Compressing Recurrent Neural Networks for FPGA-accelerated Implementation in Fluorescence Lifetime Imaging

Ismail Erbas*
Center for Modeling, Simulation, & Imaging in Medicine
Rensselaer Polytechnic Institute
Troy, NY 12206
erbasi@rpi.edu

Vikas Pandey*
Center for Modeling, Simulation, & Imaging in Medicine
Rensselaer Polytechnic Institute
Troy, NY
pandev2@rpi.edu

Aporva Amarnath
IBM T.J. Watson Research Center
Yorktown Heights, NY
aporva.amarnath@ibm.com

Naigang Wang
IBM T. J. Watson Research Center
Yorktown Heights, NY
nwang@us.ibm.com

Karthik Swaminathan
IBM T.J. Watson Research Center
Yorktown Heights, NY
kvswamin@us.ibm.com

Stefan T. Radev
Center for Modeling, Simulation, & Imaging in Medicine
Rensselaer Polytechnic Institute
Troy, NY
radevs@rpi.edu

Xavier Intes
Center for Modeling, Simulation, & Imaging in Medicine
Rensselaer Polytechnic Institute
Troy, NY
intesx@rpi.edu

Abstract

Fluorescence lifetime imaging (FLI) is an important technique for studying cellular environments and molecular interactions, but its real-time application is limited by slow data acquisition, which requires capturing large time-resolved images and complex post-processing using iterative fitting algorithms. Deep learning (DL) models enable real-time inference, but can be computationally demanding due to complex architectures and large matrix operations. This makes DL models ill-suited for direct implementation on field-programmable gate array (FPGA)-based camera hardware. Model compression is thus crucial for practical deployment for real-time inference generation. In this work, we focus on compressing recurrent neural networks (RNNs), which are well-suited for FLI time-series data processing, to en-

*Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

able deployment on resource-constrained FPGA boards. We perform an empirical evaluation of various compression techniques, including weight reduction, knowledge distillation (KD), post-training quantization (PTQ), and quantization-aware training (QAT), to reduce model size and computational load while preserving inference accuracy. Our compressed RNN model, Seq2SeqLite, achieves a balance between computational efficiency and prediction accuracy, particularly at 8-bit precision. By applying KD, the model parameter size was reduced by 98% while retaining performance, making it suitable for concurrent real-time FLI analysis on FPGA during data capture. This work represents a big step towards integrating hardware-accelerated real-time FLI analysis for fast biological processes.

1 Introduction

Non-invasive imaging methods that reliably capture dynamic cellular processes are key for disease detection, treatment monitoring, and therapy development [5, 26, 23]. Fluorescence lifetime imaging (FLI) has emerged as a powerful tool for this purpose, providing detailed insights into cellular and molecular activities by measuring fluorescence decay times. Unlike traditional intensity-based fluorescence imaging, FLI is unaffected by factors such as fluorophore concentration or excitation light intensity [13], making it suitable for investigating complex biological phenomena, including protein-protein interactions and ligand-target binding [5] in intact small animals.

To date, estimating fluorescence lifetime parameters from time-resolved data remains a computationally intensive task. Conventional approaches rely on post-processing methods after time-resolved data capture, which itself is highly time-consuming. As a result, FLI is not well-suited for applications requiring rapid inference, such as real-time monitoring of fast biological processes or fluorescence-guided surgery. Real-time processing is critical in these contexts, since immediate decisions are necessary. Deep learning models, particularly sequence-to-sequence (Seq2Seq) architectures based on Gated Recurrent Units (GRUs) [17], have shown promise in addressing the computational bottlenecks of FLI data analysis. These models are well-suited for handling time-series data and can provide faster and more efficient deconvolution of temporal point spread function (TPSF) signals. However, deploying recurrent models on hardware-constrained platforms, such as FPGAs, comes with unique implementational challenges due to limited memory and computational resources.

