CATS: Combined Activation and Temporal Suppression for Efficient Network Inference

Zeqi Zhu[*†] Arash Pourtaherian[*] Luc Waeijen[*] Ibrahim Batuhan Akkaya[*]
Egor Bondarev[†] Orlando Moreira[*]
[*]GrAI Matter Labs B.V. [†]Eindhoven University of Technology
{zzhu, apourtaherian, lwaeijen, iakkaya, omoreira}@graimatterlabs.ai, {z.zhu, e.bondarev}@tue.nl

Abstract

*Brain-inspired event-driven processors execute deep neural networks (DNNs) in a sparsity-aware manner, leading to superior performance compared to conventional platforms. In the pursuit of higher event sparsity, prior studies suppress non-zero events by either eliminating the intra-frame activations (spatially) or leveraging the redundancy in the inter-frame differences for a video (temporally). However, we have empirically observed that simultaneously enhancing activation and temporal sparsity can lead to a synergistic suppression outcome. To this end, we propose an end-to-end event suppression training approach **CATS** —— **C**ombined **A**ctivation and **T**emporal **S**uppression for efficient network inference. It utilizes a gradient-based method to search for the optimal temporal thresholds per layer while penalizing the presence of events in both spatial and temporal domains. Our experimental results show that CATS achieves* $2 \sim 6\times$ *higher event suppression compared to the inherent ReLU suppression across a wide range of vision applications, consistently outperforming the state-of-the-art (SOTA) methods by a significant margin at all accuracy levels. Furthermore, a case study on the commercial event-driven processor GrAI-VIP highlights that the induced event sparsity in SSD on the EgoHands dataset can be efficiently translated into a performance enhancement of* $2.5\times$ *in FPS,* $2.1\times$ *in latency, and* $3.8\times$ *in energy consumption, while maintaining the model accuracy.*

1. Introduction

Deep Neural Networks (DNNs) dominate AI applications but face a mismatch with industrial demands. While they succeed in academia, increasing network complexity for better performance [6, 37] conflicts with industrial requirements for low resource consumption, power efficiency, and minimal latency. Novel AI computing architectures are emerging to address these challenges. An innovative paradigm gaining attention is the event-driven architecture [2, 18, 28, 29, 32, 11, 12, 27, 19], inspired by the brain's operational principles. This approach seeks to ease computational load by capitalizing on DNN sparsity.

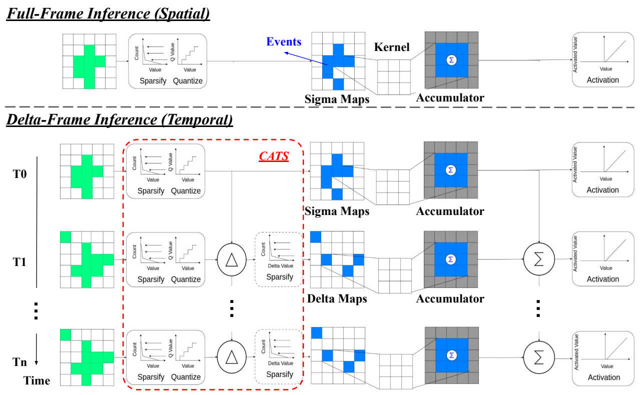

Figure 1: Two execution modes for DNN inference on hardware: Full-Frame Inference and Delta-Frame Inference.

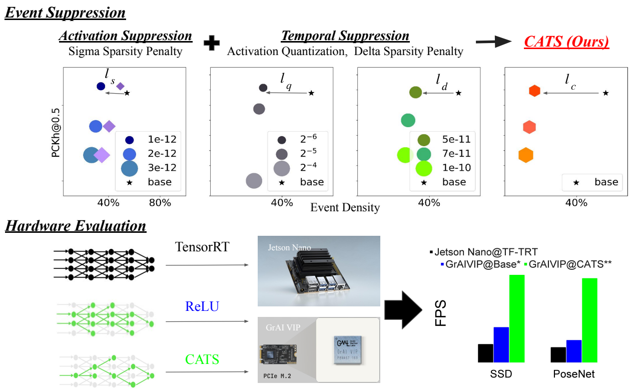

Figure 2: Combining activation suppression and temporal suppression (CATS) results in an accumulated effect on event suppression ($l_c \gg l_s, l_q, l_d$), leading to a significant performance boost on the event-driven processor [28, 33].

An essential distinction between event-driven processors and conventional hardware (e.g., GPUs, CPUs), lies in their approaches to DNN processing. Conventional hardware is limited to full-frame inference, necessitating the reprocessing of each frame within a video. In contrast, event-driven processors can perform either full-frame (spatial) or delta-frame (temporal) inference, as depicted in Fig. 1. This capability is achieved through the utilization of sigma-delta ($\Sigma - \Delta$) modulation [9, 21] to solely communicate changes in activations across video frames.

In event-driven processing, an event can be defined as either a non-zero activation in full-frame inference or an activation change in delta-frame inference. In both inference modes, activated neurons trigger event generation with values like integer or floating point. These events access the weight memory and execute the convolutions asynchronously [5]. The removal of an event can result in the elimination of accompanying memory accesses and multiply-accumulate (MAC) computations. A pivotal optimization lies in the deliberate suppression of events within the inputs of computational-heavy layers (e.g., Conv2D), thereby resulting in a diminished event density (i.e., percentage of events). This optimization process, known as event suppression, has been shown to contribute to latency and energy savings on real-world silicons [39, 40, 36], with both theoretical and experimental supports provided in Appendix A. Thus, the development of an efficient event suppression approach is imperative for enhancing the overall efficiency of event-driven processing

Prior research [7, 25] has indicated that videos captured by stationary cameras exhibit notable content correlation in the background, leading to substantial temporal sparsity during delta-frame inference. However, this temporal sparsity exploration becomes challenging when dealing with significant camera motion (see Fig. 4). To address this issue, some recent studies [9, 36, 21] attempt to lower the full-frame resolution by quantizing the activation outputs, and thereby augmenting the temporal correlation between successive frames. Nevertheless, searching for the optimal quantization scale (temporal threshold) per layer is challenging due to several factors: the layerwise threshold fine-tuning incurs a high manual cost [21], the rounding operation in quantization renders the thresholds untrainable [9], and the floating-point thresholds are not hardware-efficient [36].

