{LoRA}}}$$

and the matrix $\boldsymbol{A}_t$ from the ridge regression / NTK update. Then, for each input $\boldsymbol{x}$, the prediction error is bounded as

$$\begin{aligned}\left|f^*(\boldsymbol{x}) - f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}^t_{LoRA})\right| \le &\underbrace{\alpha_t \|\boldsymbol{\phi}(\boldsymbol{x})\|_{\boldsymbol{A}_t^{-1}}}_{\textit{Term I}} \\ &+ \underbrace{\varepsilon_{max}}_{\textit{Term II}} \\ &+ \underbrace{C_{opt} + \frac{H_{ViT}}{2}\|\boldsymbol{\theta}^t_{LoRA} - \boldsymbol{\theta}^0_{LoRA}\|_2^2}_{\textit{Term III}}.\end{aligned}$$

Equivalently, defining the aggregate bias term

$$\beta := \varepsilon_{max} + C_{opt} + \frac{H_{ViT}}{2}\|\boldsymbol{\theta}^t_{LoRA} - \boldsymbol{\theta}^0_{LoRA}\|_2^2, \tag{5}$$

the error can be compactly written as

$$\left|f^*(\boldsymbol{x}) - f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}^t_{LoRA})\right| \le \alpha_t \|\boldsymbol{\phi}(\boldsymbol{x})\|_{\boldsymbol{A}_t^{-1}} + \beta. \tag{6}$$

Proof. The decomposition into 3 terms results from Lemma 1. Then,

- Term I results from Lemma 2
- Term II results from Assumption 1
- Term III results from Lemma 4

□

6 Regret Analysis

Following the established proof methodology for neural bandit algorithms [3, 14], ViT-LoRA-UCB achieves $\tilde{O}(\sqrt{T})$ regret. The key steps are:

1. **Confidence Bound:** Theorem 1 provides a high-probability bound on the prediction error.
2. **Regret Decomposition:** The cumulative regret can be bounded by the sum of confidence widths:

$$R_T \leq 2 \sum_{t=1}^{T} \text{UCB}(\boldsymbol{x}_t)$$

3. Standard analysis shows that the sum of confidence widths scales as $\tilde{O}(\sqrt{T})$.

Therefore, ViT-LoRA-UCB has the same order-optimal regret bound as prior neural bandit algorithms, while leveraging the parameter efficiency of LoRA fine-tuning.

7 Experiments

In this section, we evaluate ViT-UCB empirically by comparing it to LinUCB and Neural UCB. Three experiments were conducted, all of which contain images as context. The first experiment is fairly simple, where all algorithms are expected to perform well. The second and third experiment would be more complicated with parallels to a real-world recommendation system. In the first two experiments, each arm has a fixed reward distribution, similar to the classic MAB. This allows us to analyze the algorithms' behavior in more detail.

7.1 Experiment 1: Handwritten Digits

The first set of experiments consists of contextual bandits with $K = 10$ arms and $T = 10000$ rounds. The image dataset used is a collection of handwritten digits from Kaggle, with each digit consisting approximately 10000 images. Each arm were numbered from $i = 0 \ldots 9$. Arm i will have context images of the handwritten digit i. At each time step, each arm would uniformly sample a random image from their respective digit images. Arm i will have reward sampled from a normal distribution $N(i, 9)$. See 1 for a visualization of the experiment.

Figure 1: Visualization of Experiment 1

7.2 Experiment 2: Anime Ratings (Fixed Arm Distribution)

The second set of experiments consists of contextual bandits with $K = 3$ arms and $T = 10000$ rounds. The images are 10000 anime thumbnails pulled from popular anime database MyAnimeList, via the Jikan API. The animes are sorted by their rating and partitioned into 3 groups, numbered $i = 0 \ldots 2$. For example, if the animes have ratings $\${7.2, 2.6, 4.4, 5.3, 9.1, 8.3\}$, group 0 would consist of $\{2.6, 4.4\}$; group 1 would consist of $\{5.3, 7.2\}$; group 2 would consist of $\{8.3, 9.1\}$. At each time step, arm i would uniformly sample a random thumbnail from group i. Arm i will have reward sample from a normal distribution $N(\text{mean rating of group } i, 1)$. Following the previous example, the reward of group 1 would be sampled from $N(6.25, 1)$. The MyAnimeList database had the following means for each group: 5.5, 6.5, 7.4. This experiment mirrors a content-based recommender system, where the agent must estimate the reward of an item given its features.

