

Figure 1: Presentation of the modules proposed in this paper. (A) shows the details of DSMS and its comparison with other residual connections, (B) illustrates the details of the proposed additive attention mechanism, PSSA, and (C) compares the proposed SPGD with the vanilla method. Detailed introductions of each module are provided in Sec.4.

ABSTRACT

Spiking Neural Networks (SNNs) have indeed shown remarkable promise in the field of computer vision, emerging as a low-energy alternative to traditional Artificial Neural Networks (ANNs). However, SNNs also face several challenges: i) Existing SNNs are not purely additive and involve a substantial amount of floating-point computations, which contradicts the original design intention of adapting to neuromorphic chips; ii) The incorrect positioning of

Unpublished working draft. Not for distribution.

convolutional and pooling layers relative to spiking layers leads to reduced accuracy; iii) Leaky Integrate-and-Fire (LIF) neurons have limited capability in representing local information, which is disadvantageous for downstream visual tasks like semantic segmentation.

To address the challenges in SNNs, i) we introduce Pure Sparse Self Attention (PSSA) and Dynamic Spiking Membrane Shortcut (DSMS), combining them to tackle the issue of floating-point computations; ii) the Spiking Precise Gradient downsampling (SPGdown) method is proposed for accurate gradient transmission; iii) the Group-LIF neuron concept is introduced to ensure LIF neurons' capability in representing local information both horizontally and vertically, enhancing their applicability in semantic segmentation tasks. Ultimately, these three solutions are integrated into the Powerful Sparse-Spike-Driven Transformer (PSSD-Transformer), effectively handling semantic segmentation tasks and addressing the challenges inherent in SNNs. The experimental results demonstrate that our model outperforms previous results on standard

classification datasets and also shows commendable performance on semantic segmentation datasets. Up to this point, PSSD is the first model in the SNN field to perform semantic segmentation on large datasets. The code will be made publicly available after the paper is accepted for publication.

CCS CONCEPTS

• Networks \rightarrow Network on chip.

KEYWORDS

117

118

119

120

121

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

174

Spiking Neuron Networks, Semantic Segmentation

1 INTRODUCTION

Introduction of SNNs. Spiking Neural Networks (SNNs) are a type of neural network that more closely mimic biological neural networks. Unlike traditional Artificial Neural Networks (ANNs) that use continuous values, SNNs operate using discrete events or "spikes." These spikes are binary events that occur at points in time, resembling the way biological neurons transmit information. SNNs are known for their energy efficiency and are considered more biologically realistic. They have potential applications in areas where low power consumption[15, 27, 33, 39] is crucial, such as edge computing and neuromorphic hardware[3, 9, 11, 35].

142 Introduction of semantic segmentation. Semantic segmentation[16, 29, 41, 46] is a computer vision task where the goal is to 143 assign a label to each pixel in an image such that pixels with the 144 same label share certain characteristics. This process essentially 145 divides the image into meaningful parts with semantic interpre-146 tations, like identifying objects, boundaries, or regions. Semantic 147 148 segmentation is widely used in various applications, including au-149 tonomous vehicles [5, 19, 23, 45], medical imaging [14, 38], and scene understanding[8, 42]. The challenge lies in accurately classifying 150 each pixel while maintaining the integrity of object shapes and 151 contextual relationships within the image. 152

Why combining the spike mechanisms with segmantic 153 segmentation tasks? Semantic segmentation, a task of assigning 154 accurate semantic categories to each pixel in an image, plays a 155 crucial role in image understanding by identifying and dividing 156 different semantic regions in a scene. This vision-intensive com-157 putation task has significant application potential in various real-158 159 world scenarios like autonomous driving, medical surgery robots, and wearable devices. In these contexts, neuromorphic chips with 160 161 sparse spike-driven properties and low-power computational rules 162 are ideal due to their efficiency and suitability for these applications. The deployment of SNNs on neuromorphic chips is considered a 163 key approach for implementing vision models, such as image seg-164 mentation models, marking a promising direction in the future 165 development of artificial intelligence. 166

How to combine the spike mechanisms with semantic
segmentation tasks? To answer this question, we consider two
perspectives: the contradiction between semantic segmentation's
computational density and the low-power nature of spike-driven
computation, and the need for spatial consistency and semantic
accuracy in semantic segmentation. i) To address the core issue
of the contradiction between the computationally dense nature of

175

176

177

178

179

180

181

182

183

184

semantic segmentation and the low-power spike-driven computation rules, on one hand, the introduction of Pure Sparse Self Attention (PSSA) allows the semantic segmentation model to selectively focus on the most important areas for the task, ignoring irrelevant features and fully utilizing the sparsity of model parameters. On the other hand, the introduction of Dynamic Spiking Membrane Shortcut (DSMS) reduces the information loss caused by the use of sparse spike-driven attention, achieving sparse spikedriven computation, significantly reducing energy consumption and easing the application of image semantic segmentation models on neuromorphic chips. ii) Regarding the need for spatial consistency and semantic accuracy in semantic segmentation, the spiking Transformer inherits the global semantic information modeling capabilities of the Transformer for long-distance dependencies. The core issue then is how to enhance spatial consistency and focus on local information. On one hand, the introduction of a locality-based Spiking Precise Gradient downsampling (SPG-down) mechanism preserves gradients at the points of maximum feature information during backpropagation, and uses DSMS to better preserve spatial information. On the other hand, by migrating Group-LIF neurons from the temporal domain to the spatial domain in the feedforward network, considering the neighborhood relationships of pixels in different directions based on feature grouping, local feature contexts are obtained, addressing the problem of weak local information relevance caused by long distances, thereby enhancing the spatial consistency and correlation of features.