Thus, we propose a novel method called Differentiable Temporal Threshold Search (DTTS) to learn power-of-two thresholds for layers during training. Inspired by differentiable neural architecture search [23, 34], we formulate the search space for temporal thresholds using a super net. Through the inclusion of a delta sparsity penalty in the training loss, the optimal threshold distribution is trained by utilizing gradient-based optimization methods like SGD. During inference, only the optimal threshold in the distribution is sampled as the quantization scale on activation, thereby enhancing the temporal sparsity. In contrast to recent work [36], our approach introduces distinct intervals between candidate thresholds to prevent local optima in the quantization bit selection. Moreover, our utilization of power-of-two thresholds enhances hardware compatibility.

In addition to temporal suppression, activation suppression [14, 22, 40] is also capable of inhibiting a significant portion of events in the spatial domain, while leading to an amplified suppression effect in the temporal domain. As shown in Fig. 2, we empirically find that the combined activation suppression and temporal suppression yield greater event sparsity compared to any single-domain event suppression approach. Our results demonstrate that activation suppression plays a crucial role in achieving comprehensive event suppression, especially when models and datasets exhibit notable spatial redundancy. This insight hasn't been addressed in prior studies paying attention solely to temporal suppression [9, 36, 25, 16, 13]. Therefore, we integrate our temporal suppression approach DTTS with a modified state-of-the-art activation suppression approach STAR [40] in an end-to-end training. It considers the learning of temporal thresholds and the penalization of events simultaneously in both intra-frame activations (sigma) and inter-frame activation differences (delta).

The contributions of this paper are as follows:

1. **A novel event suppression training method (CATS)** combines activation and temporal suppression to minimize event occurrences during delta-frame inference.
2. **An efficient differentiable temporal threshold search approach (DTTS)** mitigates the learning efforts for layerwise hardware-friendly thresholds 2^n.
3. **A case study on the commercial event-driven processor GrAI-VIP** (refer to Appendix B) illustrates that CATS substantially boosts the on-chip performance of various DNN models. Remarkably, CATS coupled with GrAI-VIP exhibits a superior performance over the TensorRT-enhanced edge-GPU (Jetson Nano).

The remainder of this paper is structured as follows. Sec. 2 gives an overview of the related work. Sec. 3 presents the quantized sigma-delta network for efficient video processing. Sec. 4 provides a detailed description of our optimization approach, CATS. Sec. 5 presents the experimental setup and results. Sec. 6 concludes the paper.

2. Related Work

2.1. Activation Suppression

Activation suppression inhibits the spiking of redundant events in the sigma maps during full-frame inference, differing from static weight pruning [17] as it is dynamic and input-dependent. Various activation suppression techniques [14, 22, 39, 40] have been developed to enhance the activation sparsity in DNNs. Georgiadis [14] and Kurtz et al. [22] introduce a sparsity penalty on the activation outputs of network and update the weights via the penalty gradients to induce activation sparsity. Zhu et al. [39] follow the idea of sparsity penalty and devise an adaptive training schedule to efficiently adjust the weight of sparsity penalty in the training loss. Zhu et al. [40] further improve the performance of activation suppression and accuracy recovery by solely penalizing and thresholding those low-magnitude activations, while preserving the learning of the large-magnitude ones. Consequently, recent research have delved into achieving $70\% \sim 80\%$ sparsity in the spatial domain. However, pushing for more aggressive suppression may risk causing irrecoverable drops in accuracy.

2.2. Temporal Suppression

Temporal suppression decreases the event firing of repetitive feature contents in the adjacent video frames, and is distinct from activation suppression, which focuses solely on redundant contents within frames. Several recent approaches consider temporal redundancy for efficient video inference. Skip-convolution networks (Skip-Conv) [16] reuse activation values that have not changed significantly between frames, but require frequent re-initialization to maintain model quality, leading to reduced efficiency on hardware, especially with a moving camera. DeltaCNN [25] and EvNets [13] address this issue by incorporating long-term changes through sigma-delta modulation. Unlike our method, they truncate small delta values and incur additional memory costs for the mask, limiting the efficiency of near-memory computing. Moreover, both methods lack network retraining, restricting the potential gains from sparsity exploration, see Fig. 7. Sigma-Delta networks [9] quantize neuron activation changes for temporal redundancy, but are limited to basic tasks like digit classification. CATS, in contrast, excels in the accuracy/computation trade-off and generalization for complex real-world tasks. DAL [36] is a concurrent work with similar goals to CATS. It employs a sparsity penalty on adjacent activation differences, thus enhancing temporal event suppression. It simultaneously learns the network weights and temporal thresholds in training, enhancing model quality and computational efficiency. However, DAL's thresholds are fine-tuned in floating points, and its resulting quantization bits are heavily influenced by the initial threshold value. In contrast, CATS autonomously explores the optimal threshold among a range of discrete power-of-two values during training, rendering the learned thresholds more suitable for hardware implementation. Moreover, CATS takes a step further by integrating temporal suppression with activation suppression, which promises an enhanced approach to event suppression.

3. Preliminaries

3.1. Quantized Sigma-Delta Network

Consider a deep neural network as a stack of N convolution blocks, each including 1 convolution layer and 1 activation layer. Given a linear function g (e.g., a convolution) with a kernel $w \in R^{c_o \times c_i \times k_h \times k_w}$ and an input $x_t \in R^{c_i \times k_h \times k_w}$ at time step t, the output feature map $z_t \in R^{c_o \times k_h \times k_w}$ is computed for each frame as $z_t = g(x_t) = w * x_t$.

Given a sequence of frames in a video as network inputs, we can use the distributive property of convolution g as a linear function (visualized in Fig. 1), the output z_t of each convolution block can be obtained by:

$$\begin{aligned} z_t &= g(x_{t-1}) + g(x_t) - g(x_{t-1}) \\ &= g(x_{t-1}) + g(x_t - x_{t-1}) \\ &= z_{t-1} + g(\Delta x_t), \end{aligned} \tag{1}$$

where Δx_t represents the delta map as the difference between the adjacent sigma maps x_{t-1}, x_t at time steps $t-1$, t. Since z_{t-1} has been already computed for the frame t, computing z_t reduces to summing the term $g(\Delta x_t)$. Owing to the strong correlation between consecutive frames in a video, Δx_t tends to exhibit sparsity, containing non-zero values only for the regions that changed across time.