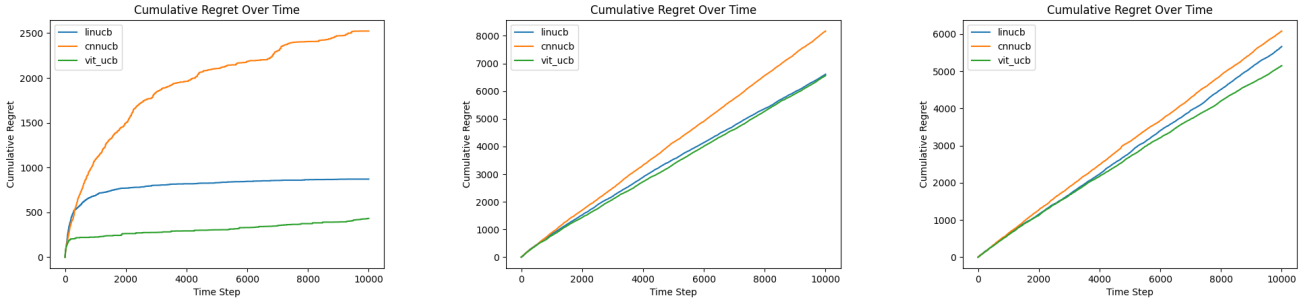
7.3 Experiment 3: Anime Ratings (Non-fixed Arm Distribution)

The third set of experiments is similar to the second. However, the animes are not grouped. At each time step, $K = 3$ animes are sampled and their thumbnails are shown to the agent as context. The reward for each action is sampled from $N(\text{Rating of anime}, 1)$. This iteration of experiment is more analogous to the experiments found in other contextual bandit literature.

7.4 Algorithm Configurations

For LinUCB, given the complex nature of images, it is expected that LinuCB would perform poorly on flattened image vectors. Instead, each image would first pass through a pre-trained vision model (vit-tiny-patch16-224 5.7M parameters), and LinUCB would take the output embedding of the vision model as input. LinUCB has hyperparameter α, which is the coefficient to the exploration bonus. α is tuned from 1 to 100, using 50 trials of Bayesian Optimization on the environment of experiment 2 but with $T = 1500$.

For CNNUCB, we use a CNN with 2 convolutional layers connected with 2 fully-connected layers. The first convolu-



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

Figure 2: Regret comparison of each experiment

tional layer consists of 32 channels and the second contains 64 channels. Both layers have kernel size of 3 and padding of 1, each followed by a ReLU activation function and a 2D max pooling with kernel size 2 and stride of 2. Similar to LinUCB, CNNUCB has tunable exploration factor α. In addition, it has a tunable regularization parameter λ. The hyperparameters are once again tuned using the same Bayesian Optimization process with $\lambda \in [0.1, 10]$.

For ViTUCB, we once again used vit-tiny-patch16-224 as our frozen pretrained model. In addition to the hyperparameters of CNNUCB, ViTUCB has LoRA Rank $r \in \{4, \ldots, 20\}$ and LoRA coefficient $\alpha_{\text{LoRA}} \in [4, 32]$.

7.5 Practical Considerations

For CNNUCB and ViTUCB, the high computation and memory cost to store and compute A_t led some constraints. As a reminder, $A_t \in \mathbb{R}^{n \times n}$, where n is the number of trainable parameters. For CNNUCB, we restricted the input image size to 50×50 to reduce n. For ViTUCB, even with LoRA—which significantly reduces the n by orders of magnitudes—and the use of the smallest available public ViT, n is still too high. To compensate for this, only a fraction of the LoRA parameters are used in the calculation of A_t. [3] suggested a way to approximate A_t by only storing the diagonal elements of A_t. However, the exploration bonus of this approach was comparably less stable, hence this approach was not used (see Appendix A for details).

Another constraint is that performing gradient descent over all previous context-reward pairs was too time consuming. Therefore, at each time step, only the most recent 50 data points are used for gradient descent.