Our contributions can be summarized as follows:

- Group-LIF for Local Information Control: Developing Group-LIF neurons to enhance control over local information representation.
- **PSSA + DSMS for Non-floating Point Operations**: Implementing Pure Sparse Self Attention (PSSA) combined with Dynamic Spiking Membrane Shortcut (DSMS) to achieve spike-based processing without floating-point computations.
- SPG-down for Precise Gradient Transmission: Introducing Spiking Precise Gradient downsampling (SPG-down) to ensure more accurate gradient propagation.
- **PSSD-Transformer Architecture**: Proposing the Powerful Sparse-Spike-Driven Transformer (PSSD-Transformer) architecture and applying it to image classification and semantic segmentation tasks, marking the first application of a spiking Transformer architecture to large-scale semantic segmentation datasets.

2 RELATED WORKS

2.1 SNNs

2.1.1 Training Strategy. There are two approaches to obtaining SNN networks. i) ANN2SNN [4, 6, 24], where an ANN is initially trained and then transformed into an SNN, which involves replacing the ReLU activation layer in a trained ANN network with a spiking neuron and adjusting certain hyper-parameters to achieve a highly accurate SNN. However, this method is characterized by long conversion time-steps as well as constraints associated with the original ANN design. ii) Directly training. In recent research, researchers predominantly employ Spike Time Dependent Plasticity (STDP) for training to simulate synaptic information transmission,

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

with more recent studies [30, 44, 48] often utilizing surrogate functions for gradient propagation.

2.1.2 Applications. Implementing above techniques has led to significant results in various fields. For instance, object detection can be achieved through Spiking-Yolo [20] and ems-yolo [37]. In terms of large language models, SpikingGPT [49] and SpikingBert [2] are preferable, and for generative models, SpikingGAN [22] is the way to go. SpikingGCN [50] and SpikingGAT [40] are recommended for Graph Neural Networks. Furthermore, neuromorphic chips such as TrueNorth [28], Loihi [10], and Tianjic [31] are now available, making it increasingly likely that SNNs will be widely used in the near future.

2.2 Semantic Segmentation

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

290

248 2.2.1 Traditional semantic segmentation. This section primarily 249 introduces semantic segmentation algorithms based on deep neural 250 networks, categorizing them into four main representative types: 251 i) Fully Convolutional Network (FCN) [26], by replacing fully con-252 nected layers with convolutional layers and adopting an encoder-253 decoder architecture, facilitates the extraction of deep discrimina-254 tive features for subsequent instance localization and segmentation 255 tasks. However, the computational demands on high-resolution 256 images may result in challenges such as insufficient memory and 257 slower processing speed, while exhibiting a relatively weaker ca-258 pability in handling detailed boundary information. ii) SegNet [1] 259 employs a pooling index mechanism for upsampling its lower-260 resolution input feature maps, aiming to accurately reconstruct 261 object contours during the decoding process. However, when deal-262 ing with large-sized images, it may encounter significant compu-263 tational and memory overhead. iii) DeepLab [7] utilizes Dilated 264 Convolution to expand the receptive field, enhancing segmentation 265 performance by capturing contextual information more effectively. 266 iv) strategies based on Visual Transformer (ViT) [13]: SETR [47] 267 achieves pixel-level segmentation through a multi-level feature ag-268 gregation module; Segmenter [36] obtains class labels and predicts 269 segmentation masks through a point-wise linear decoder or a Mask 270 Transformer decoder; Segformer [43] employs a simple yet efficient 271 MLP decoder to aggregate information from different layers, cleverly combining local attention and global attention to present pow-273 erful representations; PVT overcomes the low-resolution output 274 features of a single scale by introducing a progressively contracting 275 pyramid network backbone. 276

277 2.2.2 Sparse semantic segmentation. Unlike traditional counter-278 part, sparse networks commonly employ lightweight architectures 279 to minimize parameter count and computational load, a critical consideration for deploying semantic segmentation models on resource-280 constrained devices. ERFNet [32] is a lightweight network designed 281 for real-time semantic segmentation, employing residual connec-282 tions and decomposed convolutions to reduce computational load, 283 284 suitable for embedded systems. SGCPNet [17] proposes a strategy for spatial detail-guided context propagation, achieving sparsity by 285 avoiding the maintenance of high-resolution features throughout 286 the network. Other strategies like sparse self-attention [18], which 287 288 decomposing the dense affinity matrix into the product of two sparse matrices. Researchers also employ SNNs to achieve sparsity. 289

SpikeCalib [25] utilizes FCN for ANN to SNN conversion in semantic segmentation, introducing Burst-Spikes neuron model and proposing Lateral Inhibition Pooling (LIPooling) to address errors from max-pooling during the conversion process. SpikeSEG [21] is the first algorithm for semantic segmentation that directly trains SNNs using spike events with the STDP method, but it is effective primarily on simpler datasets. Our work addresses the gap in SNNs' capability for semantic segmentation in complex scenarios.