However, applying $(\Sigma - \Delta)$ modulation may not consistently reduce the event count within the network's delta maps compared to the sigma maps (see the examples in Fig. 4). Quantization is required to lower the resolution of sigma maps and enhance inter-frame correlation in the temporal domain [9, 36, 13]. The input x_t^n of convolution block n is obtained by applying a non-linear activation f to the previous block's output z_t^{n-1}:

$$\begin{aligned} y_{t-1} &= f(z_{t-1}), \\ y_t &= f(z_t). \end{aligned} \tag{2}$$

To quantize the sigma activation maps through temporal threshold tt, the following applies:

$$\begin{aligned} y_{t-1} &= round(f(z_{t-1})/tt) \times tt, \\ y_t &= round(f(z_t)/tt) \times tt. \end{aligned} \tag{3}$$

Notably, the outcomes of quantized full-frame inference and quantized delta-frame inference remain identical, thereby eliminating the accumulation of errors over time. This capability facilitates long-term low-event DNN inference and streamlines the hardware deployment process.

4. Proposed Method

In this paper, we propose a novel end-to-end training approach that autonomously determines layerwise temporal thresholds tt and orchestrates the sparsity penalties on both sigma and delta maps L_{sigma}, L_{delta}. Our method aims to achieve optimal event suppression while preserving the model quality. We formulate the event sparsity exploration problem as:

$$\begin{aligned} \min_{w,\theta} \quad & L_{total}(w,\theta), \\ \min_{w,\theta} \quad & L_{task}(w,\theta) + \lambda_s L_{sigma}(w,\theta) + \lambda_d L_{delta}(w,\theta), \end{aligned} \tag{4}$$

where w embodies the model parameters and θ represents the sampling parameters of temporal thresholds tt.

In our study, we concentrate on three factors of the problem: 1) how to induce sparsity in sigma maps for activation suppression; 2) how to learn layerwise temporal thresholds and mitigate delta-frame differences for temporal suppression; 3) how to balance task loss, sigma and delta sparsity penalties in loss function for an optimal accuracy-sparsity trade-off in combined activation and temporal suppression.

4.1. Activation Suppression

To induce more activation sparsity, we apply an L_1 regularization to the output of ReLUs during training. We follow the approach in the previous study [40] to penalize the non-zero activations below the temporal threshold only during finetuning, thereby allowing the growth of large-magnitude activations for better accuracy recovery. Thus, by applying partial regularization [40], weight parameters w_l in the l^{th} layer are updated by the gradients $g_{partial,l}$ as follows:

$$g_{partial,l} = \frac{\partial L_{total}}{\partial a_{out,l}} \frac{\partial a_{out,l}}{\partial w_l} + \lambda_{s,l} \frac{\partial L^{partial}_{sigma,l}}{\partial a_{out,l}} \frac{\partial a_{out,l}}{\partial w_l}, \tag{5}$$

with

$$L^{partial}_{sigma,l} = ||M(a_{out,l}, tt) \odot a_{out,l}||_1, \tag{6}$$

and

$$M(a_{out,l}, tt) = \begin{cases} 1 & a_{out,l} \in (0, tt) \\ 0 & otherwise \end{cases}, \tag{7}$$

where $a_{out,l}$ represents the l^{th} layer activation outputs, $L^{partial}_{sigma,l}$ is the partial regularization on the sigma maps, and $\lambda_{s,l}$ is the coefficient of sigma penalty used to balance sparsity exploration and accuracy recovery in network learning.

4.2. Differentiable Temporal Threshold Search

Inspired by the prior method that applies a differentiable approach to search for the optimal super net for the problem of ConvNet design [34], we represent the quantization scale (i.e., temporal threshold) search space after nonlinear activations f by a stochastic multi-threshold block (MTB), which contains M candidate branches with various power-of-2 threshold values, as described in Fig. 3.

During the inference (see Fig. 3, right), the candidate thresholds tt^l_m in the super net MTB at the l^{th} layer are executed with the sampling probability:

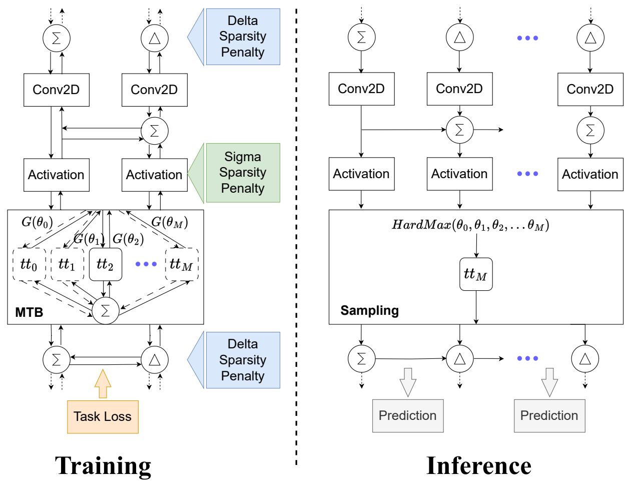

Figure 3: Differentiable Temporal Threshold Search (DTTS) in training (left) and inference (right).

$$P_{\theta_l}(tt_l = tt^m_l) = softmax(\theta^m_l; \theta_l) = \frac{exp(\theta^m_l)}{\sum_m exp(\theta^m_l)}, \tag{8}$$

θ_l indicates the trainable parameters that determine the sampling probability of each threshold. Following Eq. (3), the outputs y^t_l of l^{th} layer at time step t can be expressed as

$$mask^m_l = \begin{cases} 1 & P_{\theta^m_l} = max(P_{\theta^m_l}) \\ 0 & otherwise \end{cases}, \tag{9}$$

$$\begin{aligned} tt_l &= \sum_m mask^m_l tt^m_l, \\ y^t_l &= round(f(z^t_l)/tt_l) \times tt_l, \end{aligned} \tag{10}$$

where $mask^m_l$ is a random variable in $\{0, 1\}$ and is evaluated to 1 if threshold tt^m_l is sampled. The sampling probability is determined by Eq. (8). tt_l denotes the outputs of MTB at l^{th} layer and y^t_l denotes the quantized outputs of activation map $f(z^t_l)$ with the sampled tt_l at time step t.

