Comparison of three recurrent neural networks for rainfall-runoff modelling at a snow-dominated watershed

K Yokoo¹, K Ishida*², T Nagasato³, A Ercan⁴ and T Tu⁵

¹Graduated School of Science and Technology, Kumamoto University, Japan
²Kumamoto University, Kumamoto, Japan
³Department of Civil and Environmental Engineering, Kumamoto University
⁴University of California, Davis, One Shields Avenue Davis, California, United States
⁵School of Civil Engineering, Sun Yat-Sen University, Guangzhou, China

*E-mail: keiishida@kumamoto-u.ac.jp

Abstract. In recent years, rainfall-runoff modelling using LSTM has shown high adaptability. However, LSTM requires far more computational costs than traditional RNN. In addition, a different type of RNN, GRU, has been developed to solve this issue of LSTM. Therefore, this study compares the accuracy of the deep learning methods for rainfall-runoff modelling using three deep learning methods in a snow-dominated area. Besides, the setting of hyperparameters may affect accuracy. The accuracy of these deep learning methods was investigated by trying multiple combinations of hyperparameters. The input data were daily temperature data and precipitation data. The results show that GRU gives the highest accuracy in most combinations.

1. Introduction

Application of deep learning is a hot topic in various research fields. Deep learning is nowadays also applied to many kinds of issues in geoscience as summarised in review papers in high-impact journals [1–3]. These applications showed the high potential of deep learning in geoscience.

Rainfall-runoff modelling including flow discharge forecasting may be an issue that deep learning is frequently applied. There is a type of deep learning that is suitable for time series modelling. It is known as the recurrent neural network (RNN). A method categorized into RNN, which is called Long and Short-Term Memory (LSTM) network, has large potential to model time series that has a long-term dependency. Due to this feature of the LSTM, LSTM has been applied for time series modelling in geoscience, especially, rainfall-runoff modelling. For instance, Kratzert et al. (2019) used meteorological data such as precipitation, air temperature, and radiation as input, and then implemented flow discharge models at multiple watersheds in the United States[4]. Some other research groups applied LSTM for flow forecasting [5–7]. Furthermore, Kao et al. (2020), Li et al. (2020), and Xiang et al. (2020) applied the encoder-decoder version of LSTM for flow forecasting [8–10]. The results of these previous studies show the high applicability of LSTM for rainfall-runoff modelling.

Although LSTM has an advantage of accuracy, it has a disadvantage over the traditional RNN. It is known that LSTM requires much more computational resources than the traditional RNN because of the complex structure of LSTM. Due to this issue of LSTM, another type of RNN with a simpler structure, Gated Recurrent Unit (GRU), was developed by Cho et al. (2014)[11]. Jeong and Park (2019)[12] applied GRU and LSTM for groundwater level modelling. The accuracy of GRU is comparable to LSTM. Thus, GRU may also be another good option for rainfall-runoff modelling.
Zhang et al. (2018)[13] applied GRU for flow discharge forecasting at a sewer, and then they showed similar model accuracy between GRU and LSTM.

On the other hand, comparisons among the deep learning methods in the above studies are generally conducted with a single trained result. For the training process of a deep learning method, the initial states of its learnable parameters such as weights and biases are generally set with random values. The randomness of their initial states may affect the accuracy of the trained results. Meanwhile, a deep learning method including the traditional RNN, LSTM, GRU has model options to be set, which are called hyper-parameters. The setting of hyper-parameters may also affect accuracy. The objective of this study is to compare the model accuracy among the traditional RNN, LSTM, and GRU for rainfall-runoff modelling. In order to consider the effects of the initial states of their learnable parameters and their model configuration, the comparisons are conducted with multiple combinations of hyper-parameters. As a study watershed, the Ishikari River watershed is selected. The Ishikari River watershed is a snow-dominated watershed, which is located in the Hokkaido region, Japan. There are long-term dependencies between meteorological variables and flow discharge at a snow-dominated watershed. It may be a suitable issue to compare the three methods.