3 PRELIMINARIES

In this section, we first introduce Leaky Integrate-and-Fire (LIF) neurons, widely applied within the SNN domain. Following this, we explore traditional attention mechanisms and those already present in SNNs. Then, we introduce existing residual connections in SNNs, analyzing their advantages and limitations. Finally, we discuss the mainstream downsampling operations in SNNs and analyze their shortcomings.

3.1 LIF

In SNNs, spike neurons control the release of spikes based on a threshold. In this paper, we use LIF [12] neurons, which work in the following way:

$$U[t] = V[t-1] + \frac{1}{k_{\tau}} \left(X[t] - \left(V[t-1] - V_{reset} \right) \right)$$
(1)

$$S[t] = \mathcal{H}(U[t] - V_{th}) \tag{2}$$

$$V[t] = U[t] (1 - S[t]) + V_{reset}S[t]$$
(3)

where k_{τ} , V_{th} , and V_{reset} represent the decay factor, firing threshold, and reset membrane potential, respectively, which are pre-set to default values. The notation X[t] refers to the input at time step t, while U[t] denotes the membrane potential. The function $\mathcal{H}(\cdot)$ represents the Heaviside step function. The spike output, denoted by S[t], is calculated based on the membrane potential and the threshold. Additionally, V[t] and V[t-1] signify the temporal output at time t.

3.2 Vanilla Self-Attention

3.2.1 Self-Attention in Transformer. In the Transformer model, selfattention mechanism is employed to handle relationships between different positions in the input sequence. The self-attention mechanism in Transformer involves three key steps: computing attention weights, weighted sum, and linear mapping. Given an input sequence $X = (x_1, x_2, ..., x_n)$ where each $x_i \in \mathbb{R}^{d_x}$ is an element of the input sequence, three sets of representations Q, K, V (query, key, value) are obtained through linear mappings:

$$Q = XW_O, \quad K = XW_k, \quad V = XW_V \tag{4}$$

here, W_O , W_K , W_V are learnable weight matrices.

Next, attention weights are computed. The attention weight between positions i and j is calculated using dot product:

$$\alpha_{ij} = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \tag{5}$$

ACM MM, 2024, Melbourne, Australia

Anonymous Authors



Figure 2: Illustration of the overall architecture. Initially, the PSSD-Transformer backbone network undergoes pre-training, outputting scores for each category through a linear classification head. The Spike Mask Transformer consists of a PSSA layer and two MLP modules. These components respectively yield a decoded semantic segmentation feature matrix and a category feature embedding. By performing an inner product of these two feature matrices, the model generates K mask sequences for semantic segmentation, where K represents the number of semantic categories. Finally, a bilinear interpolation upsampling is applied to these sequences to match the original image size, resulting in the semantic segmentation output.

Each output element, z_i , is computed as weighted sum of a linearly transformed input elements:

$$z_i = \sum_{j=1}^n \alpha_{ij} V \tag{6}$$

3.2.2 Spiking Self-Attention in Spikformer. In Spiking Self-Attention (SSA) mechanism, the input undergoes linear transformations and neuron operations to be transformed into spike-form queries (Q), keys (K), and values (V) containing only 0 and 1. The computed attention map has natural non-negativity, disregarding irrelevant features. Therefore, using floating-point representations for Q, K, V, and the softmax function is redundant for modeling such spike sequences.

$$Q = SN_Q[BN(XW_Q)] \tag{7}$$

$$K = SN_K[BN(XW_K)] \tag{8}$$

$$V = SN_V[BN(XW_V)] \tag{9}$$

$$SSA(Q, K, V) = SN(\frac{QK^{I}V}{s})$$
(10)

where SN_Q , SN_K , SN_V and W_Q , W_K , W_V represent Spiking Neurons and weights of the Q, K, V branches, respectively. Q, K, $V \in \mathbb{R}^{T \times N \times d}$, in this context, T denotes the time-step, N denotes the length of the sequence, and d denotes the dimension. BN denotes the Batch Normalization layer, and X is the input of the SSA. s represents the scale factor.

3.3 Shortcut

The Membrane Shortcut (MS) has been demonstrated to satisfy the dynamic isometry theory. It adds a shortcut connection to the membrane potential of each spiking neuron, achieving an identity mapping. This ensures that the tensor of spike transmission always contains binary pulse signals.

$$X^{L} = F^{res}(X^{L-1}) + F^{short}(X^{L-1})$$
(11)

$$F^{res}(X^{L-1}) = ConvBN(SN(X^{L-1}))$$
(12)

where X^L represents the output of the layer L, $F^{res}(\cdot)$ denotes the residual path, $F^{short}(\cdot)$ denotes the shortcut path, $ConvBN(\cdot)$ represents the operation combining Convolution and Batch Normalization.

However, the MS residual connection method still has some limitations. When changing the channel number to match the feature channel number on the residual path and the shortcut path, it may easily overlook non-sparse and non-spike-driven operations in this part. Non-sparse spike operations can lead to significant energy consumption in this module.

3.4 Downsampling in Spikformer

Input undergoes a combination of convolution and batch normalization operations (ConvBN), followed by activation of spiking neurons over multiple time steps. Neurons accumulate membrane potential over time based on the received current, generating a spike signal when the membrane potential exceeds a threshold. Finally, max-pooling is performed for downsampling.

