

Single Layer Perceptron

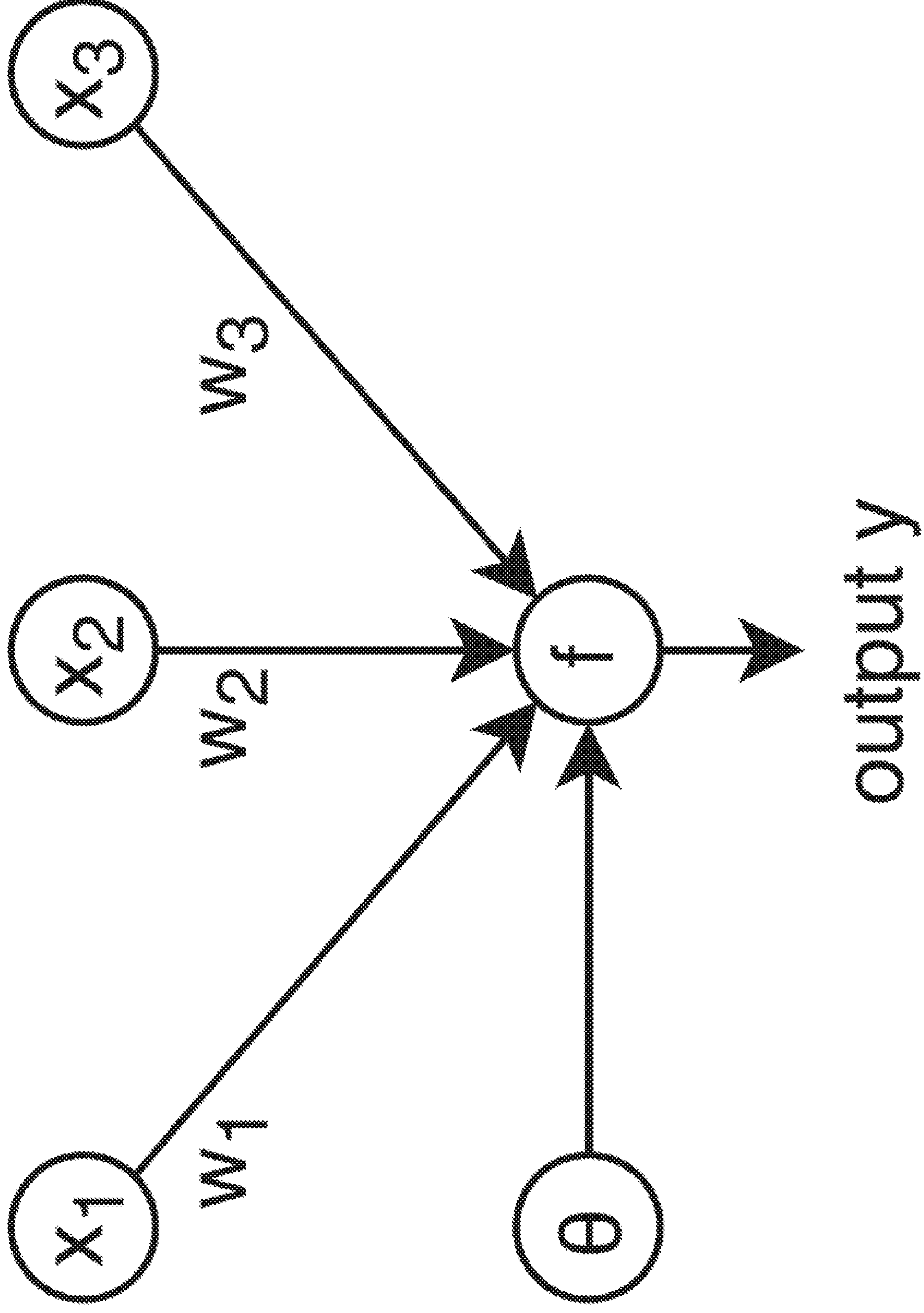


FIG. 1

Feed-Forward Neural Network

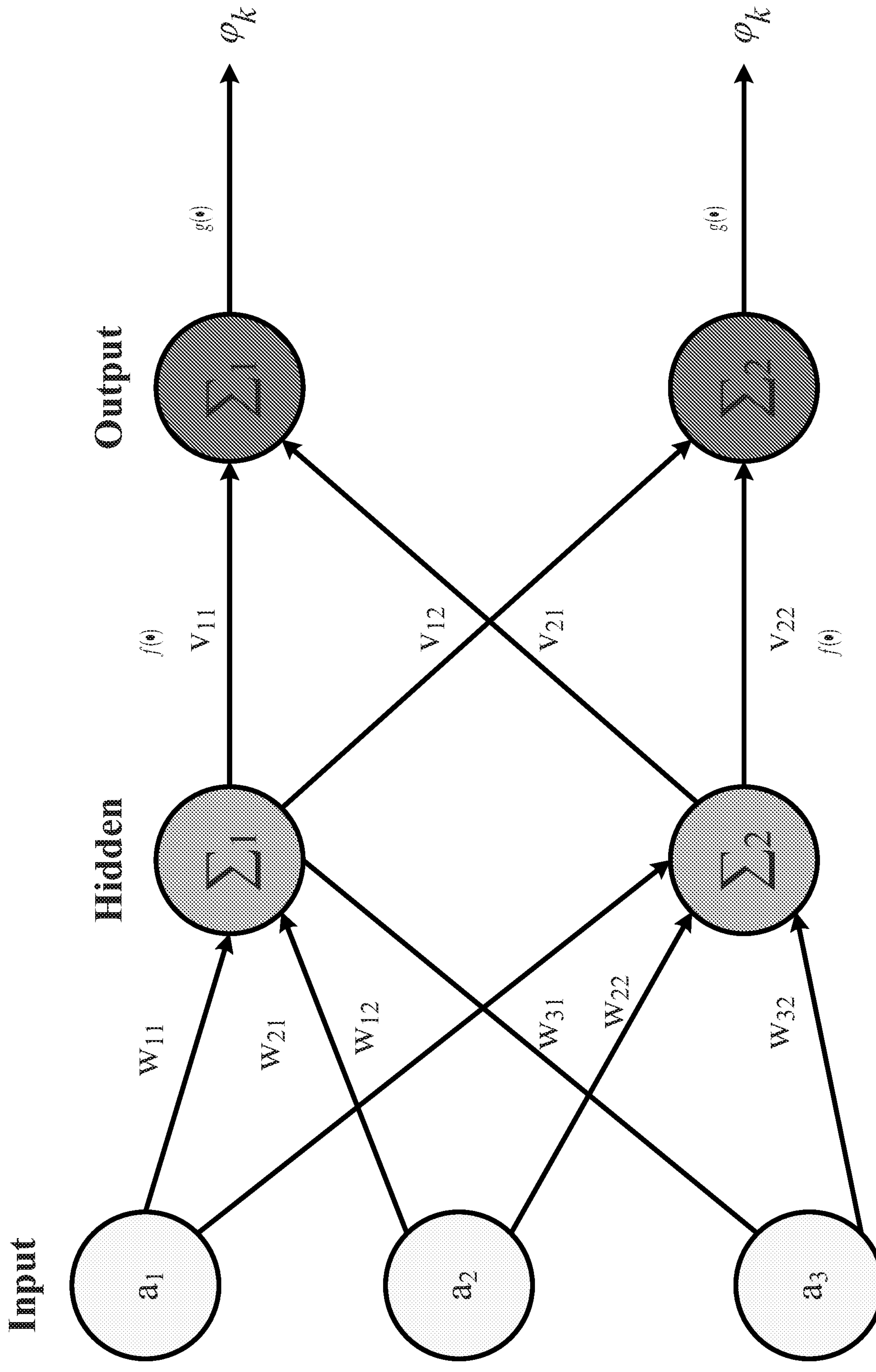
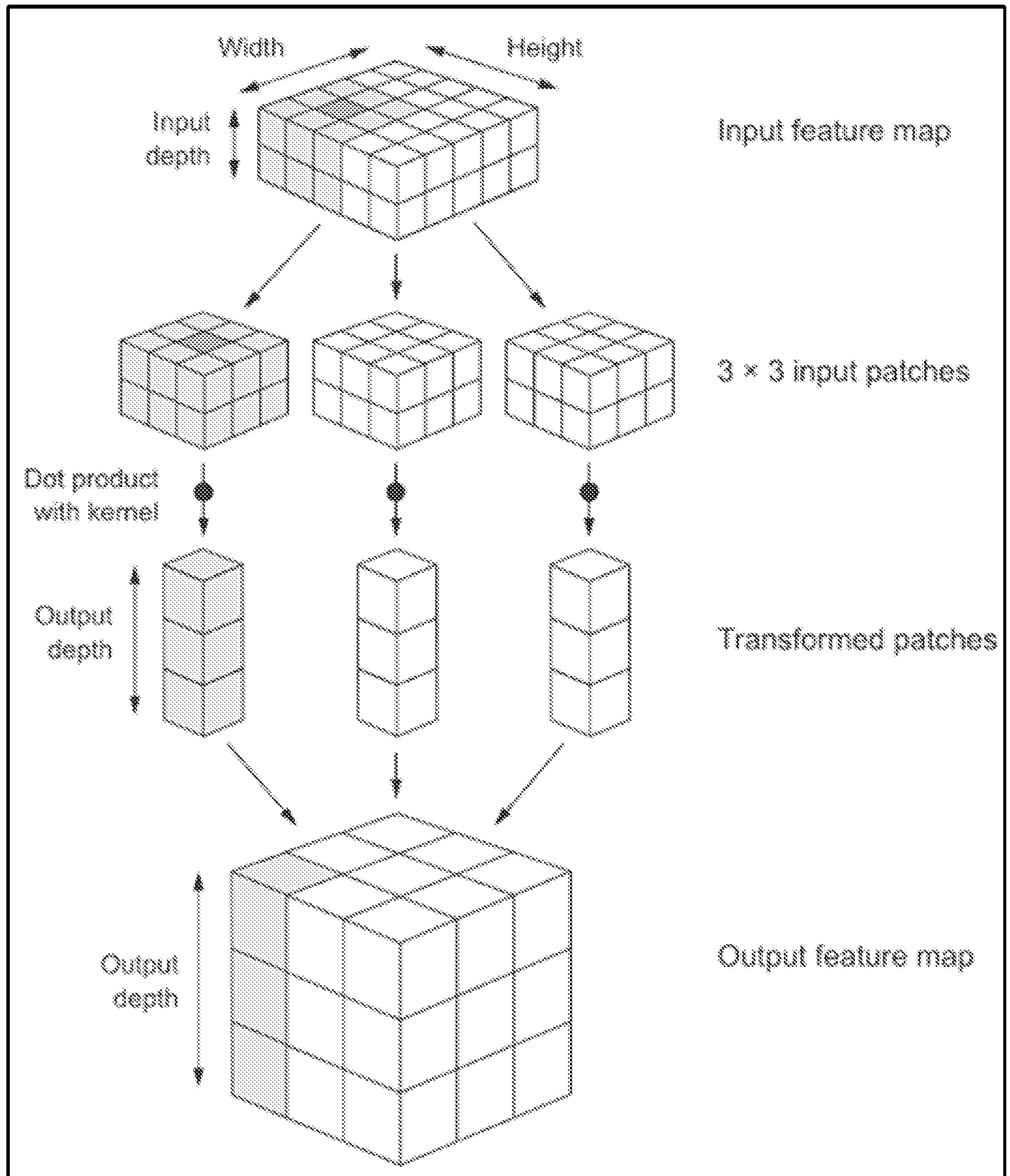


