Abstract

Most work on supervised learning research has focused on marginal predictions. In decision problems, joint predictive distributions are essential for good performance. Previous work has developed methods for assessing low-order predictive distributions with inputs sampled i.i.d. from the testing distribution. With low-dimensional inputs, these methods distinguish agents that effectively estimate uncertainty from those that do not. We establish that the predictive distribution order required for such differentiation increases greatly with input dimension, rendering these methods impractical.

To accommodate high-dimensional inputs, we introduce dyadic sampling, which focuses on predictive distributions associated with random pairs of inputs. We demonstrate that this approach efficiently distinguishes agents in high-dimensional examples involving simple logistic regression as well as complex synthetic and empirical data.

1 INTRODUCTION

We consider learning agents that are trained on data pairs \((X_t, Y_{t+1}) : t = 0, 1, ..., T - 1\). At a new input \(X_T\), such an agent can generate a predictive distribution of the outcome \(Y_{T+1}\) that is yet to be observed. This distribution characterizes the agent’s uncertainty about \(Y_{T+1}\). We refer to such a prediction as marginal to distinguish it from a joint predictive distribution over a sequence of prospective outcomes \((Y_{T+1}, ..., Y_{T+r})\) with inputs \((X_T, ..., X_{T+r-1})\).

Predictive distributions express uncertainty about future observations. The importance of such uncertainty estimation has motivated a great deal of research over recent years, much of which in the Bayesian deep learning community [Neal 2012]. With the proliferation of agents that generate predictive distributions, it is increasingly important to systematically study and improve their performance.

Recent theoretical work has highlighted the importance of joint predictive distributions in driving effective decisions [Wen et al. 2022]. This theory is supported by experiments that assess and compare agents using synthetic data generated by a random neural network and 2D inputs [Osband et al. 2022]. That work evaluates the quality of joint predictive distributions over ten inputs sampled i.i.d. from the training distribution. The results clearly distinguish agents that effectively estimate uncertainty from those that do not. This evaluation predicts agent performance when used to guide decisions in high-dimensional ‘neural bandits’.

However, as the input dimension increases, the aforementioned approach to evaluating agents becomes uninformative. As we will later discuss, the reason lies in the order of the predictive distributions being evaluated. With a two-dimensional input, the tenth order distribution suffices, but the predictive distribution order required to produce meaningful assessments increases rapidly with the input dimension. We could consider scaling the predictive distribution order as needed, but the evaluation algorithms of Osband et al. [2022] become computationally intractable.

To accommodate high-dimensional inputs, we introduce dyadic sampling, which focuses on predictive distributions associated with random pairs of inputs rather than those scattered according to the training input distribution. We demonstrate that this approach efficiently distinguishes agents in high-dimensional examples involving simple logistic regression as well as complex synthetic and empirical data. For example, agent assessments based on dyadic sampling are predictive of performance in high-dimensional neural bandits presented in results of Osband et al. [2022].

1.1 RELATED WORK

We are motivated by the importance of joint predictions in driving effective decisions [Wen et al. 2022]. Empirical analysis of joint predictions for deep learning in 2D supports this theory [Osband et al. 2022]. We provide a practical heuristic to scale these insights to high dimensions.
Our research is closely related to topics in Bayesian deep learning [MacKay, 1992] [Wilson and Izmailov, 2020], and robustness [Hendrycks and Dietterich, 2019]. For the most part, these communities have focused on the problem of marginal prediction [Nado et al., 2021] [Wilson et al., 2021]. Recent work has also highlighted a notion of cross-correlation in regression and related decision problems [Wang et al., 2021]. Our paper provides a related perspective that scales to classification and high dimensional data.

1.2 KEY CONTRIBUTIONS

We propose dyadic sampling, which evaluates high-order joint predictions at random pairs of inputs. Section 2 motivates the approach, and shows that it can mitigate some challenges in evaluating high-order predictive distributions.

Section 3 shows that dyadic sampling provides useful assessments in logistic regression. As input dimension scales, i.i.d. sampling from the training distribution does not offer a feasible approach. Dyadic sampling offers a viable path where the evaluation of [Osband et al., 2022] is inadequate.

Section 4 extends these insights to The Neural Testbed – an opensource package for the evaluation of joint predictions in deep learning. As in logistic regression, the neural network generative process is not amenable to evaluation via i.i.d. sampling when the input dimension exceeds three. In contrast, dyadic sampling scales gracefully as the input dimension grows large. As part of this project, we submit all agent and evaluation code to github.com/deepmind/neural_testbed.