The backward propagation stage of the above process can be expressed by Eq.13-15:

$$\frac{\partial y_{ij}}{\partial h_{uv}} = \begin{cases} 1 & h_{uv} = max(h) \\ 0 & h_{uv} < max(h) \end{cases}$$
(13)

$$\frac{\partial h_{uv}}{\partial x_{uv}} = \frac{\partial X^{t,n}}{\partial V^{t,n}} = \frac{1}{\alpha} H(|V_i^{t,n} - V_{th}| \le \frac{\alpha}{2})$$
(14)

$$\frac{\partial L}{\partial x_{uv}} = \sum_{i=0}^{\frac{H}{s}} \sum_{j=0}^{\frac{W}{s}} \frac{\partial L}{\partial y_{ij}} \frac{\partial y_{ij}}{\partial h_{uv}} \frac{\partial h_{uv}}{\partial x_{uv}}$$

$$= \begin{cases} \frac{1}{\alpha} \frac{\partial L}{\partial y_{ij}} H(|V_i^{t,n} - V_{th}| & h_{uv} = max(h) \\ 0 & h_{uv} < max(h) \end{cases}$$
(15)

where $h \in \mathbb{R}^{H \times W}$, $x \in \mathbb{R}^{H \times W}$, $y \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s}}$ denote the output of the SN, ConvBN and Max Pooling layer, respectively, h_{uv} , x_{uv} , y_{ij} denote the represent the feature information at a specific position for feature maps h, x, y. s is the pooling step. L represents the loss function. α is a parameter designed to ensure the integration of the gradient equals 1 and determines the steepness of the curve.

4 METHOD

In Sec.3, we refine our methods to better address semantic segmentation challenges. In this section, we start with the introduction of Group-LIF, enhancing local information processing in LIF neurons. We then present PSSA, a sparser alternative to SSA, ideal for asynchronous neural chips, with parameter integration into thresholds to minimize float count. DSMS adapts to the dynamic channel variations in semantic segmentation, extracting features from varying feature map dimensions and ensuring MS pulse-driven functionality. Additionally, SPGD is introduced to ensure accurate pulse-driven gradient backpropagation essential in downsampling operations within semantic segmentation tasks.

4.1 Group-LIF

Global LIF neurons introduce weak long-range correlations due to the larger time-steps introduced. In semantic segmentation tasks, it is crucial to maintain spatial consistency, including the interaction of contextual information and local details in the image.

To address the aforementioned issues, we proposes Group-LIF, based on a grouping mechanism. It divides the patch sequence of the feature map into several groups and applies Group-LIF neurons to facilitate information interaction within each group. This enables a more compact pattern of local context information exchange.

As shown in Fig.2(b), G-LIF attempts to transmit information from both horizontal and vertical directions, that is, by learning local contexts and spatial information in different orientations on the image through horizontal LIF and vertical LIF for different communication directions of image blocks. Ablation study for G-LIF is presented in Sec.5. Due to space limitations, details will be provided in the supplementary materials.

4.2 Pure Sparse Self-Attention

The SSA still involves non-spiking computations, making it challenging to deploy the model on neuromorphic chips. To avoid nonspiking computations and energy-consuming operations like multiplication and addition, it is essential to leverage energy-efficient operations such as logical AND, addition, masking, which require minimal energy consumption.

Lemma 1. Transforming matrix dot product using element-wise multiplication and column summation operations incurs almost negligible energy cost.

$$a^T \odot b = \sum_{j=1}^N (a_j \otimes b_j) = \sum_{a_j=1} b_j$$
(16)

$$A^{T} \odot B = \sum_{i=1}^{d} \sum_{j=1}^{N} (A_{ij} \otimes B_{ij}) = \sum_{i=1}^{d} \sum_{A_{ij=1}}^{N} B_{ij}$$
(17)

where $A, B \in \mathbb{R}^{N \times d}$, $a = A_{i1}$ and $b = B_{i2}$, $(\cdot)^T$ denotes the matrix transformation, \odot represents the matrix multiplication, \otimes is the hadamard product.

As shown in Eq.16, when performing the dot product of the transposed binary column vectors *a* and *b* with dimensions $N \times 1$, the resulting dimension is 1×1 . This is equivalent to taking the Hadamard product of column vectors *a* and *b* to obtain an $N \times 1$ tensor, followed by column-wise summation to get a 1×1 vector.

As shown in Eq.17, generalizing to the dot product of all columns of matrices A and B, it can be easily transformed using element-wise multiplication and column summation operations. Moreover, the spiking-driven Hadamard product can be considered as a logical bitwise AND operation with almost negligible energy cost, resulting in a $1 \times d$ dimensional matrix.

Extending this theoretical discovery to the dot product calculation of three binary matrices in the spiking self-attention mechanism, this process can be equivalently substituted by spiking-driven computations for the Q, K, V calculation steps in the SSA mechanism.

$$Q_s \odot K_s^T \odot V_s = SN(\sum_{i=1,j=1}^{N,d} Q_s \otimes K_s) \odot V$$
(18)

$$= SN(SUM_r(Q_s \otimes K_s)) \oplus V_s$$

where \oplus denotes the mask operation, $Q_s, K_s, V_s \in \mathbb{R}^{N \times d}$, $SUM_r(\cdot)$ is the row summary.