FIG. 2

Convolutional Neural Network

**FIG. 3**

Training Convolutional Neural Network

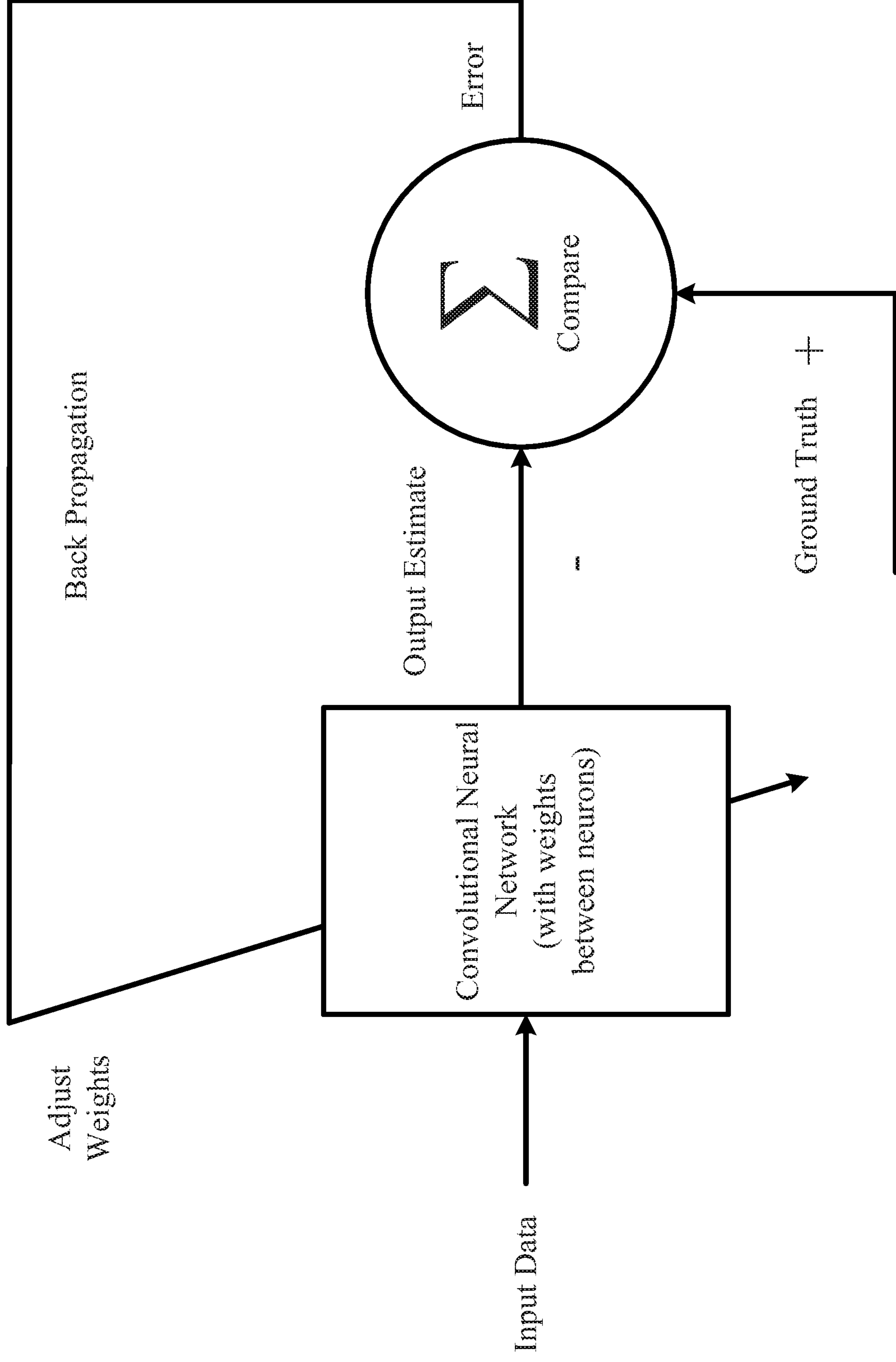
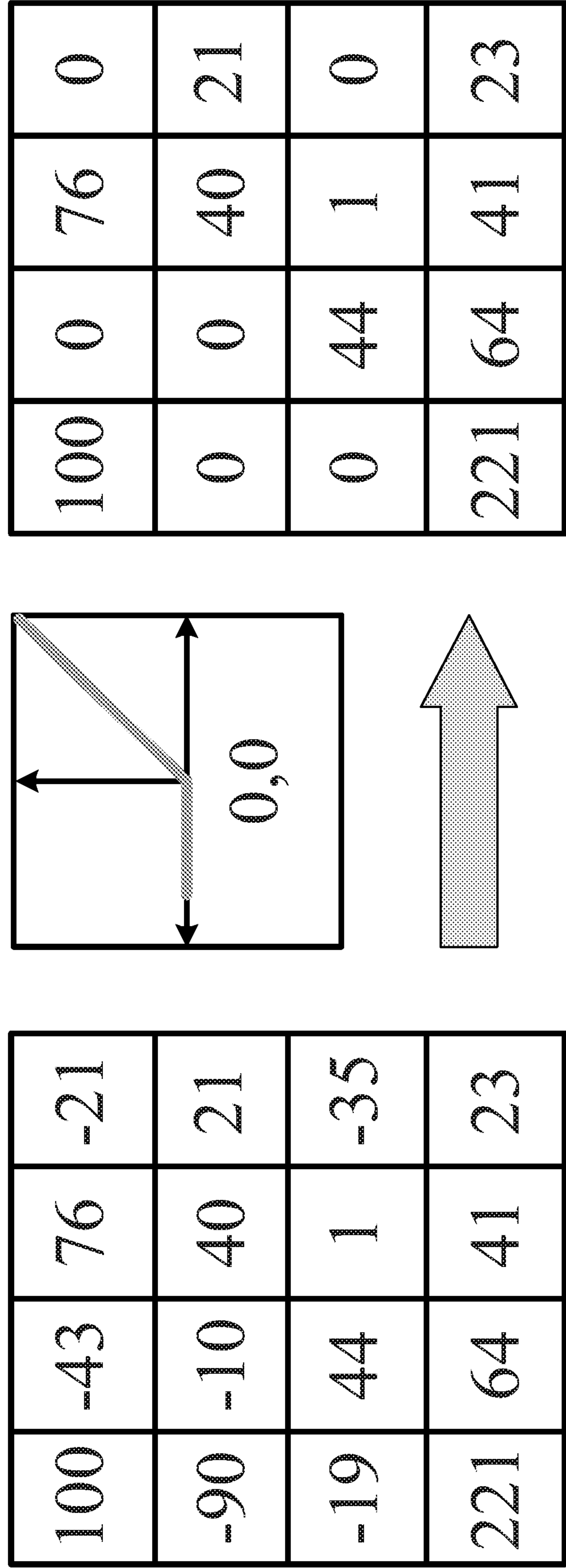