Regarding training, the overall loss from Eq. (25) is minimized by stochastic gradient descent (SGD) to update the weights w_l in convolution layers and the parameters θ_l in MTB (see Fig. 3 left). θ_l generates the sampling probability P_{θ_l} for each candidate branch, however, only one branch is activated through $mask^m_l$ in the forward pass. As a result, the gradient can not be passed through the discrete mask variable $mask^m_l$ to update θ_l. To sidestep this, previous studies [20] apply the Gumbel-Softmax technique to provide a differentiable relaxation of discrete sampling. Therefore, Eq. (8) and Eq. (9) in training can be updated as

$$\begin{aligned} mask^m_l = G(\theta^m_l) &= GumbelSoftmax(\theta^m_l | \theta_l) \\ &= \frac{exp[(\theta^m_l + Z^m_l)/\tau])}{\sum_m exp[(\theta^m_l + Z^m_l)/\tau]}, \end{aligned} \tag{11}$$

where $Z^m_l \sim Gumbel(0, 1)$ is a random noise following the Gumbel distribution. The Gumbel Softmax function is controlled by a temperature parameter τ. As τ approaches 0, it approximates the discrete categorical sampling as Eq. (9). As τ increases, $mask_l^m$ becomes a continuous random variable. Regardless of the value of τ, the mask is directly differentiable with respect to θ_l^m.

4.3. Temporal Suppression

To suppress the events in the temporal domain, we apply an L_1 regularization on the delta maps Δx_t, defined as the difference of two consecutive sigma maps. Without considering the sigma sparsity penalty, we formulate the temporal suppression problem as

$$\begin{aligned} &\min_{w,\theta} \quad L_{total}(w,\theta), \\ &\min_{w,\theta} \quad L_{task}(w,\theta) + \lambda_d L_{delta}(w,\theta). \end{aligned} \tag{12}$$

Utilizing the Gradient Descent optimization algorithm during training, the network weights w and the tt sampling parameter θ are updated after each epoch based on a learning rate μ and the gradients of w and θ with respect to the training loss $L_{total}(w,\theta)$. This process is described as follows:

$$\begin{aligned} \delta w =& -\mu \partial (L_{task}(w,\theta) + \lambda_d L_{delta}(w,\theta))/\partial w, \\ \delta \theta =& -\mu \partial (L_{task}(w,\theta) + \lambda_d L_{delta}(w,\theta))/\partial \theta, \\ w_{new} &:= w_{old} + \delta w, \theta_{new} := \theta_{old} + \delta \theta. \end{aligned} \tag{13}$$

Eq. (13) shows that the update of θ and w is affected by both task loss and delta sparsity penalty, and the optimization direction is driven by the penalty coefficient λ_d. By adjusting the parameter λ_d, the value of θ is learned to assign appropriate importance to the set of M discrete tt candidates. During inference, the sampling probability P_θ is hard-coded as a mask, enabling the selection of the optimal power-of-2 temporal threshold. This method results in an enhanced accuracy-sparsity trade-off within the temporal domain.

4.4. Combination

The core of our study involves combining activation suppression and temporal suppression to achieve a cumulative effect in event suppression. As depicted in Eq. (25), the efficacy of our suppression training hinges on three components: task loss, sigma sparsity penalty, and delta sparsity penalty. However, these three loss terms compete, since excessive optimization of one may result in the suboptimal optimization of the other two. Therefore, the optimization focus is regulated by coefficient pairs (λ_s, λ_d) associated with these loss components.

As elaborated in Sections 4.1 and 4.3, escalating λ_s / λ_d enhances sparsity in sigma and delta maps, but overly aggressive λ settings may compromise accuracy irreversibly. Consequently, extensive training iterations are required to determine optimal coefficient pairs (λ_{sigma}, λ_{delta}) for optimal optimization performance. To streamline the training effort, we employ Bayesian Optimization (BO) for efficient hyperparameter search.

4.4.1 Bayesian Optimization with SAT

This paper employs Bayesian Optimization (BO) [30] to tackle the optimization challenge:

$$max\ f(\lambda_s, \lambda_d) \tag{14}$$

Here, f represents the target function. To efficiently discover optimal coefficient pairs that maximize event suppression while staying within a specified accuracy drop boundary, we introduce a custom-designed target function termed Sparsity-Accuracy Aware Target (SAT):

$$f(\lambda_s^i, \lambda_d^i) = S_{evt}(\lambda_s^i, \lambda_d^i) * \sigma(\beta * (C(\lambda_s^i, \lambda_d^i) - C_{lim})) \tag{15}$$

In this context, i denotes the iteration index of BO, (λ_s^i, λ_d^i) refers to the coefficient pair selected in the i_{th} round, $C(\lambda_s^i, \lambda_d^i)$ signifies the downstream metric (e.g., accuracy, mIoU, mAP, etc.) of the suppressed model on the validation dataset, C_{lim} sets the boundary for the downstream metric, and $S_{evt}(\lambda_s^i, \lambda_d^i)$ indicates the average event sparsity of the network across the validation dataset. The function σ corresponds to the sigmoid function, generating a soft mask on the resulting map of the downstream metric and filtering values below the boundary C_{lim}. Its behavior is modulated by the temperature parameter β, which approximates discrete sampling at low β and transitions to a continuous random variable at higher β. Notably, both $S_{evt}(\lambda_s^i, \lambda_d^i)$ and $C(\lambda_s^i, \lambda_d^i)$ are outcomes from the same optimization round, and thus, they are computed simultaneously during training. Illustrated in Fig. 3 from Appendix C, the SAT function takes into account both model quality and event sparsity, furnishing BO with an accurate optimization direction.

