Structured State Space Models for Multiple Instance Learning in Digital Pathology

Leo Fillioux[(✉)], Joseph Boyd, Maria Vakalopoulou, Paul-henry Cournède, and Stergios Christodoulidis

MICS Laboratory, CentraleSupélec, Université Paris-Saclay,
91190 Gif-sur-Yvette, France
{leo.fillioux,joseph.boyd,maria.vakalopoulou,
paul-henry.cournede,stergios.christodoulidis}@centralesupelec.fr

Abstract. Multiple instance learning is an ideal mode of analysis for histopathology data, where vast whole slide images are typically annotated with a single global label. In such cases, a whole slide image is modelled as a collection of tissue patches to be aggregated and classified. Common models for performing this classification include recurrent neural networks and transformers. Although powerful compression algorithms, such as deep pre-trained neural networks, are used to reduce the dimensionality of each patch, the sequences arising from whole slide images remain excessively long, routinely containing tens of thousands of patches. Structured state space models are an emerging alternative for sequence modelling, specifically designed for the efficient modelling of long sequences. These models invoke an optimal projection of an input sequence into memory units that compress the entire sequence. In this paper, we propose the use of state space models as a multiple instance learner to a variety of problems in digital pathology. Across experiments in metastasis detection, cancer subtyping, mutation classification, and multitask learning, we demonstrate the competitiveness of this new class of models with existing state of the art approaches. Our code is available at https://github.com/MICS-Lab/s4_digital_pathology.

Keywords: Multiple instance learning · Whole slide images · State space models

1 Introduction

Precision medicine efforts are shifting cancer care standards by providing novel personalised treatment plans with promising outcomes. Patient selection for such

---

L. Fillioux and J. Boyd—These authors contributed equally to this work.

---

treatment regimes is based principally on the assessment of tissue biopsies and the characterisation of the tumor microenvironment. This is typically performed by experienced pathologists, who closely inspect chemically stained histopathological whole slide images (WSIs). Increasingly, clinical centers are investing in the digitisation of such tissue slides to enable both automatic processing as well as research studies to elucidate the underlying biological processes of cancer. The resulting images are of gigapixel size, rendering their computational analysis challenging. To deal with this issue, multiple instance learning (MIL) schemes based on weakly supervised training are used for WSI classification tasks. In such schemes, the WSI is typically divided into a grid of patches, with general purpose features derived from pretrained ImageNet [18] networks extracted for each patch. These representations are subsequently pooled together using different aggregation functions and attention-based operators for a final slide-level prediction.

State space models are designed to efficiently model long sequences, such as the sequences of patches that arise in WSI MIL. In this paper, we present the first use of state space models for WSI MIL. Extensive experiments on three publicly available datasets show the potential of such models for the processing of gigapixel-sized images, under both weakly and multi-task schemes. Moreover, comparisons with other commonly used MIL schemes highlight their robust performance, while we demonstrate empirically the superiority of state space models in processing the longest of WSI sequences with respect to commonly used MIL methods.

2   Related Work

Using pretrained networks for patch-wise feature extraction is a well established strategy for histopathology analysis [4,20]. An extension of this approach is with MIL, where the patch-wise features of an entire slide are digested simultaneously by an aggregator model, such as attention-based models CLAM [17] and TransMIL [19], the latter being a variant of self-attention transformers [21]. [3] proposes another transformer-based method in the form of a hierarchical ViT. Similar to our multitask experiments, [6] explores combining slide-level and tile-level annotations with a minimal point-based annotation strategy. One of the key components of MIL methods is the aggregation module that pools together the set of patch representations. Mean or max pooling operations are among the simplest and most effective for aggregating predictions over a whole slide [2]. In contrast, recurrent neural networks (RNN) with long short-term memory (LSTM) [14] model the patches more explicitly as a set of tokens in sequence. In particular, LSTM networks have been shown to work well in different MIL settings including both visual cognition [22] and computational pathology [1].