7.6 Results

Figures 2a, 2b and 2c show each algorithm's cumulative regret on each experiment, respectively. ViTUCB achieves the best performance across all experiments. Surprisingly, in the experiments, LinUCB achieves results similar to ViTUCB

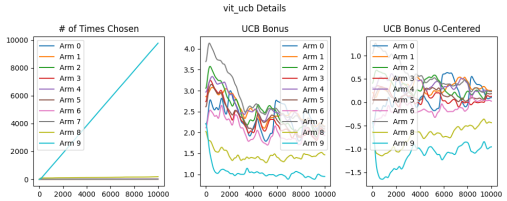


Figure 3: Details of ViTUCB on Experiment 1

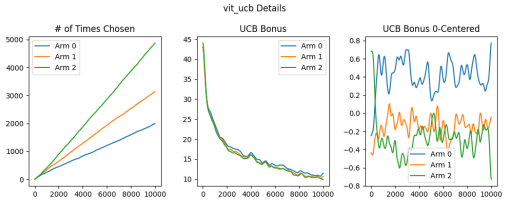


Figure 4: Details of ViTUCB on Experiment 2

and outperforms CNNUCB. This suggests that there may be a linear correlation between the embedding of the images to the rewards. Figures 3, 4, and 5 show the details of the first two experiments. The left-most subplot show the number of times each arm is selected over time. An optimal bandit algorithm should have the curve for better arms outgrowing the curve for worse arms. The middle subplots represent the UCB exploration bonus over time, while the right-most subplots are the curves but with each time step t normalized such that the sum is 0. Both the middle and right-most graphs are

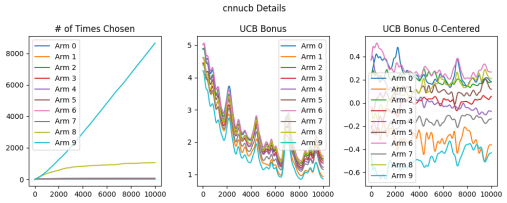
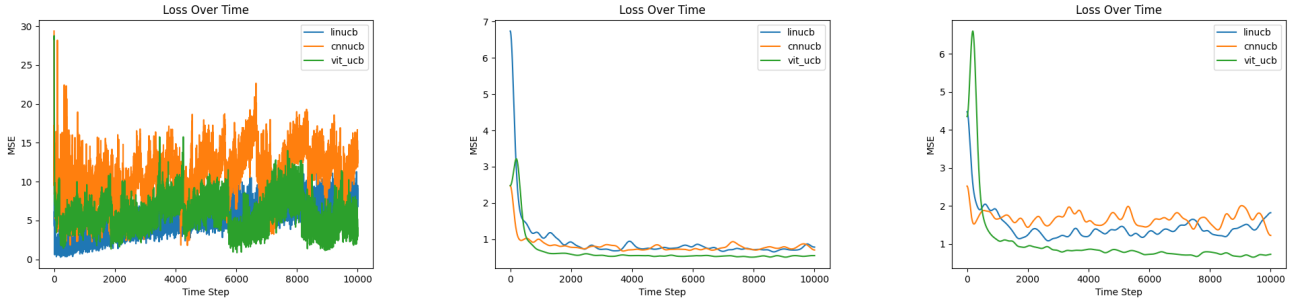


Figure 5: Details of CNNUCB on Experiment 1



(a) Experiment 1 (b) Experiment 2 (smoothed) (c) Experiment 3 (smoothed)

Figure 6: Each model's reward estimate error. The error at time step t is $\frac{1}{k}\sum_{k=1}^{K}\left(f^*(x_{t,k}) - f_{\text{ViT}}(x_{t,k})\right)^2$. ViTUCB having the lowest loss in all 3 experiments show that it was able to capture the context-reward relationship better than CNNUCB and LinUCB.

smoothed out with Gaussian Filtering ($\sigma = 10$) so that it is easier to notice overall trends. For these subplots, the curves for better arms should be lower because the agent should have picked these arms more often, leading to less uncertainty.

In experiment 1, all 3 algorithms achieved $\tilde{O}(\sqrt{T})$ looking regret curves, suggesting that all algorithms were able to learn the optimal policy. It is surprising that CNNUCB performed much worse than the other two algorithms on this task, given that CNNs have proven to have really strong performances in handwritten digit classification [7]. This performance gap likely stems from the difficulty of training the CNN feature extractor from scratch under high-variance Gaussian reward noise, whereas ViTUCB benefits from stable, pre-trained representations. Comparing figures 3 and 5 elucidates this failure mode: CNNUCB continued to pick arm 8 after the initial attempts while ViTUCB essentially picked arm 9 exclusively (recall that arm i has expected value of i). The middle and right-most subplots of figure 5 show that exploration bonus for arm 8 is unusually high, leading to its high pick rate. In contrast, figure 3 shows that ViTUCB's exploration bonuses more aligned with our expectations.