Lemma 2. The scaling factor (s) can be incorporated into the threshold V_{th} .

$$\boldsymbol{o}^{(t)} = \begin{cases} 1 & \text{if } \frac{x}{s} > V_{\text{th}} \\ 0 & \text{otherwise} \end{cases}$$
(19)

where x and $o^{(t)}$ are the input and the output of the Spiking Neuron, respectively.

As shown in Eq.19, we update the V_{th} to $V_{th}^{new} = s \cdot V_{th}$.

$$PSSA(Q, k, V) = SN_{V_{th} = V_{th}^{new}} [SUM_r(Q \otimes K) \oplus V]$$
(20)

Thus, in PSSA, the computation process is pure sparse, focusing attention on relevant features to reduce computational overhead. It is also spike-driven, introducing nearly energy-free fully binary

spiking computation, significantly reducing power resources and energy costs, aligning with the computational rules of neuromorphic chips.

4.3 Dynamic Spike-Driven Membrane Shortcut

DSMS retains the concept of MS connections in the main residual path. In the shortcut path, when the feature channel number needs to change, an additional LIF neuron model is added before ConvBN to transform information into sparse spiking signals.

In the context of semantic segmentation, DSMS adapts to backbone networks that extract features through continuous downsampling using Maxpooling to reduce feature map size and parameters. When the channel number remains the same or decreases, DSMS employs a single-branch concatenation of LIF neurons and ConvBN after Maxpooling. In cases requiring an increase in channel number, DSMS introduces a Concat operation for comprehensive spiking feature reuse. This dynamic control of channels ensures consistency with the residual main branch, allowing DSMS to efficiently extract target features from diverse dimensions and channel numbers. The above process can be described by the following equations.

$$X' = MaxPool(X^{L-1})$$
(21)

$$F^{short}(X^{L-1}) = Concat(X', ConvBN(SN(X')))$$
(22)

where X' is the computation result of the maxpool branch.

The quantity and channel width of DSMS can be dynamically adjusted based on specific tasks. In semantic segmentation tasks, optimizing DSMS improves information flow, and the output flow on the shortcut path conceptually approximates the sum of synaptic inputs to the neuron membrane. This representation of complete spiking features effectively reflects the energy-efficient characteristics of the network, ensuring that all modules in the network are spike-driven.

4.4 Spiking Precise-Gradient Downsampling

Unlike the sequence of operations from the order discussed in3.4, the input in SPG-Down undergoes the ConvBN operation to obtain the feature map *x*. Following this, it undergoes max-pooling to obtain a downsized feature map *h*. Finally, the pulse neuron activation produces the feature map *y*.

The backward propagation stage of the above process can be expressed by the Eq.23-25 :

$$\frac{\partial y_{ij}}{\partial x_{uv}} = \begin{cases} x_{uv} & x_{uv} = max(x) \\ 0 & x_{uv} < max(x) \end{cases}$$
(23)

$$\frac{\partial y_{ij}}{\partial h_{ij}} = \frac{\partial X^{t,n}}{\partial V^{t,n}} = \frac{1}{\alpha} H(|V_i^{t,n} - V_{th}|)$$
(24)

$$\frac{\partial L}{\partial x_{uv}} = \sum_{i=0}^{\frac{H}{s}} \sum_{j=0}^{\frac{W}{s}} \frac{\partial L}{\partial y_{ij}} \frac{\partial y_{ij}}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial x_{uv}}$$
$$= \begin{cases} \frac{1}{\alpha} \frac{\partial L}{\partial y_{ij}} H(|V_i^{t,n} - V_{th}| & x_{uv} = max(x) \\ 0 & x_{uv} < max(x) \end{cases}$$
(25)

where *x* denotes the output of "ConvBN". The output of the LIF neuron is its membrane potential $h \in \mathbb{R}^{H \times W}$, with the maximization goal being the output vector $y \in \mathbb{R}^{S \times S}$, making the gradients $\frac{\partial y}{\partial h}$ and $\frac{\partial L}{\partial y}$ non-zero. The activation function and the non-differentiable membrane potential update equation pose key challenges for gradient propagation. During backpropagation, the non-zero nature of $\frac{\partial L}{\partial y_{ij}}$ and $\frac{\partial y_{ij}}{\partial h_{uv}}$ results in the gradient $\frac{\partial L}{\partial h_{uv}}$ becoming zero, where x_{uv} , h_{uv} , y_{ij} represent the respective gradients from $\frac{\partial h_{uv}}{\partial x_{uv}}$, then all gradients would vanish during propagation. Each term of the gradient is the product of the derivative of the previous term and the current term, where α is an amplification factor to apply this amplification to all gradient calculations, with max(h) being amplified to a sufficiently large value.

Evidently, the maximum element after max-pooling in the feature map x is selected as the maximum value, representing the position with the most feature information in x. During back propagation, this position retains the gradient, preserving the maximum feature information in each local region before max-pooling (aggregation). Additionally, since max-pooling is applied to feature map x first, the feature map h has a reduced size, saving energy and memory resources.

5 EXPERIMENT

In this work, the datasets used include ImageNet, CIFAR-10, and ADE20K. The introductions to these datasets and basic experimental setup are provided in the supplementary materials. In this section, ablation studies are conducted on the various modules proposed in this paper; followed by an analysis of the performance of the proposed PSSD-TFSeg on semantic segmentation datasets. It's worth noting that, as we are the first to apply an SNN model on large-scale semantic segmentation datasets, there are no other models for comparison. The chapter concludes with the presentation of Grad-Cam images and semantic segmentation result images from different layers of the model.