Section 5 shows that our methodology can extend beyond synthetic data. Dyadic sampling can feasibly evaluate joint predictions on high-dimensional real datasets. We evaluate benchmark approaches to Bayesian deep learning and show that the insights from the Testbed carry over to real data. We see that, after tuning, all agents perform similarly in terms of marginal predictions. However, there are significant differences in the quality of joint predictions per agent, evaluated via dyadic sampling. Further, Testbed performance is highly predictive of performance on empirical data.

2 EVALUATING PREDICTIVES

This section introduces notation for the standard supervised learning framework we will consider as well as our evaluation metric: KL-loss. We show that estimating KL divergence in high dimensional distributions can be challenging, and present dyadic sampling as an effective heuristic.

2.1 ENVIRONMENT AND PREDICTIONS

Consider a sequence of pairs \(((X_t, Y_{t+1}) : t = 0, 1, 2, \ldots)\), where each \(X_t\) is a feature vector and each \(Y_{t+1}\) is its target label. Each target label \(Y_{t+1}\) is produced by an environment \(\mathcal{E}\), which we formally take to be a conditional distribution \(\mathcal{E}(\cdot|X_t)\). The environment \(\mathcal{E}\) is a random variable; this reflects the agent’s uncertainty about how labels are generated. Note that \(P(Y_{t+1} \in \cdot|\mathcal{E}, X_t) = \mathcal{E}(\cdot|X_t)\) and \(P(Y_{t+1} \in \cdot|X_t) = \mathbb{E}[\mathcal{E}(\cdot|X_t)|X_t]\).

We consider an agent that learns about the environment from training data \(D_T = ((X_t, Y_{t+1}) : t = 0, 1, \ldots, T - 1)\). After training, the agent predicts testing class labels \(Y_{T+1:T+\tau} \equiv (Y_{T+1}, \ldots, Y_{T+\tau})\) from unlabeled feature vectors \(X_{T:T+\tau-1} \equiv (X_T, \ldots, X_{T+\tau-1})\).

We describe the agent’s predictions in terms of a generative model, parameterized by a vector \(\theta_T\) that the agent learns from the training data \(D_T\). Specifically, \(\theta_T\) parameterizes a distribution \(\mathbb{P}(\hat{\mathcal{E}} \in \cdot|\theta_T)\) over imagined environment \(\hat{\mathcal{E}}\), which is also a conditional distribution. For any inputs \(X_{T:T+\tau-1}\), to generate the imagined labels \(Y_{T+1:T+\tau}\), the agent first samples an imagined environment \(\hat{\mathcal{E}}\) from \(\mathbb{P}(\hat{\mathcal{E}} \in \cdot|\theta_T)\), then generates \(Y_{t+1} : t \sim \mathcal{E}(\cdot|X_t)\) conditionally i.i.d. for each \(t = T, \ldots, T + \tau - 1\).

The agents \(\tau\)th-order predictive distribution is given by

\[
\hat{P}_{T+1:T+\tau} = \mathbb{P}(\hat{Y}_{T+1:T+\tau} \in \cdot|\theta_T, X_{T:T+\tau-1}),
\]

which represents an approximation to what would be obtained by conditioning on the environment:

\[
P_{T+1:T+\tau} = \mathbb{P}(Y_{T+1:T+\tau} \in \cdot|\mathcal{E}, X_{T:T+\tau-1}).
\]

If \(\tau = 1\), this represents a marginal prediction of a single label for a single feature vector. For \(\tau > 1\), this is a joint prediction over \(\tau\) labels for \(\tau\) different feature vectors.

2.2 EVALUATING JOINT PREDICTIONS

A learning agent can be assessed through the quality of its predictive distribution \(\hat{P}_{T+1:T+\tau}\). A canonical approach is to evaluate the KL-divergence [Wen et al., 2022]

\[
\Delta_{\tau} \equiv d_{KL}(P_{T+1:T+\tau} \parallel \hat{P}_{T+1:T+\tau}) \quad (1)
\]

\[
d_{KL} \equiv \mathbb{E}[\Delta_{\tau}] \quad (2)
\]

Recall that the expectation represents an integral over all random variables. The minimum of \(d_{KL}\) over all agents that depend on the environment only through \(D_T\) is attained by the posterior agent, whose predictive distribution is

\[
P_{T+1:T+\tau} = \mathbb{P}(Y_{T+1:T+\tau} \in \cdot|D_T, X_{T:T+\tau-1}). \quad (3)
\]

Let \(\Delta_{\tau}^*\) denote the minimum achievable KL-divergence.

