EMPIRICAL ANALYSIS OF REPRESENTATION LEARNING AND EXPLORATION IN NEURAL KERNEL BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural bandits have been shown to provide an efficient solution to practical sequential decision tasks that have nonlinear reward functions. The main contributor to that success is approximate Bayesian inference, which enabled neural network (NN) training with uncertainty estimates. However, Bayesian NNs often suffer from a prohibitive computational overhead or operate on a subset of parameters. Certain classes of infinite neural networks were shown to directly correspond to Gausian processes (GP) with neural kernels (NK). NK-GPs provide accurate uncertainty estimates and can be trained faster than most Bayesian NNs. We propose to guide common bandit policies with NK distributions and show that NK bandits achieve state-of-the-art performance on nonlinear structured data. Moreover, we propose a framework for measuring independently the ability of a bandit algorithm to explore and exploit, and use it to analyze the impact of NK distributions w.r.t. those two aspects. We consider policies based on a GP and a Student's t-process (TP). Furthermore, we study practical considerations, such as training frequency and model partitioning. We believe our work will help better understand the impact of utilizing NKs in applied settings.

1 INTRODUCTION

Contextual bandit algorithms, like upper confidence bound (UCB) (Auer et al., 2002) or Thompson sampling (TS) (Thompson, 1933), typically utilize Bayesian inference to facilitate both representation learning and uncertainty estimation. Neural networks (NN) are applied increasingly often to model non-linear relations between contexts and rewards (Allesiardo et al., 2014; Collier & Llorens, 2018). Due to lack of a one-size-fits-all solution to model uncertainty with neural networks, various NN models result in trade offs in bandits framework. Riquelme et al. (2018) compared a comprehensive set of modern Bayesian approximation methods in their benchmark, and observed that state-of-the-art Bayesian NNs require training times prohibitive for practical bandit applications. Classic approaches, on the other hand, lack complexity to accurately guide a nonlinear policy. Riquelme et al. (2018) showed that their neural-linear method provides the best practical performance and strikes the right balance between computational efficiency and Bayesian parameter estimation.

Bandit policies balance exploration and exploitation with two terms: (1) a mean reward estimate and (2) an uncertainty term (Lattimore & Szepesvári, 2020). From a Bayesian perspective those two terms represent the first two moments of a posterior predictive distribution, e.g. a Gaussian process (GP). Research on neural kernels (NK) has recently established a correspondence between deep networks and Gaussian processes (GP) (Lee et al., 2018). The resulting model can be trained more efficiently than most Bayesian NNs. In this work we focus on the conditions in which NK-GPs provide a competitive advantage over other NN approaches in bandit settings. Even though NKs have been shown to lack the full representational power of the corresponding NNs, they outperform finite fully-connected networks in small data regimes (Arora et al., 2019b), and combined with GPs, successfully solve simple reinforcement learning tasks (Goumiri et al., 2020). NK-GPs provide a fully probabilistic treatment of infinite NNs, and therefore result in more accurate predictive distributions than those used in most of the state-of-the-art neural bandit models. We hypothesized that NKs outperform NNs for datasets possessing certain characteristics, like data complexity, the need for exploration, or reward type. The full list of considered characteristics can be found in Tab. 2. We ran an empirical assessment of NK bandits using the contextual bandit benchmark proposed by

Riquelme et al. (2018). The obtained results show that NKs provide the biggest advantage in bandit problems derived from balanced classification tasks, with high non-linear feature entanglement.

Measuring empirical performance of bandits may require a more detailed analysis than rendered in standard frameworks, in which the task dimensions are usually entangled. Practitioners choosing the key components of bandits (e.g. forms of predictive distributions, policies, or hyperparameters), looking for the best performance on a single metric, may be missing a more nuanced view along the key dimensions of representation learning and exploration. Moreover, interactions between a policy and a predictive distribution may differ depending on a particular view. In order to provide a detailed performance assessment we need to test these two aspects separately. For most real-world datasets, this is not feasible as we do not have access to the data generating distribution. In order to gain insight into the capability to explore, Riquelme et al. (2018) created the "wheel dataset", which lets us measure performance under sparse reward conditions. We propose to expand the wheel dataset by a parameter that controls the representational complexity of the task, and use it to perform an ablation study on a set of NK predictive distributions (He et al., 2020; Lee et al., 2020) with stochastic policies. We remark that our approach provides a general framework for evaluating sequential decision algorithms.

Our contributions can be summarized as follows:

- We utilize recent results on equivalence between NKs and deep neural networks in the bandit setting.
- We show that NK bandits achieve state-of-the-art performance on complex structured data.
- We propose an empirical framework that decouples evaluation of representation learning and exploration in sequential decision processes.

Within this framework:

- We evaluate the most common NK predictive distributions and bandit policies and assess their impact on the key aspects of bandits — exploration and exploitation.
- We propose to analyze the efficacy of a Student's t-process (TP), as proposed by Shah et al. (2013); Tracey & Wolpert (2018), in improving exploration in NK bandits.

We make our work fully reproducible by providing the implementation of our algorithm and the experimental benchmark. [1] The explanation of the mathematical notation can be found in Sec. A.

2 Background

In this section we provide a necessary background on NK bandits from a perspective of Bayesian inference. Our approach focuses heavily on the posterior predictive derivation and the Bayesian interpretation of predictive distributions obtained as ensembles of inite-width deep neural networks. We build on material related to neural bandits, NKs, and the posterior predictive. We also point out the unique way in which the methods are combined and utilized in our approach.

2.1 Contextual bandit policies

Contextual multi-armed bandits are probabilistic models that at each round $t \in [T]$ of a sequential decision process receive a set of k arms and a global context $\mathbf{x}_t$. The role of a *policy* π is to select an action $a \in [k]$ and observe the associated reward $y_{t,a}$. We name the triplet $(\mathbf{x}_t, a_t, y_{t,a_t})$ an *observation* at time t. Observations are stored in a dataset, which can be joint ($\mathcal{D}$) or separate for each arm ($\mathcal{D}_a$). The objective is to minimize the cumulative regret $R_T = \mathbb{E}\big[\sum_{t=1}^{T} \max_a y_{t,a} - \sum_{t=1}^{T} y_{t,a_t}\big]$ or, alternatively, to maximize the cumulative reward $\sum_{t=1}^{T} y_{t,a_t}$.

Thompson sampling (TS) is a policy that operates on the Bayesian principle. The reward estimates for each arm are computed in terms of a posterior predictive distribution $p(y_*|a, D_a, \boldsymbol{\theta}) = \int p(y_*|\boldsymbol{\theta})p(\boldsymbol{\theta}|D_a)d\boldsymbol{\theta}$. Linear, kernel, and neural TS (Agrawal & Goyal, 2013; Chowdhury & Gopalan, 2017; Riquelme et al., 2018) are all contextual variants of TS, which commonly model

[1] Code link temporarily hidden for blind review.