2. Methods

2.1. Long and Short-Term Memory (LSTM) Network[14]

LSTM is described as follows:

\[ i_s = \sigma(W_{ii} x_s + b_{ii} + W_{hi} h_{s-1} + b_{hi}) \]
\[ o_s = \sigma(W_{io} x_s + b_{io} + W_{ho} h_{s-1} + b_{ho}) \]
\[ f_s = \sigma(W_{if} x_s + b_{if} + W_{hf} h_{s-1} + b_{hf}) \]
\[ g_s = \tanh(W_{ig} x_s + b_{ig} + W_{hg} h_{s-1} + b_{hg}) \]
\[ c_s = f_s \cdot c_{s-1} + i_s \cdot g_s \]
\[ h_s = o_s \cdot \tanh(c_s) \]

where \( i_s, f_s, g_s, \) and \( o_s \) are the input, forget, cell, and output gate at time \( s \). \( h_s, c_s \) are the hidden state and cell state at time \( s \). \( W_{ii}, W_{io}, W_{if}, W_{ig}, W_{hi}, W_{ho}, W_{hf}, W_{hg} \) and \( W_{hg} \) are weights. \( b_{ii}, b_{io}, b_{if}, b_{ig}, b_{hi}, b_{ho}, b_{hf}, \) and \( b_{hg} \) are biases. When \( x_s \) is input, \( x_s \) is used \( h_{s-1} \) is the hidden state at time \( s \). \( h_s \) is a linear transformation of \( c_s \). The internal information of \( c_{s-1} \) is updated by inputting \( x_s \) and \( h_{s-1} \). Finally, \( h_s \) of each LSTM block is linearly transformed, and the total value becomes the \( o_s \).

2.2. Traditional Recurrent Neural Network (RNN)

The traditional RNN can be written by the following equation:

\[ h_s = \tanh(W_{lh} x_s + b_{lh} + W_{hh} h_{s-1} + b_{hh}) \]

where \( h_s, x_s \) are the hidden state and the input at time \( s \). \( h_{s-1} \) is the hidden state at the previous time \( (s - 1) \). \( W_{lh}, W_{hh} \) are input weights. \( b_{lh}, b_{hh} \) are biases. The hidden state is updated each time. Finally, the value obtained by linearly transforming the hidden state is given as output.

2.3. Gated Recurrent Unit (GRU)[15]

GRU consists of the following equations:

\[ y_s = \sigma(W_{iy} x_s + b_{iy} + W_{iy} h_{s-1} + b_{iy}) \]
\[ z_s = \sigma(W_{iz} x_s + b_{iz} + W_{hz} h_{s-1} + b_{hz}) \]
\[ n_s = \tanh(W_{in} x_s + b_{in} + y_s \ast (W_{hn} h_{s-1} + b_{hn})) \]
\[ h_s = (1 - z_s) * n_s + z_s * h_{s-1} \]

Where \( \gamma_s, z_s, \) and \( n_s \) are the reset, update, and new gates at time \( s \). \( h_s \) is the hidden state at time \( s \). GRU is a method that has a simpler structure than LSTM.

2.4. Hyperparameters

The above three methods have various hyper-parameters which are their model options to be set. The input data length (IDL) may be an important hyper-parameter. Rainfall-runoff modelling at a snow-dominated watershed is affected by the snow accumulation and melting processes which leads long term dependencies between the meteorological variables and flow discharge. The hidden state length (HSL) may also be an important hyper-parameter. HSL is equivalent to the number of neurons of a traditional neural network. A deep learning method is generally trained by the mini-batch gradient descent method. This study also utilized the mini-batch gradient descent method. In this case, the batch size needs to be set. Meanwhile, the mini-batch gradient descent method requires a loss function to update its learnable parameters, and an optimization algorithm to adjust the learning rate. Among these hyper-parameters, this study focused on IDL, HSL, and Optimization algorithm. The combinations of hyper-parameters used in this study are tabulated in Table 1-9.

3. Study area and Data

3.1. Study area

The Ishikari River originates from Mt.Ishikari whose altitude is 1,967 meters above sea level. This channel length is 268km, which is the third-largest in Japan. This basin is 14,330km\(^2\), which is the second-largest in Japan. This basin is located at E141°20’45” N43°14’14” alt (Figure 1). This basin which is a snow-accumulated area accumulates a lot of snow in winter. The accumulated-snow period is from November to March, and the snow-melting period is from April to May. The flow discharge in this basin increases significantly compared to other periods.