To address these challenges, the current paper focuses on compressing GRU-based Seq2Seq models for real-time FLI data processing on FPGAs by reducing memory usage and computational complexity. We explore quantization techniques for weight reduction, such as PTQ [7, 12, 24] and QAT [10, 8], to lower the precision of model precision, improving efficiency without a significant drop in accuracy. Additionally, we implement knowledge distillation [19, 18, 14] to further compress the model by transferring knowledge from a larger model to a smaller one without compromising performance.

2 Background

FLI operates through time-resolved imaging, where a fluorescent sample is excited with a short pulse of light and the emitted fluorescence signal is captured over time using time-resolved/time-gated detectors such as time-correlated single photon counting (TCSPC), time-gated instensified charge-couple device (ICCD) [25], and single photon avalanche diode (SPAD) [2]. This process generates the TPSF, representing photon-arrival temporal distribution after the excitation pulse (see Figure 1). The TPSF is distorted by the Instrument Response Function (IRF), representing the temporal response of the imaging system on the delta input signal. Hence, the observed TPSF can be modeled as a convolution of the sample fluorescence decay (SFD) and the IRF [3]. In biological samples, multiple fluorophore components or changes in local environmental conditions can alter the fluorescence decay rate. These variations contribute to the SFD, which is modeled as the sum of the exponential decay of each fluorophore component.

Despite the advantages of FLI, accurately extracting SFD is computationally expensive. Traditional approaches, such as nonlinear least-squares fitting, center-of-mass, and maximum likelihood estimation methods, require significant computational resources and depend on initial guess parameters[1]. These iterative computation methods require IRF estimation prior to the re-convolution and fitting approach. Hence, they are less practical for fast FLI parameter estimation; DL methods are slowly

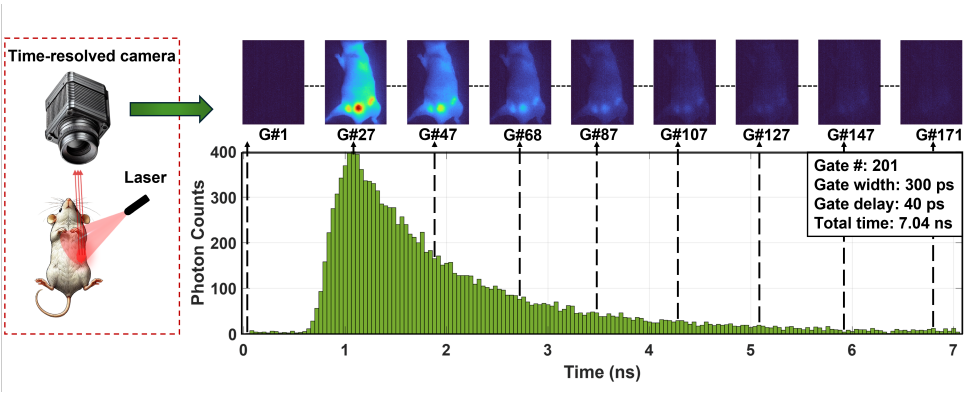

Figure 1: **Experimental set-up and data**. Schematic illustration of fluorescence lifetime imaging (FLI), time-resolved data capture, and the temporal point spread function (TPSF). The top-right panel shows experimental time-resolved fluorescence images of HER2+ tumor xenografts labeled with Alexa Fluor 700 conjugated to Trastuzumab in a nude mouse, captured at different time gates. The bottom-right panel presents the corresponding TPSF for a single pixel.

replacing these computational methods[21, 6, 15]. Recently, recurrent neural networks (RNNs), specifically GRU networks [4], have demonstrated potential in deconvolving SFD from the TPSF without the need of IRF, allowing for faster and more accurate fluorescence lifetime parameter extraction [17]. GRUs excel at handling sequential time-series data and reduce the computational complexity compared to traditional methods, making real-time analysis feasible.