Algorithm 1 provides a simple Monte-Carlo approach to evaluate \(\Delta_{\tau}^*\). As the order of the predictive distribution \(\tau\) grows, this provides a more nuanced evaluation of agent beliefs than just marginals. However, even for simple problems, the magnitude of \(\tau\) required to provide additional insight beyond marginals can become intractably large. To anchor our thinking on this matter we consider a simple coin tossing example.
Algorithm 1 KL-Loss Estimation [Osband et al., 2022].

\begin{algorithm}
\begin{algorithmic}
\State for $j = 1, 2, \ldots, J$ do
\State \hspace{1em} sample environment and training data
\State \hspace{1em} train agent on training data
\State for $n = 1, 2, \ldots, N$ do
\State \hspace{2em} compute environment likelihood $p_{j,n}$
\State \hspace{2em} compute agent likelihood $\hat{p}_{j,n}$
\State end for
\State end for
\State return $\frac{1}{JN} \sum_{j=1}^J \sum_{n=1}^N \log (p_{j,n}/\hat{p}_{j,n})$
\end{algorithmic}
\end{algorithm}

Example 1 (Bag of coins). Let each $X_i$ be a sample from coins $\{1, \ldots, M\}$. Let the probability of heads $p_x \sim \text{Unif}(0, 1)$ i.i.d. for each coin $x$. Each observation $Y_{i+1}$ is the outcome from tossing coin $X_i$, so that $\mathcal{E}(1|X_i) = px_i$.

Let us consider a predictive distribution for which $\mathbb{P}(Y_{1:T}|X_{0:T-1}) = \prod_{t=1}^T \mathbb{P}(Y_{t+1}|X_t) = 1/2^T$. Suppose an agent uses this to select coins sequentially with the aim of maximizing the expected number of heads. While $\hat{Y}_{1:T}$ accurately minimizes $d^2_{\text{KL}}$, the agent assumes that toss outcomes are independent and therefore does not learn from history to improve successive choices. Accounting for dependencies arising in the joint distribution, as would be captured by $d^2_{\text{KL}}$, is essential to maximizing performance.

Proposition 1 (Small $\tau$ approximately marginal). If the agent defined above is applied to Example 1 with $\tau \ll M$,

$$d^2_{\text{KL}} = \overline{d}^2_{\text{KL}} + O \left( \tau^3/M \right).$$

\textit{Proof.} Note that under the event that there are no repeated inputs in $X_{0:T-1}$, the posterior agent is equivalent to the agent defined above. For $\tau \ll M$, this event occurs with high probability. The detailed proof is in Appendix A.1.

Proposition 1 shows that if $\tau \ll M$, then $d^2_{\text{KL}}$ is unable to distinguish agents that only match marginals from those that are useful for decision making. When the cardinality of the input space $M$ is much larger than the test distribution order $\tau$, it is unlikely that any correlated inputs will be sampled. The metric $d^2_{\text{KL}}$ punishes agents that impose an erroneous correlation, but is unlikely to reward agents that correctly capture this dependence until $\tau$ is sufficiently large.

In Example 1 it may suffice to use a value of $\tau$ that grows cubicly in $M$. However, due to the curse of dimensionality, the required magnitude of $\tau$ can grow exponentially in problem dimension. To handle such cases, we introduce a practical evaluation metric that correctly identifies high quality predictive distributions with modest values of $\tau$.

For this result, we introduce notation for assignment: for random variables $A, B, C$ and a function $f(c) \equiv \mathbb{E}[A|B = c]$, let $\mathbb{E}[A|B \leftarrow C] = f(C)$. Note that, in general, if $C$ is a random variable then $\mathbb{E}[A|B \leftarrow C] \neq \mathbb{E}[A|B = C]$.

Definition 1 (Polyadic test sampling (of order $\kappa$)). For any $\kappa \in \mathbb{N}$, let ‘anchor points’ $\mathbf{X}_{1:k}$ be drawn i.i.d. from $\mathbb{P}(X_t \in \cdot)$, and let $\mathbf{X}_{T:T+\tau-1}^\kappa \sim \text{Unif}\{\mathbf{X}_{1:k}\}$. We define,

$$d^{\kappa}_{\text{KL}} \equiv \mathbb{E} \left[ \Delta_\tau \mid X_{T:T+\tau-1} \leftarrow \mathbf{X}_{T:T+\tau-1}^\kappa \right].$$

Polyadic sampling is motivated by a desire to investigate an agent’s predictions in situations where correlation between predictions is more likely to play an important role. Under reasonable regularity assumptions $\lim_{\kappa \to \infty} d^{\kappa}_{\text{KL}} = d_{\text{KL}}$ for all $\tau$. In the case of $\kappa = 1$, we can ensure that at least one input will be sampled multiple times. In the special case of $\kappa = 1$, we call this monadic sampling.