FIG. 4

Non-Linear Layers of Convolutional Neural Networks



ReLU Activation Function

FIG. 5

Dilated Convolutions

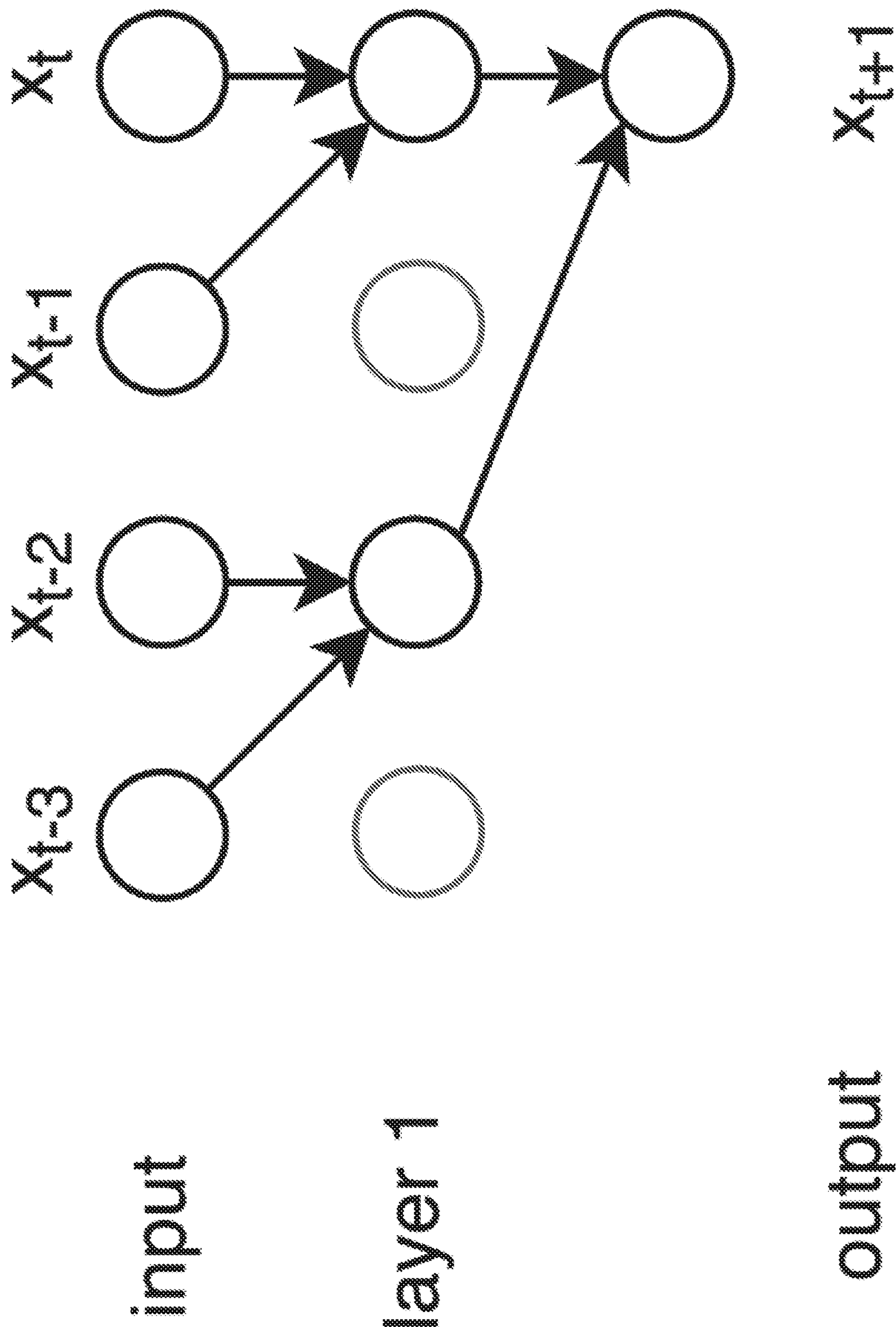


FIG. 6

Sub-Sampling Layers of Convolutional Neural Networks

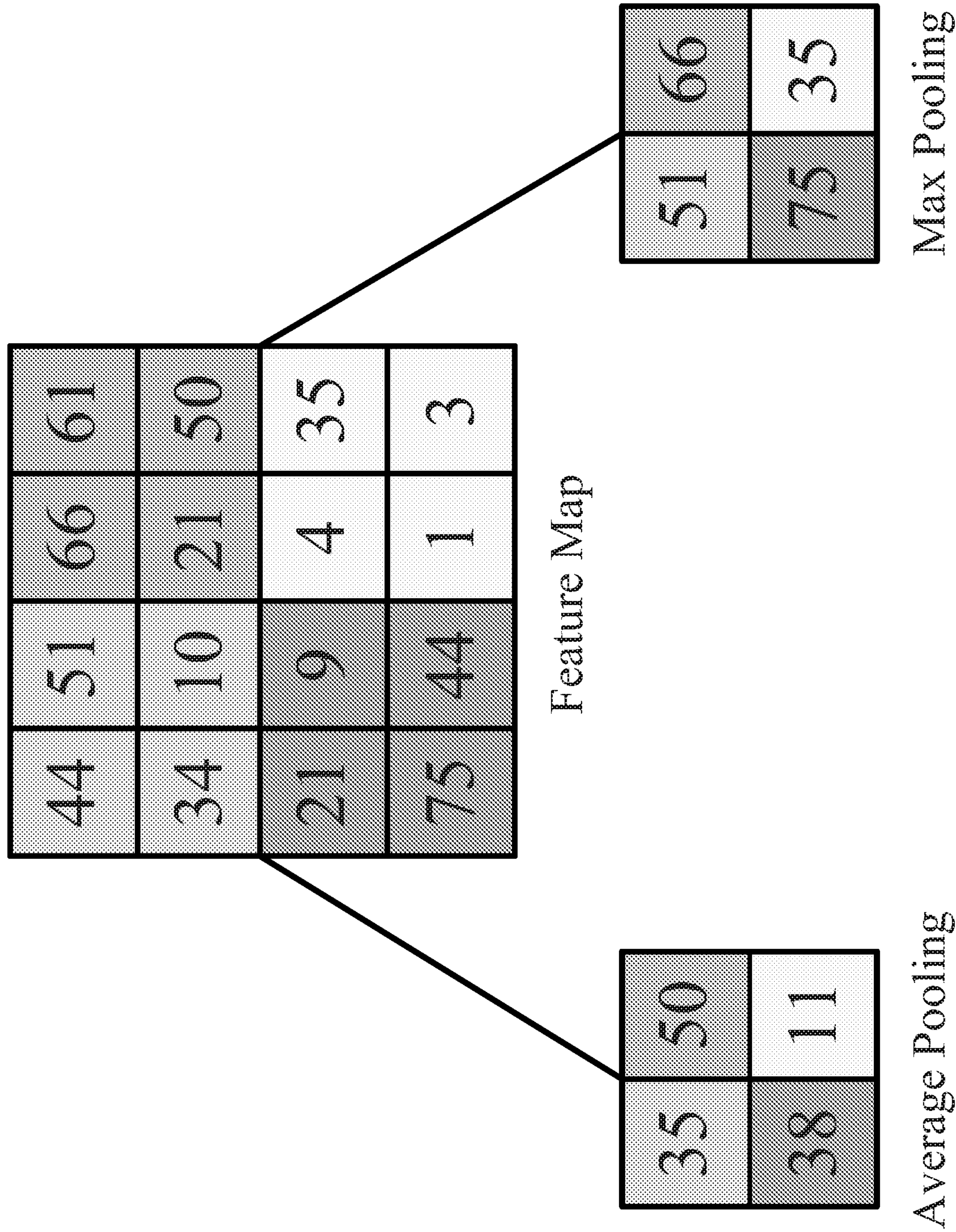


FIG. 7

Convolution Layers of Convolutional Neural Networks

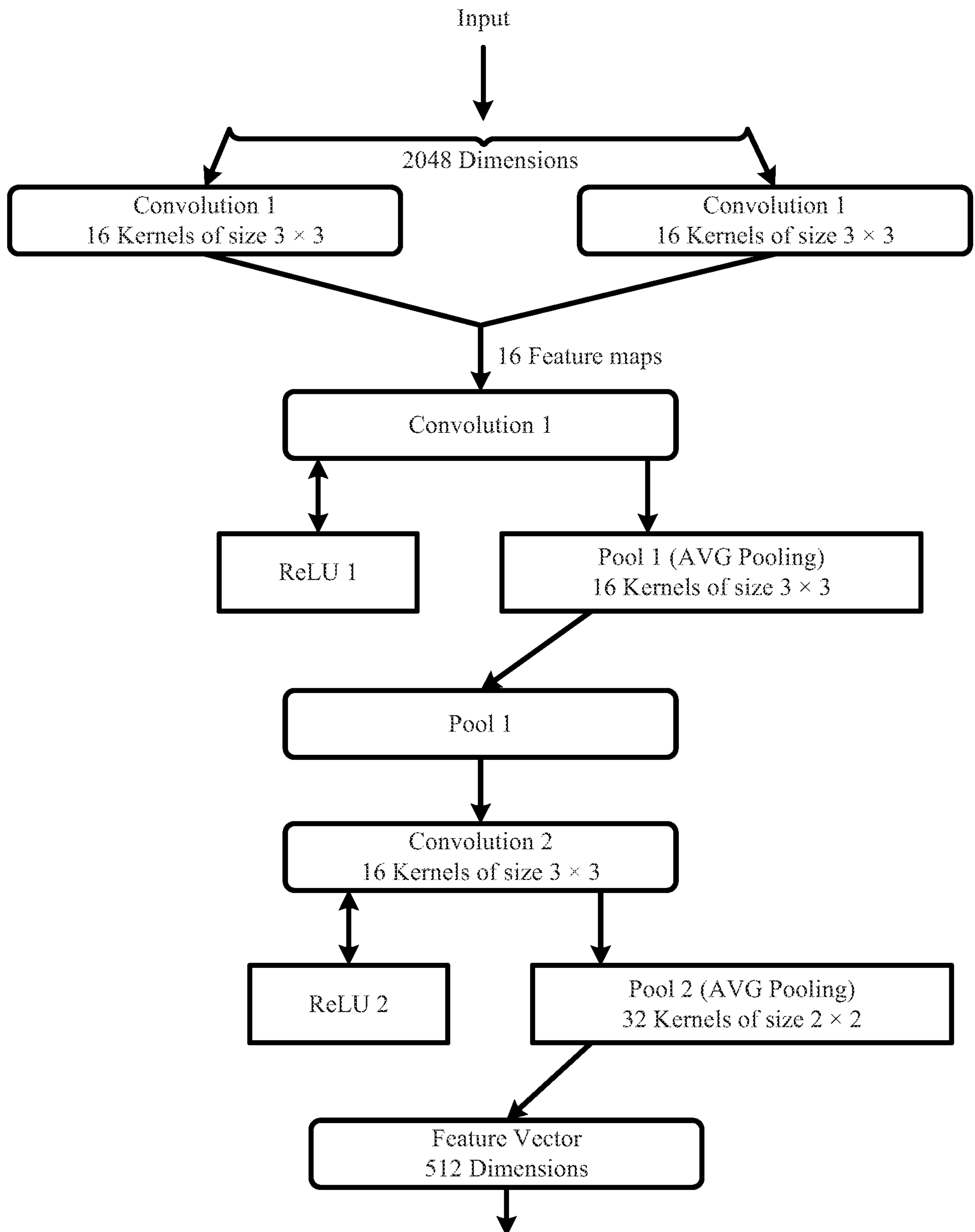


FIG. 8

Residual Connections used in Convolutional Neural Networks

9 / 59

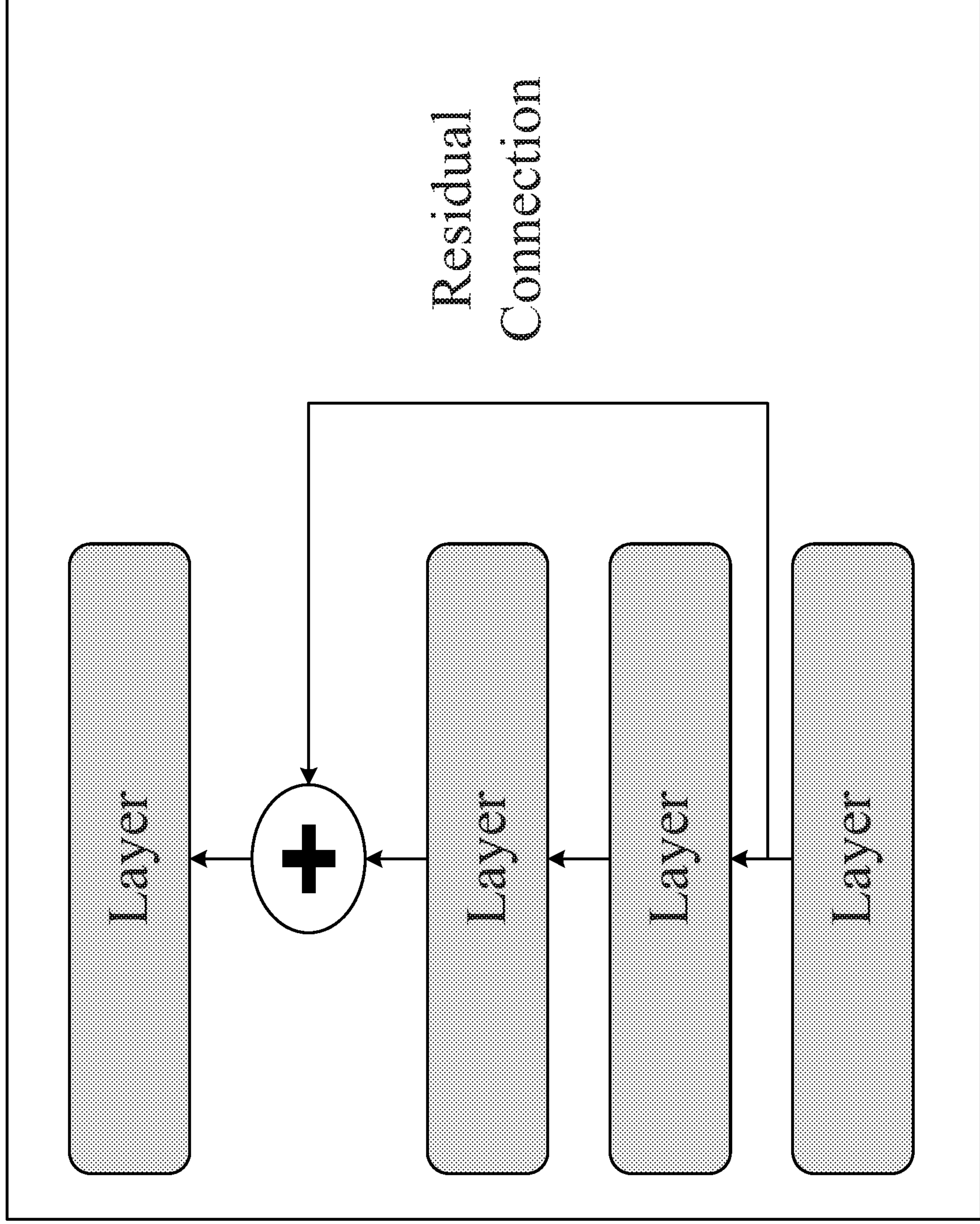
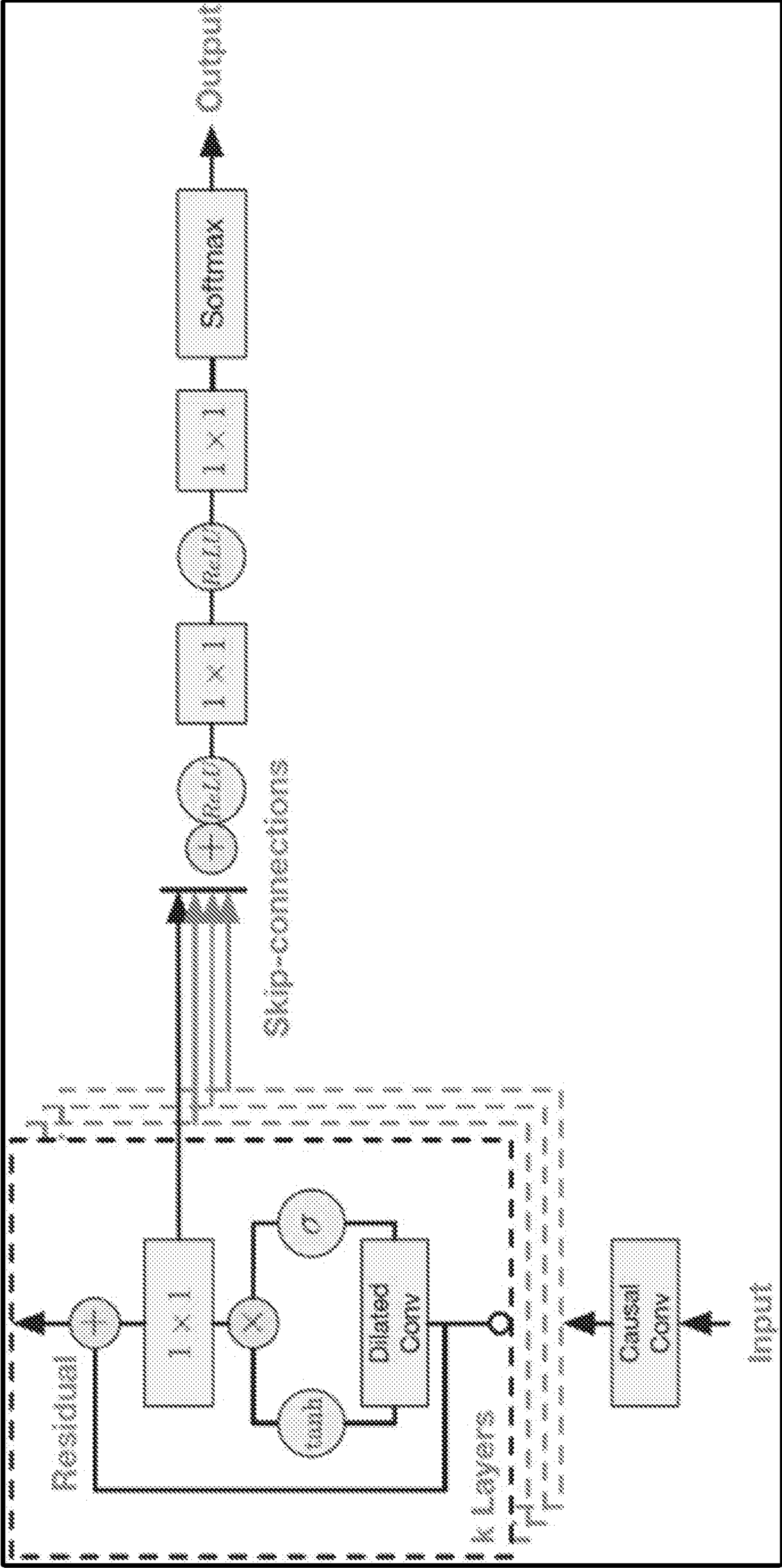


FIG. 9



Residual Block and Skip-Connections used in
Convolutional Neural Networks

FIG. 10

Stacked Dilated Convolutions

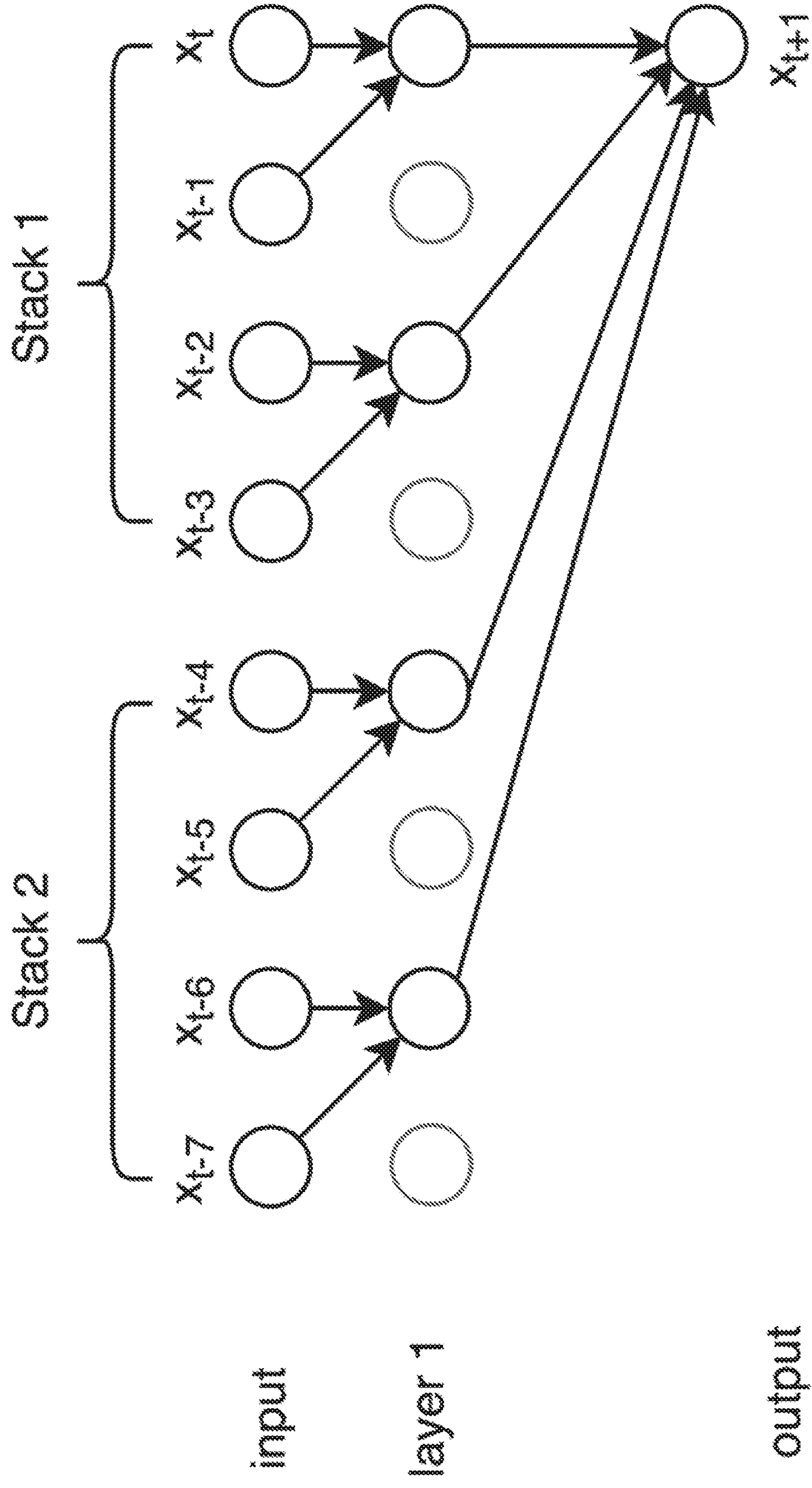


FIG. 11

Batch Normalization Forward Pass with Convolutional Neural Networks

$$\begin{aligned}\mu_{\mathcal{B}} &= \frac{1}{n} \sum_{i=1}^n x_i^{(\ell-1)} \\ \sigma_{\mathcal{B}}^2 &= \frac{1}{n} \sum_{i=1}^n (x_i^{(\ell-1)} - \mu_{\mathcal{B}})^2 \\ \hat{x}^{(\ell-1)} &= \frac{x^{(\ell-1)} - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \\ x^{(\ell)} &= \gamma^{(\ell)} \hat{x}^{(\ell-1)} + \beta^{(\ell)}\end{aligned}$$

FIG. 12

Batch Normalization – Inference with Convolutional Neural Networks

$$\hat{x}^{(\ell-1)} = \frac{x^{(\ell-1)} - \mu_{\mathcal{D}}}{\sqrt{\sigma_{\mathcal{D}}^2 + \epsilon}}$$
$$x_i^{(\ell)} = \gamma^{(\ell)} \hat{x}_i^{(\ell-1)} + \beta^{(\ell)}$$

13 / 59

FIG. 13

Batch Normalization Backward Pass with Convolutional Neural Networks

$$\begin{aligned}\nabla_{\gamma^{(\ell)}} \mathcal{L} &= \sum_{i=1}^n (\nabla_{x^{(\ell+1)}} \mathcal{L})_i \cdot \hat{x}_i^{(\ell)} \\ \nabla_{\beta^{(\ell)}} \mathcal{L} &= \sum_{i=1}^n (\nabla_{x^{(\ell+1)}} \mathcal{L})_i\end{aligned}$$