This deconvolution method [17], integrated in the camera FPGA, can be used for simultaneous temporal data capture and deconvolution operation, which can be a step towards real-time FLI parameter analysis. Such a hardware-accelerated camera will be suitable for clinical diagnostics and dynamic biological process monitoring, where rapid, accurate processing of FLI data is important. However, integration of this DL model architecture into FPGA is highly challenging due to its size and complexities of computation operation. One promising approach, as demonstrated by [11], involves integrating RNNs into SPAD camera systems to improve the speed and accuracy of FLI data processing. The proposed method bypasses conventional histogram-based techniques by directly estimating fluorescence lifetimes from raw photon timestamps using GRU and LSTM models. While the system, using a 32×32 pixel SPAD camera, achieved real-time processing with up to 4 million photons per second and frame rates of 10 frames per second, the lower resolution of the camera limited the overall data volume.

Deploying deep learning model on large-format time resolved detector such as the SwissSPAD2 (SS2) [2] introduces new challenges due to the increased data volume, which places additional demands on FPGA for memory and computation. The SS2 camera with its large-format 512 x 512 time-gated SPAD detector arrays, sub-10-cps dark count rate (DCR) per pixel and 50% maximum photon detection probability (PDP) has been shown [20] most suitable for fast FLI applications.

To address these challenges and constraints, DL model compression techniques including quantization methods become highly important [9]. Quantization reduces the precision of model parameters from floating-point to lower-precision fixed-point formats. This reduces the memory footprint and computational demand, making the models more suitable for FPGA deployment [22]. Quantization can be performed through two main approaches: PTQ and QAT. PTQ involves applying quantization to a pre-trained model without further training, offering a quick but sometimes less accurate solution [7, 12, 24]. In contrast, QAT incorporates quantization during the training process, allowing the model to adapt to reduced precision and maintain higher accuracy, albeit at the cost of additional training time [10, 8]. Another key model compression strategy is knowledge distillation [19, 18, 14], where a smaller, more computation efficient model (student) is trained to replicate the performance of a larger, more complex model (teacher). The student model learns from the original training data as well as teacher model's output, that is helpful to capture key data features while reducing computational requirements. This approach is particularly valuable for deploying deep learning models on hardware-constrained platforms like FPGAs, where maintaining accuracy while reducing model size is critical for FLI applications.