![Figure 1. Ishikari basin in Japan.](image)

3.2. Input and target data

This study utilizes precipitation and air temperature data as input. The target data are flow discharge data at the study watershed. These data were obtained from different data sources as follows.
3.2.1. Precipitation data. This study obtained daily precipitation from the Asian Precipitation – Highly Resolved Observational Data Integration Towards Evaluation of Water Resource (APHRODITE) [16]. APHRODITE is a gridded observed precipitation dataset at the daily-scale. Its spatial coverage is entire Japan. Its spatial resolution is approximately 5 km. The temporal coverage is from 1900 to 2015.

3.2.2. Air temperature data. The air temperature data except precipitation data were extracted from an atmospheric reanalysis dataset, the fifth generation of the European Centre for Medium-Range Weather Forecasts Re-analysis (ERA5)[17]. The horizontal resolution of ERA5 is approximately 25 km x 25 km (0.25° x 0.25°). The temporal resolution of the provided data is hourly. The ERA5 dataset contains many variables. This study obtained air temperature from the single-level data. The basin average value at the study watershed was calculated from the gridded information at each hour. Then the hourly data were converted into the daily scale after adjusting the time difference between the global time and Japan standard time.

3.2.3. Flow Discharge data. The flow discharge data that are used as the target data in this study were obtained at the Ishikari Ohashi station, which is provided by Water Information System (WIS). WIS is operated by the Ministry of Land, Infrastructure, Transport and Tourism, Japan (http://www1.river.go.jp/). The Ishikari Ohashi station is located at E141°32'32" N43°07'20". The river mouth is 26.60 km downstream from the station. The original data were obtained at hourly-scale. The original flow discharge data were also converted to the daily-scale.

4. Comparison procedure
This study utilized a deep learning framework, PyTorch[18], to develop rainfall-runoff models with LSTM, RNN, and GRU. Daily precipitation and daily mean air temperature were given together to the model as input. The target data are the daily flow discharge at the Ishikari Ohashi station. The dataset was divided into three subsets: the training dataset (1998-2009), the validation dataset (2010-2012), and the test dataset (2013-2015). The models by each deep learning method are trained with the dataset with each combination of the hyperparameters (Table 1-9). To consider the randomness contained in the initial states of the learnable parameters, each method with each hyperparameter combination is trained 100 times. The best result with respect to the losses for the validation dataset is picked up and then evaluated with the test dataset. Then, the three methods are compared with respect to statistical values such as root mean square error (RMSE) and Nash-Sutcliffe efficiency (NSE) for the test dataset. The RMSE decreases as the observed and predicted values approach each other. The accuracy of NSE increases as it approaches 1.0. The accuracy is high when NSE = 0.5 or more. The accuracy of R increases as it approaches 1.0. Meanwhile, these statistical values for the training and validation datasets are also investigated to check the fluctuations in the results due to the randomness in the initial states.

5. Results and Discussions
First, IDL and the optimization algorithm were set to be 365 and Adam, respectively. Then the five HSLs were tried with all the methods. Table 1-3 shows the three statistical values (R, NSE, and RMSE) for the test dataset. For each of the three methods, there is not a linear relationship between the model accuracy and the HSL. All three methods show the highest accuracy with HSL=25. For LSTM, R, NSE, and RMSE are 0.899, 0.801, and 173.8 m³/s, respectively. For RNN, R, NSE, and RMSE are 0.873, 0.759, and 191.1 m³/s, respectively. For GRU, R, NSE, and RMSE are 0.917, 0.837, and 157.2 m³/s, respectively. However, the worst results are found with HSL=100 for LSTM, with HSL=100 for RNN, and with HSL=50 for GRU. On the other hand, GRU shows better results than LSTM and RNN with all the HSL. The R-value for GRU is 0.017-0.041 higher than that for LSTM, and 0.043-0.083 higher than that for RNN. The NSE value for GRU is 0.035-0.085 higher than that for LSTM, and 0.078-0.179 higher than that for RNN. The RMSE value for GRU is 15.6-36.5 m³/s small than that for LSTM, and 33.9-69.8 m³/s smaller than that for RNN.
Figures 2-4 show R, NSE, and RMSE, respectively, for the training and validation datasets that were obtained by three methods with the five HSLs. LSTM and GRU clearly show better results than RNN with respect to all three statistical values. Furthermore, GRU is slightly better than LSTM with respect to its median values. Meanwhile, LSTM and GRU obtained better accuracy with a longer HSL although there is not such a relation between the model accuracy and HSL for the test dataset as shown above. The simulation for the test dataset was obtained with the trained model that shows the best results with respect to the losses for the validation dataset among the 100 trained results as explained above. This indicates that the best result for the validation dataset may not be the best for the test dataset although there is some consistency between them.