5. Experiments

5.1. Experimental Setup

Models and Datasets: We extensively evaluate various event suppression approaches using the SSD object detector [24] and the Egohands dataset [4] to demonstrate the effectiveness and superiority of CATS. However, solely validating object detection may not address complex industrial application needs. To prove the versatility of our approach, we extend its application to other domains, such as object tracking (FairMOT-yolov5s [38] on MOT17 [26]), pose estimation (MobileNet-PoseNet [35] on MPII [3]) and semantic segmentation (ResNet18-DeepLabV3+ [8] on Cityscapes [10]).

Environment: We use TensorFlow (TF) for CATS implementation, conducting event suppression training on an Nvidia Quadro RTX 5000 GPU. After optimization, models are evaluated on GrAI-VIP [33], a commercial event-driven DNN processor by GrAI Matter Labs (more details in Appendix B). For inference performance comparison, the models are further evaluated on Nvidia Jetson Nano with TensorRT [31] acceleration, running in MAXN power mode.

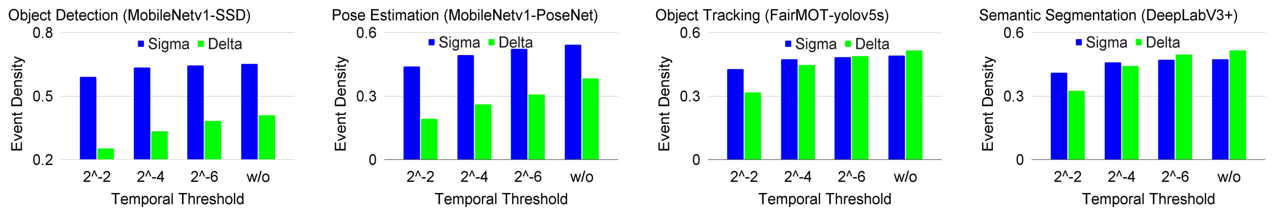

Figure 4: Network event density in full-frame (Sigma) and delta-frame (Delta) inference with different temporal thresholds.

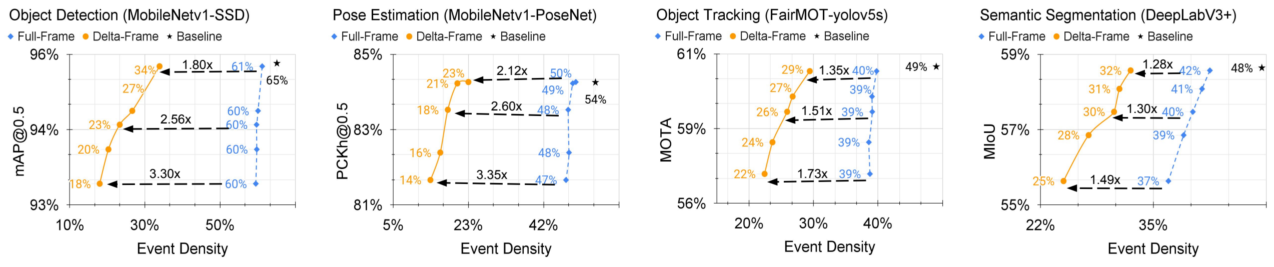

Figure 5: Event suppression via Differentiable Temporal Threshold Search (DTTS) with increased delta sparsity penalties.

Baseline and Evaluation: Baseline models (standard training) serve as reference and starting points for event suppression experiments. The standard-trained model ($base$), alongside its sparsity-augmented variants ($spar$), can operate in either full-frame or delta-frame inference on GrAI-VIP. Metrics are defined to quantify the suppression performance of CATS and its tangible advantages in on-chip evaluation. Suppression metrics for CATS include Event Density (D_{ev}), Improvement Factor for Event Density ($Imp_{D_{ev}}$), and Accuracy (Acc). Performance metrics for hardware evaluation encompass Frame-per-Second (FPS), Latency, Energy Consumption, Improvement Factors for FPS (Imp_{fps}), Latency (Imp_{lat}), and Energy Consumption (Imp_{ene}). Notably, the Improvement Factor is calculated as the ratio of $Metric_{base}$ to $Metric_{spar}$.

5.2. Results

Initially, we assess the accuracy vs. event density trade-off across four computer vision applications: object detection, object tracking, pose estimation, and semantic segmentation. Subsequently, we analyze how the CATS-optimized models perform on an event-driven processor GrAI-VIP.

5.2.1 DTTS via Delta Sparsity Penalty

We present the effect of temporal threshold on network event density in Fig. 4. Utilizing low temporal thresholds ($tt \leq 2^{-6}$) for datasets like MOT17 and Cityscapes, which exhibit limited temporal redundancy, results in ($\Sigma - \Delta$) modulation generating more events within the network's delta maps than the sigma maps. Nevertheless, increasing the temporal threshold value effectively induces more zeros in the delta maps, thereby achieving fewer event counts compared to the sigma maps. This highlights that temporal threshold enhances the performance of delta suppression, even for datasets lacking substantial temporal redundancy.

As illustrated in Fig. 5, the occurrence of event suppression in networks trained using our DTTS method can be attributed to the following three pivotal factors. Firstly, the incorporation of the ReLU activation function (baseline) serves to eliminate negative activations during standard training, thereby suppressing the total volume of events by approximately $47.51\% \sim 65.13\%$. Secondly, the quantization of activation outputs (Full-Frame) compels low-magnitude events to be eliminated, resulting in an extra event reduction by $4.79\% \sim 9.26\%$ beyond the ReLU suppression. Thirdly, the existence of redundant temporal content across consecutive time-steps within the network (Delta-Frame) drives the delta differences towards zero, leading to a substantial reduction of $1.28 \sim 2.12\times$ in event density without compromising the model quality. This reduction can be further enlarged to $1.49 \sim 3.35\times$ with more tolerance on accuracy drop ($1\% \sim 3\%$). The results show that both temporal redundancy and temporal thresholds are crucial in boosting the performance of event suppression.