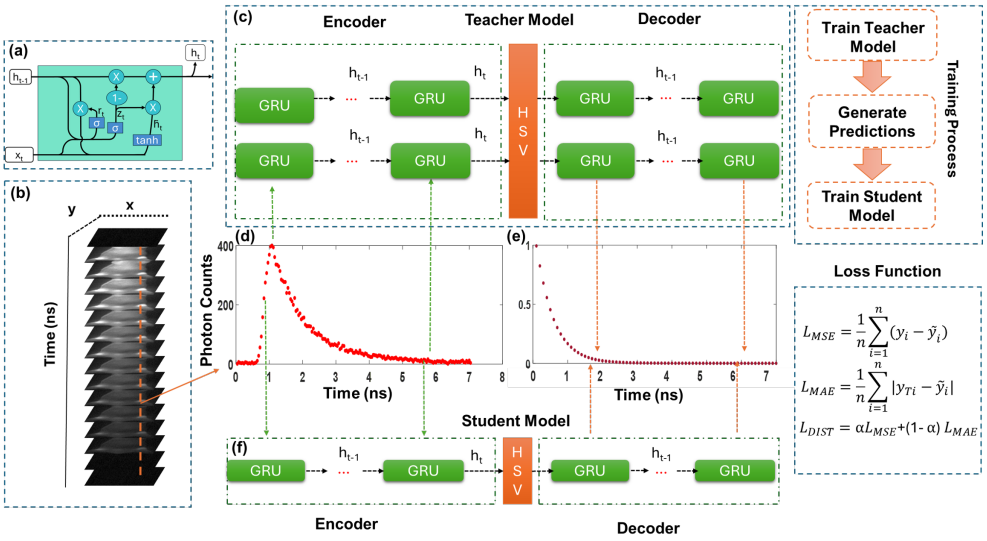

Figure 2: **Model and training setup**. (a) Gated Recurrent Unit (GRU); (b) Time-resolved fluorescence images; (c) Deep GRU-based encoder-decoder architecture (teacher), trained for TPSF deconvolution to pixel-wise SFD from (b), with the resulting deconvolved SFDs shown in (e); (f) Single-layer encoder-decoder RNN model (Student), derived from (c) using the knowledge distillation (KD) method. The stack of time-resolved fluorescence images (b) and deconvolved SFDs (e) are used to train (f). The student model learns hidden features using a combined loss function.

In this work, we explore optimization techniques such as quantization and weight size reduction for the deployment of efficient FLI deconvolution on large-format time resolved detector using an FPGA board. By optimizing these processes, we aim to enhance the performance of real-time FLI in resource-constrained environments, broadening its potential applications in both biomedical research and clinical diagnostics.

3 Methods

3.1 Synthetic data

To train and validate GRU-based model for FLI, we generated synthetic data simulating time-resolved SFDs. Fluorescence decays were modeled using a bi-exponential function:

$$f(t) = A_R \exp\left(-\frac{t}{\tau_1}\right) + (1 - A_R) \exp\left(-\frac{t}{\tau_2}\right) + \epsilon_t, \tag{1}$$

where $\tau_1 \in [0.2, 0.8]$ and $\tau_2 \in [0.8, 1.5]$ represent short and long lifetime components typically observed in near-infrared (NIR) applications (measured in nanoseconds; ns), respectively, and $A_R \in [0, 1]$ denotes the amplitude fraction. The residual term ϵ_t denotes system-generated Poisson-distributed noise. We used the MNIST dataset to create 28×28 pixel images with simulated fluorescence decays assigned to each pixel. To mimic the experimental conditions, we convolved these decays with pixel-wise instrument response functions (IRFs) obtained by illuminating a diffused white paper with a 700 nm laser and capturing the reflected light through a neutral density filter. A total of 200 synthetic image sets were generated, yielding 3,920,000 TPSFs for training.

3.2 Experimental data

For experimental validation, we imaged a $10M$ solution of Alexa Fluor 700 (AF700) dye in PBS using the large-format time resolved detector. AF700 is a mono-exponential NIR dye with excitation and emission maxima at 702nm and 723nm, respectively, and a fluorescence lifetime of approximately

Seq2Seq model size	(64×64)	(64×32)	(64×16)	(45×45)	(32×32)	(16×16)
RMSE	0.071±0.01	0.084±0.01	**0.032±0.001**	0.063±0.01	0.1±0.01	0.1±0.01
R^2 **Score**	0.86±0.02	0.81±0.03	**0.96±0.01**	0.89±0.02	0.74±0.04	0.76±0.03
L2 norm	0.60±0.05	0.71±0.05	**0.32±0.04**	0.53±0.05	0.83±0.05	0.79±0.05
DTW distance	0.49±0.01	0.47±0.03	**0.35±0.02**	0.46±0.02	0.55±0.04	0.48±0.02

Table 1: Performance metrics for different Seq2Seq model configurations on experimental data.

1ns in PBS. The dye was excited at 700nm, and emission was collected using a 740 ± 10nm filter. The collected TPSFs were used to evaluate the model's performance on real-world data.

3.3 Model development and optimization

Seq2Seq model We implemented the GRU-based sequence-to-sequence (Seq2Seq) architecture from [16] for fast estimation of SFDs. Figure 2 The encoder consists of two GRU layers, each with 128 hidden units. The input to the encoder is the TPSF sequence for each pixel. Each GRU cell sequentially processes an input time point x_t and the hidden state h_{t-1}, producing an output y_t and an updated hidden state h_t. The decoder mirrors the encoder architecture, generating the output SFD sequences from the encoded representation. A final linear dense layer refines the output. The model was trained using the Adam optimizer with a learning rate of 0.001, with mixed loss function. All the models were trained using Windows 11 system with Intel i9-13900K CPU and Nvidia RTX 4090.