The state space model is a linear differential equation,

$$\dot{x}(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t) \tag{1}$$

that is widely studied in control theory, and describes a continuous time process for input and output signals $u(t) \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^q$, and state signal $x(t) \in \mathbb{R}^n$, and where the process is governed by matrices $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{B} \in \mathbb{R}^{n \times p}$, $\boldsymbol{C} \in \mathbb{R}^{q \times n}$, $\boldsymbol{D} \in \mathbb{R}^{q \times p}$. In HiPPO [9] (high-order polynomial projection operator), continuous time memorisation is posed as a problem of function approximation in a Hilbert space defined by a probability measure μ. For a *scaled Legendre* probability measure, one obtains the HiPPO matrix $\boldsymbol{A}$, which enforces uniform weight in the memorisation of all previously observed inputs, in contrast to the exponentially decaying weighting of the constant error carousel of LSTMs [14]. The HiPPO mode of memorisation is shown empirically to be better suited to modeling long-range dependencies (LRD) than other neural memory layers, for which it serves as a drop-in replacement.

Whereas in HiPPO, the state matrix $\boldsymbol{A}$ is a fixed constant, the linear state space layer (LSSL) [12] incorporates $\boldsymbol{A}$ as a learnable parameter. However, this increased expressiveness introduces intractable powers of $\boldsymbol{A}$. In [10], the LSSL is instead reparameterised as the sum of diagonal and low-rank matrices, allowing for the efficient computation of the layer kernel in Fourier space. This updated formulation is known as the *structured* state space sequence layer (S4). Note that as a linear operator, the inverse discrete Fourier transform is amenable to backpropagation in the context of a neural network. Note also that under this formulation, the hidden state $x(t)$ is only computed implicitly. Finally, [11] presents a simplification of the S4 layer, known as diagonal S4 (S4D), in which $\boldsymbol{A}$ is approximated by a diagonal matrix.

3 Method

Given that the patch extraction of whole slide images at high magnifications results in long sequences of patches, we propose to incorporate a state space layer in a MIL aggregation network to better represent each patch sequence.

3.1 Neural State Space Models

In practice, neural state space models (SSM) simulate Eq. 1 in discrete time, invoking a recurrence relation on the discretised hidden state,

$$
\begin{aligned}
x_t &= \overline{\boldsymbol{A}} x_{t-1} + \overline{\boldsymbol{B}} u_t \\
y_t &= \overline{\boldsymbol{C}} x_t + \overline{\boldsymbol{D}} u_t
\end{aligned}
\tag{2}
$$

where the sequences u_t, x_t, and y_t are the discretised $u(t)$, $x(t)$, and $y(t)$, and the modified model parameters arise from a bilinear discretisation [12]. As such, SSMs bear an inherent resemblance to RNNs, where the hidden representation x_t can be interpreted as a memory cell for the observed sequence over the interval $[0, t]$, and with $\overline{\boldsymbol{D}} u_t$ acting as a skip connection between the input and output at point t. Due to their lack of non-linearities, state space models can also be viewed as a convolution between two discrete sequences. Playing out the recurrence in Eq. 2, one obtains,

$$y = \overline{K} * u + \overline{D}u, \tag{3}$$

where $u \in \mathbb{R}^L$ and $y \in \mathbb{R}^L$ are the full input and output sequences, and the sequence $\overline{K} \in \mathbb{R}^L$ is defined as,

$$\overline{K} = (\overline{CB}, \overline{CAB}, \dots, \overline{CA}^{L-1}\overline{B}), \tag{4}$$