Next, HSL and the optimization algorithm were set to be 50 and Adam, respectively. Then the five IDLs were tried with all the methods. Table 4-6 shows the three statistical values (R, NSE, and RMSE) for the test dataset. For LSTM, the best results were obtained with IDL=180. R, NSE, and RMSE are 0.894, 0.796, and 176.1 m$^3$/s, respectively. RNN obtained the best accuracy with IDL=100. R, NSE, and RMSE are 0.877, 0.766, and 188.3 m$^3$/s, respectively. GRU showed the best with IDL=365. R, NSE, and RMSE are 0.909, 0.813, and 168.5 m$^3$/s, respectively. LSTM and GRU seem to show better accuracy with a longer IDL. Similar to the above comparisons, GRU shows better results than LSTM and RNN with all the IDLs. The R-value for GRU is 0.004-0.031 higher than that for LSTM, and 0.004-0.059 higher than that for RNN. The NSE value for GRU is 0.005-0.044 higher than that for LSTM, and 0.005-0.100 higher than that for RNN. The RMSE value for GRU is 2.4-18.7 m$^3$/s small than that for LSTM, and 1.6-39.1 m$^3$/s smaller than that for RNN. Figures 5-7 show the corresponding R, NSE, and RMSE values, respectively, for the training and validation datasets. The three statistical values are worse with IDL=10 for all three methods. With the other IDLs, the results show similar features to those with different HSLs as shown above.

For the last comparison, HSL and IDL were set to be 50 and 365, respectively, and then four optimization algorithms were tried with all three methods. With respect to the statistical values (R, NSE, and RMSE) for the test period (Table 7-9), LSTM and GRU show the highest accuracy with Adagrad while RNN shows the best with Adam. R, NSE, and RMSE for LSTM with Adagrad are 0.903, 0.808, and 170.8 m$^3$/s, respectively. These for GRU with Adagrad are 0.912, 0.823, and 164.0 m$^3$/s, respectively. Unlike the above two comparisons, GRU is not always the best among the three methods. LSTM is the best with RMSprop and Adadelta.

Figures 8-10 show R, NSE, and RMSE, respectively, for the training and validation datasets obtained by the three methods with the four optimization algorithms. The figures show that RNN was stably trained only with Adagrad and Adam. Especially, RNN cannot properly be trained with RMSprop. The training process of GRU was also very unstable with RMSprop. Only LSTM is relatively stable with all the optimization algorithms.

Comparisons between the three methods and each Optimizer have been performed, but none have been compared at the same time. There is not much comparison by HSL or IDL. Therefore, by making this comparison, it was clarified which method is suitable for each hyperparameter.

The results in this study show that GRU has a high potential to obtain high accuracy for rainfall-runoff modelling at a snow-dominated watershed although LSTM has the most complex structure among the three methods. GRU showed the best accuracy with most combinations of hyper-parameters. However, GRU was unstable with RMSprop for the optimization algorithms. These results indicate that the comparisons among the methods should be conducted with some combinations of hyper-parameters.