Predictive GP	μ	σ^2
NNGP	$\boldsymbol{\mathcal{K}}_{\boldsymbol{x}_*\boldsymbol{X}}(\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{X}}+\gamma\mathbf{I})^{-1}\mathbf{y}$	$\boldsymbol{\mathcal{K}}_{\boldsymbol{x}_*\boldsymbol{x}_*}-\boldsymbol{\mathcal{K}}_{\boldsymbol{x}_*\boldsymbol{X}}(\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{X}}+\gamma\mathbf{I})^{-1}\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{x}_*}$
Deep Ensembles	$\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}^{-1}\mathbf{y}$	$\boldsymbol{\mathcal{K}}_{\boldsymbol{x}_*\boldsymbol{x}_*}+\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}^{-1}\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{X}}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}^{-1}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{x}_*}-2\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}^{-1}\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{x}_*}$
Randomized Prior	$\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}(\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}+\gamma\mathbf{I})^{-1}\mathbf{y}$	$\boldsymbol{\mathcal{K}}_{\boldsymbol{x}_*\boldsymbol{x}_*}+\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}(\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}+\gamma\mathbf{I})^{-1}\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}+\gamma\mathbf{I})^{-1}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{x}_*}$ $-2\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}(\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}+\gamma\mathbf{I})^{-1}\boldsymbol{\mathcal{K}}_{\boldsymbol{X}\boldsymbol{x}_*}$
NTKGP	$\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}(\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}+\gamma\mathbf{I})^{-1}\mathbf{y}$	$\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{x}_*}-\boldsymbol{\Theta}_{\mathbf{x}_*\mathbf{X}}(\boldsymbol{\Theta}_{\mathbf{X}\mathbf{X}}+\gamma\mathbf{I})^{-1}\boldsymbol{\Theta}_{\mathbf{X}\mathbf{x}_*}$

Table 1: NK predictive distributions. Table reproduced from (He et al., 2020).

the rewards in terms of GP feature regression $y = \boldsymbol{\Phi}\boldsymbol{\theta} + \epsilon$, where $\boldsymbol{\Phi} = \phi(\mathbf{X})$ and $\epsilon \sim \mathcal{N}(0, \sigma_y)$. The analytical solution to the model is given by: $\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \gamma\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{y}$, where γ is a regularizer, and a covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$. Together $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}}$ parametrize the marginal predictive distribution used to obtain reward estimates $p_{a,t}$.

If the number of parameters p is larger than the number of samples n, it is advantageous to apply the kernel trick to reduce the amount of computation, that is to use the identity $\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \gamma\mathbf{I}\right)^{-1}\boldsymbol{\Phi}^T = \boldsymbol{\Phi}^T\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \gamma\mathbf{I}\right)^{-1}$ to convert the $p \times p$ covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} = \phi(\mathbf{X})^T\phi(\mathbf{X})$ into a smaller $n \times n$ kernel matrix $\mathbf{K}_{\mathbf{X}\mathbf{X}} = \phi(\mathbf{X})\phi(\mathbf{X})^T$. This results in a posterior predictive distribution given by:

$$p(y_*|\mathbf{x}_*, D) = \mathcal{N}\big(\mathbf{K}_{\mathbf{x}_*\mathbf{X}}(\mathbf{K}_{\mathbf{X}\mathbf{X}} + \gamma\mathbf{I})^{-1}\mathbf{y}, \quad \sigma_\theta^2(\mathbf{K}_{\mathbf{x}_*\mathbf{x}_*} - \mathbf{K}_{\mathbf{x}_*\mathbf{X}}(\mathbf{K}_{\mathbf{X}\mathbf{X}} + \gamma\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}\mathbf{x}_*}\big). \quad (1)$$

This distribution was originally applied to bandits by Srinivas et al. (2010); Valko et al. (2013).

A contextual bandit can be either joint or disjoint. A joint model (e.g. Chu et al. (2011)) uses a single kernel and makes predictions based on action-specific contexts, while a disjoint model uses a separate kernel per action, which is more suited for a global context scenario.

While TS offers an explicit Bayesian interpretation, other stochastic policies, like upper confidence bound (UCB) (Auer et al., 2002), can take advantage of the same inference process, and use the moments of the predictive distribution to compute the reward estimates.

Neural bandits. Neural bandits are a special case of nonlinear contextual bandits, in which the function $f(x)$ is represented by a neural network. The neural-greedy (e.g. Allesiardo et al. (2014)) approach uses predictions of a NN directly to choose the next action. It is considered a naive approach, tantamount to exploitation in feature space. Neural-linear, given by Riquelme et al. (2018), utilize probabilistic regression with $\phi(x)$ representing its penultimate layer features. Nabati et al. (2021) proposed a limited memory version of neural-linear (LiM2), which uses likelihood matching to propagate information about the predictive parameters. This approach is in contrast to storing the data and re-training the network at each iteration in order to prevent catastrophic forgetting.

2.2 Neural kernel distributions

We define a neural kernel (NK) to be any compositional kernel (Cho & Saul, 2009; Daniely et al., 2017) mimics a state or behavior of a neural network. We consider two recently proposed kernels - neural network Gaussian process kernel (NNGP) (Lee et al., 2018), $\boldsymbol{\mathcal{K}}$, and neural tangent kernel (NTK) (Jacot et al., 2020), $\boldsymbol{\Theta}$. Both were shown to model the behavior of an infinitely wide network, and allow for training by means of Bayesian inference. He et al. (2020) distinguished several NK predictive distributions, corresponding to common approaches in Bayesian deep learning (Tab. 1).

The NNGP kernel models the network's covariance at random initialization. It can be composed recursively, such that the kernel of each subsequent layer $\boldsymbol{\mathcal{K}}^{(l)}$ denotes the expected covariance between the outputs of the current layer, for inputs sampled from a GP induced by a kernel of the previous layer $\boldsymbol{\mathcal{K}}^{(l-1)}$. The output of the last layer $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{K}}^{(L)}$ induces a prior GP of the Bayesian regression model and the NNGP posterior predictive can be obtained by inference (Tab. 1).

Jacot et al. (2020) showed that dynamics of the gradient descent optimization, when examined from functional perspective, can be captured by another kernel, NTK, with a general form $\nabla_{\boldsymbol{\theta}} f(\mathbf{X})\nabla_{\boldsymbol{\theta}} f(\mathbf{X})^T$, and with the corresponding feature space covariance $\mathbf{Z} = \nabla_{\boldsymbol{\theta}} f(\mathbf{X})^T\nabla_{\boldsymbol{\theta}} f(\mathbf{X})$, referred to as the neural tangent feature matrix (NTF). In practice we compute NTK recursively

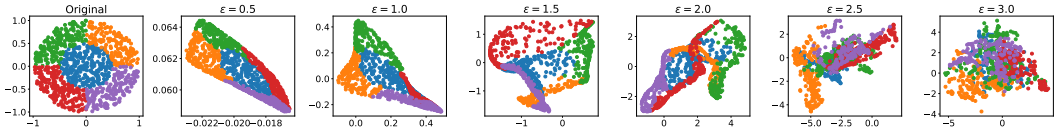

Figure 1: Noisy wheel dataset. The coordinates represent input features, and the colors represent classes. ε regulates the complexity of the problem linearly w.r.t. the common baselines (Sec. B.3).

$\mathbf{\Theta} = \sum_{l=1}^{L+1} \left(\boldsymbol{\mathcal{K}}^{(l-1)} \prod_{l'=l}^{L+1} \dot{\boldsymbol{\mathcal{K}}}^{(l')} \right)$, with dot denoting a gradient. For networks with ReLU activations both NNGP and NTK become variants of the arc-cosine kernel (Cho & Saul, 2009).

In the infinite width limit the parameters change so little during the training ($O(1/\sqrt{p})$, where p is the number of parameters) that the network can be approximated with a first order Taylor expansion about the initial parameters (Lee et al., 2020). That approximation is referred to as a "linearized network". By putting a Gaussian prior of $\mathcal{N}(\mathbf{0}; \boldsymbol{\mathcal{K}})$ on the functional form of the neural network, the linearized NTK model results in a GP predictive distribution with moments outlined in Tab. 1 - Deep Ensembles. This formulation is given by Lee et al. (2020) (eq 16), who point out, however, that it does not admit the interpretation of a Bayesian *posterior*, yet can still be used as a predictive distribution. When regularization term is added to the GP, He et al. (2020) showed that this approach also corresponds to networks with randomized priors (Osband et al., 2018) (Tab. 1). He et al. (2020) further attempted to remedy the lack of admittance as a posterior by adding a missing variance of the parameters of all layers except the last one, to the linearized network formulation, which resulted in *NTKGP* distribution (Tab. 1).