FIG. 14

Batch Normalization in Convolution Layers

```
conv_model.add(layers.Conv2D(32, 3, activation='relu'))  # After a Conv layer
conv_model.add(layers.BatchNormalization())

dense_model.add(layers.Dense(32, activation='relu'))  # After a Dense layer
dense_model.add(layers.BatchNormalization())
```

15 / 59

FIG. 15

1D Convolution used in Convolutional Neural Networks

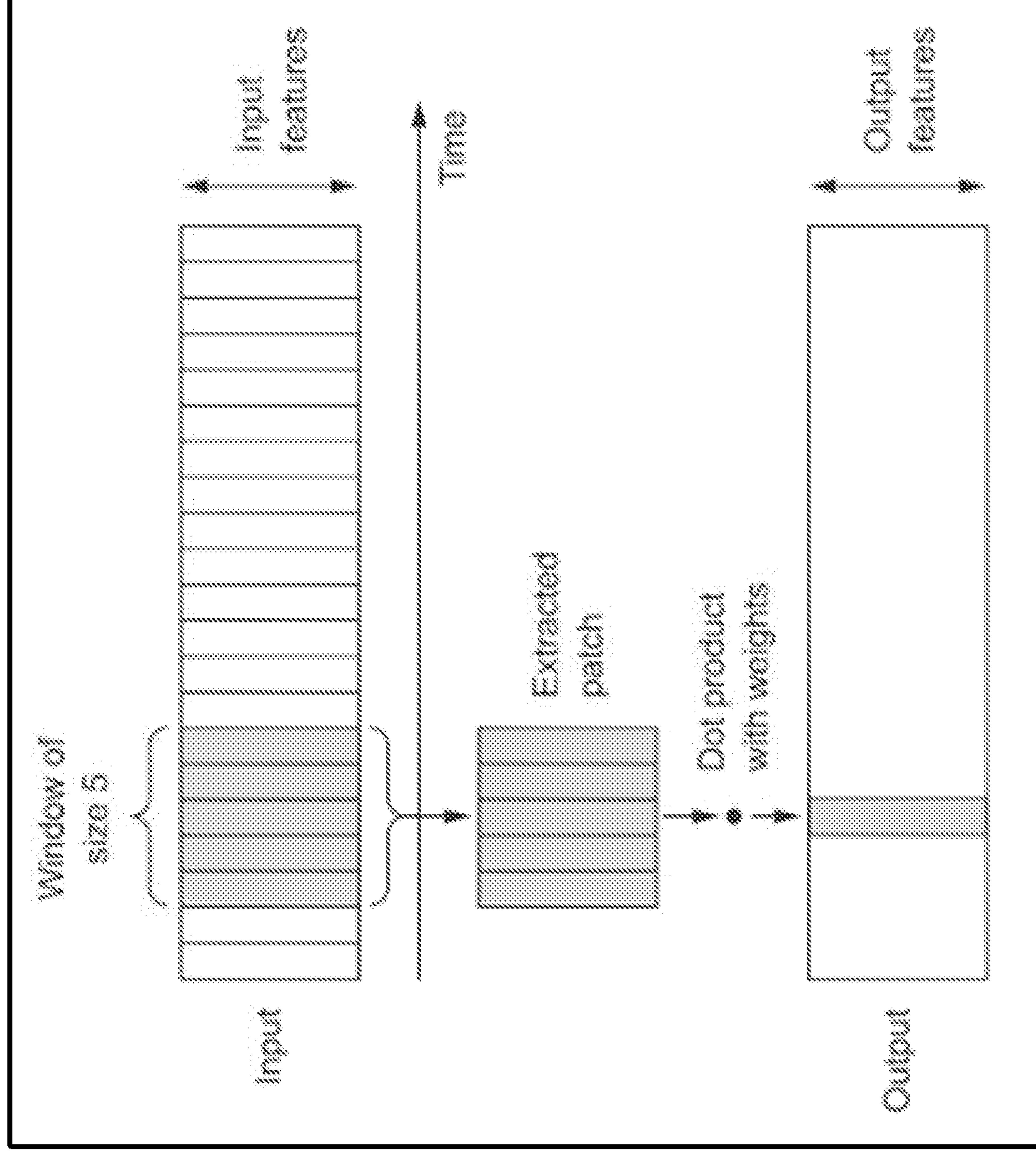


FIG. 16

Global Average Pooling (GAP) in Convolutional Neural Networks

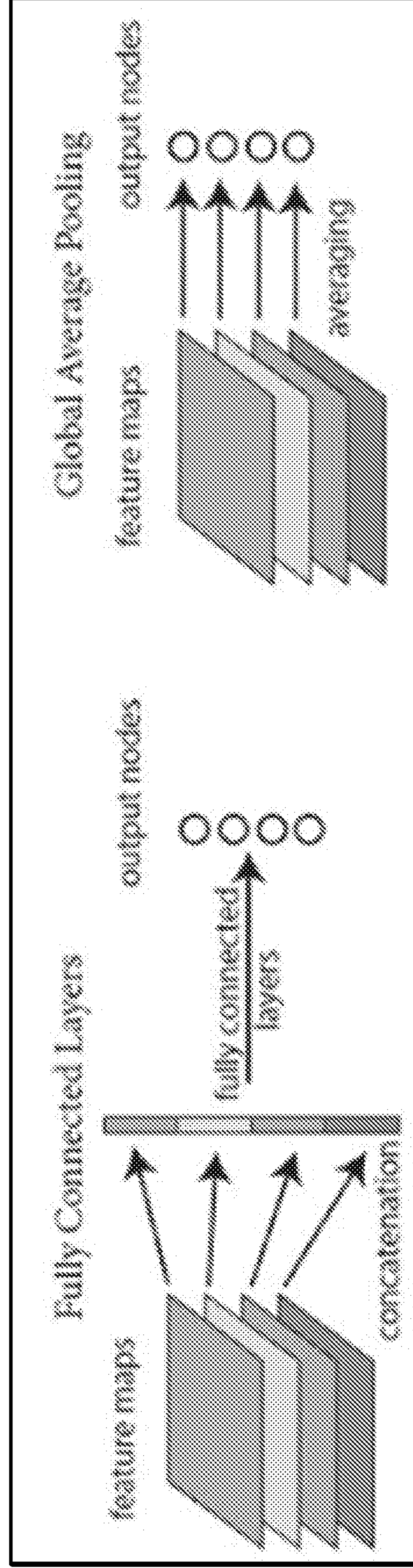
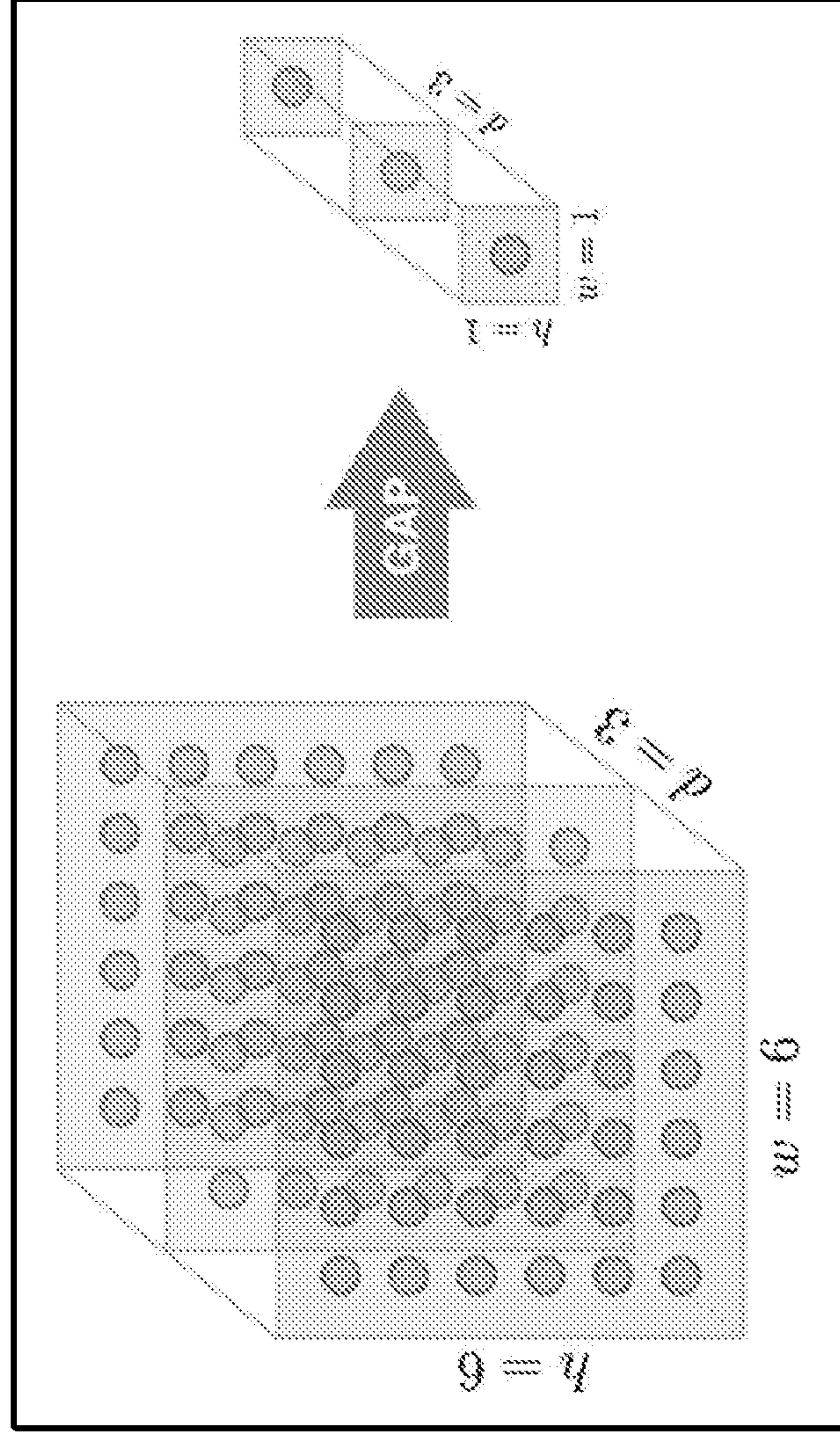


FIG. 17

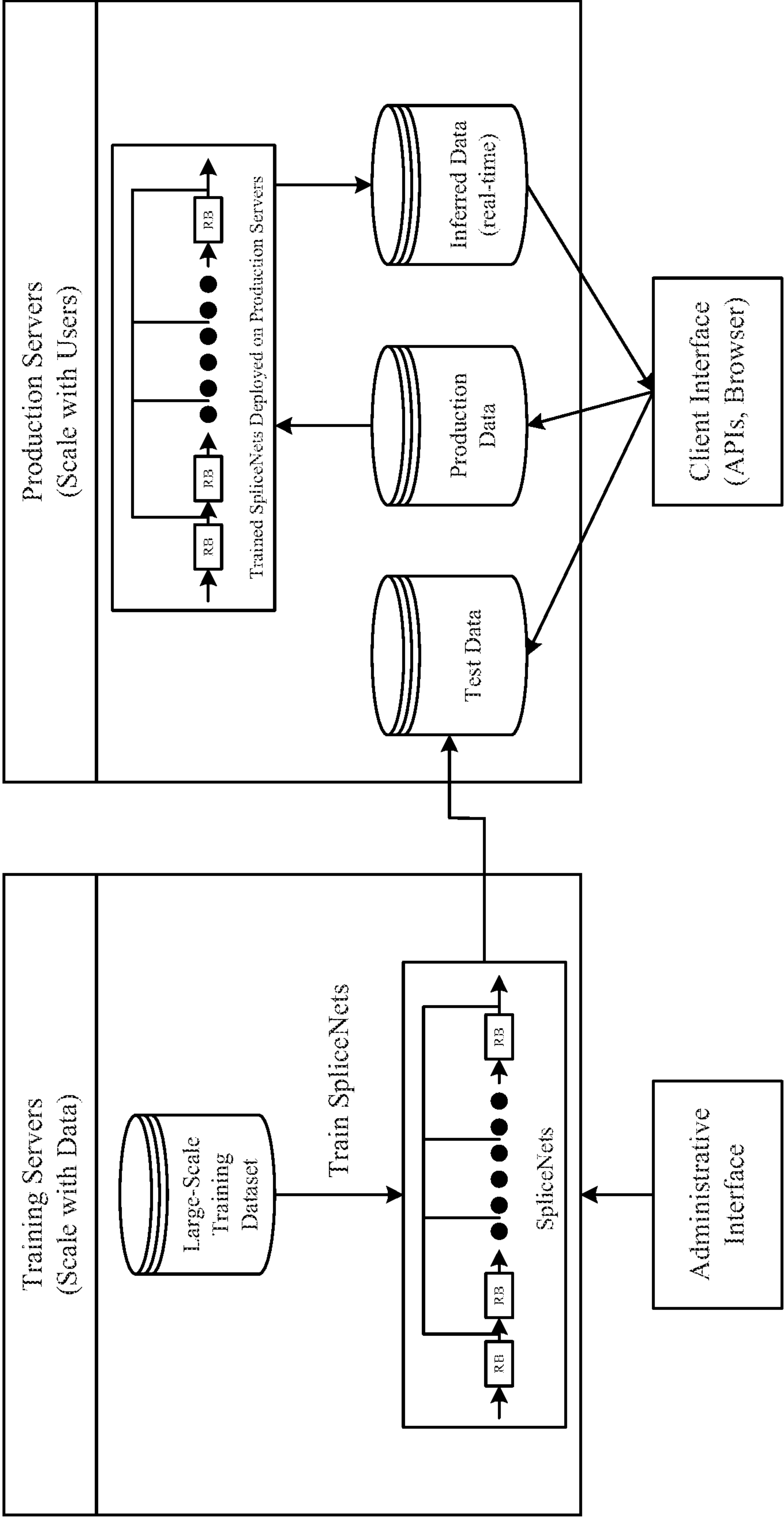
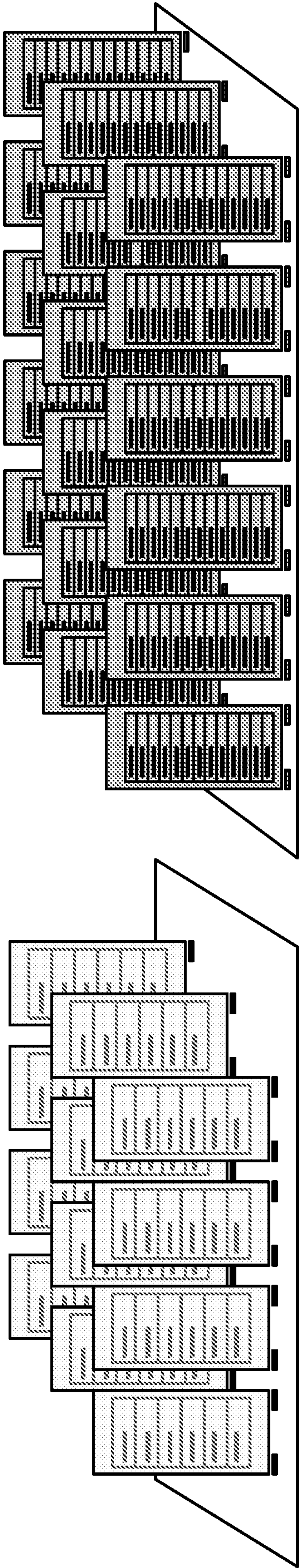