Weight reduction and model quantization To enable deployment on resource-constrained hardware, we explored weight reduction and quantization techniques to reduce the model size and computational complexity. We experimented with various configurations of hidden units in the GRU layers, including sizes of 128, 64, 45, 32, and 16. This resulted in weight matrices of sizes 128×128, 128×64, 64×32, 64×16, 45×45, 32×32, and 16×16. We trained and tested these models to find the optimal settings within a reasonable error margin. For model quantization, we applied PTQ, reducing the precision of the weights from 32-bit floating-point to 16-bit and 8-bit integers. The quantization process is defined as $q = \text{round}\,(x/s)$, where x is the floating-point weight, q is the quantized integer value, and s is the scale factor determined by the range of the floating-point weights, typically calculated as $s = \max(|x|) \,/\, \left(2^b - 1\right)$ where b represents the bit-width of the target precision (e.g., 8-bit or 16-bit). This approach ensures the quantized values maintain the highest possible precision within the given range.

Knowledge distillation To further reduce model complexity, we developed a simplified model called Seq2SeqLite, which consists of a single GRU layer in both the encoder and decoder. We experimented with hidden unit sizes of 128, 64, 32, and 16. During the training process, QAT was applied to the Seq2SeqLite model. QAT incorporates quantization effects by simulating reduced-precision arithmetic during training, allowing the model to adapt effectively to quantization. This approach helps maintain accuracy after quantization to 16-bit or 8-bit precision. In addition to QAT, we also employed KD to compensate for the reduced model capacity. In this setup, the Seq2Seq model serves as the teacher, while Seq2SeqLite acts as the student. The student model was trained to minimize a combined loss function as shown in Figure 2.

4 Empirical Evaluation

All models were evaluated using RMSE, R^2 score, L2 norm, and DTW distance. Together, these metrics provide a comprehensive assessment of model performance in terms of accuracy, error magnitude, and temporal consistency. All models were trained on the simulated data and evaluated on the experimental well plate data, aggregating the metrics across all pixels.

4.1 Weight reduction results with Seq2Seq model

The results of the weight reduction experiments for the Seq2Seq model using 32-bit floating point precision are shown in the Table 1, illustrating the performance across various sizes, including (128×64), (64×64), (64×32), (64×16), (45×45), (32×32), and (16×16). RMSE values range

from 0.0374 to 0.0995, with the (64×16) model showing the lowest error. The R^2 score is highest at 0.96 for the (64×16) configuration, while more compressed models like (32×32) and (16×16) show lower performance. The L2 norm is smallest with the (64×16) model, whereas compressed models exhibit larger norms. DTW distance is lowest for the (64×16) model and highest for the (32×32) model. Overall, these results highlight the trade-offs between model size and performance, with the (64×16) configuration consistently achieving superior results across most metrics.

4.2 Quantization results

Seq2Seq Models (16-bit and 8-bit)							
Type	**Metrics**	$\mathbf{64 \times 64}$	$\mathbf{64 \times 32}$	$\mathbf{64 \times 16}$	$\mathbf{45 \times 45}$	$\mathbf{32 \times 32}$	$\mathbf{16 \times 16}$
16-bit	**RMSE**	0.071±0.01	0.084±0.01	**0.032±0.001**	0.063±0.01	0.1±0.01	0.095±0.01
	R^2 **Score**	0.87±0.02	0.81±0.03	**0.96±0.01**	0.89±0.02	0.74±0.04	0.76±0.03
	L2 norm	0.59±0.05	0.71±0.05	**0.31±0.04**	0.53±0.05	0.83±0.05	0.79±0.05
	DTW distance	0.49±0.01	0.47±0.03	**0.34±0.02**	0.46±0.02	0.55±0.04	0.48±0.02
8-bit	**RMSE**	0.063±0.01	0.089±0.01	**0.032±0.001**	0.071±0.01	0.105±0.01	0.089±0.01
	R^2 **Score**	0.89±0.02	0.79±0.03	**0.97±0.01**	0.87±0.02	0.70±0.04	0.79±0.03
	L2 norm	0.54±0.05	0.74±0.05	**0.27±0.04**	0.58±0.05	0.88±0.05	0.75±0.05
	DTW distance	0.69±0.02	0.72±0.04	**0.52±0.03**	0.53±0.02	0.61±0.04	0.44±0.02