3 METHODS

Our extensions are designed to provide insights to the behavior of the NK bandits along the axes of Bayesian inference and exploration-exploitation trade-off. We begin this section by a proposal to expand the wheel dataset in a way that lets us decouple representation learning from exploration, and provides grounds for analysis of differences among the predictive posteriors. We then discuss details of the main algorithm, with the emphasis on the implementation considerations and the choice of posterior predictive. We close by discussing the conjugate prior approach to the posterior derivation.

3.1 WHEEL DATASET AND ITS EXTENSION

The wheel dataset was originally proposed by Riquelme et al. (2018), and can be seen on the far left side of Fig. 1. The parameter δ controls the proportion of the inner circle to the peripheral classes, which are given the highest rewards. By increasing δ we ask the algorithm to explore more in order to find those classes. Our reward distribution follows directly from Riquelme et al. (2018) and is detailed in Sec. B.1. Datasets similar to wheel were proposed in the past to measure the capabilities of machine learning models to learn representations (Li & Abu-Mostafa, 2006).

We propose to extend the wheel dataset by an orthogonal parameter that regulates the representation learning difficulty. Following a body of work on measuring the complexity of machine learning problems (Li & Abu-Mostafa, 2006; Lorena et al., 2019), we define the representation learning difficulty as a complexity of an algorithm needed to achieve a predefined level of accuracy. In order to generate the data, we reverse that definition, and morph the original wheel dataset such that it predictably decays the performance of an algorithm of fixed complexity. In particular, we propagate the wheel dataset through a fully-connected MLP, initialized with random weights, $w_{ij}^{(l)} \sim \mathcal{N}(0, \frac{\varepsilon}{\sqrt{d^{(l)}}})$, where $d^{(l)}$ is the number of input dimensions to a particular layer, and ε is a scalar. In the experiments we use ReLU activation functions and $L = 5$ fully-connected layers. The newly introduced "noise parameter" ε causes the dataset to be warped nonlinearly and injects a moderate level of noise around samples. The effect of increasing ε can be seen from left to right in Fig. 1. By varying δ and ε we create tasks of varying exploration and representation learning difficulty. More details on how the morphing influences complexity can be found in Sec. B.3 and on Fig. 6.

3.2 NEURAL KERNEL BANDITS

Algorithm 1 Neural kernel bandit

Require: Number of arms k, number of rounds T, kernel regularization parameter γ, exploration parameter η, NNGP kernel function $\mathcal{K}(\cdot,\cdot)$, neural tangent kernel function $\Theta(\cdot,\cdot)$, initial number of steps ι, policy π

Play each arm sequentially for ι steps to accumulate data $\mathbf{X}_a \in \mathbb{R}^{\iota \times d}, \mathbf{y}_a \in \mathbb{R}^{\iota} \ \forall a$
for round $t = 1, 2, ..., T$ **do**
  Observe context $\mathbf{x}_t$
  for arm $a = 1, 2, ..., k$ **do**
    $\boldsymbol{\mathcal{K}}_{a,t} \leftarrow \mathcal{K}(\mathbf{X}_a, \mathbf{X}_a)$
    $\boldsymbol{\Theta}_{a,t} \leftarrow \Theta(\mathbf{X}_a, \mathbf{X}_a) + \gamma \mathbf{I}$
    // Calculate predictive distribution moments
    $\mu_{a,t} \leftarrow \boldsymbol{\Theta}_{\mathbf{x_t}\mathbf{X_a}} \boldsymbol{\Theta}_{a,t}^{-1} \mathbf{y}_a$
    $\sigma^2_{a,t} \leftarrow \boldsymbol{\mathcal{K}}_{\mathbf{x_t}\mathbf{x_t}} + \boldsymbol{\Theta}_{\mathbf{x_t}\mathbf{X_a}} \boldsymbol{\Theta}_{a,t}^{-1} \boldsymbol{\mathcal{K}}_{a,t} \boldsymbol{\Theta}_{a,t}^{-1} \boldsymbol{\Theta}_{\mathbf{X_a}\mathbf{x_t}} - 2\boldsymbol{\Theta}_{\mathbf{x_t}\mathbf{X_a}} \boldsymbol{\Theta}_{a,t}^{-1} \boldsymbol{\mathcal{K}}_{\mathbf{X_a}\mathbf{x_t}}$
    if π is UCB **then**
      $p_{a,t} \leftarrow \mu_{a,t} + \frac{\eta}{\gamma^{1/2}} \sigma_{a,t}$
    else if π is TS **then**
      $p_{a,t} \sim \mathcal{N}(\mu_{a,t}, \frac{\eta}{\gamma^{1/2}} \sigma_{a,t})$
    end if
  end for
  Choose $a_t \leftarrow \arg\max_a p_{a,t}$ and obtain reward y_t
  Update ($\mathbf{X}_{a_t}$,$\mathbf{y}_{a_t}$) with $(\mathbf{x}_t, y_t)$
end for

Our algorithm (Alg. 1) is built using a similar structure to classic kernel and GP bandits (Chowdhury & Gopalan, 2017; Srinivas et al., 2010; Valko et al., 2013). We compute GP moments $\mu_{a,t}$ and $\sigma_{a,t}$ according to Tab. 1. As an example, in Alg. 1 we use the randomized prior approach. We provide both UCB and TS variants of our algorithm. In each round an arm is chosen based on the highest estimate of the reward $p_{a,t}$. We use the disjoint approach as our primary model, which means that the data $(\mathbf{X}_a, \mathbf{y}_a)$ and kernels (NNGP $\boldsymbol{\mathcal{K}}_{a,t}$ and NTK $\boldsymbol{\Theta}_{a,t}$) are collected and computed separately for each arm a. The disjoint strategy preserves memory, while ensuring that better performing arms, chosen more frequently, have larger associated datasets and thus more certainty over time. This approach is very much in line with classic optimism in face of uncertainty (OFU) methods. A common alternative is the joint model, which, in the case of GP, uses a single kernel and produces separate posteriors, albeit with a common uncertainty estimate, for each arm at prediction time. Using NKs in this manner for multi-class problems is common in the NNGP/NTK literature (e.g. Arora et al. (2019b); Novak et al. (2019)). However, the approach is not feasible in the bandit setting as it assumes access to rewards associated with all the arms. A common solution (e.g. Zhou et al. (2020)) is to use zero-padding to create a separate context vector for each arm $\mathbf{x}_a = [\mathbf{0}, \ldots, \mathbf{0}, \mathbf{x}_{orig}, \mathbf{0}, \ldots, \mathbf{0}]$. In the feature space this has an effect of splitting the model's parameters per arm yielding separate uncertainty estimates. For further details and background please refer to Sec. 3.

3.3 STUDENT'S T-PROCESS

While our GP predictive distributions represent the posterior predictives with the fixed observation noise σ_y, we also analyze the behavior of the algorithm in conjunction with the joint prior. Linear and neural-linear models were shown by Nabati et al. (2021) and Riquelme et al. (2018) to benefit from applying a joint conjugate prior to both parameters of the likelihood, resulting in a Normal-inverse-Gamma distribution $p(\boldsymbol{\theta}, \sigma_y^2) = p(\boldsymbol{\theta}|\sigma_y^2)p(\sigma_y^2)$. When σ_y is not fixed, it continues to evolve with the data, which results in more accurate uncertainty estimates over time. The joint prior inference can be performed either by (1) a two-stage sampling process, where parameters are sampled individually, and then used in the regression model, or (2) by a full predictive distribution derivation. We use the first approach in linear and neural-linear bandits. When using kernels, the parameters cannot be sampled, and require a full derivation, which results in a Student's t process (TP), Eq. 2. TP is related to GP by having the same mean, but a scaled covariance, by a factor of

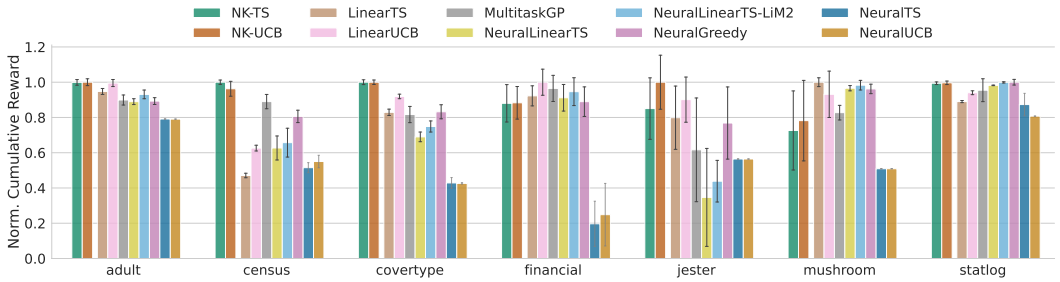

Figure 2: Final normalized cumulative rewards (y-axis) of NK bandits compared against all other methods after 5,000 steps rollout on the respective UCI datasets (x-axis). Each bar shows the mean and standard deviation over 10 rollouts.