which is computed efficiently by the S4D algorithm [11]. Note that although SSM layers are linear, they may be combined with other, non-linear layers in a neural network. Note also that although Eq. 3 is posed as modeling a one-dimensional signal, in practice multi-dimensional inputs are modelled simply by stacking SSM layers together, followed by an affine "mixing" layer.
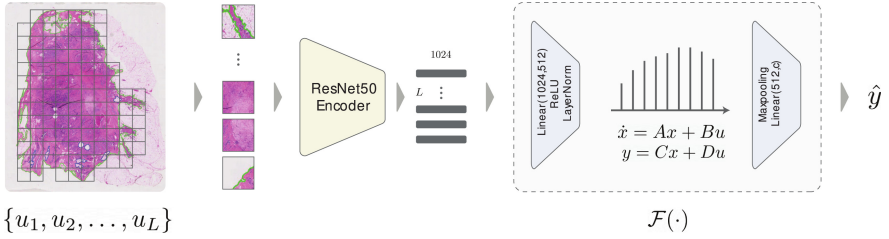

Fig. 1. Overview of the proposed pipeline. In the first step, patches are extracted from a regular grid on a WSI. These patches are embedded using a pre-trained ResNet50 and are aggregated by a sequence model based on a state space layer.

3.2 MIL Training

In our pipeline (Fig. 1) WSIs are first divided into a sequence of L patches $\{u_1, u_2, \dots, u_L\}$, where L will vary by slide. A pretrained ResNet50 is then used to extract a 1024-dimensional feature vector from each patch $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L\}$, which constitute the model inputs. We define a SSM-based neural network $\mathcal{F}$ to predict a WSI-level class probability given this input sequence,

$$\hat{y} = \mathcal{F}(\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L\}). \tag{5}$$

The architecture of $\mathcal{F}$ is composed of an initial linear projection layer, used to lower the dimensionality of each vector in the input sequence. A SSM layer is then applied feature-wise by applying the S4D algorithm. That is, Eq. 3, including the skip connection, transforms the sequence $\{u_{1,d}, u_{2,d}, \dots, u_{L,d}\}$ for all features d, and the resulting sequences are concatenated. A linear "mixing" layer is applied token-wise, doubling the dimensionality of each token, followed by a gated linear unit [5] acting as an output gate, which restores the input dimensionality. For

the SSM layer, we used the official implementation of S4D[1]. A max pooling layer merges the SSM layer outputs into a single vector, which is projected by a final linear layer and softmax to give the class probabilities $\hat{y}$. The model is trained according to,

$$\mathcal{L}_{MIL} = -\frac{1}{M} \sum_{m=1}^{M} \log \hat{y}_{c_m}, \tag{6}$$

where $\hat{y}_{c_m}$ denotes the probability corresponding to c_m, the slide-level label of the sequence corresponding to the mth of M whole slide images.

3.3   Multitask Training

One advantage of processing an entire slide as a sequence is the ease with which additional supervision may be incorporated, when available. A patch-level ground truth creates the opportunity for multitask learning, which can enhance the representations learned for slide-level classification. As an extension of our base model in Eq. 6, we train a multitask model to jointly predict a slide-level and patch-level labels. Prior to the max pooling layer of the base model, an additional linear layer is applied to each sequence token, yielding L additional model outputs. This multitask model is trained according to a sum of log losses,

$$\mathcal{L}_{MT} = -\frac{1}{M} \sum_{m=1}^{M} \left(\log \hat{y}_{c_m} + \frac{\lambda}{L} \cdot \sum_{l=1}^{L} \log \hat{y}_{c_{m,l}} \right), \tag{7}$$

where $c_{m,l}$ indexes the class of the lth patch in the mth training slide and λ is a tunable hyperparameter used to modulate the relative importance of each task.

3.4   Implementation Details

We extracted patches of size 256×256 from the tissue regions of WSIs at 20x magnification. Following CLAM [17], the third residual block of a pretrained ResNet50 [13] was used as a feature extractor, followed by a mean pooling operation, resulting in a 1024-dimensional representation for each patch. These features were used as inputs to all models. All model training was performed under a 10-fold cross-validation, and all reported results are averaged over the validation sets of the folds, aside from CAMELYON16, for which the predefined test set was utilized. Thus, for CAMELYON16, we report test set performances averaged over the validation.