5.2.2 Combined Temporal and Activation Suppression

Temporal suppression consistently achieves greater event sparsity than activation suppression while maintaining the model quality (refer to Fig. 6). However, the impact of activation suppression on total event suppression should not be disregarded due to the substantial number of events it eliminates in some particular applications. For instance, it reduces event density by 38.25% in Mobilenetv1-SSD (EgoHands) and 23.49% in FairMOT-yolov5s (MOT17). However, comparing the full-frame temporal suppression curves in Fig. 5 with the activation suppression curves in Fig. 6, it's evident that relying solely on a delta sparsity penalty has a limited impact on mitigating spatial redundancy, as

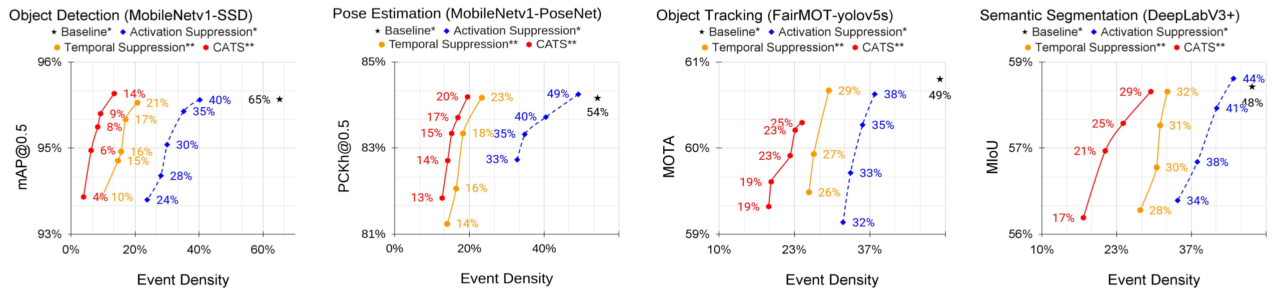
Figure 6: Comparing event suppression across activation suppression, temporal suppression, and CATS for various AI applications. * denotes models executed in full-frame inference, ** indicates models executed in delta-frame inference.

indicated by the steeper slope of full-frame temporal suppression. As the activation suppression performed in the sigma maps can also amplify the gains in sparsity within the delta maps (see Fig. 2), we combine the activation suppression and temporal suppression (CATS, red curve) to yield more event sparsity on top of those single-domain suppression approaches (see Fig. 6). Additionally, we observe that a higher level of activation suppression leads to more event suppression gains in the context of CATS, building upon the foundation of temporal suppression. Activation suppression becomes instrumental in achieving more event suppression, particularly in scenarios where models and datasets display pronounced spatial redundancy.

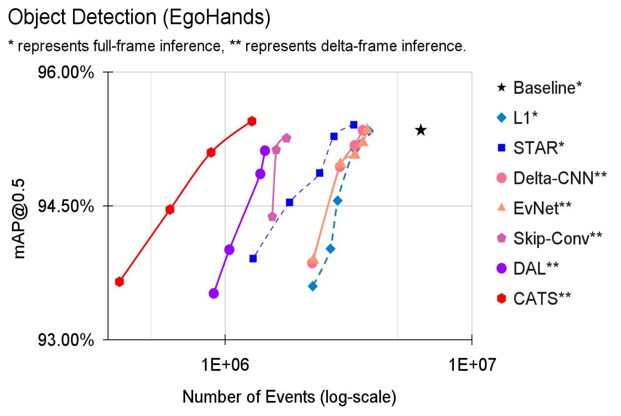
Figure 7: Comparisons with the state-of-the-art (SOTA).

5.2.3 Comparison with State-Of-The-Art (SOTA)

We conducted a replication of recent research involving temporal and activation suppression techniques on Mobilenetv1-SSD (EgoHands). The methods can be categorized into four groups: (1) Activation Suppression: L1 [14], STAR [40], (2) Temporal Suppression without Training: Delta-CNN [25], EvNet [13], (3) Temporal Suppression with Training: DAL [36], Skip-Conv [16], (4) Combining Activation Suppression and Temporal Suppression through Training: CATS.

In Fig. 7, one salient trend is that the training approaches (L1, STAR, DAL, Skip-Conv, and CATS) exhibit significant superiority over the post-training methods (Delta-CNN, EvNet) in terms of event suppression. The second noticeable pattern is that temporal suppression (Skip-Conv, DAL, CATS) methods yield greater event reduction compared to activation suppression methods (L1, STAR), primarily due to the considerable temporal redundancy present in the dataset. A third observation underscores that CATS, featuring an improved temporal threshold search algorithm and benefiting from additional sparsity gain through activation suppression, consistently outperforms existing methods by a significant margin across different accuracy levels. Notably, the disparities in event suppression between CATS and other techniques become even more evident when permitting increased tolerance for model quality degradation.

5.2.4 Ablation Study

We conduct a comprehensive ablation study on MobileNetv1-SSD (EgoHands) for object detection in Tab. 1. A detailed breakdown of the experimental settings is provided below: The baseline serves as the foundation for comparison, reflecting the initial performance under inherent ReLU suppression. Exp-1 has the configuration of CATS, combining all the suppression tricks in training. It achieves a mAP@.5 of 95.39%, reduces event density ($\mathrm{D}ev$) to only 9.43%, and improves the event suppression by a factor of $6.91\times$ while maintaining the model quality. Exp-2 omits the partial regularization, causing a slight mAP@.5 decrease of 0.29% compared to Exp-1, while retaining the event density level. Exp-3 downgrades DTTS to a global non-trainable threshold configuration, resulting in a 43.33% rise in event density compared to Exp-2. Comparing Exp-2 and Exp-4, the absence of activation quantization causes a significant increase of 66.96% in the event counter. Exp-5 removes the sigma sparsity penalty on top of Exp-4, causing a 92.41% increase in event density. On the other hand, the removal of the delta sparsity penalty in Exp-7 also leads to 66.81% event augmentation. Last but not least, Exp-8 removes $(\Sigma - \Delta)$ modulation on top of Exp-7, showing an increase of 30.72% in event density. By systematically varying suppression components, we observe that activation quan-

Table 1: Ablation study on various suppression methods through MobileNetv1-SSD on EgoHands (Object Detection).