$\frac{\nu}{\nu-2}$. ν is known as "degrees of freedom", and controls the kurtosis. The lower ν values result in heavier tails, and thus lead to more aggressive exploration. TP approaches GP as $\nu \to \infty$, and can therefore be interpreted as a generalization of GP (Tracey & Wolpert, 2018). Correspondence to GP allows us to use the same NK-based distribution parameters in an unchanged form. To introduce finer control over performance, we make use of a regularizer γ, although we note that it does not have the same interpretation as in GP. The key advantage of TP over GP is that its variance depends on both input and output data, rather than just inputs. The output variance is introduced through the $\mathbf{y}^T(\mathbf{K_{XX}} - \gamma\mathbf{I})^{-1}\mathbf{y}$ term, which represents the data complexity. This term will drift from n the more the outputs deviate from the GP modelling assumption (Tracey & Wolpert, 2018).

$$p(y_*|\mathbf{x}_*, D) = \mathcal{T}\left(\mathbf{K}_{\mathbf{x}_*\mathbf{X}}(\mathbf{K_{XX}}-\gamma\mathbf{I})^{-1}\mathbf{y}, \quad \frac{\nu + \mathbf{y}^T(\mathbf{K_{XX}} - \gamma\mathbf{I})^{-1}\mathbf{y} - 2}{\nu + n - 2}\widehat{\mathbf{K}}_{\mathcal{T}}, \quad \nu + n\right),$$
$$\widehat{\mathbf{K}}_{\mathcal{T}} = \frac{\nu - 2}{\nu}\left(\mathbf{K}_{\mathbf{x}_*\mathbf{x}_*} - \mathbf{K}_{\mathbf{x}_*\mathbf{X}}(\mathbf{K_{XX}} - \gamma\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}\mathbf{x}_*}\right). \tag{2}$$

4 Experiments

While NK bandits were proposed in the past, the work presented by Zhou et al. (2020); Zhang et al. (2020) focuses mainly on theoretical analysis and offers only a limited empirical evaluation of an approach that can be considered a feature space variant of NTKGP. We conduct experiments to show that NKs provide a competitive approach in bandit settings. We begin by evaluating our method on a Bayesian bandit benchmark (Riquelme et al., 2018) and show that it achieves on-par or better performance than state-of-the-art baselines on most of the tested UCI datasets. We conclude that the method should be considered by the community as one of the standard approaches for bandits with nonlinear reward models. Next, we utilize the noisy wheel dataset to assess the impact of utilizing NK predictive distributions to accurately trade off exploration and exploitation. While no approach can adapt to all settings, we observe that NTKGP can provide significant improvement in sparse environments requiring large exploration. In addition to the main results, we assess the ability of the noisy wheel dataset to model complexity, and measure the implications of practical implementation considerations — training frequency and model partitioning (Sec. B.4, Sec. B.3, and Sec. B.5).

4.1 NK bandits outperform neural-linear and NTF bandits on complex data

We tested the performance of NK bandits on a set of UCI datasets (Dua & Graff, 2019) used in the Bayesian bandit benchmark developed by Riquelme et al. (2018). We compared our method against the state-of-the-art neural-linear (Riquelme et al., 2018) and neural-linear LiM2 (Nabati et al., 2021) bandits. We followed the experimental setup of (Riquelme et al., 2018; Nabati et al., 2021) and report the mean and standard deviation of the cumulative reward over 10 runs. We included important baselines, linear TS / UCB and multitask GP, due to their significant performance on simpler datasets. According to a common practice in the field (Riquelme et al., 2018; Nabati et al., 2021), we did not tune the hyperparameters of any algorithm. Unlike in supervised learning, bandit benchmarks typically do not provide a separate test set. Instead, we pre-set the hyper-parameters to established values. For consistency, we used the hyperparameters reported by Riquelme et al. (2018), whenever

applicable. Due to differences in the NN architecture between Riquelme et al. (2018) and Nabati et al. (2021), in Tab. 3 we report the results for both settings. We used the neural-tangents library (Novak et al., 2019) to obtain the NTK, NNGP kernel, and the mean and covariance of the predictive distributions. We computed the NKs based on a two-layer fully-connected architecture with ReLU activations and regularizer $\gamma = 0.2$. We chose the the randomized prior GP (He et al., 2020; Lee et al., 2020; Osband et al., 2018) as our preliminary experiments showed it to perform best on the majority of the selected UCI datasets (this is not shown in the paper). In light of findings presented in Sec. 4.2 and the assessment done by Riquelme et al. (2018) we suspect that deep ensemble and randomized prior distributions are more suitable for datasets like UCI, that require fewer exploration steps. We tested both UCB and TS policies with the exploration parameter η set to 0.1. As established by Nabati et al. (2021), normalized cumulative rewards are computed w.r.t. uniform sampling and the best algorithm for each dataset: $\text{norm_cum_rew}_{\text{alg}} = \frac{\text{cum_rew}_{\text{alg}} - \text{cum_rew}_{\text{uniform}}}{\text{cum_rew}_{\text{best}} - \text{cum_rew}_{\text{uniform}}}$.

The results (Fig. 2) show that NK bandits constitute a competitive approach, achieving state-of-the-art results on most datasets. The improvement is most visible on datasets with high non-linear entanglement and large number of features (> 50). Interestingly, we get the highest performance boost on covertype dataset, which is typically considered the hardest. We attribute the increase to the high accuracy of exploitation, achieved by application of NTK to nonlinear structured data problems (Arora et al., 2019b), and exploration, achieved through the appropriate choice of the predicitve distribution. For linear datasets, linear methods tend to outperform non-linear ones, including ours. The same trend was noted by Riquelme et al. (2018) w.r.t. previous methods.

We recognize that low scalability is a limitation of our method. Computing NTK requires large computational resources. In the appendix B we show two avenues for scaling up our approach, with relatively low impact on performance. Disjoint arm treatment (Sec. B.5) provides opportunity for distributed computing, while lowering the training frequency (Sec. B.4) allows for large computational gain with a low ratio of performance drop. We also note that current research directions provide promising avenues for significant scaling of our approach in the near future (e.g. Zandieh et al. (2021)). We expand further on possible future expansions in Sec. 6.