FIG. 18

SpliceNet

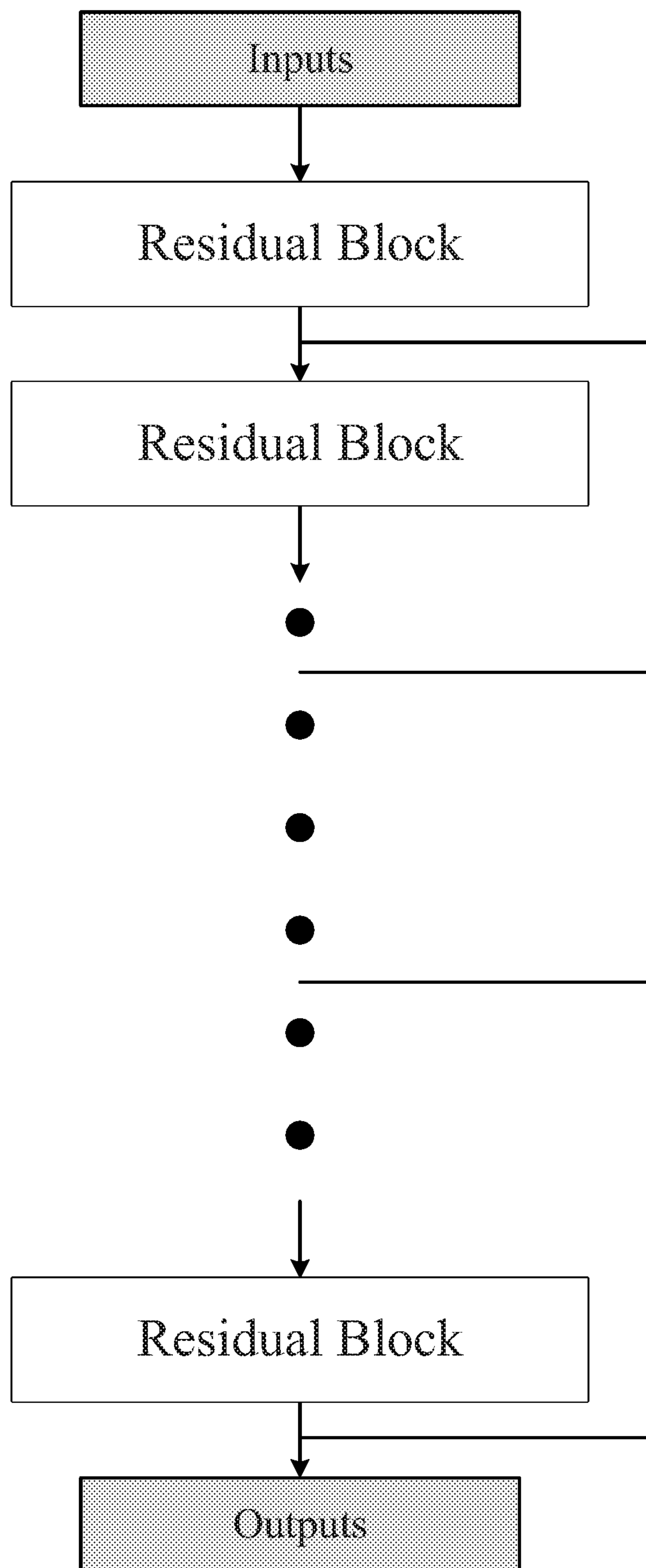
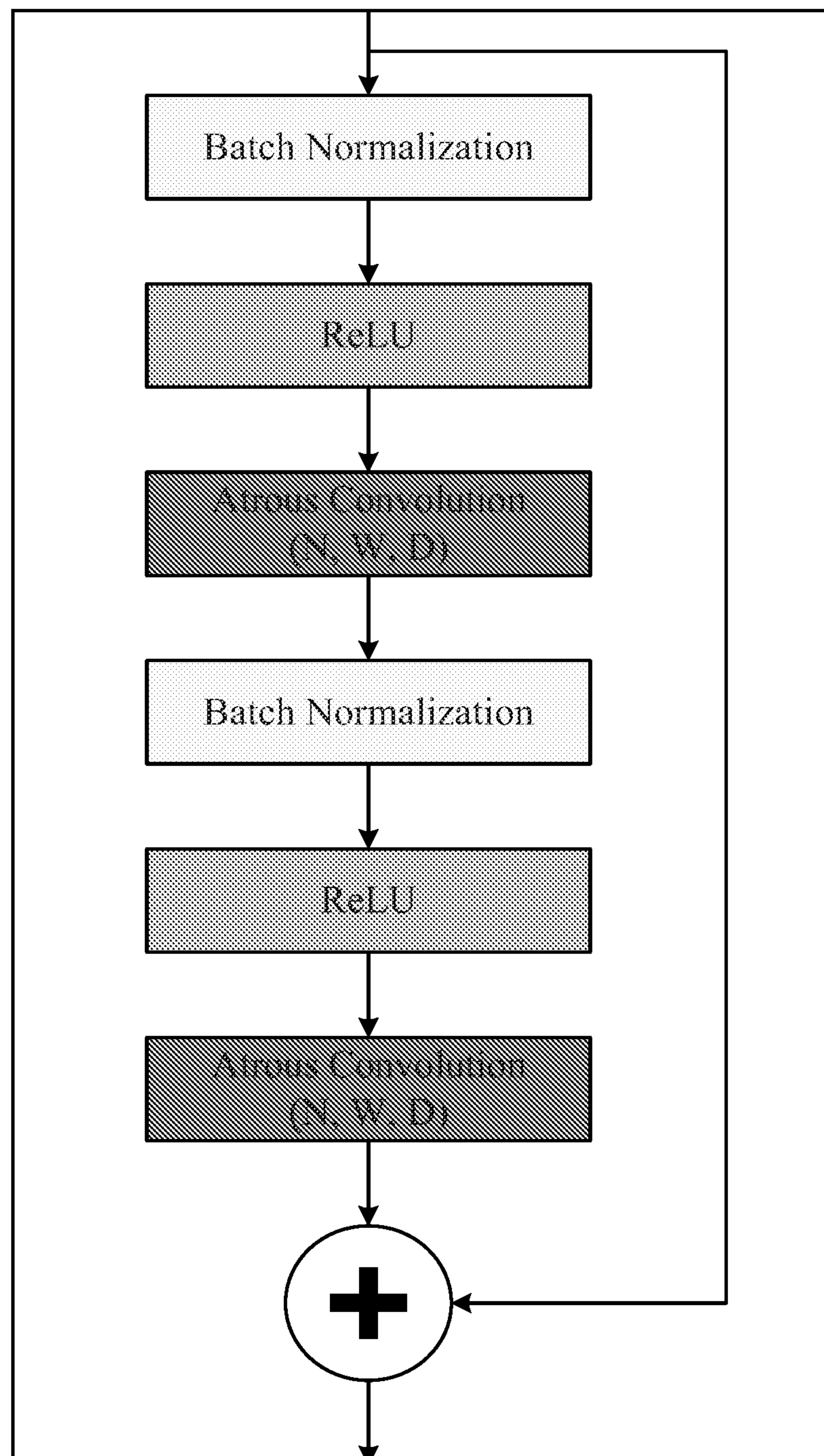


FIG. 19

Residual Block (RB) (N, W, D)