Proposition 2 (Monadic sampling cannot spot bad agents). Consider an agent that ignores the inputs and predicts

$$\mathbb{P}(\hat{Y}_{1:T} \mid X_{0:T-1}) = \mathbb{P}(\hat{Y}_{1:T}),$$

where $\hat{Y}_1, \ldots, \hat{Y}_\tau$ are sampled independently from $\text{Ber}(\hat{p})$ with a shared parameter $\hat{p} \sim \text{Unif}(0, 1)$. Then, for any $\tau \in \mathbb{N}$ in Example 1, this agent achieves the minimum $d^{\kappa=1}_{\text{KL}}$ over all agents.

\textit{Proof.} This agent is constructed so that for any repeated inputs $X_{0:T-1} = X_{0:T-1}$, this agent’s predictive distribution matches that of the posterior agent.

Monadic sampling can examine whether an agent understands the correlation structure at a single input. However, Proposition 2 shows that it does not punish agents that erroneously ascribe correlation to independent input-output pairs. This agent achieves the best possible score in $d^{\kappa=1}_{\text{KL}}$ but is useless for driving decisions. In order to weed out these agents it is crucial to sample more than one input point.

2.3 DYADIC SAMPLING ($\kappa = 2$)

This paper introduces dyadic test input sampling ($\kappa = 2$) as a practical heuristic for assessing the quality of joint predictions in high dimensions. This sampling scheme samples two random anchor points from the input space, and then randomly resamples the $\tau$ inputs from these anchor points. Even with moderate $\tau = 10$, we can be sure that most batches will contain a mix of points that are highly correlated to each other, as well as some others which may be quite different.

Dyadic sampling is a heuristic approach designed to work well in practical problems. The choice of $\kappa = 2$ addresses the extreme shortcomings of $d^{\kappa=1}_{\text{KL}}$ by Propositions 1 and 2 in the settings $\kappa \to \infty$ and $\kappa = 1$ respectively. However, it is certainly not a perfect substitute for evaluating $d^{\kappa=2}_{\text{KL}}$ with very large $\tau$. Depending on the setting, it is certainly possible to design agents that fare very well according to $d^{\kappa=2}_{\text{KL}}$, but very poorly according to $d^{\kappa=1}_{\text{KL}}$. 

1554
One might ask, ‘Does some other $1 < \kappa < \infty$ provide a better candidate for practical evaluation of posterior predictions?’ Could there be an analogous result to Proposition 2 when considering $\kappa = 2$, but evaluating posterior predictions at three anchor points? Note that, since $d_{KL}^{T,2}$ already evaluates the quality of the joint predictions at any pair of inputs, then for most problems the distribution over any three inputs will also be estimated well. In particular, for any Gaussian process, the first two moments are enough to determine the entire distribution of $\mathcal{E}$. We push details to Appendix A.2.

### 2.4 Joint Predictions and Information

So far, we have motivated dyadic sampling mostly through appeal to Example 1 together with some heuristic arguments. In this subsection we expand on this intuition through the lens of information theory.

To illustrate this, let’s consider the posterior agent, which is optimal for generating predictive distributions. Note that under the posterior agent,

$$d_{KL}^{t} = \mathbb{I}(Y_{t+1:T+\tau}; X_{D:T:t+1}) = \sum_{t}^{T+\tau-1} \mathbb{I}(Y_{t+1}; X_{D:T:t+1}), \quad (5)$$

where $\mathbb{I}(\cdot)$ denote the (conditional) mutual information [Cover 1999] and $D_{T:t} = (X_{T:t-1}, Y_{T:t+1})$. Note that the second equality follows from the chain rule of mutual information.

On the other hand,

$$\tau d_{KL}^{1} = \tau \mathbb{I}(Y_{t+1}; X_{D:T:t+1}) = \sum_{t}^{T+\tau-1} \mathbb{I}(Y_{t+1}; X_{D:T:t+1}), \quad (6)$$

where the second equality follows from $X_{D:T:t+1}$ are i.i.d.