Table 2: Performance metrics for Seq2Seq models (16-bit and 8-bit) across various configurations on experimental data.

The performance of different Seq2Seq model architectures under PTQ with 16-bit and 8-bit precision was evaluated as shown in Table 2. The models assessed include configurations such as (64×64), (64×32), (64×16), (45×45), (32×32), and (16×16). For the 16-bit models, the (64×16) architecture achieved the lowest RMSE of 0.032, along with an R^2 score of 0.96 and a relatively low L2 norm of 0.31, indicating efficient weight scaling. The model also exhibited a low DTW distance of 0.34, making it suitable for applications where both accuracy and computational efficiency are required. In contrast, the (32×32) model, while showing reasonable performance with an R^2 score of 0.74, demonstrated higher RMSE and L2 norm values, indicating larger errors and weight magnitudes. In the 8-bit setting, the (64×16) model continued to perform well, with an RMSE of 0.032 and an R^2 score of 0.97. The L2 norm decreased slightly to 0.27, suggesting that the 8-bit quantization did not introduce significant degradation in performance. While DTW distance increased to 0.52, the model maintained competitive performance compared to other configurations. Overall, 8-bit quantized models showed a slight increase in RMSE and DTW distance compared to their 16-bit counterparts, but still performed competitively. The (64×16) configuration consistently delivered strong results across both precision levels, making it a favorable choice for memory-constrained deployment scenarios.

The results for the quantized Seq2SeqLite models, focused on memory-efficient architectures, are summarized in Table 3, comparing different configurations under 16-bit and 8-bit quantization, both with and without KD. For the 16-bit models, the (32×32) architecture with KD achieves the lowest RMSE (0.068), while the (16×16) model shows moderate improvement with KD, reducing its RMSE from 0.088 to 0.072. R^2 scores indicate better fit when KD is applied, with the (32×32) model improving from 0.73 without KD to 0.88 with it, and similar trends are observed for the (16×16) architecture. Metrics such as L2 norm and DTW distance also improve with KD. In the 8-bit setting, the impact of KD is more pronounced. The (32×32) model with KD achieves an RMSE of 0.01, a significant improvement over the non-KD configuration (0.138). Similarly, the (16×16) model shows improved performance with KD, reducing RMSE from 0.13 to 0.047. R^2 scores and L2 norms confirm the overall benefit of KD across the architectures.

In summary, applying KD improves performance across all architectures in both 16-bit and 8-bit quantized models, particularly in terms of the RMSE, R^2, L2 norm, and DTW metrics, making them more suitable for deployment in memory-constrained environments.

Seq2SeqLite Models (16-bit and 8-bit)						
Type	**Metrics**	128×128	32×32	32×32 **w/ KD**	16×16	16×16 **w/ KD**
16-bit	**RMSE**	0.03±0.000	0.1±0.01	**0.068±0.01**	0.088±0.01	0.072±0.01
	R^2 **Score**	0.98±0.01	0.73±0.39	**0.88±0.29**	0.79±0.04	0.86±0.03
	L2 norm	0.25±0.04	0.84±0.05	**0.57±0.06**	0.74±0.06	0.60±0.06
	DTW distance	0.37±0.03	0.50±0.04	**0.44±0.04**	0.49±0.03	0.45±0.04
8-bit	**RMSE**	0.12±0.01	0.138±0.02	**0.01±0.001**	0.13±0.01	0.047±0.01
	R^2 **Score**	0.62±0.08	0.49±0.11	**0.99±0.01**	0.55±0.08	0.94±0.02
	L2 norm	1.00±0.10	1.15±0.12	**0.10±0.02**	1.09±0.09	0.39±0.06
	DTW distance	0.61±0.07	0.71±0.08	**0.52±0.05**	0.71±0.06	0.55±0.03