Exp ID	Ablation Settings						mAP@.5 ↑	D_{ev} ↓	Imp$_{D_{ev}}$ ↑
	$\Sigma-\Delta$ modulation	partial penalty	differentiable threshold search	hard threshold	delta penalty	sigma penalty			
baseline							95.35%	65.13%	1.00×
1 (CATS)	✓	✓	✓		✓	✓	**95.39**%	9.43%	6.91×
2	✓		✓		✓	✓	95.10%	**9.23**%	**7.06**×
3	✓			✓	✓	✓	95.09%	13.23%	4.92×
4	✓				✓	✓	95.04%	15.41%	4.22×
5	✓				✓		95.14%	29.65%	2.19×
7	✓	✓		✓		✓	95.12%	22.07%	2.95×
8		✓		✓		✓	95.12%	28.85%	2.26×

tization, sigma sparsity penalty, and temporal sparsity penalty are the most influential, each yielding extra event suppression by 40.18%, 48.03%, and 40.05%, respectively. Despite potential overlapping effects, the experimental findings affirm that employing these methods concurrently within an end-to-end training framework enables maximal event suppression.

5.2.5 Hardware Performance

The charts in Fig. 8 demonstrate event/MAC suppression in CATS models alongside their performance advantage over their standard-trained counterparts. On average, CATS achieves improvements of 2.4× in FPS, 2.1× in latency, and 3.1× in energy consumption, showcasing a substantial lead over the on-chip performance of the state-of-the-art activation suppression method STAR [40]. Notably, energy savings through CATS are approximately proportional to its event/MAC reduction, although this relationship does not apply directly to FPS and latency. We speculate this divergence may stem from the poor segmentation of the network for hardware mapping due to real-life hardware constraints, such as limited on-chip memory and varying compilation strategies. Thus, a more detailed analysis of the processing bottlenecks will be conducted in future study.

Moreover, we conduct FPS and latency performance comparisons between GrAI-VIP and the widely-used Nvidia GPU Jetson Nano (see charts in Fig. 9). Taking Jetson TensorRT-enhanced performance as a reference, we observe that GrAI-VIP exhibits slightly better performance than Jetson on the standard-trained models. However, CATS can significantly boost the base model performance by $\sim 3\times$, clearly surpassing the capabilities of Jetson in both latency and FPS. Thus, CATS can elevate event-driven processors as a superior alternative to GPUs for edge computing.

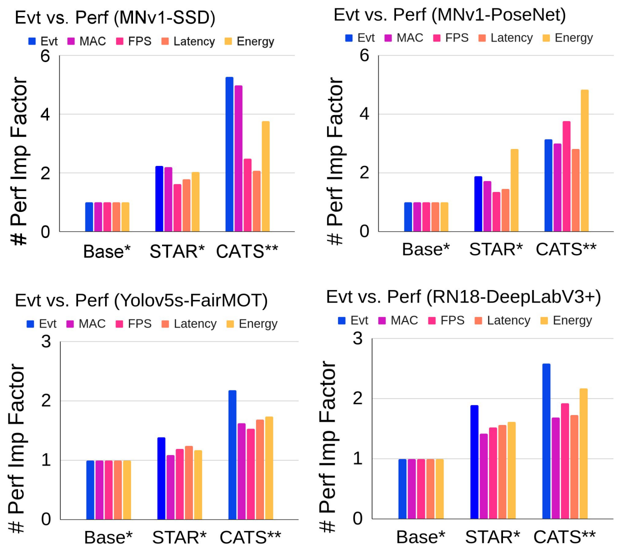

Figure 8: The speedups and energy-savings of activation suppressed full-frame inference (STAR*) and CATS optimized delta-frame inference (CATS**) at batch size 1 on GrAI-VIP compared to the standard-trained model (Base*).

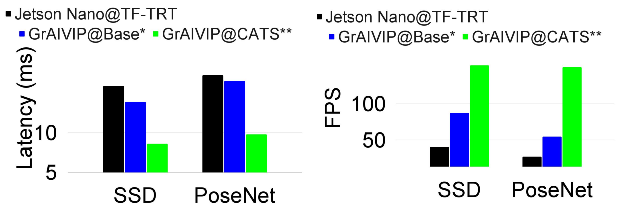

Figure 9: Latency and FPS comparisons between Jetson Nano and GrAI-VIP (with the vertical axis on a log-scale).

We combine activation suppression and temporal suppression by penalizing the presence of non-zero events in spatial and temporal domains simultaneously, which increases the overall event suppression in the network and improves the stability of event counts. Our method outperforms the SOTA on event suppression by a significant margin, demonstrating its strong ability to induce event sparsity. Furthermore, the achieved event suppression efficiently translates into up to 2.4× FPS increase, up to 2.1× latency reduction, and up to 3.1× energy savings on an event-driven processor. These findings highlight the significant value of CATS optimizing DNNs for efficient event-driven processing.

6. Conclusions

This paper introduces CATS, a novel event suppression training method aimed at fulfilling the booming need for sparsity exploration in event-driven processing. Our submethod DTTS automates the determination of layerwise hardware-friendly temporal thresholds, achieving a harmonious balance between accuracy and temporal suppression.

References

[1] Near-memory computing: Past, present, and future. *Microprocessors and Microsystems*, 71:102868, 2019. 9886

[2] Filipp Akopyan, Jun Sawada, Andrew S. Cassidy, Rodrigo Alvarez-Icaza, John V. Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, Brian Taba, Michael P. Beakes, Bernard Brezzo, Jente Benedict Kuang, Rajit Manohar, William P. Risk, Bryan L. Jackson, and Dharmendra S. Modha. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34:1537–1557, 2015. 9876

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 9880

[4] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 9880

[5] Lennart Bamberg, Arash Pourtaherian, Luc Waeijen, Anupam Chahar, and Orlando Moreira. Synapse compression for event-based convolutional-neural-network accelerators. *CoRR*, abs/2112.07019, 2021. 9877, 9888

[6] Simone Bianco, Rémi Cadène, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018. 9876

[7] Lukas Cavigelli and Luca Benini. Cbinfer: Exploiting frame-to-frame locality for faster convolutional network inference on video streams. *IEEE Transactions on Circuits and Systems for Video Technology*, PP:1–1, 03 2019. 9877

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. ECCV*, 2018. 9880

[9] K.F. Cheung and P.Y.H. Tang. Sigma-delta modulation neural networks. In *IEEE International Conference on Neural Networks*, pages 489–493 vol.1, 1993. 9877, 9878