For $d_{KL}^{t}$ to be significantly different from $\tau d_{KL}^{1}$, we need for at least one $t$, the dataset $D_{T:t}$ is informative about the target label $Y_{t+1}$ at $X_{t}$. For practical problems with $\tau$ small relative to the input space, the $D_{T:t}$ is not informative about $Y_{t+1}$. In such cases, we have $d_{KL}^{t} \approx \tau d_{KL}^{1}$. One way to think about dyadic sampling is a heuristic approach to sample $X_{D:T:t+1}$ so that $D_{T:t}$ is particularly informative about $Y_{t+1}$ and so evaluate the quality of the posterior approximation. Depending on the problem settings, other input sampling schemes may also be appropriate to accomplish this goal.

### 3 Logistic Regression

The results of Section 2 provide a motivation for dyadic sampling where it can sidestep the curse of dimensionality in higher-order predictive distributions. In this section, we show that this effect can occur in practical settings, not just obtuse problems cooked up for theory. In fact, even for the canonical problem of logistic regression, the benefits of dyadic sampling can be significant.

#### 3.1 Problem Formulation

We consider the familiar problem of $D$-dimensional logistic regression. Inputs are sampled i.i.d. $X_{t} \sim N(0, I_{D})$ and the environment $\mathcal{E}$ is determined by parameter $\phi \sim N(0, I_{D})$. Outputs $Y_{t+1} \in \{0, 1\}$ are then sampled according to

$$P(Y_{t+1} = 1|\mathcal{E}, X_{t}) = \frac{\exp(\rho\phi^{T}X_{t})}{\exp(\rho\phi^{T}X_{t}) + 1}.$$ 

Here $\rho > 0$ is the temperature controlling signal to noise ratio (SNR). We set $\rho = 0.01$ for a high SNR setting.

In this simple setting, we can compare three agents that make predictions $\hat{Y}_{t:T}$ given inputs $X_{0:t-1}$.

1. **uniform**: $P(\hat{Y}_{t+1} = 1|X_{t}) = \frac{1}{2}$ for $t = 0, 1, \ldots$
2. **marginal**: Samples $\lambda \sim N(0, 1)$, and then predicts $P(\hat{Y}_{t+1} = 1|\lambda, X_{t}) = \frac{\exp(\rho\lambda|X_{t}|)}{\exp(\rho|X_{t}|)^{2} + 1}$ for $t = 0, 1, \ldots$
3. **prior**: Samples $\phi \sim N(0, I_{D})$, and then predicts $P(\hat{Y}_{t+1} = 1|\phi, X_{t}) = \frac{\exp(\rho\phi^{T}X_{t})}{\exp(\rho\phi^{T}X_{t}) + 1}$ for $t = 0, 1, \ldots$

The agents are chosen to highlight specific properties of the logistic regression problem. The **uniform** agent makes the correct marginal predictions at any input, but does not capture any correlation among $Y_{t:T}$. The **marginal** agent makes the correct marginal predictions, and it also gets the correct joint distribution if inputs $X_{0:t-1}$ are all sampled at a single point (monadic sampling). However, it introduces spurious correlation among the predicted outputs if the inputs are not all equal. The **prior** agent samples from the true prior, and so is optimal for all $d_{KL}^{1}$. We would like to have a practical evaluation metric that can separate this optimal agent from these sub-optimal approximations.

We consider metrics $d_{KL}^{\tau}$ and $d_{KL}^{\kappa}$ for $\kappa = 1, 2$, all of which are estimated through Monte Carlo sampling according to Algorithm 1. Despite the simplicity of this problem, only dyadic sampling ($d_{KL}^{\tau}$ when $\tau = 2$) can correctly separate the agents once the input dimension grows.

#### 3.2 Results

As the **prior** agent makes optimal predictions in this problem, in principle, this agent will outperform all others according to $d_{KL}^{\tau}$ as the order of the predictive distribution $\tau$ grows. However, to separate the agents, the required $\tau$ can quickly become intractable in the input dimension.

Figure 1 shows that, in logistic regression, for dimension $D \geq 5$, even $\tau = 10,000$ is insufficient to give a factor of 2 separation between the optimal **prior** agent and the uninformmed **uniform** agent. The computational cost of evaluating $d_{KL}^{\tau}$ grows with $\tau$, so that this can quickly becomes impractical even for relatively small-scale problems. By contrast, evaluation with $d_{KL}^{\kappa = 2}$ is able to identify this separation with only $\tau = 10$ even as the input dimension grows.
Figure 1: $d_{KL}^\tau$ can separate prior agent from uniform, but the required $\tau$ is intractable in the input dimension. Dyadic sampling $d_{KL}^{\tau,\kappa=2}$ can distinguish these agents with a small value of $\tau$.