Table 3: Performance metrics for Seq2SeqLite quantized models (16-bit and 8-bit) with and without KD on experimental data.

5 Conclusion

In this work, we compressed DL models for real-time FLI data processing on resource-constrained hardware, such as FPGAs. By focusing on GRU-based Seq2Seq architecture, we explored model weight reduction and quantization techniques to enable efficient deployment of these models in FPGAs. Specifically, we applied PTQ and QAT to reduce the precision of model parameters to 16-bit and 8-bit formats. Furthermore, we incorporated KD as a model compression strategy to reduce computational complexity while maintaining high performance. The (32×32) Seq2SeqLite model with KD demonstrated an optimal balance between model size and performance in both 16-bit and 8-bit quantized versions. The 8-bit version, in particular, showed strong suitability for real-time applications, making it ideal for FPGA deployment. These improvements make the (32×32) KD model particularly suited for FPGA deployment in clinical and research settings where both memory efficiency and computational speed are critical.

References

[1] Wolfgang Becker. Fluorescence lifetime imaging–techniques and applications. *Journal of microscopy*, 247(2):119–136, 2012.

[2] Claudio Bruschini, Harald Homulle, Ivan Michel Antolovic, Samuel Burri, and Edoardo Charbon. Single-photon avalanche diode imagers in biophotonics: review and outlook. *Light: Science & Applications*, 8(1):87, 2019.

[3] Sez-Jade Chen, Nattawut Sinsuebphon, Alena Rudkouskaya, Margarida Barroso, Xavier Intes, and Xavier Michalet. In vitro and in vivo phasor analysis of stoichiometry and pharmacokinetics using short-lifetime near-infrared dyes and time-gated imaging. *Journal of biophotonics*, 12(3):e201800185, 2019.

[4] Kyunghyun Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] Ruslan I Dmitriev, Xavier Intes, and Margarida M Barroso. Luminescence lifetime imaging of three-dimensional biological objects. *Journal of Cell Science*, 134(9):1–17, 2021.

[6] Ismail Erbas, Vikas Pandey, Navid Ibtehaj Nizam, Nanxue Yuan, Amit Verma, Margarida Barroso, Xavier Intes, et al. Transformer-based deep learning model for fluorescence lifetime parameter estimations using pixelwise instrument response function. *Research Square*, 2024.

[7] Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

[8] Ekaterina Govorkova, Ema Puljak, Thea Aarrestad, Thomas James, Vladimir Loncar, Maurizio Pierini, Adrian Alan Pol, Nicolò Ghielmetti, Maksymilian Graczyk, Sioni Summers, et al. Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics

detection at 40 mhz at the large hadron collider. *Nature Machine Intelligence*, 4(2):154–161, 2022.

[9] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[10] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30, 2018.

[11] Yang Lin, Paul Mos, Andrei Ardelean, Claudio Bruschini, and Edoardo Charbon. Coupling a recurrent neural network to spad tcspc systems for real-time fluorescence lifetime imaging. *Scientific Reports*, 14(1):3286, 2024.

[12] Johannes Maly and Rayan Saab. A simple approach for quantizing neural networks. *Applied and Computational Harmonic Analysis*, 66:138–150, 2023.