5.1 Ablation Study

To investigate the characteristics and effectiveness of several modules proposed in this chapter, including the spiking computing approach (PSSA self-attention, MS and DSMS residual connections), SPG-Down precise gradient downsampling, and neuron mechanisms (Group Horizontal G-HLIF and Group Vertical G-VLIF), a series of ablation experiments were conducted. These experiments further assessed the contributions of the proposed algorithms.

5.1.1 Feasibility of Sparse spiking. Experiments were conducted on the ImageNet1K dataset to assess the accuracy and performance of the PSSA self-attention mechanism and MS/DSMS residual connections. This involved ablation studies on SSA and PSSA mechanisms, SEW versus MS/DSMS, and the combined effect of these two modules.

Tab.1 reveals that although replacing SSA with PSSA incurs a slight loss in accuracy, the energy consumption is less than 50% of that of the SSA-based Spikformer. Essentially, the pulse-driven self-attention mechanism utilizes binary self-attention scores to mask non-essential channels in sparse pulse value tensors, converting all multiplications into sparse computational additions. While this

PSSD-Transformer: Powerful Sparse Spike-Driven Transformer for Image Semantic Segmentation

ACM MM, 2024, Melbourne, Australia



Figure 3: Grad-CAM visualization of the PSSD-Transformer's mid-layer reveals how it divides images into semantic regions, focusing on key objects within its encoder. This targeted learning enhances class label prediction for pixel segmentation by honing in on significant object areas, improving semantic category understanding.



Figure 4: Showcased in the image are semantic segmentation results from the PSSD-TFSeg model on the ADE20K dataset, featuring an MLP-based segmentation head and MS residual connections in its encoder-decoder structure. The model adeptly handles a variety of settings, effectively focusing on objects requiring segmentation in both crowded indoor scenes and complex outdoor environments.

leads to a minor accuracy reduction, the PSSA operator consumes almost no energy. This not only achieves sparse pulse driving but also aligns with low-power computing principles, bringing the deployment of neuromorphic chips closer to reality.

5.1.2 *Effectiveness of spatial consistency.* Experiments on image classification using backbone networks on the CIFAR10 dataset. These experiments involved replacing the downsampling methods in Spikformer and PSSD-Transformer with either standard Vanilla Downsample or SPG-Down, maintaining a time step of 4.

As shown in Tab.2, this setting demonstrated the effectiveness and rationality of the optimized precise downsampling method.

5.2 Comparative quantitative experiments

In this section, comparative quantitative experiments were conducted using the ADE20K dataset (with a resolution of 512×512) and three evaluation metrics. The semantic segmentation backbone network employed was PSSD-Transformer, where residual connections utilized both MS and DSMS. Additionally, the model incorporated the pulse downsampling method SPG-Down and neuron

Table 1: Experiments of the Feasibility of Sparse spiking (PSSA and DSMS) on Imagenet

	Self-Attention	Shortcut	Power(mJ)	Acc (%)
1	*SSA	*SEW	4.96	65.1
2	SSA	MS	4.92(-0.04)	66.9(+1.8)
3	PSSA	SEW	2.18(-2.78)	64.5(-0.6)
4	PSSA	MS	2.14(-2.82)	66.2(+1.1)
5	PSSA	DSMS+MS	2.21(-2.75)	66.8 (+1.7)

Table 2: Ablation Study of Spatial Consistency (SPG-Down and G-LIF) on ADE20K

_	Architecture	Downsampling	Neuron	MIoU (%)
1	Spikformer	Vanilla Down	LIF	26.7
2	Spikformer	SPG-Down	LIF	26.9(+0.2)
3	Spikformer	SPG-Down	G-HLIF	27.4(+0.7)
4	Spikformer	SPG-Down	G-MLIF	27.8(+1.1)

combinations based on G-VLIF and G-HLIF. The semantic segmentation decoders used in these experiments included a pulse linear MLP module (MLP) and a Spike Mask Transformer (SMT).

Table 3: Comparative quantitative experiments on ADE20K

Architecture	Decoder	Power(mJ)	MIoU(%)
*Spikformer	MLP	19.12	26.7
Spikformer	SMT	20.81(+1.69)	26.9(+0.2)
PSSD + MS	MLP	12.61(-6.51)	28.6(+1.9)
PSSD + MS	SMT	14.04(-5.08)	28.9(+2.2)
PSSD + DSMS	MLP	13.50(-5.62)	29.1(+2.4)
	Architecture *Spikformer PSSD + MS PSSD + MS PSSD + DSMS	ArchitectureDecoder*SpikformerMLPSpikformerSMTPSSD + MSMLPPSSD + MSSMTPSSD + DSMSMLP	Architecture Decoder Power(mJ) *Spikformer MLP 19.12 Spikformer SMT 20.81(+1.69) PSSD + MS MLP 12.61(-6.51) PSSD + MS SMT 14.04(-5.08) PSSD + DSMS MLP 13.50(-5.62)

5.3 Visualization

Clearly shown in Fig.3, the Grad-CAM[34] of the PSSD-Transformer effectively segments the image into distinct semantic regions, aligning with the semantic categories of each area. Within each encoder block of the model, the focus is gradually refined to emphasize key object areas in the actual data. This precise targeting significantly bolsters the model's ability to learn various semantic categories, thereby enhancing its capability to more accurately assign class labels to individual pixels in later segmentation tasks.