N: Number of Convolution Filters

W: Convolution Window Size

D: Atrous Convolution Rate

FIG. 20

SpliceNet80

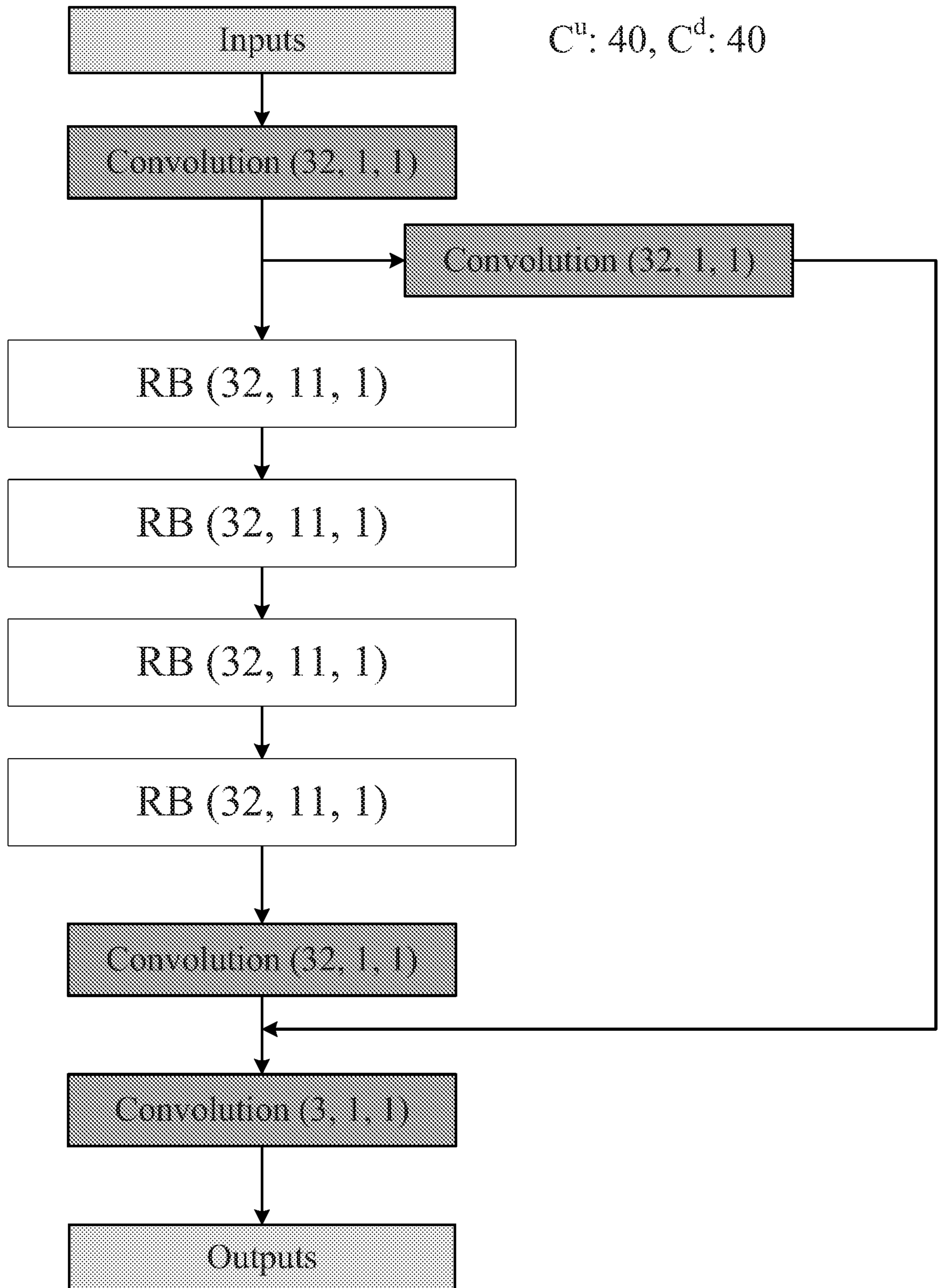


FIG. 21

SpliceNet400

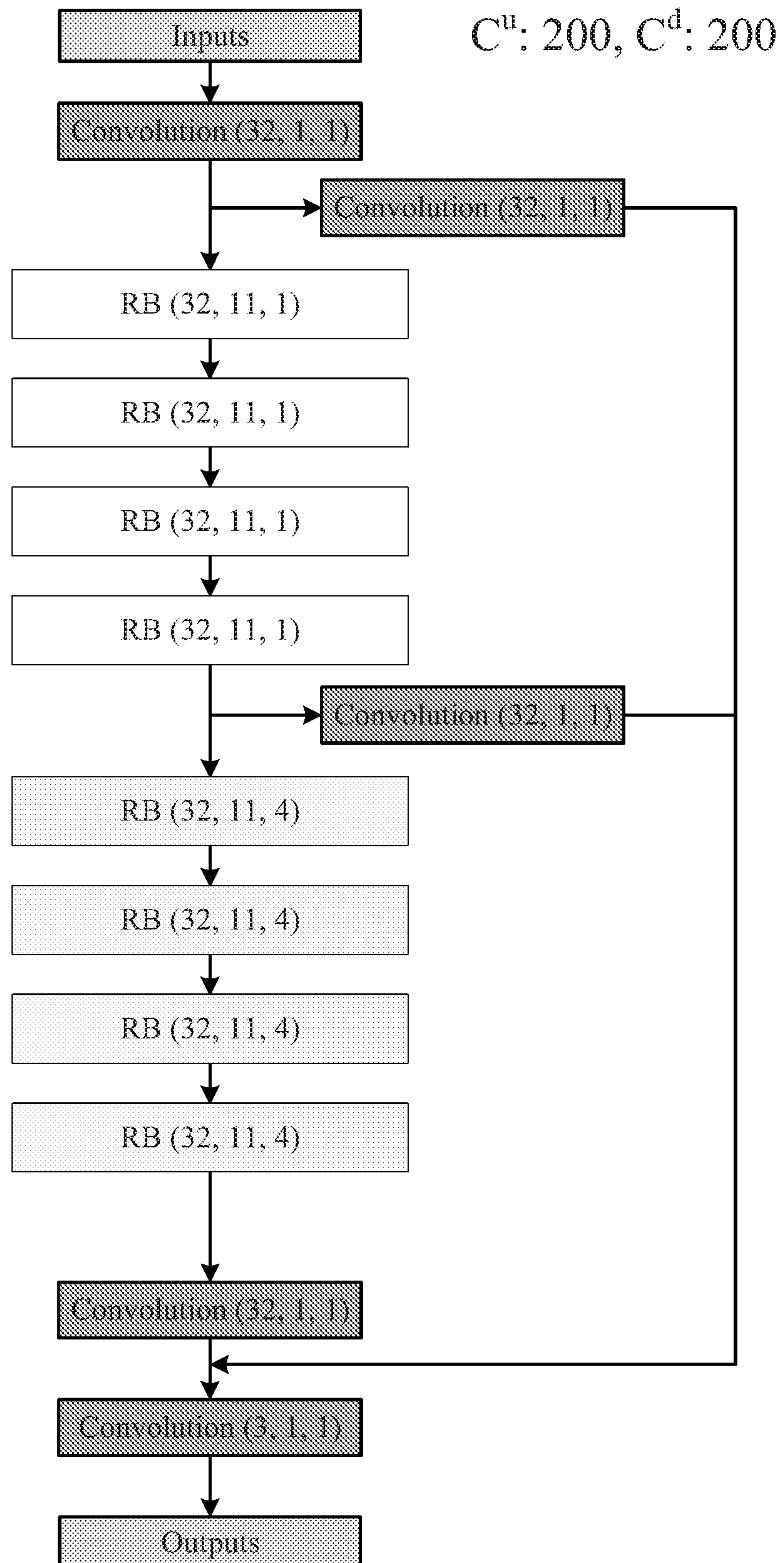


FIG. 22

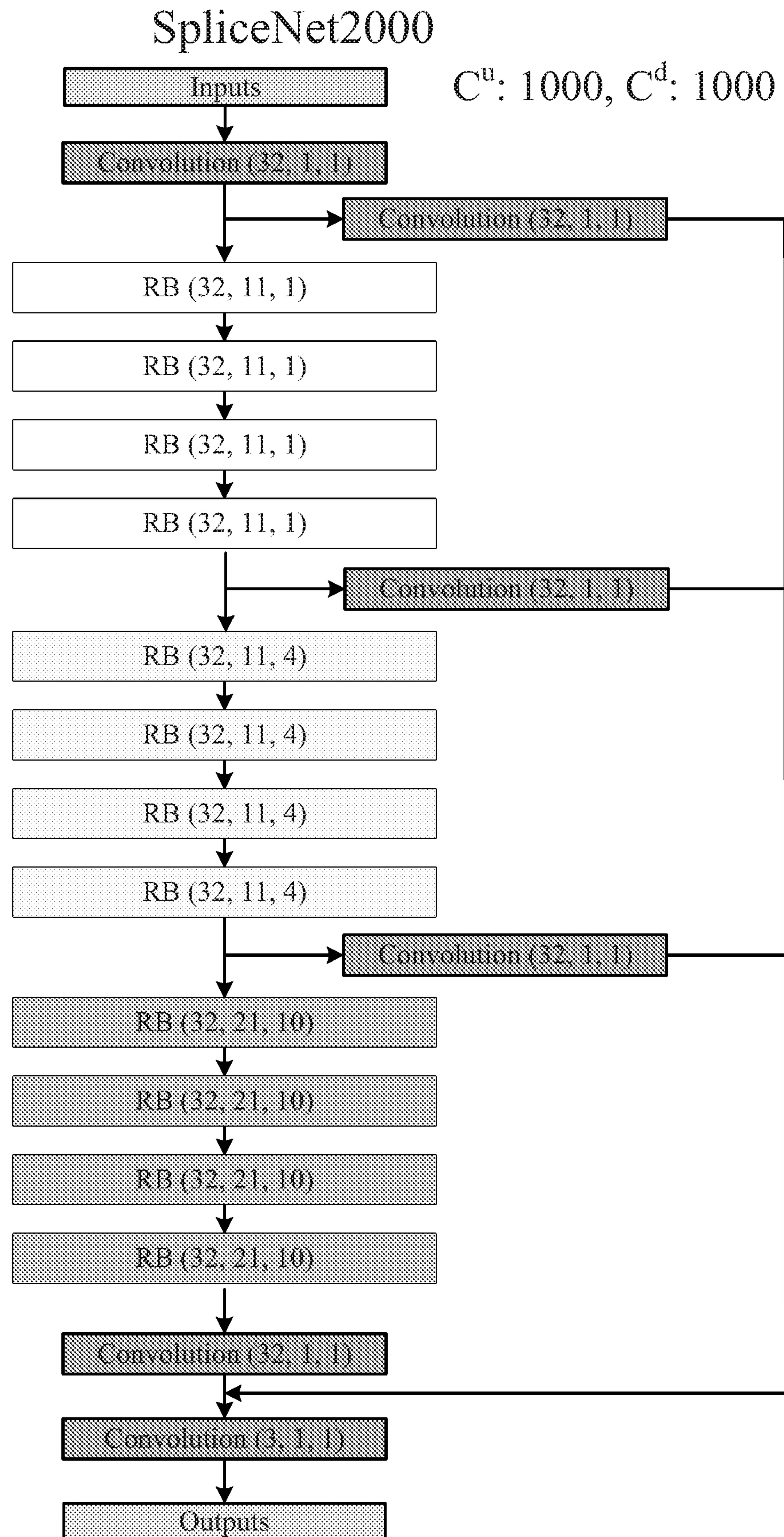


FIG. 23

SpliceNet10000

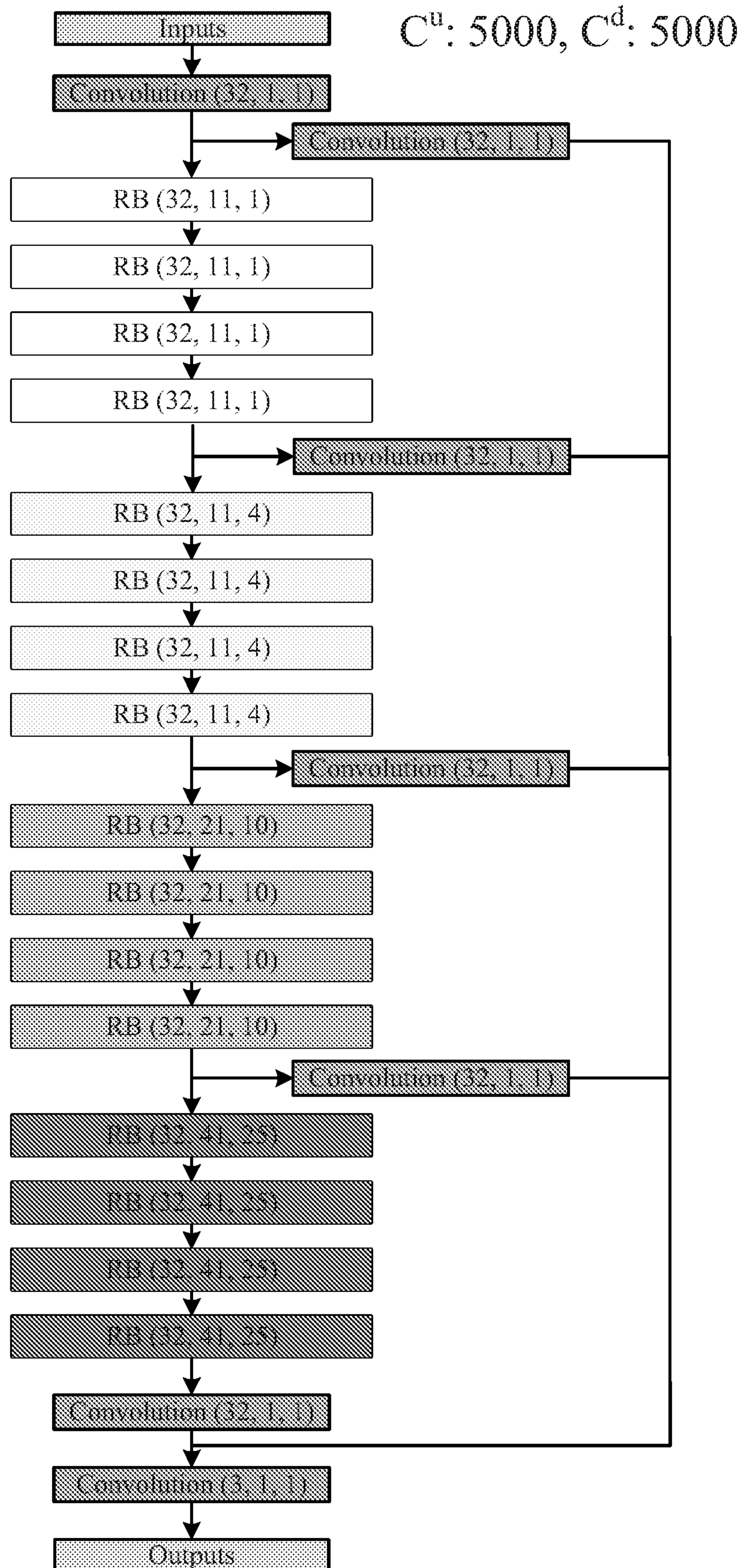
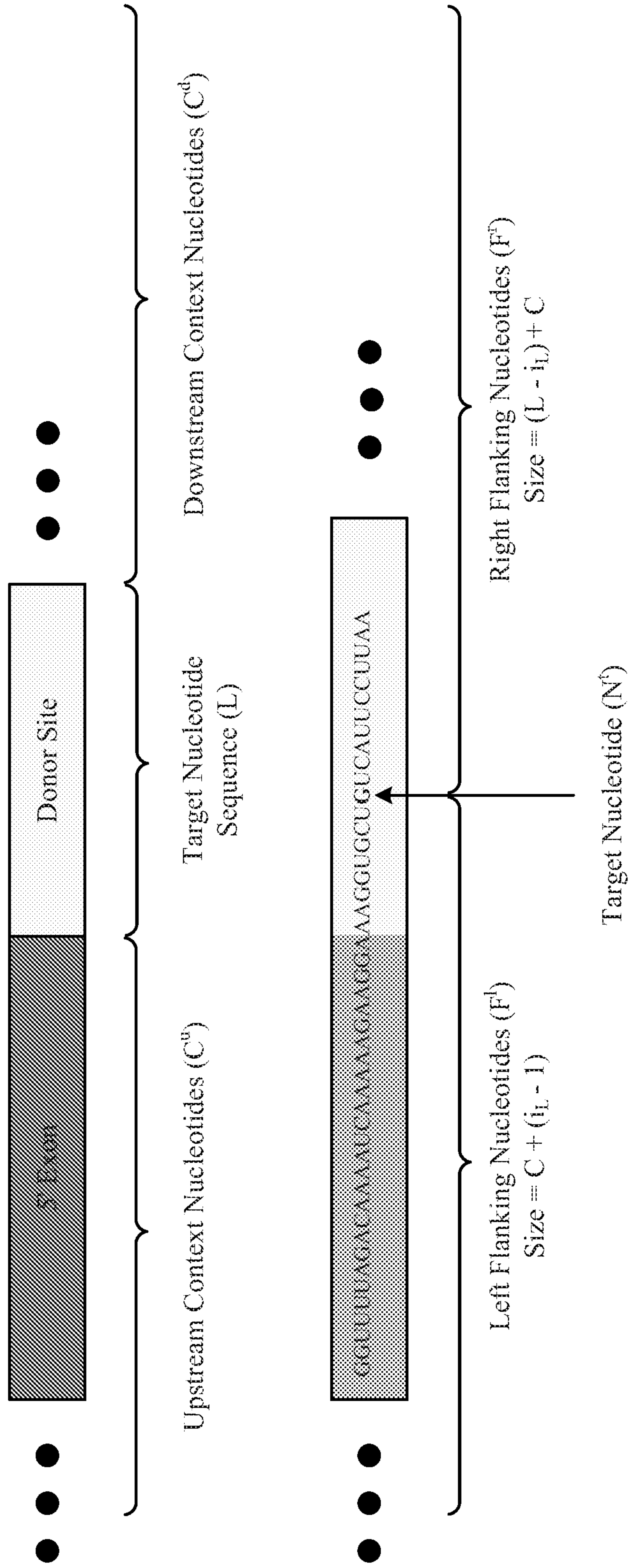


FIG. 24

Reference Genome (annotated)