In experiment 2 and 3, the regret curves look less ideal, but ViTUCB still remains the best. This is expected because ViT's advantage in global context allows it to better relationship between a thumbnail and its corresponding rating. As seen from left-most subplot 4, ViTUCB once again picks the arm 2 the most times. And although subtle, we can see that arm 2's curve seem to concave up while those of other arms concave down, suggesting that the agent is prioritizing the best arm more and more. The exploration bonus also displays expected behavior, with arms being picked more having a lower exploration bonus.

To further investigate the source of ViTUCB's superior performance, we analyze the accuracy of the underlying reward estimates. Figure 6 visualizes the Mean Squared Error (MSE) between the model's predicted reward and the true reward function over time. Across all three experiments, ViTUCB consistently achieves the lowest estimation error. This indicates that the Vision Transformer's self-attention mechanism captures the complex non-linear relationship between the global visual context and the reward signal more effectively than the baseline methods. Since the UCB strategy relies heavily on the accuracy of the mean reward estimate to construct valid confidence bounds, this lower estimation error directly translates to the reduced cumulative regret observed in Figure 2. Conversely, the higher MSE observed in CNNUCB lead to less reliable upper confidence bounds and suboptimal arm selection.

8 Conclusion

In this work, we introduced ViTUCB, a novel algorithm for visual contextual bandits that leverages the global representation power of Vision Transformers. By utilizing ViTs' innate strength in modeling long-range dependencies and holistic image composition, ViTUCB more effectively captures the complex relationships between image-based contexts and their rewards. We empirically validated our approach on multiple benchmarks, including a scenario that mirrors recommender systems, demonstrating consistent and superior performance over state-of-the-art baselines. Our results underscore the significant potential of modern transformer architectures in improving decision-making systems reliant on visual data.

9 Future Work

Several improvements could be made to the theoretical aspect of this paper. First, the UCB bound for can be elaborated upon. As it stands, the UCB bound hinges on several values such as C_{opt} and H_{ViT}. These values can be further explored to provide a more concrete UCB bound. Also, similar to [3], the UCB bound can involve analysis on the nature of Gradient Descent. Next, a rigorous analysis for the regret bound

was not provided in this paper, so adapting the regret bound analysis from [3, 14] to the ViT case is necessary. As with other related UCB works such as LinUCB and NeuralUCB, a natural extension to this paper would be to explore the Thompson Sampling Variant of ViTUCB. As noted in Section 7.5, many modifications were made in the experiment due to the hardware constraints. Therefore, if given more computation power, the authentic algorithm should be reconducted.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24:2312–2320, 2011.

[2] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. *CoRR*, abs/1209.3352, 2012.

[3] Yikun Ban and Jingrui He. Convolutional neural bandit for visual-aware recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1–9, 2021.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Sarah Filippi, Olivier Cappé, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. *Advances in Neural Information Processing Systems*, 23:586–594, 2010.

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[7] 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.

[8] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[9] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548, 2016.

[10] José Maurício, Inês Domingues, and Jorge Bernardino. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, 13(9), 2023.

[11] Weronika Ormaniec, Felix Dangel, and Sidak Pal Singh. What does it mean to be a transformer? insights from a theoretical hessian analysis, 2025.

[12] Michal Valko, Nathan Korda, Remi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[13] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