Figure 2 shows that this scaling carries over to high dimensions, fixing $\tau = 10$. Sampling test inputs i.i.d. cannot distinguish the uniform agent from the prior agent in dimensions greater than 100. At a high level, this result matches the spirit of Proposition 1. Figure 2 also shows that monadic sampling $d_{KL}^{\tau,\kappa=1}$ cannot distinguish the prior agent from the marginal agent. This mirrors Proposition 2 but in a setting with generalization. Dyadic sampling $d_{KL}^{\tau,\kappa=2}$ correctly identifies that prior agent is a superior agent across all input dimensions.

These results clearly demonstrate that the theoretical concerns raised in Section 2 actually occur in practical problems. Further, these concerns can occur even in the most simple settings of logistic regression, rather than contrived scenarios. We push the details on the robustness/sensitivity of these results to Appendix B.

4 THE NEURAL TESTBED

In this section we show that the insights observed in the linear setting of Section 3 extend to nonlinear function approximation and neural networks. Osband et al. [2022] introduce the Neural Testbed as a simple synthetic 2D problem to evaluate posterior predictives in deep learning. We show that, using the existing $d_{KL}^\tau$ evaluation, this approach does not scale to higher dimensions. However, using dyadic sampling we are able to extend these insights to practical scales. As part of our work we contribute these changes to [github.com/deepmind/neural_testbed](https://github.com/deepmind/neural_testbed).

4.1 PROBLEM FORMULATION

The Neural Testbed works with a synthetic data generating process around random 2-layer MLPs [Osband et al. 2022]. For each random seed, a random neural network is sampled according to standard Xavier initialization [Glorot and Bengio 2010]. Then, random train/test inputs are sampled $X_{1:T+T'} \sim N(0, I)$ and labels assigned randomly according to the probabilities of the generative MLP. We follow the exact settings in the existing opensource package except for two key changes.

First, we supplement the existing evaluation by $d_{KL}^1, d_{KL}^{10}$ to also evaluate according to $d_{KL}^{10,\kappa=2}$. Then, we vary the input dimension of the problem (which is fixed at $D = 2$ in the original Neural Testbed release). To account for the different data requirements in higher dimensions we similarly increase the number of training pairs in low, medium, high data regimes to scale with the input dimension.

The full testbed sweep is defined over input dimensions $D \in \{2, 10, 100\}$, number of training pairs $T = \lambda D$ for $\lambda \in \{1, 10, 100, 1000\}$, temperature $\rho \in \{0.01, 0.1, 0.5\}$ with 5 random seeds in each setting. We push full details, together with opensource implementation, to Appendix C.
Table 1: Summary of benchmark agents, full details in Appendix C.2.

<table>
<thead>
<tr>
<th>agent</th>
<th>description</th>
<th>hyperparameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>mlp</td>
<td>Vanilla MLP</td>
<td>$L_2$ decay</td>
</tr>
<tr>
<td>ensemble</td>
<td>‘Deep Ensemble’ [Lakshminarayanan et al., 2017]</td>
<td>$L_2$ decay, ensemble size</td>
</tr>
<tr>
<td>dropout</td>
<td>Dropout [Gal and Ghahramani, 2016]</td>
<td>$L_2$ decay, network, dropout rate</td>
</tr>
<tr>
<td>bbb</td>
<td>Bayes by Backprop [Blundell et al., 2015]</td>
<td>prior mixture, network, early stopping</td>
</tr>
<tr>
<td>hypermodel</td>
<td>Hypermodel [Dwaracherla et al., 2020]</td>
<td>$L_2$ decay, prior, bootstrap, index dimension</td>
</tr>
<tr>
<td>ensemble+</td>
<td>Ensemble + prior functions [Osband et al., 2018]</td>
<td>$L_2$ decay, ensemble size, prior scale, bootstrap</td>
</tr>
<tr>
<td>sgmcmc</td>
<td>Stochastic Langevin MCMC [Welling and Teh, 2011]</td>
<td>learning rate, prior, momentum</td>
</tr>
</tbody>
</table>

Figure 3: Comparing different agents on the testbed problems with input dimension $D = 100$. We see that the results for marginal $d_{KL}^{10}$ and joint $d_{KL}^{10,2}$ with i.i.d. test sampling do not show any significant difference in performance. By contrast, dyadic sampling $d_{KL}^{10,2}$ clearly separates agent performance in joint versus marginals.

4.2 BENCHMARK AGENTS

To compare the performance of benchmark agents we make use of the opensource agents developed by Osband et al. [2022]. Table 1 lists agents that we study and compare as well as hyperparameters that we tune. In our experiments, we optimize these hyperparameters via grid search. The choices start from the defaults released in github.com/deepmind/neural_testbed, but extend and tweak some hyperparameter choices for high dimensional problems. Further detail on these agents is provided in Appendix C.2.