C^u : 40, 200, 1000, 5000
 C^d : 40, 200, 1000, 5000
 L : 5000

FIG. 25

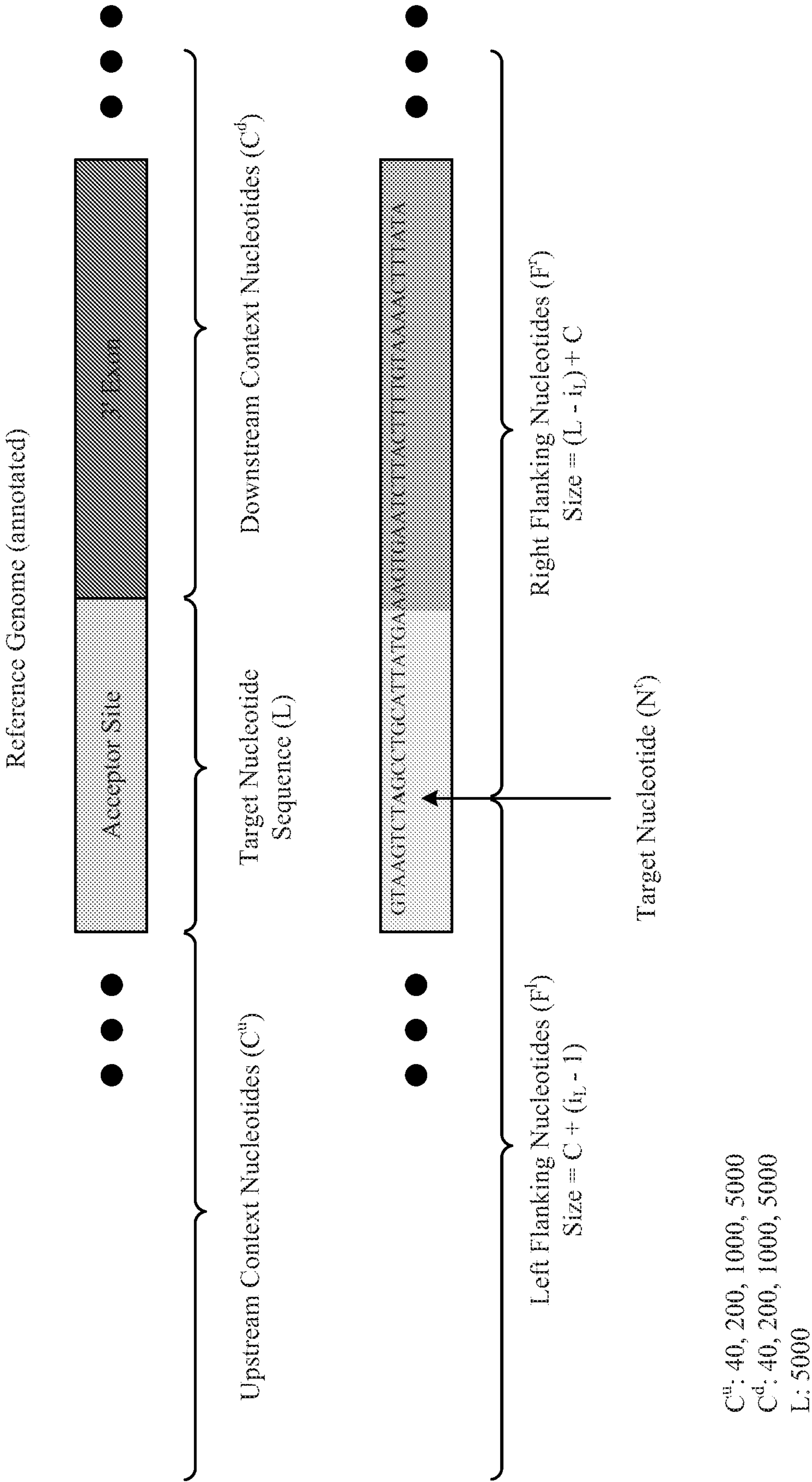


FIG. 26

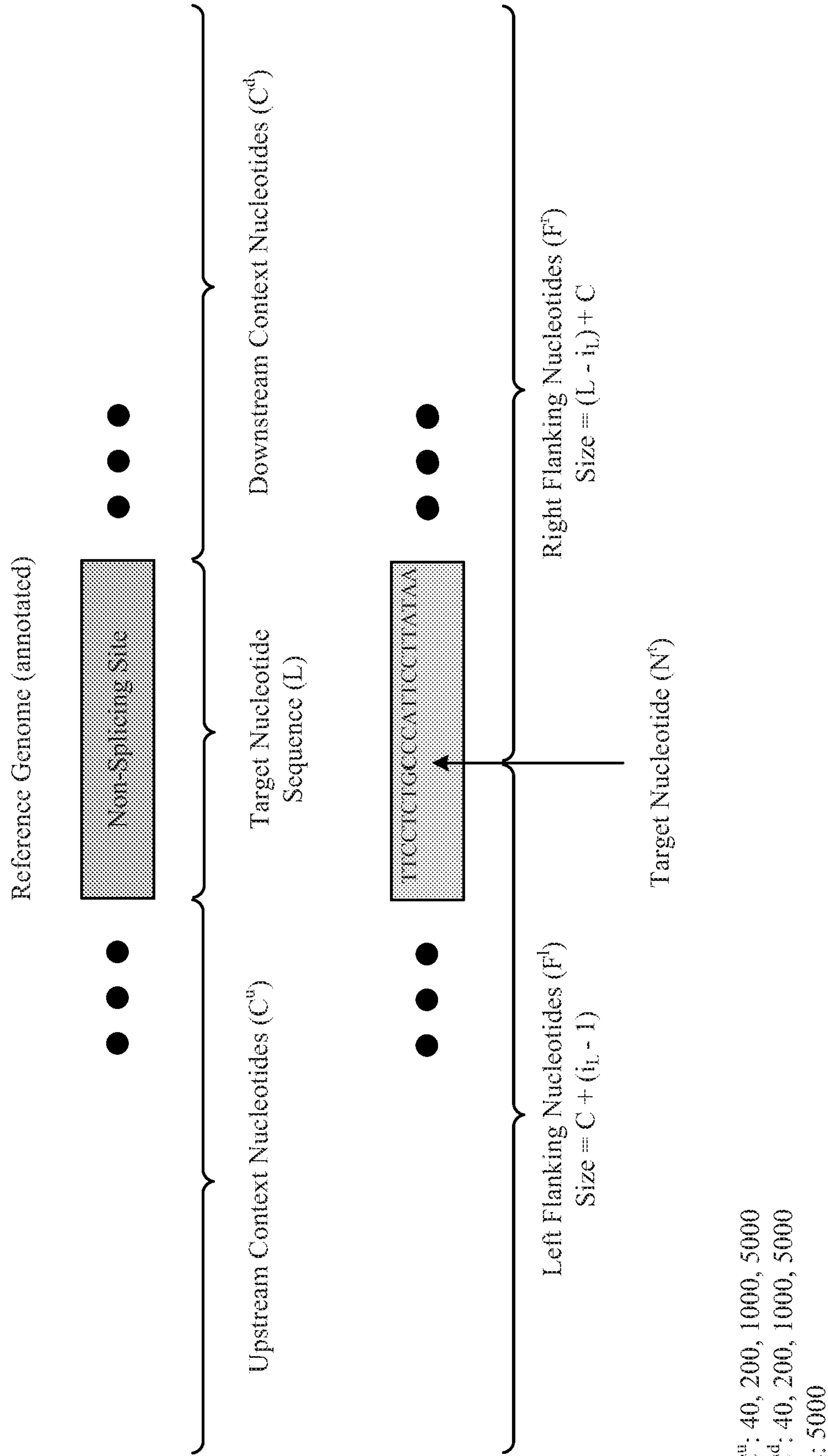
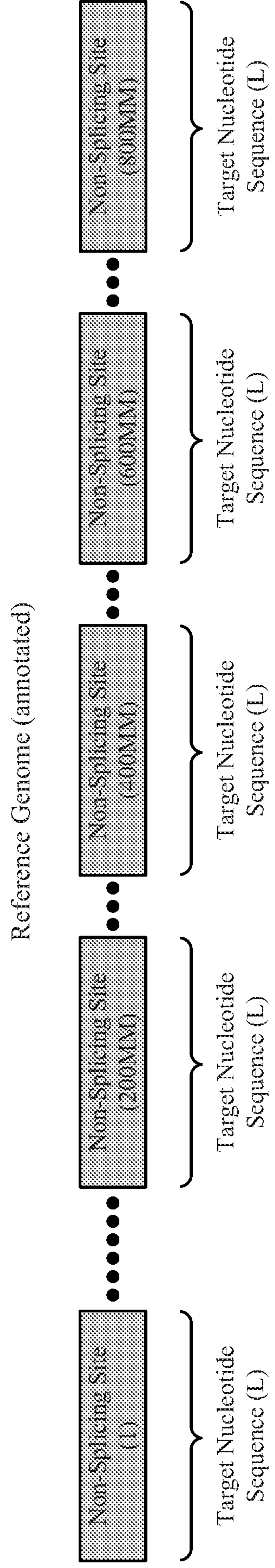


FIG. 27



L: 5000

FIG. 28

	A	C	G	T	T	G	C	A	C	A	C	G	?	G	T	A	T	A	T	
A	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	1	0
C	0	1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0
G	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
T	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