4.2 Decoupling representation learning from uncertainty estimation

Decoupling representation learning from uncertainty estimation allows us to study each factor separately and understand their impact on the overall performance of bandits. In the following experiment we utilize the noisy wheel dataset to measure the performance of TS and UCB policies combined with NK-induced TPs and GPs (Tab. 1), w.r.t. the varying factors of complexity ε and the need to explore δ. We compared all methods by measuring their *peripheral accuracy*, i.e. the percentage of correctly chosen actions on the periphery of the wheel: $pacc = \sum_i I[a_i^* \neq a^0] I[a_i = a_i^*] / \sum_i [a_i^* \neq a^0]$. We evaluated each algorithm on a grid of (ε, δ) pairs, where ε's were spaced linearly, due to their linear impact on performance (see Sec. B.3), and δ's were spaced according to Riquelme et al. (2018). The grid was computed separately for each {distribution,policy} pair. Each method was run for 5,000 steps and the results were averaged over 10 runs. Due to the large number of runs, we updated the model each 20 steps to speed up the computation (see Sec. B.4). All the experiments were performed using a disjoint model with exploration parameter $\eta = 0.1$. Apart from the "deep ensemble" distribution, all others were regularized with $\gamma = 0.2$. The Student's t-process was initialized with parameter $\nu = 12$. The results can be seen in Fig. 3.

While there is no single method that achieves the highest score in all circumstances, in average the NTKGP distribution provides the best performance w.r.t. the exploration-exploitation trade-off. All predictive distributions and policies exhibit good performance close to the original wheel problem with $\delta = 0.5$ (Fig. 3; top left corner of each grid), and gradually worsen as problems become more complicated on both axes. The algorithms eventually fail when $\delta = 0.99$. We argue that a good bandit method should be able to adapt and sustain its performance longer as the problem becomes more complex and requires more exploration. In Fig. 3 we can see that NTKGP clearly dominates on harder problems, especially in combination with Thompson sampling. This is indicated by almost uniformly higher scores (3rd column, 1st and 2nd row) and can be further verified by observing the histograms (rhs). The result is in contrast with our finding that the randomized prior distribution performed best on the UCI datasets. We believe that the difference is due to the low need of exploration in the UCI tasks (Riquelme et al., 2018). Fig. 4 and Fig. 3 (top histograms) further confirm that intuition by showing little distinction in performance when the need for exploration is small.

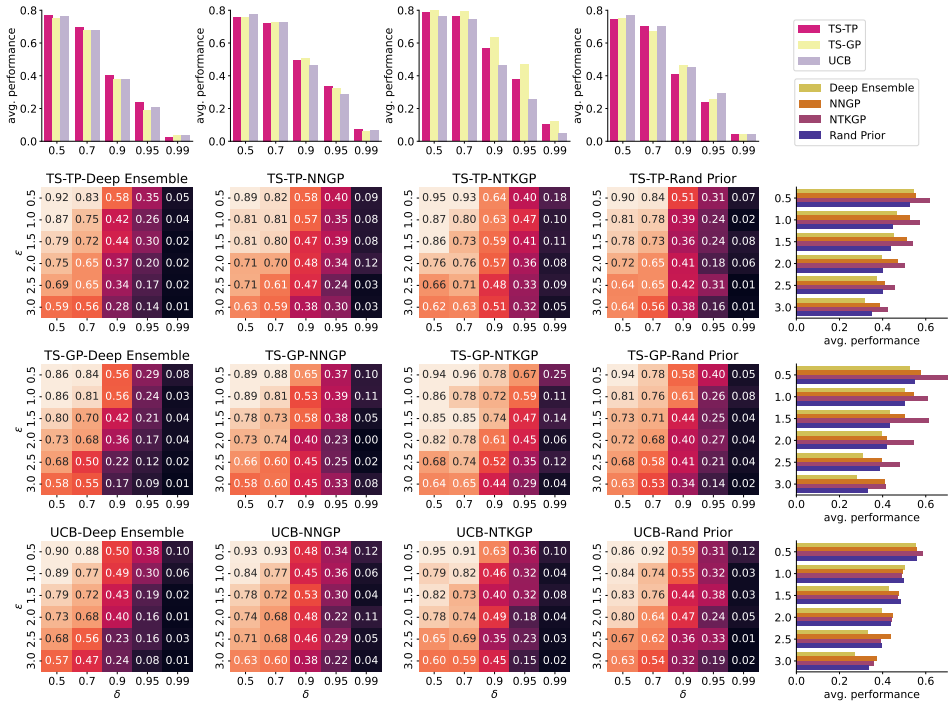

Figure 3: Exploration and exploitation of NK predictive distribution (columns) in combination with bandit policies (rows). Grid cells represent peripheral accuracies. Histograms represent mean results (by method) over columns or rows respectively.

NTKGP was shown to provide more accurate out-of-distribution (OOD) and occasionally worse in-distribution predictions (He et al., 2020). In bandits, the improved OOD performance is crucial, due to its effect on initial exploration. After the uncertainty converges, it plays significantly lesser role in overall performance. NTKGP was also shown by He et al. (2020) to make more "conservative" decisions on samples it is uncertain about. In the ensemble settings, studied by He et al. (2020), this means that the algorithm utilizes uncertainty as its measure of confidence, and tends to take more confident, less extreme actions. In bandits we use the larger uncertainty to produce an opposite effect, and decide to sample the uncertain regions more frequently, which facilitates exploration.

NNGP often performs better than deep ensemble and randomized prior distributions, but worse that NTKGP. It is interesting to see that a method based purely on NNGP kernel outperforms more sophisticated distributions that combine both kernels. Deep ensemble and randomized prior approaches are also well founded in representing NNs after training (Lee et al., 2018), while NNGP performs Bayesian feature regression on a randomly initialized network without training. NTKGP and NNGP are fully Bayesian approaches, while deep ensemble and randomized priors are not. The results suggest that the Bayesian interpretation may be advantageous in uncertainty estimation.

The fact that the deep ensemble distribution performs much worse than other methods on average, while not being the simplest to compute (NNGP is), shows a large impact of the regularizer on the overall performance. Interestingly, also the disparity in performance between deep ensemble and all other methods grows significantly as the data complexity increases. This disparity is less noticeable when changing the exploration parameter delta (compare with Fig. 5).

Fig. 3 shows that UCB has less variance in performance between the distributions. We attribute the lower variance to the fewer number exploration steps the UCB performs over the rollout. Fewer exploration steps mean less reliance on uncertainty estimation, and thus smaller distinction between the tested predictive distributions. While we tried a limited sensitivity analysis for setting the η parameter to bring TS and UCB to similar performance, the results were inconclusive. We set $\eta = 0.1$ for both methods and leave a more extensive assessment for future work.

5 Related Work

The work of (Chu et al., 2011; Li et al., 2010) on linear UCB and (Agrawal & Goyal, 2013) on linear TS motivated an abundance of research on regression bandits. GP-UCB (Srinivas et al., 2010) and GP-TS (Chowdhury & Gopalan, 2017) are special cases of kernel bandits (Valko et al., 2013), in which the ridge regularizer is set to Gaussian noise to form a GP. In practical applications bandits are traditionally combined with well-known kernels, like linear, RBF or Matérn (Rasmussen & Williams, 2006). More recently a new family of neural kernels (NK) has emerged, which directly approximates the behavior of deep neural networks (NN) (Cho & Saul, 2009; Daniely et al., 2017). The neural network Gaussian process (NNGP) kernel is derived from a forward pass of a NN, and corresponds to a NN with randomly initialized weights (Lee et al., 2018). The finding of a direct GP correspondence was later followed by establishing a similar relation for linearized NNs and the neural tangent kernel (NTK), where NTK represents a network's dynamics during gradient descent training (Arora et al., 2019a; He et al., 2020; Jacot et al., 2020; Lee et al., 2020). A review of all recent NK-GP models, with connections to other existing uncertainty models, like deep ensembles and randomized priors (Osband et al., 2018), can be found in (He et al., 2020).

NeuralUCB (Zhou et al., 2020) and NeuralTS (Zhang et al., 2020) are two related neural bandit policies that can be considered feature space variants of NK bandits. They have an NTKGP posterior (He et al., 2020), whose moments are defined by a standard NN estimate and the neural tangent features (NTF) covariance (Sec. 2.2). These algorithms were shown to achieve a regret bound comparable to kernel bandits with the effective dimension induced by a corresponding NTK (Zhou et al., 2020; Zhang et al., 2020). Yet, they were shown to underperform in practical applications (Zhou et al., 2020; Zhang et al., 2020; Nabati et al., 2021), which was linked to overparametrization and poor covariance approximation (Xu et al., 2020).