Baseline models were chosen to be prior art CLAM [17] and TransMIL [19]. The official code of these two models was used to perform the comparison. In addition, we included a vanilla transformer, a LSTM RNN, and models based on mean and max pooling. Our vanilla transformer is composed of two stacked self-attention blocks, with four attention heads, a model dimension of 256, and

---

[1] https://github.com/HazyResearch/state-spaces.

a hidden dimension of 256. For the LSTM, we used an embedding size of 256 and a width of 256. The pooling models applied pooling feature-wise across each sequence, then used a random forest with 200 trees for classification. For the S4 models, the dimension of the state matrix A was tuned to 32 for CAMELYON16 and TCGA-RCC, and 128 for TCGA-LUAD. Our models were trained using the Adam [15] optimizer with the lookahead method [23], with a learning rate of $2 \cdot 10^{-4}$, and weight decay of 10^{-4} for TCGA-LUAD and TCGA-RCC and 10^{-3} for CAMELYON16. Early stopping with a patience of 10 was used for all our training. Our implementation is publicly available[2].

4   Experiments and Discussion

4.1   Data

CAMELYON16 [16] is a dataset that consists of resections of lymph nodes, where each WSI is annotated with a binary label indicating the presence of tumour tissue in the slide, and all slides containing tumors have a pixel-level annotation indicating the metastatic region. In multitask experiments, we use this annotation to give each patch a label indicating local tumour presence. There are 270 WSIs in the training/validation set, and 130 WSIs in the predefined test set. In our experiments, the average patch sequence length arising from CAMELYON16 is 6129 (ranging from 127 to 27444).

TCGA-LUAD is a TCGA lung adenocarcinoma dataset that contains 541 WSIs along with genetic information about each patient. We obtained genetic information for this cohort using Xena browser [7]. As a MIL task, we chose the task of predicting the patient mutation status of TP53, a tumor suppressor gene that is highly relevant in oncology studies. The average sequence length is 10557 (ranging from 85 to 34560).

TCGA-RCC is a TCGA dataset for three kidney cancer subtypes (denoted KICH, KIRC, and KIRP). It consists of 936 WSIs (121 KICH, 518 KIRC, and 297 KIRP). The average sequence length is 12234 (ranging from 319 to 62235).

4.2   Results

Multiple Instance Learning Results. We evaluate our method on each dataset by accuracy and area under receiver operating characteristic curve (AUROC). For multiclass classification, these were computed in a one-versus-rest manner.

Table 1 summarises the comparison between our proposed model and baselines. For the CAMELYON16 dataset, our method performs on par with Trans-MIL and the CLAM models, while it clearly outperforms the other methods. Similarly, in the TCGA-LUAD dataset the proposed model achieves comparable performance with both CLAM models, while outperforming TransMIL and the other methods. We note that TCGA-LUAD proves to be a more challenging

---

dataset for all models. Moreover, our method outperforms CLAM models on the
TCGA-RCC dataset, while reporting very similar performance with respect to
TransMIL. Overall, looking at the average metrics per model across all three
datasets, our proposed method achieves the highest accuracy and the second
highest AUROC, only behind CLAM-MB. A pairwise t-test between the pro-
posed method, CLAM, and TransMIL shows that there is no statistical signifi-
cance performance difference (see supplementary material).

We further compare our method with respect to model and time complexity.
In Table 2 we report the number of trainable parameters, as well as the inference
time for all models. The number of parameters is computed with all models con-
figured to be binary classifiers, and the inference time is computed as the average
time over 100 samples for processing a random sequence of 1024-dimensional vec-
tors of length 30000. For our proposed method, we report both models with the
different state dimensions (Ours (SSM_{32})) and (Ours (SSM_{128})). Compared

Table 1. Comparison of accuracy and AUROC on three datasets CAMELYON16,
TCGA-LUAD, TCGA-RCC, and on average. All metrics in the table are the average
of 10 runs. Best performing methods are indicated in **bold** and second best in *italics*.
* indicates results from [19].