Fig.4 presents a glimpse of the PSSD-TFSeg model's semantic segmentation capabilities on the ADE20K dataset. This model features an MLP-based semantic segmentation head and MS for residual connections within its encoder-decoder framework. Excelling in a variety of scenes, the model adeptly focuses on key objects for segmentation. This includes intricate indoor settings with numerous people and objects, as well as outdoor landscapes populated with cars, buildings, and animals.

DISCUSSION

6.1 Energy Consumption Estimation

The essence of SNN algorithms is to reconcile the trade-off between precision and power usage. This segment delves into how spike-based models achieve efficiency, utilizing FLOPS and SOPs as performance indicators. SOPs, indicating spikes per second, mirror an SNN's complexity; fewer SOPs mean lower power draw. PSSD-Transformer's efficiency stems from Self-Attention, ConvBN, and MLP modules, which skip computations when inputs are inactive. Energy use for ConvBN and MLP is calculable via operator energy, ANN FLOPS, SFR, and timestep count. The energy consumption of each component is reflected in Tab.1 and Tab.3. Above process can be described by Eq.26-28.

$$E_{\text{Base}} = E_{\text{AC}} \times \text{FL} \times R \times T \tag{26}$$

$$E_{\text{Attn}} = E_{\text{AC}} \times \text{MSFR} \times N_{\text{LIF}} \times T \tag{27}$$

$$E_{\rm sum} = E_{\rm ConvBn} + E_{\rm MLP} + E_{\rm Attn}$$
(28)

Here, FL represents the known FLOPS of the corresponding architecture ANN, reflecting the spiking rate, E_{AC} represents the energy consumption of a single addition operation in the operators. Generally, in ANN models, E_{MAC} = 4.6pJ, E_{AC} = 0.9pJ, T represents the simulation time step, and NLIF represents the number of spiking neurons.

Limitation and Future work 6.2

The application of our work on other datasets has not yet been completed, but it has already demonstrated strong performance on the ADE20K dataset. Future work will focus on both depth and breadth: depth refers to deeper and more powerful networks, while breadth refers to more fields and datasets. Another point is that, similar to other works in this field, energy consumption estimation is still in a preliminary phase. Experiments are conducted on GPUs and have not yet been actually deployed on neuromorphic chips. We believe that in the near future, neuromorphic chips will support a wider range of networks and find extensive applications across various fields, shining brightly. By then, PSSD will become even more meaningful.

CONCLUSION

In this work, we introduce the PSSD-Transformer, a novel approach that effectively marries SNNs with the demanding task of semantic segmentation. By addressing the challenges faced by SNNs in computer vision, such as their intensive floating-point computations, inaccurate layer positioning, and limited local information representation, we have paved the way for the first large-scale application of SNNs in semantic segmentation. Our proposed solutions, including PSSA, DSMS, SPG-down, and Group-LIF neurons, collectively enhance the model's efficiency, accuracy, and applicability. Demonstrating significant improvements on various datasets, the PSSD-Transformer not only surpasses existing models in performance but also highlights the untapped potential of SNNs in energy-efficient, high-precision visual tasks.

PSSD-Transformer: Powerful Sparse Spike-Driven Transformer for Image Semantic Segmentation

ACM MM, 2024, Melbourne, Australia

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043 1044

929 **REFERENCES**

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

- Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE trans*actions on pattern analysis and machine intelligence 39, 12 (2017), 2481–2495.
- [2] Malyaban Bal and Abhronil Sengupta. 2023. SpikingBERT: Distilling BERT to Train Spiking Language Models Using Implicit Differentiation. arXiv preprint arXiv:2308.10873 (2023).
- [3] Kwabena A Boahen. 2000. Point-to-point connectivity between neuromorphic chips using address events. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 47, 5 (2000), 416–434.
- [4] Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. 2022. Optimized potential initialization for low-latency spiking neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 11–20.
- [5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 11621–11631.
- [6] Yongqiang Cao, Yang Chen, and Deepak Khosla. 2015. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision* 113 (2015), 54–66.
- [7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.
- [8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition. 3213–3223.
- [9] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* 38, 1 (2018), 82–99.
- [10] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* 38, 1 (2018), 82–99.
- [11] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R Risbud. 2021. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proc. IEEE* 109, 5 (2021), 911–934.
- [12] Peter Dayan and Laurence F Abbott. 2005. Theoretical neuroscience: computational and mathematical modeling of neural systems. MIT press.
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
- [14] Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. 2017. Machine learning for medical imaging. *radiographics* 37, 2 (2017), 505–515.
- [15] Steve Furber. 2016. Large-scale neuromorphic computing systems. Journal of neural engineering 13, 5 (2016), 051001.
- [16] Shijie Hao, Yuan Zhou, and Yanrong Guo. 2020. A brief survey on semantic segmentation with deep learning. *Neurocomputing* 406 (2020), 302–321.
- [17] Shijie Hao, Yuan Zhou, Yanrong Guo, Richang Hong, Jun Cheng, and Meng Wang. 2022. Real-Time semantic segmentation via spatial-detail guided context propagation. *IEEE Transactions on Neural Networks and Learning Systems* (2022).
- [18] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. 2019. Interlaced sparse self-attention for semantic segmentation. arXiv preprint arXiv:1907.12273 (2019).
- [19] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. 2018. The apolloscape dataset for autonomous driving. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 954–960.
- [20] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. 2020. Spikingyolo: spiking neural network for energy-efficient object detection. In Proceedings of the AAAI conference on artificial intelligence, Vol. 34. 11270–11277.
- [21] Paul Kirkland, Gaetano Di Caterina, John Soraghan, and George Matich. 2020. SpikeSEG: Spiking segmentation via STDP saliency mapping. In 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 1–8.
- [22] Vineet Kotariya and Udayan Ganguly. 2022. Spiking-GAN: A spiking generative adversarial network using time-to-first-spike coding. In 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 1–7.
- [23] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. 2011. Towards fully autonomous driving: Systems and algorithms. In 2011 IEEE intelligent vehicles symposium (IV). IEEE, 163–168.
- [24] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. 2021. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration.