The extended Kalman filter (EKF) provides another fully Bayesian treatment of all the model's parameters that was applied to neural bandits (Duran-Martin et al., 2021). The EKF approach was shown to be particularly suitable to online learning, as it operates on constant memory. In practice, however, the approach relies on dimensionality reduction to approximate parameters of a linearized network. The NK approach (NTK and NNGP), on the other hand, applies the law of large numbers and central limit theorem to construct an exact predictive distribution recursively from an infinite network, which is also linearized, but the method does not rely on further approximation. In this work we are interested in the NK approach, with the goal of testing the performance of neural bandits in the presence of exact uncertainty estimates.

6 Conclusions

In this paper we evaluated NK bandits on a popular benchmark that measures performance on real-world (UCI) and synthetic (wheel) datasets. Specifically, we proposed to guide the bandit policy by Student's t or Gaussian processes corresponding to common classes of infinite neural networks. We showed that our model outperforms the state-of-the-art methods on the majority of tasks. We also note that it performs best when the number of features is small, or when the reward is highly non-linear. In addition, we expanded the wheel dataset, originally proposed by Riquelme et al. (2018), to measure both ability to explore and to learn complex representations. We used our proposed framework to evaluate the impact of NK predictive distributions on the two key aspects of bandits, exploration and exploitation, and showed that while there is no single distribution applicable to all scenarios, NTKGP in combination with Thompson sampling, is most robust to changes in the complexity and the need for exploration.

We propose several direction to continue the study on practical aspects of neural kernal bandits:

- Utilizing kernel approximation methods to scale computations up to larger datasets, and bring computation times further down, e.g. Zandieh et al. (2021).
- Investigating further the difference between TP and GP in handling the exploration and exploitation trade-off under varied conditions.
- Benchmarking other fully Bayesian neural bandits, e.g. Duran-Martin et al. (2021).
- Comparing the exploration and exploitation balance for other methods from Riquelme et al. (2018)'s framework, especially neural-linear that was said to benefit from decoupling.

REFERENCES

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135. PMLR, 2013.

Robin Allesiardo, Raphael Feraud, and Djallel Bouneffouf. A Neural Networks Committee for the Contextual Bandit Problem. *arXiv:1409.8191 [cs]*, September 2014. URL http://arxiv.org/abs/1409.8191. arXiv: 1409.8191.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On Exact Computation with an Infinitely Wide Neural Net. *arXiv:1904.11955 [cs, stat]*, November 2019a. URL http://arxiv.org/abs/1904.11955. arXiv: 1904.11955.

Sanjeev Arora, Simon S. Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, and Dingli Yu. Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks. *arXiv:1910.01663 [cs, stat]*, October 2019b. URL http://arxiv.org/abs/1910.01663. arXiv: 1910.01663.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2):235–256, May 2002. ISSN 1573-0565. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Youngmin Cho and Lawrence Saul. Kernel Methods for Deep Learning. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://papers.nips.cc/paper/2009/hash/5751ec3e9a4feab575962e78e006250d-Abstract.html.

Sayak Ray Chowdhury and Aditya Gopalan. On Kernelized Multi-armed Bandits. *arXiv:1704.00445 [cs]*, May 2017. URL http://arxiv.org/abs/1704.00445. arXiv: 1704.00445.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214. JMLR Workshop and Conference Proceedings, 2011.

Mark Collier and Hector Urdiales Llorens. Deep Contextual Multi-armed Bandits. *arXiv:1807.09809 [cs, stat]*, July 2018. URL http://arxiv.org/abs/1807.09809. arXiv: 1807.09809.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity. *arXiv:1602.05897 [cs, stat]*, May 2017. URL http://arxiv.org/abs/1602.05897. arXiv: 1602.05897.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2019. URL http://archive.ics.uci.edu/ml.

Gerardo Duran-Martin, Aleyna Kara, and Kevin Murphy. Efficient Online Bayesian Inference for Neural Bandits. *arXiv:2112.00195 [cs]*, November 2021. URL http://arxiv.org/abs/2112.00195. arXiv: 2112.00195.

Imène R. Goumiri, Benjamin W. Priest, and Michael D. Schneider. Reinforcement Learning via Gaussian Processes with Neural Network Dual Kernels. *arXiv:2004.05198 [cs, eess, stat]*, April 2020. URL http://arxiv.org/abs/2004.05198. arXiv: 2004.05198.

Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian Deep Ensembles via the Neural Tangent Kernel. *arXiv:2007.05864 [cs, stat]*, October 2020. URL http://arxiv.org/abs/2007.05864. arXiv: 2007.05864.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv:1806.07572 [cs, math, stat]*, February 2020. URL http://arxiv.org/abs/1806.07572. arXiv: 1806.07572.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. URL https://tor-lattimore.com/downloads/book/.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. *arXiv:1711.00165 [cs, stat]*, March 2018. URL http://arxiv.org/abs/1711.00165. arXiv: 1711.00165.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12): 124002, December 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc62b. URL http://arxiv.org/abs/1902.06720. arXiv: 1902.06720.

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 - WWW '10*, pp. 661, 2010. doi: 10.1145/1772690.1772758. URL http://arxiv.org/abs/1003.0146. arXiv: 1003.0146.

Ling Li and Yaser S. Abu-Mostafa. Data complexity in machine learning. 2006. Publisher: California Institute of Technology.

Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. How Complex Is Your Classification Problem? A Survey on Measuring Classification Complexity. *ACM Computing Surveys*, 52(5):107:1–107:34, September 2019. ISSN 0360-0300. doi: 10.1145/3347711. URL https://doi.org/10.1145/3347711.

Ofir Nabati, Tom Zahavy, and Shie Mannor. Online Limited Memory Neural-Linear Bandits with Likelihood Matching. *arXiv:2102.03799 [cs]*, June 2021. URL http://arxiv.org/abs/2102.03799. arXiv: 2102.03799.

Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. *arXiv preprint arXiv:1912.02803*, 2019.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *arXiv preprint arXiv:1806.03335*, 2018.

C. E. Rasmussen and C. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006. ISBN 0-262-18253-X. URL http://gaussianprocess.org/gpml/.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. *arXiv:1802.09127 [cs, stat]*, February 2018. URL http://arxiv.org/abs/1802.09127. arXiv: 1802.09127.

Amar Shah, Andrew G. Wilson, and Zoubin Ghahramani. Bayesian optimization using Student-t Processes. In *NIPS Workshop on Bayesian Optimization*, 2013.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *arXiv:0912.3995 [cs]*, June 2010. doi: 10.1109/TIT.2011.2182033. URL http://arxiv.org/abs/0912.3995. arXiv: 0912.3995.

William 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. Publisher: JSTOR.

Brendan D. Tracey and David H. Wolpert. Upgrading from Gaussian Processes to Student's-T Processes. *2018 AIAA Non-Deterministic Approaches Conference*, January 2018. doi: 10.2514/6.2018-1659. URL http://arxiv.org/abs/1801.06147. arXiv: 1801.06147.

Michal Valko, Nathaniel Korda, Remi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-Time Analysis of Kernelised Contextual Bandits. *arXiv:1309.6869 [cs, stat]*, September 2013. URL http://arxiv.org/abs/1309.6869. arXiv: 1309.6869.

Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. Neural Contextual Bandits with Deep Representation and Shallow Exploration. *arXiv:2012.01780 [cs, stat]*, December 2020. URL http://arxiv.org/abs/2012.01780. arXiv: 2012.01780.