| Table 1. R for each HSL in the test period. |
|---------------------------------------------|
| HSL  | 10   | 25   | 50   | 75   | 100  |
|------|------|------|------|------|------|
| LSTM | 0.899| 0.899| 0.878| 0.891| 0.873|
| RNN  | 0.855| 0.873| 0.850| 0.859| 0.831|
| GRU  | 0.917| 0.916| 0.909| 0.914| 0.914|
| Table 2. NSE for each HSL in the test period. |
|------------------|-----|-----|-----|-----|-----|
| **HSL** | **10** | **25** | **50** | **75** | **100** |
| LSTM | 0.782 | 0.801 | 0.769 | 0.792 | 0.747 |
| RNN | 0.724 | 0.759 | 0.721 | 0.716 | 0.653 |
| GRU | 0.837 | 0.837 | 0.813 | 0.827 | 0.832 |

| Table 3. RMSE for each HSL in the test period. |
|------------------|-----|-----|-----|-----|-----|
| **HSL** | **10** | **25** | **50** | **75** | **100** |
| LSTM | 181.918 | 173.837 | 187.159 | 177.638 | 195.989 |
| RNN | 204.611 | 191.108 | 205.596 | 207.464 | 229.327 |
| GRU | 157.293 | 157.220 | 168.482 | 162.006 | 159.474 |

| Table 4. R for each IDL in the test period. |
|------------------|-----|-----|-----|-----|-----|
| **IDL** | **10** | **50** | **100** | **180** | **365** |
| LSTM | 0.721 | 0.887 | 0.868 | 0.894 | 0.878 |
| RNN | 0.722 | 0.869 | 0.877 | 0.840 | 0.850 |
| GRU | 0.726 | 0.894 | 0.887 | 0.898 | 0.909 |

| Table 5. NSE for each IDL in the test period. |
|------------------|-----|-----|-----|-----|-----|
| **IDL** | **10** | **50** | **100** | **180** | **365** |
| LSTM | 0.512 | 0.787 | 0.747 | 0.796 | 0.769 |
| RNN | 0.522 | 0.754 | 0.766 | 0.701 | 0.721 |
| GRU | 0.527 | 0.793 | 0.779 | 0.801 | 0.813 |

| Table 6. RMSE for each IDL in the test period. |
|------------------|-----|-----|-----|-----|-----|
| **IDL** | **10** | **50** | **100** | **180** | **365** |
| LSTM | 272.157 | 179.836 | 197.190 | 176.133 | 187.159 |
| RNN | 269.401 | 193.100 | 188.316 | 212.840 | 205.596 |
| GRU | 267.827 | 177.416 | 183.051 | 173.710 | 168.482 |

| Table 7. R for each optimizer in the test period. |
|------------------|-----|-----|-----|-----|
| **OPT** | Adadelta | Adagrad | Adam | RMSprop |
| LSTM | 0.845 | 0.903 | 0.878 | 0.894 |
| RNN | 0.847 | 0.841 | 0.850 | 0.280 |
| GRU | 0.837 | 0.912 | 0.909 | 0.808 |

| Table 8. NSE for each optimizer in the test period. |
|------------------|-----|-----|-----|-----|
| **OPT** | Adadelta | Adagrad | Adam | RMSprop |
| LSTM | 0.714 | 0.808 | 0.769 | 0.796 |
| RNN | 0.707 | 0.693 | 0.721 | 0.00 |
| GRU | 0.701 | 0.823 | 0.812 | 0.631 |
Table 9. RMSE for each optimizer in the test period.

| OPT   | Adadelta | Adagrad | Adam   | RMSprop |
|-------|----------|---------|--------|---------|
| LSTM  | 208.473  | 170.787 | 187.159| 175.764 |
| RNN   | 210.711  | 215.660 | 205.596| 388.869 |
| GRU   | 213.057  | 163.997 | 168.482| 236.647 |

**Figure 2.** $R$ values for the training and validation period.

**Figure 3.** NSE values for the training and validation period.

**Figure 4.** RMSE values for the training and validation period.

**Figure 5.** $R$ values for the training and validation period.

**Figure 6.** NSE values for the training and validation period.

**Figure 7.** RMSE values for the training and validation period.
6. Conclusions
This study compared three deep learning methods, the traditional RNN, LSTM, and GRU for rainfall-runoff modelling at a snow-dominated watershed, which has long-term dependencies between input data (meteorological data) and the target data (flow discharge). In order to consider the effects of the selections of hyper-parameters, the comparisons were conducted with several combinations of hyper-parameters. The results in this study show that the GRU of the three methods is the best accurate in most cases. Therefore, GRU is the best among the three methods with most of the combinations of the hyper-parameters. On the other hand, GRU is worse than LSTM with a few combinations. This indicates that the comparisons among the methods should be conducted with various combinations of hyper-parameters.