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE CVPR*, 2016. 9880

[11] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. 9876

[12] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021. 9876

[13] Matthew Dutson, Yin Li, and Mohit Gupta. Event neural networks. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 276–293, Cham, 2022. Springer Nature Switzerland. 9877, 9878, 9882

[14] Georgios Georgiadis. Accelerating convolutional neural networks via activation map compression. In *Proc. IEEE CVPR*, pages 7078–7088, Jun. 2019. 9877, 9878, 9882

[15] Xavier Glorot, Antoine Bordes, and Y. Bengio. Deep sparse rectifier neural networks. volume 15, 01 2010. 9886

[16] Amirhossein Habibian, Davide Abati, Taco Cohen, and Babak Ehteshami Bejnordi. Skip-convolutions for efficient video processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 9877, 9878, 9882

[17] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *CVPR*, 2016. 9878

[18] Sebastian Höppner, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Marco Stolba, Florian Kelber, Bernhard Vogginger, Felix Neumärker, Georg Ellguth, Stephan Hartmann, Stefan Schiefer, Thomas Hocker, Dennis Walter, Gengting Liu, Jim D. Garside, Stephen B. Furber, and Christian Mayr. The spinnaker 2 processing element architecture for hybrid digital neuromorphic computing. *ArXiv*, abs/2103.08392, 2021. 9876

[19] Xabier Iturbe, Nassim Abderrahmane, Jaume Abella, Sergi Alcaide, Eric Beyne, Henri-Pierre Charles, Christelle Charpin-Nicolle, Lars Chittka, Angélica Dávila, Arne Erdmann, Carles Estrada, Ander Fernández, Anna Fontanelli, José Flich, Gianluca Furano, Alejandro Hernán Gloriani, Erik Isusquiza, Radu Grosu, Carles Hernández, Daniele Ielmini, David Jackson, Maha Kooli, Nicola Lepri, Bernabé Linares-Barranco, Jean-Loup Lachese, Eric Laurent, Menno Lindwer, Frank Linsenmaier, Mikel Luján, Karel Masařík, Nele Mentens, Orlando Moreira, Chinmay Nawghane, Luca Peres, Jean-Philippe Noel, Arash Pourtaherian, Christoph Posch, Peter Priller, Zdenek Prikryl, Felix Resch, Oliver Rhodes, Todor Stefanov, Moritz Storring, Michele Taliercio, Rafael Tornero, Marcel van de Burgwal, Geert van der Plas, Elisa Vianello, and Pavel Zaykov. Nimbleai: Towards neuromorphic sensing-processing 3d-integrated chips. In *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1–6, 2023. 9876

[20] Eric Jang, Shixiang Shane Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144, 2016. 9879

[21] Mina A Khoei, Amirreza Yousefzadeh, Arash Pourtaherian, Orlando Moreira, and Jonathan Tapson. Sparnet: Sparse asynchronous neural network execution for energy efficient inference. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 256–260, 2020. 9877

[22] Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Goin, William Leiserson, Sage Moore, Nir Shavit, and Dan Alistarh. Inducing and exploiting activation sparsity for fast inference on deep neural net-

works. In *Proc. ICML*, pages 5533–5543, Jul. 2020. 9877, 9878

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *ArXiv*, abs/1806.09055, 2018. 9877

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander Berg. Ssd: Single shot multibox detector. volume 9905, pages 21–37, 10 2016. 9880

[25] Christopher D. Twigg Cem Keskin Robert Wang Markus Steinberger Mathias Parger, Chengcheng Tang. Deltacnn: End-to-end cnn inference of sparse frame differences in videos. *CVPR 2022*, June 2022. 9877, 9878, 9882

[26] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831. 9880

[27] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE Trans. Biomed. Circuits Syst.*, pages 106–122, Aug. 2017. 9876

[28] Orlando Moreira, Amirreza Yousefzadeh, Fabian Chersi, Gokturk Cinserin, Rik-Jan Zwartenkot, Ajay Kapoor, Peng Qiao, Peter Kievits, Mina A. Khoei, Louis Rouillard, Aimee Ferouge, Jonathan C. Tapson, and Ashoka Visweswara. Neuronflow: a neuromorphic processor architecture for live AI applications. In *Proc. DATE*, pages 840–845, 2020. 9876, 9886, 9888

[29] O. Moreira, A. Yousefzadeh, F. Chersi, A. Kapoor, R.-J. Zwartenkot, P. Qiao, G. Cinserin, M.A. Khoei, M. Lindwer, and J. Tapson. Neuronflow: A hybrid neuromorphic – dataflow processor architecture for ai workloads. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 01–05, 2020. 9876, 9886

[30] Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014–. 9880

[31] Nvidia. Nvidia® tensorrt™: An sdk for high-performance deep learning inference. 9880

[32] Anup Vanarse, Adam Osseiran, Alexander Rassau, and Peter van der Made. A hardware-deployable neuromorphic solution for encoding and classification of electronic nose data. *Sensors*, 19:4831, 11 2019. 9876

[33] Sally Ward-Foxton. GrAI Matter research gives rise to AI processor for the edge, Jan. 2020. 9876, 9880

[34] B. Wu, K. Keutzer, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, and Y. Jia. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10726–10734, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. 9877, 9879

[35] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018. 9880

[36] Amirreza Yousefzadeh and Manolis Sifalakis. Delta activation layer exploits temporal sparsity for efficient embedded video processing. 07 2022. 9877, 9878, 9882

[37] Dalin Zhang, Kaixuan Chen, Yan Zhao, B. Yang, Li-Ping Yao, and Christian S. Jensen. Design automation for fast, lightweight, and effective deep learning models: A survey. *ArXiv*, abs/2208.10498, 2022. 9876

[38] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 9880

[39] Zeqi Zhu, Arash Pourtaherian, Luc J.W. Waeijen, Egor Bondarau, and Orlando Moreira. Arts: An adaptive regularization training schedule for activation sparsity exploration. 2022. DSD2022: Euromicro Conference on Digital Systems Design 2022. 9877, 9878

[40] Zeqi Zhu, Arash Pourtaherian, Luc Waeijen, Egor Bondarev, and Orlando Moreira. Star: Sparse thresholded activation under partial-regularization for activation sparsity exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4553–4562, June 2023. 9877, 9878, 9879, 9882, 9883