Amir Zandieh, Insu Han, Haim Avron, Neta Shoham, Chaewon Kim, and Jinwoo Shin. Scaling neural tangent kernels via sketching and random features. *Advances in Neural Information Processing Systems*, 34:1062–1073, 2021.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Thompson Sampling. *arXiv:2010.00827 [cs, stat]*, October 2020. URL http://arxiv.org/abs/2010.00827. arXiv: 2010.00827.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Contextual Bandits with UCB-based Exploration. *arXiv:1911.04462 [cs, stat]*, July 2020. URL http://arxiv.org/abs/1911.04462. arXiv: 1911.04462.

A MATHEMATICAL NOTATION

We denote the training data by $(\mathbf{X}, \mathbf{y}) \in (\mathcal{X}, \mathcal{Y})$, and the test points by $(\mathbf{x}_*, y_*)$. The input data is composed of $(\mathbf{x}_i, y_i)$, where $i \in [n]$, when we talk about a predetermined set. Alternatively the data points can be indexed by time $t \in [T]$ in the context of bandits, where the data is collected sequentially. Parameter vectors are denoted by $\boldsymbol{\theta}$ and the reward estimation models by $f : \mathcal{X} \to \mathcal{Y}$. The dimensionality of both the feature vector and the parameter vector is denoted by p, i.e. $\boldsymbol{\theta} \in \mathbb{R}^p$ and $\phi(\mathbf{x}) \in \mathbb{R}^p$. In the linear case $\phi(\mathbf{x}) = \mathbf{x}$, so also $\mathbf{x} \in \mathbb{R}^p$. We denote a generic kernel matrix by $\mathbf{K}$ and the associated kernel function by $K(\cdot, \cdot)$, such that $\mathbf{K}_{\mathbf{xy}} \equiv K(\mathbf{x}, \mathbf{y})$. We use $\mathbf{x}_*$ for a general test input in the context of Bayesian inference, or $\mathbf{x}_t$ when the test input is given at a specific time t. We denote test predictions by y_* or y_t.

B EXPERIMENTAL SETUP

B.1 DATASETS

Table 2 provides an summary of the key properties associated with each of the UCI datasets we considered. While Riquelme et al. (2018) and Nabati et al. (2021) used the same datasets, there are subtle differences in the way that they were implemented. Most notably, Nabati et al. (2021) use the "adult" dataset to classify income bracket, while Riquelme et al. (2018) use it to classify occupation. In case of any discrepancies we followed Nabati et al. (2021)'s set up for UCI datasets. In the (noisy) wheel dataset we used the same reward distribution parameters as proposed by Riquelme et al. (2018). The reward for guessing the inner circle class (i.e. where $||x|| \leq \delta$) is $r_{small} \sim \mathcal{N}(1.2; 0.05)$. The reward for the correct peripheral class (sample classified correctly) is $r_{big} \sim \mathcal{N}(50; 0.01)$, while $r_{peripheral} \sim \mathcal{N}(1; 0.05)$.

B.2 UCI EXPERIMENTATION TIME ANALYSIS

We complement our results by reporting the average (over 10 runs) times per round that our algorithms took w.r.t. the neural-linear approach (Tab. 4). As the times grow with increasing number of collected data points, we report the shortest, median, and the longest average round. Both TS and UCB variants of our method tend to be about 5 times slower than the LiM2 approach, taking around 2 seconds for the longest round. The total times can be further reduced for some applications by performing less frequent updates, as described in Sec. B.4. However, additional study would need to be conducted to check for a similar effect in neural-linear approaches.

B.3 COMPLEXITY ASSESSMENT OF THE NOISY WHEEL DATASET

SVM performance is one of the standard metrics for empirical assessment of the complexity of a problem (Li & Abu-Mostafa, 2006; Lorena et al., 2019). In Fig. 6 we measured performance of 3 baseline supervised approaches - NK ridge regression, SVM with RBF kernel, and SVM with polynomial kernel. Each method was run 10 times. The results show that the noise parameter increases the difficulty of the problem linearly in the chosen range from the performance perspective when no additional hyperparameter tuning is done.

[2]For adult and census datasets the categorical features are subsequently encoded with one hot vectors, resulting in ~ 80 features for adult and ~ 370 features for census. The numbers vary due to subsampling.

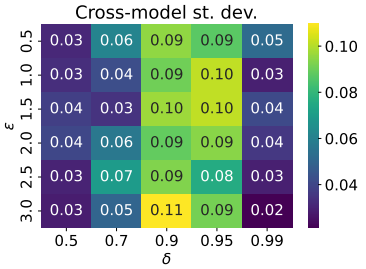

Figure 4: Standard deviations across tested methods (Fig. 3) reveal regions greatest differences in performance, which can be used to rank methods.

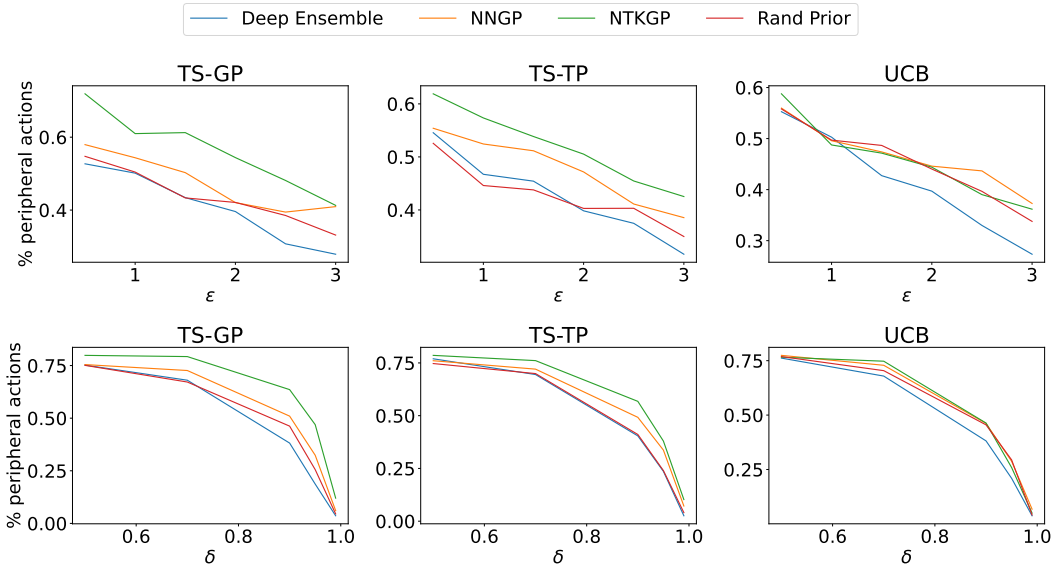

Figure 5: Percentage of correct peripheral actions, averaged across policies and a missing dimension (δ in top row, and ϵ in bottom row) for the experiment presented in Fig. 3. The aggregates provide clearer view of the average rank of each NK distribution.

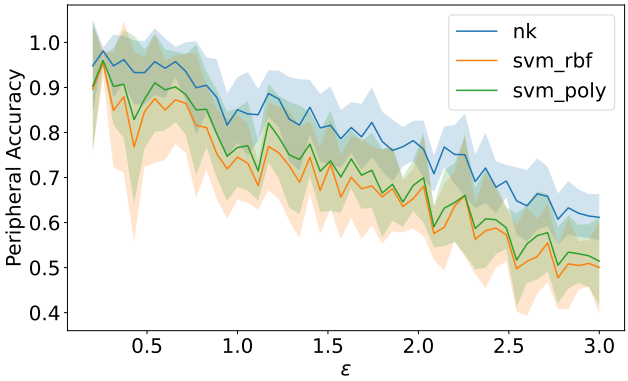

Figure 6: Baseline peripheral accuracy drops linearly as we increase the level of noise ε in noisy wheel dataset. The result further supports ε as a sufficient proxy for the rate of problem's complexity.