Acknowledgments
This study was partially supported by JSPS KAKENHI Grant-in-Aid for Special Purposes 20K2191600.

References
[1] Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N and Prabhat 2019 Deep learning and process understanding for data-driven Earth system science Nature 566 195–204
[2] Shen C 2018 A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists Water Resour. Res. 54 8558–93
[3] Karpatne A, Ebert-Uphoff I, Ravela S, Babaie H A and Kumar V 2019 Machine Learning for the Geosciences: Challenges and Opportunities IEEE Trans. Knowl. Data Eng. 31 1544–54
[4] Kratzert F, Klots D, Brenner C, Schulz K and Herrnegger M 2018 Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks Hydrology and Earth System Sciences 22 6005–22
[5] Tian Y, Xu Y-P, Yang Z, Wang G and Zhu Q 2018 Integration of a Parsimonious Hydrological Model with Recurrent Neural Networks for Improved Streamflow Forecasting Water 10 1655
[6] Song T, Ding W, Wu J, Liu H, Zhou H and Chu J 2019 Flash Flood Forecasting Based on Long Short-Term Memory Networks Water 12 109
[7] Zhu S, Luo X, Yuan X and Xu Z 2020 An improved long short-term memory network for streamflow forecasting in the upper Yangtze River Stoch. Environ. Risk Assess. 1–17
[8] Li W, Kiaghadi A and Dawson C N 2020 High Temporal Resolution Rainfall Runoff Modelling Using Long-Short-Term-Memory (LSTM) Networks
[9] Kao I-F, Zhou Y, Chang L-C and Chang F-J 2020 Exploring a Long Short-Term Memory based Encoder-Decoder Framework for Multi-Step-Ahead Flood Forecasting J. Hydrol. 124631
[10] Xiang Z, Yan J and Demir I 2020 A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning Water Resour. Res. 56
[11] Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
[12] Jeong J and Park E 2019 Comparative applications of data-driven models representing water table fluctuations J. Hydrol. 572 261–73
[13] Zhang D, Lin J, Peng Q, Wang D, Yang T, Sorooshian S, Liu X and Zhuang J 2018 Modeling and simulating of reservoir operation using the artificial neural network, support vector regression, deep learning algorithm J. Hydrol. 565 720–36
[14] Hochreiter S and Schmidhuber J 1997 Long Short-Term Memory Neural Comput. 9 1735–80
[15] Dey R and Salenf F M 2017 Gate-variants of Gated Recurrent Unit (GRU) neural networks Midwest Symposium on Circuits and Systems vol 2017-August (Institute of Electrical and Electronics Engineers Inc.) pp 1597–600
[16] Kamiguchi K, Arakawa O, Kitoh A, Yatagai A, Hamada A and Yasutomi N 2010 Development of APHRO_JP, the first Japanese high-resolution daily precipitation product for more than 100 years Hydrol. Res. Lett. 4 60–4
[17] Hersbach H, Bell B, Berrisford P, Hirahara S, Horányi A, Muñoz-Sabater J, Nicolas J, Peubey C, Radu R, Schepers D, Simmons A, Soci C, Abdalla S, Abellan X, Balsamo G, Bechtold P, Biavati G, Bidlot J, Bonavita M, Chiara G, Dahlgren P, Dee D, Diamantakis M, Dragani R, Fleming J, Forbes R, Fuentes M, Geer A, Haimberger L, Healy S, Hogan R J, Holm E, Janisková M, Keeley S, Laloyaux P, Lopez P, Lupu C, Radnoti G, Rosnay P, Rozum I, Vamborg F, Villaume S and Thépaut J 2020 The ERA5 global reanalysis Q. J. R. Meteorol. Soc. 146 1999–2049
[18] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J and Chintala S 2019 PyTorch: An Imperative Style, High-Performance Deep Learning Library Advances in Neural Information Processing Systems 32 ed H Wallach, H Larochelle, A Beygelzimer, F dometrics single Alché-Buc, E Fox and R Garnett (Curran Associates, Inc.) pp 8024–35