Dataset	CAMELYON16		TCGA-LUAD		TCGA-RCC		Average	
Metric	Acc.	AUROC	Acc.	AUROC	Acc.	AUROC	Acc.	AUROC
Mean-pooling	0.5969	0.5810	0.6261	0.6735	0.8608	0.9612	0.6946	0.7386
Max-pooling	0.7078	0.7205	0.6328	0.6686	0.8803	0.9659	0.7403	0.7850
Transformer [21]	0.5419	0.5202	0.5774	0.6214	0.7932	0.9147	0.6375	0.6854
LSTM [8]	0.5310	0.5053	0.5389	0.5208	0.6654	0.7853	0.5784	0.6038
CLAM SB [17]	0.8147	0.8382	0.6859	*0.7459*	0.8816*	0.9723*	0.7941	0.8532
CLAM MB [17]	*0.8264*	*0.8523*	**0.6901**	**0.7573**	0.8966*	0.9799*	*0.8044*	**0.8632**
TransMIL [19]	**0.8287**	**0.8628**	0.6348	0.7015	**0.9466***	*0.9882**	0.8034	0.8508
Ours	0.8217	0.8485	*0.6879*	0.7304	*0.9426*	**0.9885**	**0.8174**	*0.8558*

Table 2. Comparison of parameter count and inference time for all methods.

Model	Number of parameters	Inference time (ms)
Mean-pooling	1 025	5.60
Max-pooling	1 025	77.49
Transformer [21]	1 054 978	2.60
LSTM [8]	789 250	320.52
CLAM SB [17]	790 791	0.84
CLAM MB [17]	791 048	5.85
TransMIL [19]	2 672 146	8.58
Ours (SSM_{128})	1 184 258	2.01
Ours (SSM_{32})	1 085 954	1.97

with TransMIL, our method runs four times faster and has less than half the parameters. The CLAM models are more efficient in terms of number of trainable parameters, yet CLAM MB is slower.

Table 3 shows the effect of modifying parts of the architecture on the results for TCGA-RCC. Most modifications had very little impact on AUROC, but a more significant impact can be seen on the accuracy of the model. Models A and B show that stacking multiple SSM layers results in lower accuracy, which was observed over all three datasets, while models C and D show that modifying the state dimension of the SSM module can have an impact on the accuracy. The optimal state space dimension varies depending on the dataset.

Multitask Learning Results. We explored the ability of our model to combine slide- and patch-level information on the CAMEYLON16 dataset. We compared our model with the best performing model on CAMELYON16, TransMIL. Both models were trained according to Eq. 7 with $\lambda = 5$ tuned by hand. In Table 4 we give slide-level accuracy and AUROC for the two models. We observe that all accuracies and AUROC increase compared with those reported in Table 1. This indicates that the use of patch-level annotations complements the learning of the slide-level label. We furthermore observe that our model outperforms TransMIL when combining slide- and patch-level annotations. We map the sequence of output probabilities to their slide coordinates giving a heatmap localising metastasis (see supplementary material).

Table 3. Ablation study for the different SSM components on the TCGA-RCC dataset. Best results in **bold**.

Model	SSM layers	State dimension	Accuracy	AUROC
A	2	32	0.9236	0.9813
B	3	32	0.9179	0.9834
C	1	16	0.9352	0.9846
D	1	64	0.9352	0.9861
Ours	1	32	**0.9426**	**0.9885**

Table 4. Comparison of accuracy and AUROC for models trained as multitask classifiers on the CAMELYON16 dataset. Best results in **bold**.

Model	Accuracy	AUROC
TransMIL [19]	0.8403	0.8828
Ours	**0.8488**	**0.8998**

Performance on Longest Sequences. In order to highlight the inherent ability of SSM models to effectively model long sequences, we performed an experiment on only the largest WSIs of the TCGA-RCC dataset. Indeed, this dataset contains particularly long sequences (up to 62235 patches at 20x). We evaluated the trained models for each fold on a subset of the validation set, only containing sequences with a length in the 85th percentile. Table 5 shows the obtained average accuracy (weighted by the number of long sequences in each validation set) and AUROC on both CLAM models, TransMIL, and our proposed method. Both in terms of AUROC and accuracy, our method outperforms the other methods on long sequences, while the performances are comparable to Table 1, albeit slightly lower, illustrating the challenge of processing large WSIs.