4.3 OVERALL RESULTS

Figure 4 shows the KL estimates for these agents, normalized so that the baseline MLP has a score of 1. In each case, these agents are tuned for performance on the Neural Testbed for input dimension 100. We can see that in this setting evaluation in $d_{KL}^{10}$ is statistically indistinguishable from that of marginal predictions. We also see that, for the most part, the quality of these marginal predictions is not massively improved versus the MLP. However, unlike the 2D testbed results, we do see that some of these more advanced approaches can improve marginal predictions.

However, we see that evaluating agents according to dyadic sampling leads to massive distinctions in their evaluations. Interestingly, these qualitative results match the $d_{KL}^{10}$ with i.i.d. test sampling ordering in the 2D setting. Osband et al. [2022] showed that this order was highly correlated with performance in sequential decision problems, even for high input dimension. Our results provide a significant new finding; in high dimensional problems dyadic sampling sampling can provide a more targeted signal for the suitability in downstream tasks.

4.4 PRIORS IN HIGH DIMENSIONS

One of the most clear and interesting pairs of agents to compare is ensemble and ensemble+. These agents are identical except for the addition of randomized, fixed prior networks. Prior work has shown that this difference can be crucial in high-dimensional decision problems [Osband et al., 2018, Burda et al., 2019]. Comparison of joint predictions $d_{KL}^{10}$ in 2D problems also showed a significant difference, but only for very small training sets $T \leq 30$. The question remained, do these randomized priors provide value in large scale supervised learning?

Figure 4 shows that, according to $d_{KL}^{10}$ the benefits of ensemble+ appear to evaporate for input dimensions $\geq 2$. However, using dyadic sampling and $\kappa = 2$ we can see there are huge differences in the quality of their posterior approximation that extend to high dimensional problems. Figure 5 shows that, as we increase the dimensionality of
the problem, so too we increase the size of the largest training sets where prior functions afford significant advantages. Rather than becoming irrelevant in large problems, the importance of good inductive bias actually increases with input dimension.

![Figure 4: Global input sampling separates ensemble from ensemble+ only for low input dimension. Local sampling \( \kappa = 2 \) scales to high dimensions.](image)

![Figure 5: The benefits of ensemble+ over ensemble occur in the ‘low data regime’. However, the amount of data that constitutes as ‘low data’ grows with input dimension.](image)

5 REAL DATA

In this section we show that the key insights gained from the synthetic neural testbed can carry over to real datasets. We replace the neural network generative process of Section 4 with small challenge datasets drawn from the deep learning literature. We then tune the agents of Table 1 for each of these settings and analyse the results. We find that all agents can be tuned to perform roughly equivalently in terms of marginal predictions. However, their performance difference greatly in terms of their joint performance as measured by dyadic sampling. Further, agent performance on the testbed is highly correlated with performance on real datasets.

5.1 PROBLEM FORMULATION

Progress in the field of deep learning has been driven in large part through evaluation on shared, fixed datasets [Krizhevsky et al., 2012]. We repeat the analysis of Section 4 but replace the synthetic data generating process with a collection of datasets drawn from the literature [TFD].

Table 2 outlines the ten datasets we include in our analysis. We wanted to choose datasets that might provide an analogous challenge to the Neural Testbed and so selected them based on their popularity in the literature, and suitability for training with a 2-layer MLP. For this reason, large scale challenges such as ImageNet or language modelling, which typically require different classes of models were not included in our selection [Deng et al., 2009].

To mirror our evaluation in the Neural Testbed we begin with datasets \( D^n T_n = ((X_t, Y_{t+1}) : t = 0, \ldots, T_n - 1) \) for \( n = 1, \ldots, 10 \). To evaluate different data regimes we create subsampled datasets \( \hat{D}^n T \) for \( T = 10, 100, 1000, T_n \) to evaluate different data regimes. We then evaluate \( d_{KL}\) in the ‘low temperature’ limit, taking the labels in the supplied test set as probability 1 or, equivalently, the negative log-likelihood [Wen et al., 2022].

As in Section 4.2, we evaluate the agents outlined in Table 1 across each of these datasets in each data regime. We then tune the hyperparameters per dataset, per data regime and aggregate the performance by taking the average over all evaluations. This mirrors the procedure that we applied in Section 4. We push full details to Appendix D.