[13] J Sven D Mieog, Friso B Achterberg, Aimen Zlitni, Merlijn Hutteman, Jacobus Burggraaf, Rutger-Jan Swijnenburg, Sylvain Gioux, and Alexander L Vahrmeijer. Fundamentals and developments in fluorescence-guided cancer surgery. *Nature reviews Clinical oncology*, 19(1):9–22, 2022.

[14] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *International Conference on Learning Representations*, 2018.

[15] Navid Ibtehaj Nizam, Vikas Pandey, Ismail Erbas, Jason T Smith, and Xavier Intes. A novel technique for fluorescence lifetime tomography. *bioRxiv*, pages 2024–09, 2024.

[16] Vikas Pandey, Ismail Erbas, Xavier Michalet, Arin Ulku, Claudio Bruschini, Edoardo Charbon, Margarida Barroso, and Xavier Intes. Deep learning-based temporal deconvolution for photon time-of-flight distribution retrieval. 2024.

[17] Vikas Pandey, Ismail Erbas, Xavier Michalet, Arin Ulku, Claudio Bruschini, Edoardo Charbon, and Xavier Intes. Temporal point spread function deconvolution in time-resolved fluorescence lifetime imaging using deep learning model. pages OM1D–4, 2024.

[18] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

[19] Sungho Shin, Yoonho Boo, and Wonyong Sung. Knowledge distillation for optimization of quantized deep neural networks. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6. IEEE, 2020.

[20] Jason T Smith, Alena Rudkouskaya, Shan Gao, Juhi M Gupta, Arin Ulku, Claudio Bruschini, Edoardo Charbon, Shimon Weiss, Margarida Barroso, Xavier Intes, et al. In vitro and in vivo nir fluorescence lifetime imaging with a time-gated spad camera. *Optica*, 9(5):532–544, 2022.

[21] Jason T Smith, Ruoyang Yao, Nattawut Sinsuebphon, Alena Rudkouskaya, Nathan Un, Joseph Mazurkiewicz, Margarida Barroso, Pingkun Yan, and Xavier Intes. Fast fit-free analysis of fluorescence lifetime imaging via deep learning. *Proceedings of the National Academy of Sciences*, 116(48):24019–24030, 2019.

[22] Xiao Sun, Naigang Wang, Chia-Yu Chen, Jiamin Ni, Ankur Agrawal, Xiaodong Cui, Swagath Venkataramani, Kaoutar El Maghraoui, Vijayalakshmi Viji Srinivasan, and Kailash Gopalakrishnan. Ultra-low precision 4-bit training of deep neural networks. *Advances in Neural Information Processing Systems*, 33:1796–1807, 2020.

[23] Amit Verma, Catherine Sherry, Nanxue Yuan, Vikas Pandey, John Williams, Xavier Intes, and Margarida M Barroso. Using meditope-based antibody labeling to improve fluorescence lifetime fret imaging. In *Multiphoton Microscopy in the Biomedical Sciences XXIV*, page PC128470S. SPIE, 2024.

[24] Naigang Wang, Chi-Chun Charlie Liu, Swagath Venkataramani, Sanchari Sen, Chia-Yu Chen, Kaoutar El Maghraoui, Vijayalakshmi Viji Srinivasan, and Leland Chang. Deep compression of pre-trained transformer models. *Advances in Neural Information Processing Systems*, 35:14140–14154, 2022.

[25] Nanxue Yuan, Vikas Pandey, Xavier Michalet, and Xavier Intes. Experimental study of fluorescence lifetime uncertainty in time-gated iccd-based macroscopic fluorescence lifetime imaging. In *Clinical and Translational Biophotonics*, pages TM5B–4. Optica Publishing Group, 2024.

[26] Nanxue Yuan, Vikas Pandey, Amit Verma, John C Williams, Xavier Intes, and Margarida Barroso. Antibody-target binding quantification in living tumors using macroscopy fluorescence lifetime forster resonance energy transfer imaging (mfli fret). In *Visualizing and Quantifying Drug Distribution in Tissue VIII*, volume 12821, pages 17–20. SPIE, 2024.