Table 5. Results of CLAM SB, CLAM MB, TransMIL, and our proposed method on long sequences. Best results in **bold**.

Model	Accuracy	AUROC
CLAM SB [17]	0.9149	0.9635
CLAM MB [17]	0.8936	0.9654
TransMIL [19]	0.9007	0.9652
Ours	**0.9220**	**0.9737**

5 Conclusions

In this work we have explored the ability of state space models to act as multiple instance learners on sequences of patches extracted from histopathology images. These models have been developed for their ability to memorise long sequences, and they have proven competitive with state of the art MIL models across a range of pathology problems. Additionally, we demonstrated the ability of these models to perform multiclass classification, which furthermore allowed us to visualise the localisation of metastasic regions. Finally, we demonstrated that on the longest sequences in our datasets, state space models offer better performance than competing models, confirming their power in modeling long-range dependencies.

References

1. Agarwalla, A., Shaban, M., Rajpoot, N.M.: Representation-aggregation networks for segmentation of multi-gigapixel histology images. In: British Machine Vision Conference (BMVC) (2017)

2. Campanella, G., et al.: Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. Nature Med. **25**(8), 1301–1309 (2019)

3. Chen, R.J., et al.: Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16144–16155 (2022)

4. Coudray, N., et al.: Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning. Nature Med. **24**(10), 1559–1567 (2018)

5. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: International Conference on Machine Learning, pp. 933–941. PMLR (2017)

6. Gao, Z., et al.: A semi-supervised multi-task learning framework for cancer classification with weak annotation in whole-slide images. Med. Image Anal. **83**, 102652 (2023)

7. Goldman, M.J., et al.: Visualizing and interpreting cancer genomics data via the Xena platform. Nat. Biotechnol. **38**(6), 675–678 (2020)

8. Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks, pp. 37–45 (2012)

9. Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C.: HiPPO: recurrent memory with optimal polynomial projections. Adv. Neural Inf. Process. Syst. **33**, 1474–1487 (2020)

10. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)

11. Gu, A., Gupta, A., Goel, K., Ré, C.: On the parameterization and initialization of diagonal state space models. arXiv preprint arXiv:2206.11893 (2022)

12. Gu, A., et al.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. Adv. Neural Inf. Process. Syst. **34**, 572–585 (2021)

13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

16. Litjens, G., et al.: 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. GigaScience **7**(6), giy065 (2018)

17. Lu, M.Y., Williamson, D.F., Chen, T.Y., Chen, R.J., Barbieri, M., Mahmood, F.: Data-efficient and weakly supervised computational pathology on whole-slide images. Nat. Biomed. Eng. **5**(6), 555–570 (2021)

18. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)

19. Shao, Z., Bian, H., Chen, Y., Wang, Y., Zhang, J., Ji, X., et al.: TransMIL: transformer based correlated multiple instance learning for whole slide image classification. Adv. Neural Inf. Process. Syst. **34**, 2136–2147 (2021)

20. Tellez, D., Litjens, G., van der Laak, J., Ciompi, F.: Neural image compression for gigapixel histopathology image analysis. IEEE Trans. Pattern Anal. Mach. Intell. **43**(2), 567–578 (2019)

21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30 (2017)

22. Wang, K., Oramas, J., Tuytelaars, T.: In defense of LSTMs for addressing multiple instance learning problems. In: Proceedings of the Asian Conference on Computer Vision (2020)

23. Zhang, M.R., Lucas, J., Hinton, G., Ba, J.: Lookahead optimizer: K steps forward, 1 step back. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA (2019)