5.2 RESULTS

We begin by assessing the quality of the agents’ performance in marginal predictions, when averaged over all datasets, for all data regimes. Figure 6 shows that, once agents are optimized for each setting, the differences between agents is not statistically significant. This finding mirrors our observation in the case of synthetic data and Figure 3. These agents perform similarly at marginal prediction in the testbed, and overall they perform similarly in the real datasets as well.

Once you consider the quality of joint predictions however, there is a significant difference in the quality of predictive distributions evaluated on real data. Further, Figure 7 shows...
Table 2: Summary of benchmark datasets studied, full details in Appendix D.

<table>
<thead>
<tr>
<th>dataset name</th>
<th>type</th>
<th># classes</th>
<th>input dimension</th>
<th># training pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>structured</td>
<td>3</td>
<td>4</td>
<td>120</td>
</tr>
<tr>
<td>wine quality</td>
<td>structured</td>
<td>11</td>
<td>11</td>
<td>3,918</td>
</tr>
<tr>
<td>german credit numeric</td>
<td>structured</td>
<td>2</td>
<td>24</td>
<td>800</td>
</tr>
<tr>
<td>mnist</td>
<td>image</td>
<td>10</td>
<td>784</td>
<td>60,000</td>
</tr>
<tr>
<td>fashion-mnist</td>
<td>image</td>
<td>10</td>
<td>784</td>
<td>60,000</td>
</tr>
<tr>
<td>mnist-corrupted/shot-noise</td>
<td>image</td>
<td>10</td>
<td>784</td>
<td>60,000</td>
</tr>
<tr>
<td>emnist/letters</td>
<td>image</td>
<td>37</td>
<td>784</td>
<td>88,800</td>
</tr>
<tr>
<td>emnist/digits</td>
<td>image</td>
<td>10</td>
<td>784</td>
<td>240,000</td>
</tr>
<tr>
<td>cmaterdb</td>
<td>image</td>
<td>10</td>
<td>3,072</td>
<td>5,000</td>
</tr>
<tr>
<td>cifar10</td>
<td>image</td>
<td>10</td>
<td>3,072</td>
<td>50,000</td>
</tr>
</tbody>
</table>

Figure 6: None of the agents perform significantly better than MLP baseline in marginal likelihood.

Figure 7: The quality of joint predictions on the testbed is highly correlated with performance in real data.

that this difference is highly correlated with performance on the Neural Testbed. Agents that perform better in the setting with synthetic data also tend to perform better when evaluated on real data. This finding is particularly significant since the differences in $d^{KL}_{10.2}$ are quite large even for these state of the art agents. These results provide strong indications that the issues observed in sequential decision problems [Osband and Van Roy, 2015] and synthetic data [Osband et al., 2022] can extend to real data.

Now, in some sense the results we have presented are ‘non-standard’ in that our evaluation includes averages over restricted-data versions of the canonical datasets in Table 2. We believe that this is a sensible approach if you are interested in designing learning agents that work in online decision making and are robust to different data regimes. However, in some supervised learning settings it is more common from practitioners to care only about the ‘full’ datasets with $T = T_n$. In fact, the findings of Figure 6 and Figure 7 are essentially unchanged when restricting only to the ‘full data’ regime. That is, the differences in marginal predictions $\tau = 1$ are quite minor, but the differences in $\tau = 10, \kappa = 2$ are extreme. Further, that these differences in joint performance are highly correlated with agent performance in the testbed. We push full details to Appendix D.

6 CONCLUSION

Good predictions are essential for good decisions. Crucially, the quality of these decisions depends on the quality of joint predictions and not just the marginals [Wen et al., 2022]. In this paper, we highlight the difficulties in evaluating high-order predictive distributions that are essential for decision making. We introduce dyadic sampling as a practical heuristic to sidestep the curse of dimensionality.

We motivate dyadic sampling through a simple discrete example, and show that the key insights extend to linear and then nonlinear systems. We show that the Neural Testbed cannot effectively scale to high dimensions with i.i.d. sampling, but that it can with dyadic sampling. Importantly, this approach also scales to challenge datasets, and we show that testbed performance is highly correlated with real data.

A major contribution of our work is the opensource effort at [github.com/deepmind/neural_testbed]. This includes all the code used to generate the paper, and helps to provide clear and reproducible benchmarks for the community. We believe that this paper can provide an stimulating base for future research into agents that make predictions in high-dimensional problems, and drive effective AI systems.
References


Zheng Wen, Ian Osband, Chao Qin, Xiuyuan Lu, Morteza Ibrahimi, Vikranth Dwaracherla, Mohammad Asghari, and Benjamin Van Roy. From predictions to decisions: The importance of joint predictive distributions, 2022.