[14] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

A Approximating the Matrix A

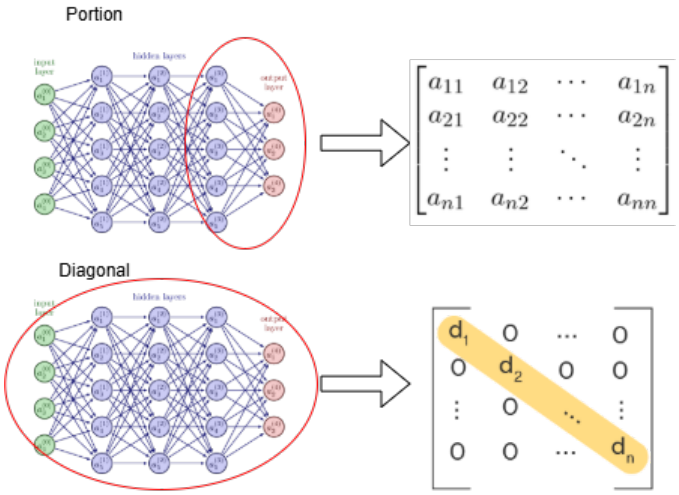


Figure 7: Visualization of methods to approximate A

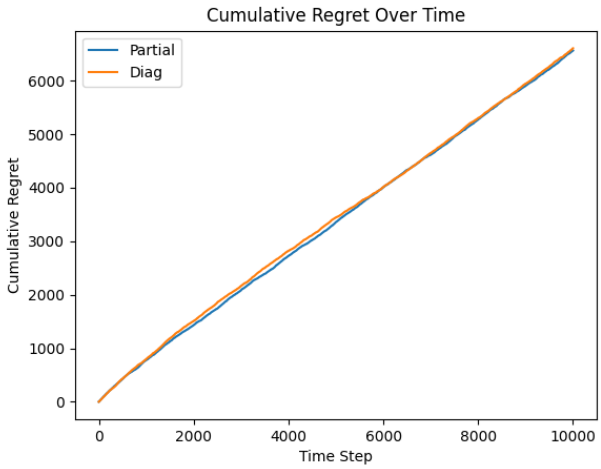


Figure 8: Regret Comparison of Storing Partial Parameters vs. Storing Diagonal of A

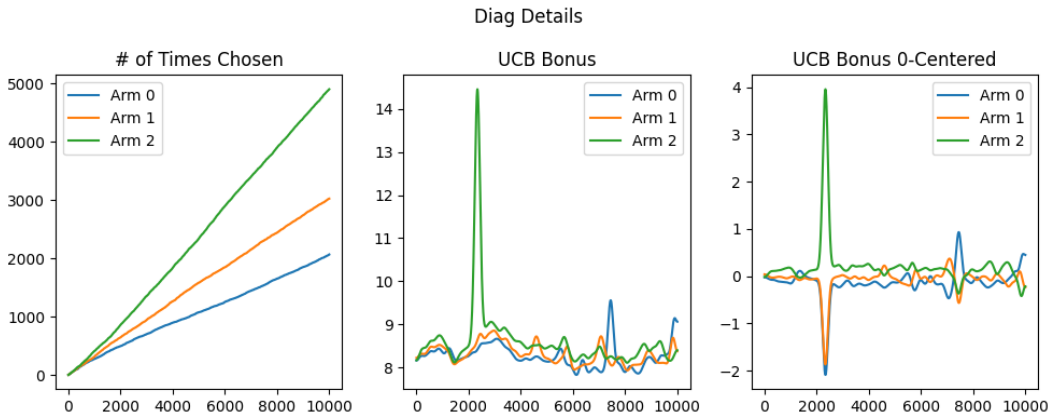


Figure 9: Details of Diagonal Method

Here, two different methods of approximating the full matrix A are compared (See 7). The first way is to only track a portion of the gradient features. Recall the matrix A is defined as $A = \lambda I + \sum_{t=1}^{T} g_{t,a_t} g_{t,a_t}^\top$. Instead of using the full g, we only use $g_{t,a_t}(i:j)$ where $i:j$ are the indices we choose to track, so $A \approx \lambda I + \sum_{t=1}^{T} g_{t,a_t}(i:j) g_{t,a_t}(i:j)^\top$. The other method is to only store the diagonal of the matrix. Looking at the recursive definition of $A_t = A_{t-1} g_{t,a_t} g_{t,a_t}^\top$, we can see that Sherman-Morrison formula can be used to update $A^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} g_{t,a_t} g_{t,a_t}^\top A_{t-1}^{-1}}{1 + g_{t,a_t}^\top A_{t-1}^{-1} g_{t,a_t}}$. Storing only the diagonal, we have $d_t = d_{t-1} - \frac{(d_{t-1} \odot g)^{\circ 2}}{1 + g^\top (d_{t-1} \odot g)}$. Figure 8 shows that the two ways of approximating A yields similar regret. However, figure 7 reveals that the exploration bonus of storing only diagonals is quite noisy.