- In International conference on machine learning. PMLR, 6316–6325.
- [25] Yang Li, Xiang He, Yiting Dong, Qingqun Kong, and Yi Zeng. 2022. Spike calibration: Fast and accurate conversion of spiking neural network for object detection and segmentation. arXiv preprint arXiv:2207.02702 (2022).
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.
- [27] Danijela Marković, Alice Mizrahi, Damien Querlioz, and Julie Grollier. 2020. Physics for neuromorphic computing. *Nature Reviews Physics* 2, 9 (2020), 499– 510.
- [28] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 6197 (2014), 668–673.
- [29] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. 2022. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing* 493 (2022), 626–646.
- [30] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine* 36, 6 (2019), 51–63.
- [31] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* 572, 7767 (2019), 106–111.
- [32] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. 2017. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems* 19, 1 (2017), 263–272.
- [33] Catherine D Schuman, Thomas E Potok, Robert M Patton, J Douglas Birdwell, Mark E Dean, Garrett S Rose, and James S Plank. 2017. A survey of neuromorphic computing and neural networks in hardware. arXiv preprint arXiv:1705.06963 (2017).
- [34] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision. 618–626.
- [35] Jae-sun Šeo, Bernard Brezzo, Yong Liu, Benjamin D Parker, Steven K Esser, Robert K Montoye, Bipin Rajendran, José A Tierno, Leland Chang, Dharmendra S Modha, et al. 2011. A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In 2011 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 1–4.
- [36] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. 2021. Segmenter: Transformer for semantic segmentation. In Proceedings of the IEEE/CVF international conference on computer vision. 7262–7272.
- [37] Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. 2023. Deep Directly-Trained Spiking Neural Networks for Object Detection. arXiv preprint arXiv:2307.11411 (2023).
- [38] Kenji Suzuki. 2017. Overview of deep learning in medical imaging. Radiological physics and technology 10, 3 (2017), 257–273.
- [39] Yoeri van De Burgt, Armantas Melianas, Scott Tom Keene, George Malliaras, and Alberto Salleo. 2018. Organic electronics for neuromorphic computing. *Nature Electronics* 1, 7 (2018), 386–397.
- [40] Beibei Wang and Bo Jiang. 2022. Spiking GATs: Learning Graph Attentions via Spiking Neural Network. arXiv preprint arXiv:2209.13539 (2022).
- [41] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. 2018. Understanding convolution for semantic segmentation. In 2018 IEEE winter conference on applications of computer vision (WACV). Ieee, 1451–1460.
- [42] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*. 418–434.
- [43] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems 34 (2021), 12077–12090.
- [44] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. 2023. Spike-driven Transformer. arXiv preprint arXiv:2307.01694 (2023).
- [45] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. 2020. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access* 8 (2020), 58443–58469.
- [46] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. 2018. Context encoding for semantic segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 7151–7160.
- [47] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. 2021. Re-thinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and*

1045 1046	[48]	pattern recognition. 6881–6890. Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong	[49]	Rui-Jie Zhu, Qihang Zhao, and Jason K Eshraghian. 2023. Spikegpt: Genera- tive pre-trained language model with spiking neural networks. <i>arXiv preprint</i>	1103 1104
1047		Tian, and Li Yuan. 2022. Spikformer: When spiking neural network meets trans- former arXiv preprint arXiv:2209 15425 (2022)	[50]	arXiv:2302.13939 (2023). Zulun Zhu, Jiaving Peng, Jintang Li, Liang Chen, Qi Yu, and Sigiang Luo, 2022	1105
1048		10111101. uritiv proprint uritivi2009.19129 (2022).	[30]	Spiking graph convolutional networks. <i>arXiv preprint arXiv:2205.02767</i> (2022).	1106
1049					1107
1050					1108
1051					1109
1052					1110
1053					1111
1054					1112
1055					1113
1056					1114
1057					1115
1058					1116
1059					1117
1060					1118
1061					1119
1062					1120
1063					1121
1064					1122
1065					1123
1066					1124
1067					1125
1068					1126
1069					1127
1070					1128
1071					1129
1072					1130
1073					1131
1074					1132
1075					1133
1076					1134
1077					1135
1078					1136
1079					1137
1080					1138
1081					1139
1082					1140
1083					1141
1084					1142
1085					1143
1086					1144
1087					1145
1088					1146
1089					1147
1090					1148
1091					1149
1092					1150
1093					1151
1094					1152
1095					1153
1096					1154
1097					1155
1098					1156
1099					1157
1100					1158
1101					1159