B.4 TRAINING FREQUENCY

Frequent updates of complex models in bandit settings put high demand on compute resources. We measure the sensitivity of our method to the number of actions taken between subsequent model updates. We ran our NK bandit for numbers of actions specified in the recent literature (Nabati et al., 2021; Riquelme et al., 2018): $\{1, 5, 20, 100, 400\}$, and report their respective cumulative rewards on the UCI datasets after 5,000 steps. Fig. 7 shows a smooth degradation in performance as the number of actions increases, which is expected. We note, however, that the degradation is relatively small and preserves competitive performance on all tested datasets. As large numbers of actions help to save significant amounts of computational resources, we believe that the sustained performance provides a strong argument for applicability to real world scenarios. It also positions our method as a good candidate for a limited memory approach, which we leave for future work. For some datasets, we observe a significant spike in performance at 5 actions before retraining. The result suggests an additional role the training frequency plays in the exploration-exploitation trade-off. In Fig. 7 we also show the results of the best performing models, other than our method, as reported in Fig. 2. We observe that for some datasets of higher complexity (e.g. covertype), our method's rank w.r.t. other algorithms is not affected, even with large numbers of actions before retraining, which additionally confirms its robustness.

B.5 JOINT MODEL

Following the zero-padding approach (Sec. 3.2), we compared the performance of joint and disjoint models and noticed a considerable improvement in the last epoch performance for linear and stochastic datasets (financial and mushroom), slight improvement for datasets of medium complexity, and a large deterioration for some of the most difficult datasets — covertype and jester. When examined from a resource usage perspective, however, the joint model underperforms considerably. This result is significant, as most neural bandits use a joint (NeuralUCB, NeuralTS) or partially joint (neural-linear) model. It is important to note that in functional space a disjoint model reduces the overall number of kernel entries, while in the case of parameter space models we observe the opposite effect. We can only create a disjoint model at the expense of introducing additional $(k-1)|\boldsymbol{\theta}|$ number of parameters, which puts a burden on resources. Our result highlights this difference, and can help practitioners assess the impact of choosing joint or disjoint models when using NK bandits.

Table 2: Summary of the UCI datasets used in the experiments.

Dataset	**Description**	**Distinguishing factor(s)**	d^2	k	$\sim n$	**Reward**	**Context**
Adult	predict income based on personal information	binary classification	13	2	50k	binary	categorical / integer
Census	predict occupation based on personal information	multivariate classification; large number of features	67	9	250k	binary	categorical / integer
Covertype	type of forest coverage in various areas	multivariate classification; requires nonlinear models (Riquelme et al., 2018)	54	7	150k	binary	categorical / integer
Financial	stock prices from NYSE and NASDAQ	linear problem; easy to over-explore	21	8	4k	continuous	continuous
Jester	joke recommender system	inputs and outputs in the same domain	32	8	20k	continuous	continuous
Mushroom	poisonous vs. safe mushrooms	imbalanced stochastic rewards; captures aspects of classification and regression	117	2	10k	continuous	categorical
Statlog	space shuttle flight indicators	multivariate classification; tests exploration; one arm optimal 80% of the time; requires nonlinear models (Riquelme et al., 2018)	9	7	45k	binary	integer

Table 3: Neural-linear methods compared over two significant hyperparameter settings.

	adult	census	covertype	financial	jester	mushroom	statlog
NK-TS $L = 1$ $\gamma = 0.2$	**4119 ± 16**	3152 ± 41	3402 ± 41	4407 ± 344	16779 ± 1234	4052 ± 4974	4720 ± 132
NK-TS $L = 2$ $\gamma = 0.2$	4113 ± 19	**3186 ± 29**	**3441 ± 36**	4162 ± 276	**16856 ± 1212**	2016 ± 3677	4820 ± 32
NeuralLinearTS $L = 1$ $n_l = 50$	3970 ± 36	2557 ± 54	2698 ± 76	4358 ± 363	12552 ± 1875	**11111 ± 336**	4771 ± 13
NeuralLinearTS $L = 2$ $n_l = 100$	3927 ± 26	2223 ± 181	2588 ± 74	4196 ± 350	11542 ± 2365	10516 ± 345	4801 ± 7
NeuralLinearTS-LiM2 $L = 1$ $n_l = 50$	4044 ± 17	2726 ± 36	2745 ± 141	**4485 ± 360**	14333 ± 1339	10962 ± 1070	4816 ± 66
NeuralLinearTS-LiM2 $L = 2$ $n_l = 100$	3993 ± 39	2304 ± 217	2748 ± 85	4357 ± 366	12325 ± 1005	10915 ± 631	**4872 ± 14**

Table 4: Times per epoch [sec] (min / median / max) rounded to 2 significant figures.

	adult	census	covertype	financial	jester	mushroom	statlog
NTK-TS	0.0 / 0.8 / 2.25	0.0 / 0.07 / 1.93	0.0 / 0.64 / 2.47	0.0 / 0.02 / 1.79	0.0 / 0.48 / 1.62	0.0 / 0.82 / 2.66	0.0 / 0.85 / 2.42
NTK-UCB	0.0 / 0.8 / 1.85	0.0 / 0.07 / 2.04	0.0 / 0.71 / 2.01	0.0 / 0.02 / 2.04	0.0 / 0.44 / 3.15	0.0 / 0.82 / 4.81	0.0 / 0.85 / 2.81
JointNTK-TS	0.67 / 0.9 / 5.68	0.0 / 2.0 / 5.5	0.0 / 0.97 / 2.43	0.0 / 0.83 / 2.51	0.0 / 0.92 / 2.87	0.68 / 0.94 / 2.47	0.0 / 0.83 / 4.13
JointNTK-UCB	0.67 / 0.91 / 3.16	0.0 / 2.0 / 5.49	0.0 / 0.97 / 3.56	0.0 / 0.83 / 2.27	0.0 / 0.89 / 2.38	0.67 / 0.94 / 2.45	0.0 / 0.81 / 2.82
NeuralLinearTS	0.0 / 0.0 / 1.2	0.0 / 0.0 / 1.3	0.0 / 0.0 / 1.24	0.0 / 0.0 / 0.96	0.0 / 0.0 / 1.22	0.0 / 0.0 / 1.19	0.0 / 0.0 / 1.21
NeuralLinearTS-LiM2	0.02 / 0.03 / 0.33	0.04 / 0.1 / 0.36	0.03 / 0.06 / 0.36	0.04 / 0.08 / 0.38	0.03 / 0.09 / 0.37	0.02 / 0.03 / 0.32	0.03 / 0.08 / 0.36

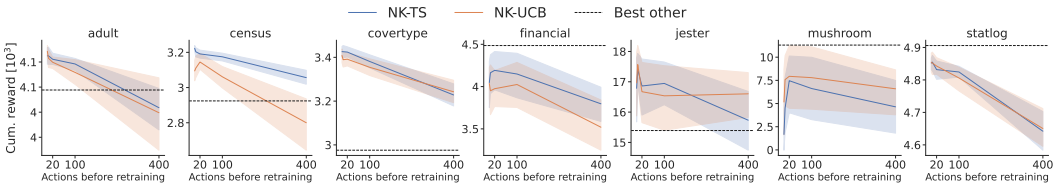

Figure 7: Performance w.r.t. $\{1, 5, 20, 100, 400\}$ actions before retraining.

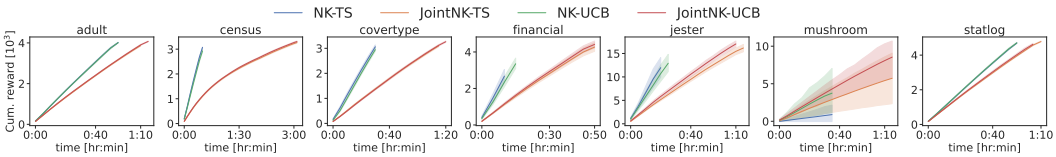

Figure 8: Cumulative rewards w.r.t. running times of a joint and a disjoint variant of our method.