One-hot Encoding

One-hot Encoding

A C G T T T G C A C A C A C G ? G T A T A T

FIG. 29

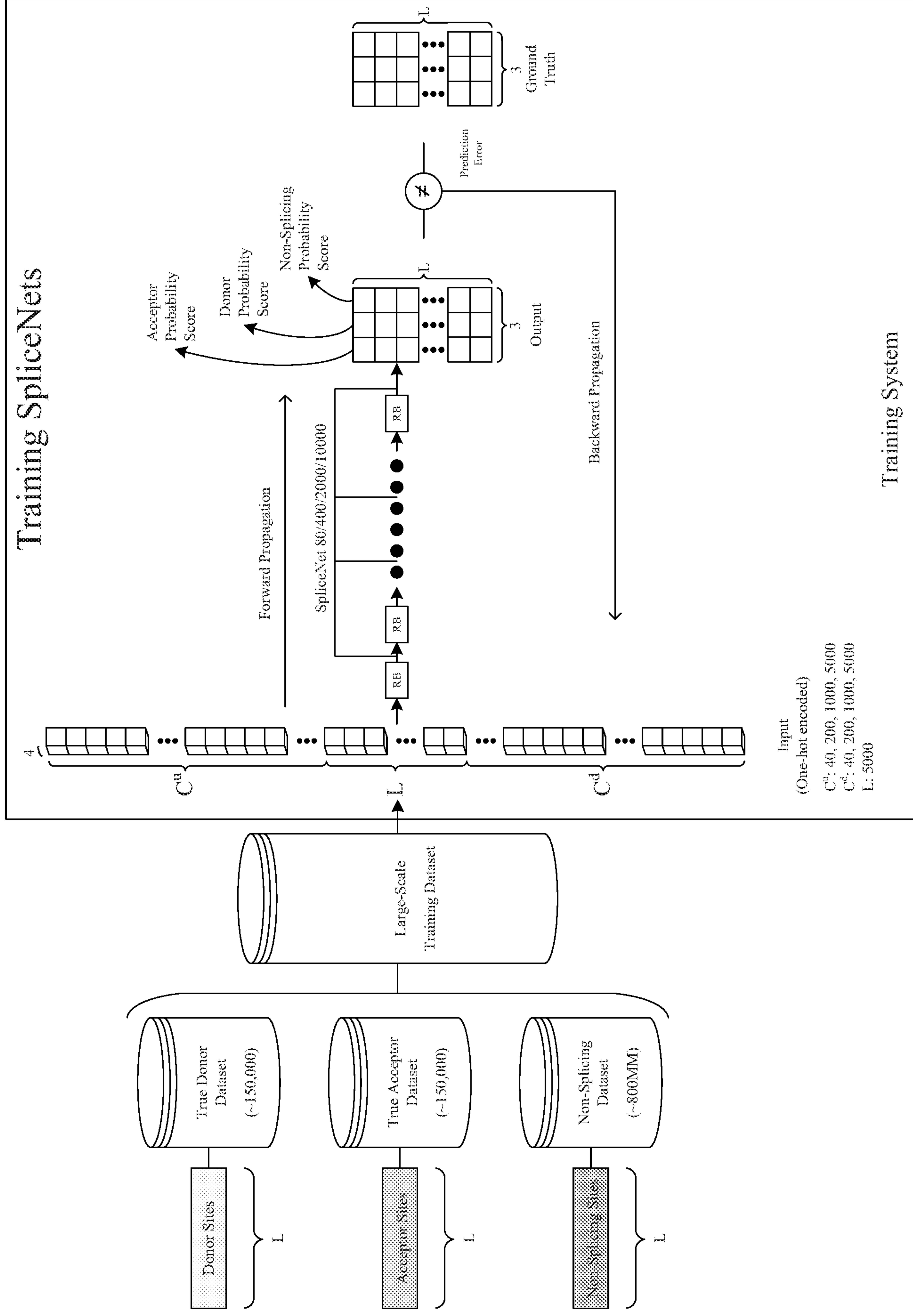


FIG. 30

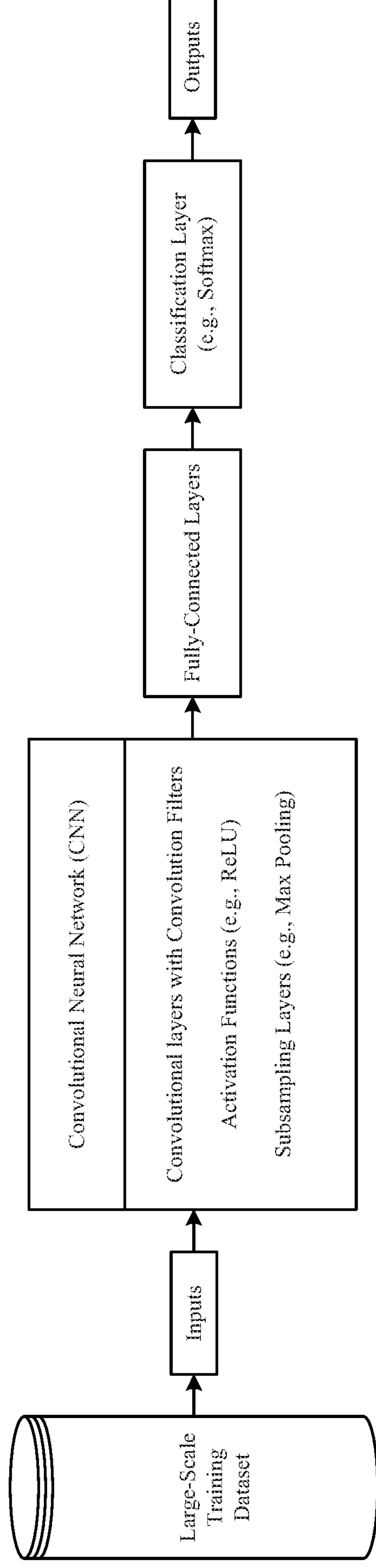


FIG. 31

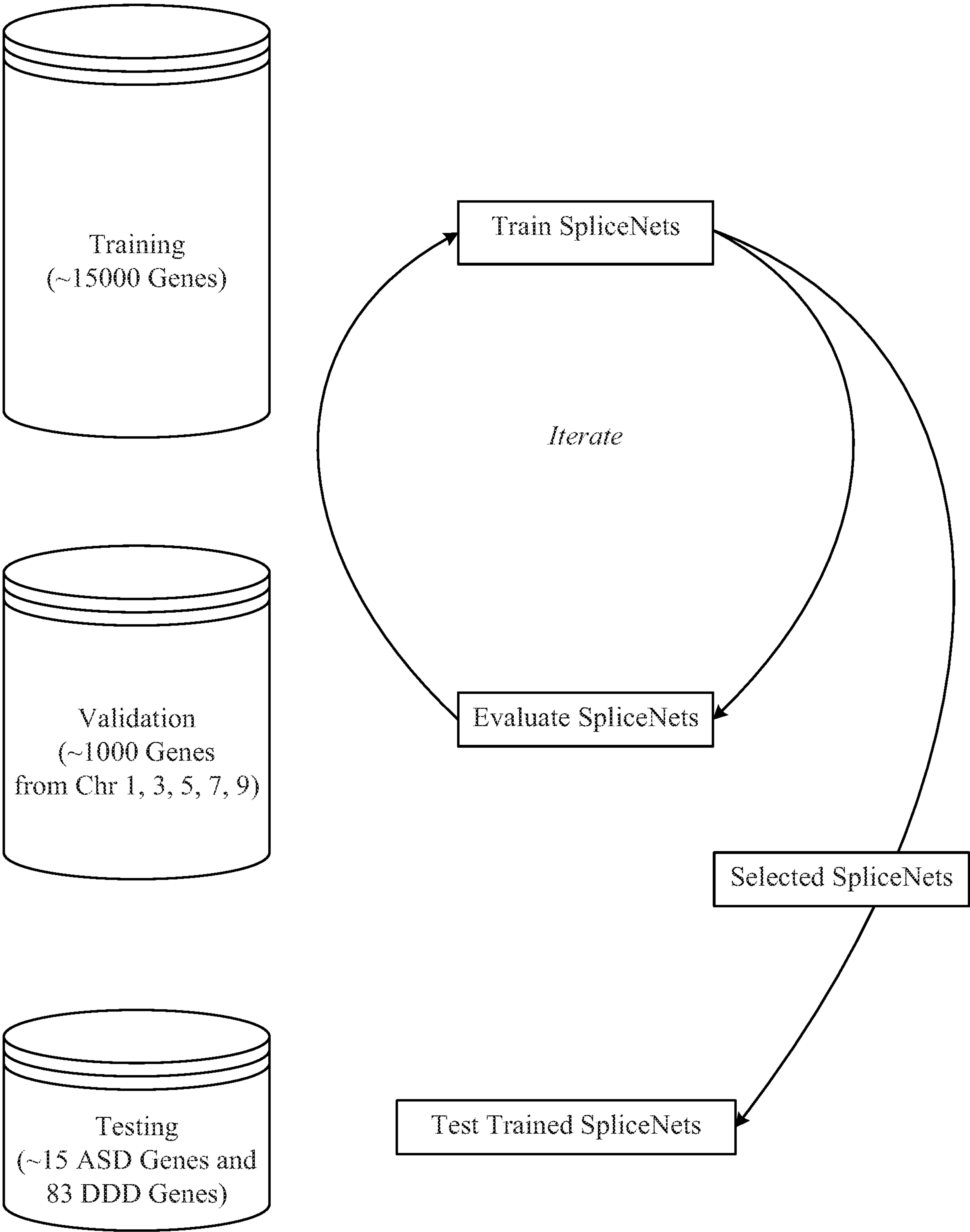


FIG. 32

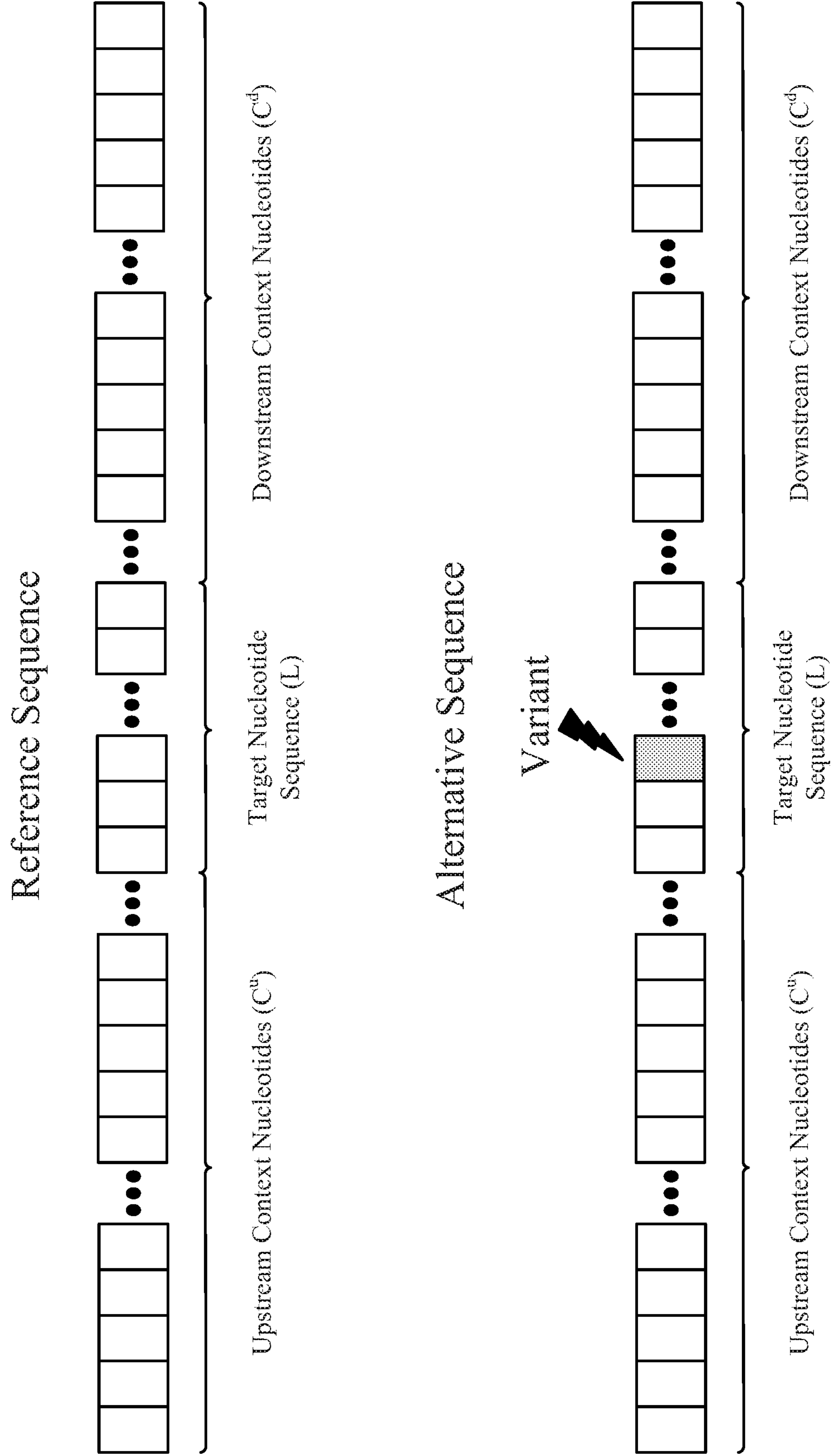
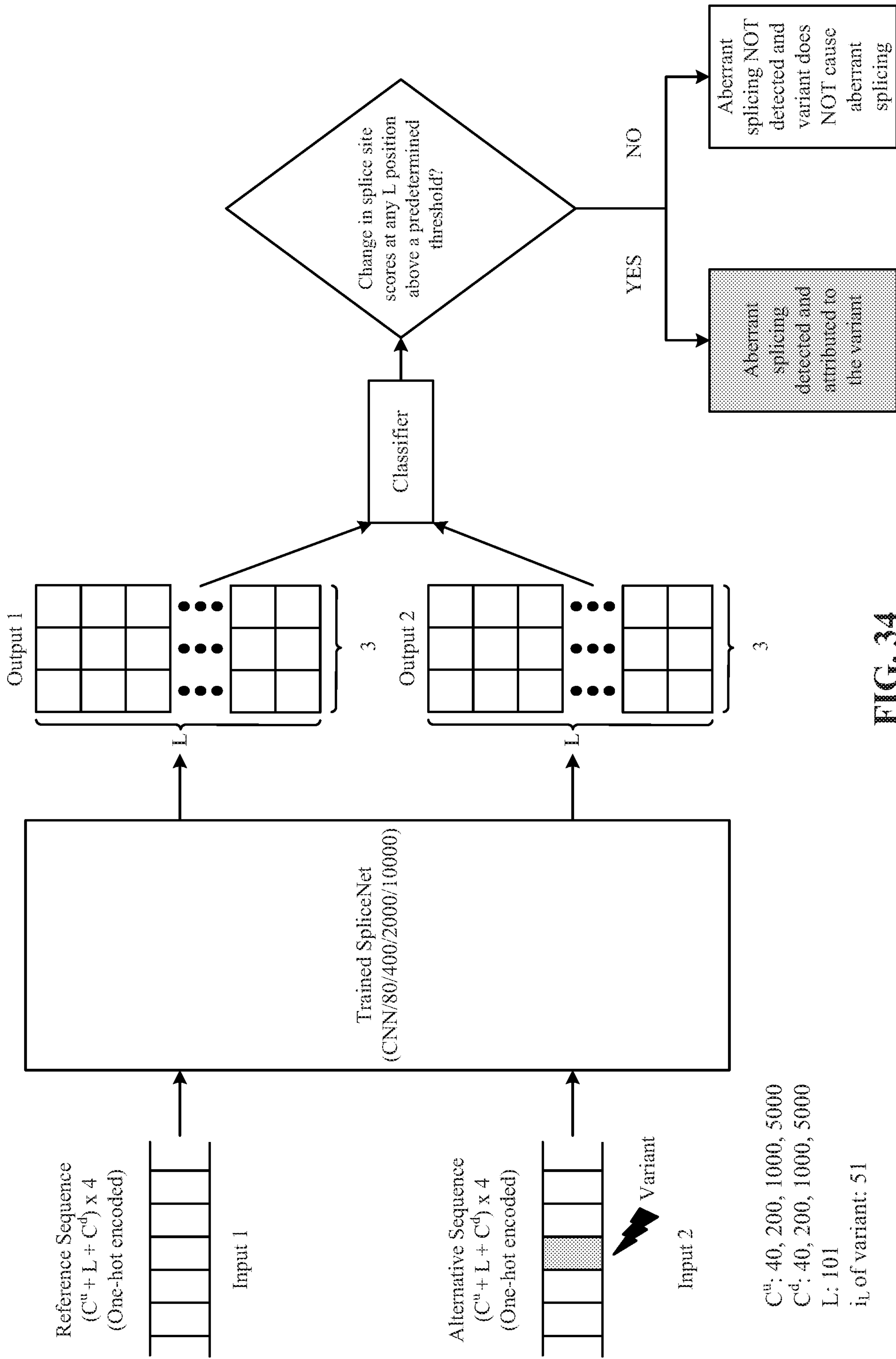


FIG. 33

Aberrant Splicing Detection



C^u : 40, 200, 1000, 5000
 C^d : 40, 200, 1000, 5000
 L : 101
 i_L of variant: 51

FIG. 34

SpliceNet10000 Processing Pyramid for Splice Site Classification

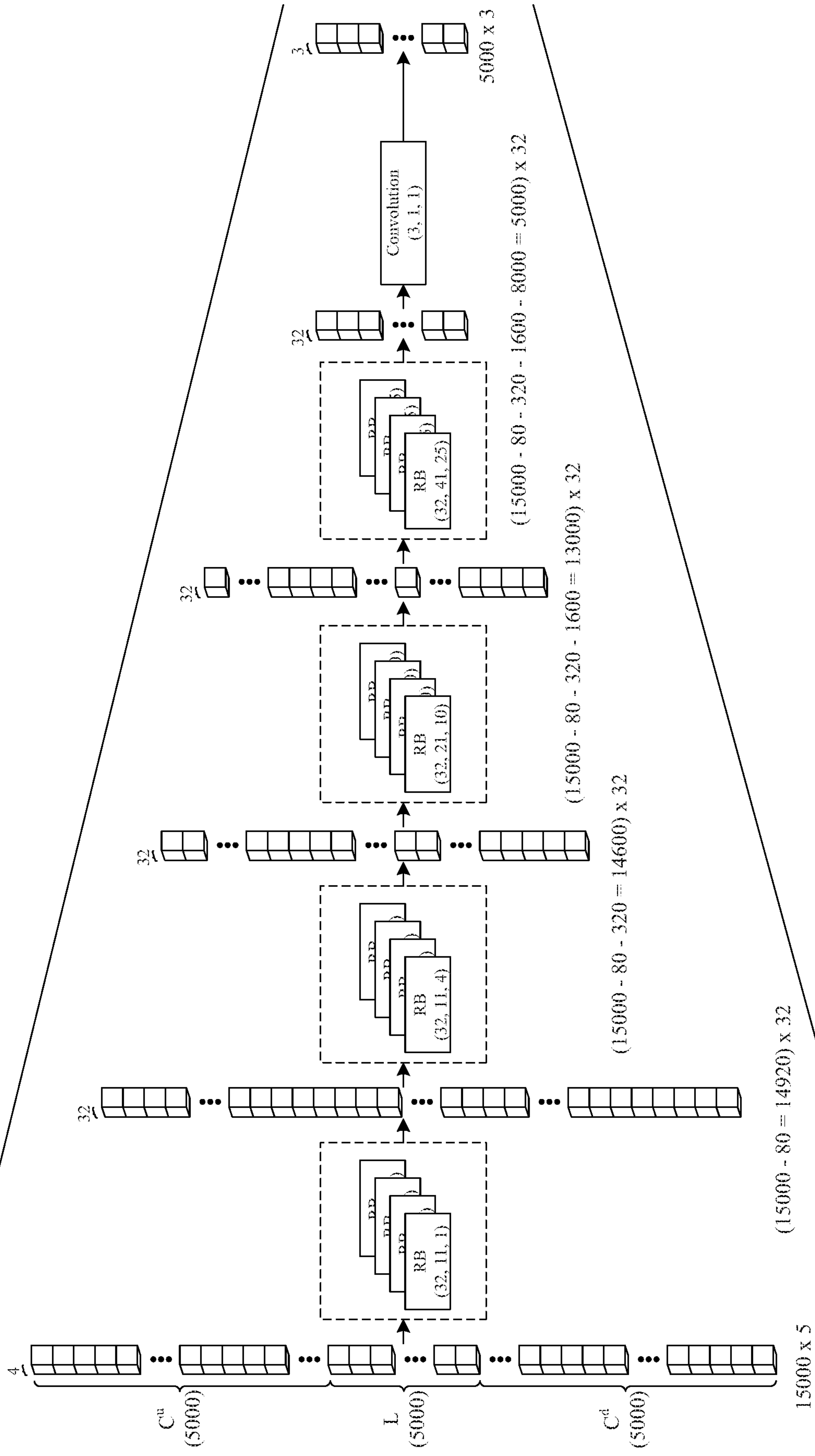


FIG. 35

SpliceNet10000 Processing Pyramid for Aberrant Splicing Detection

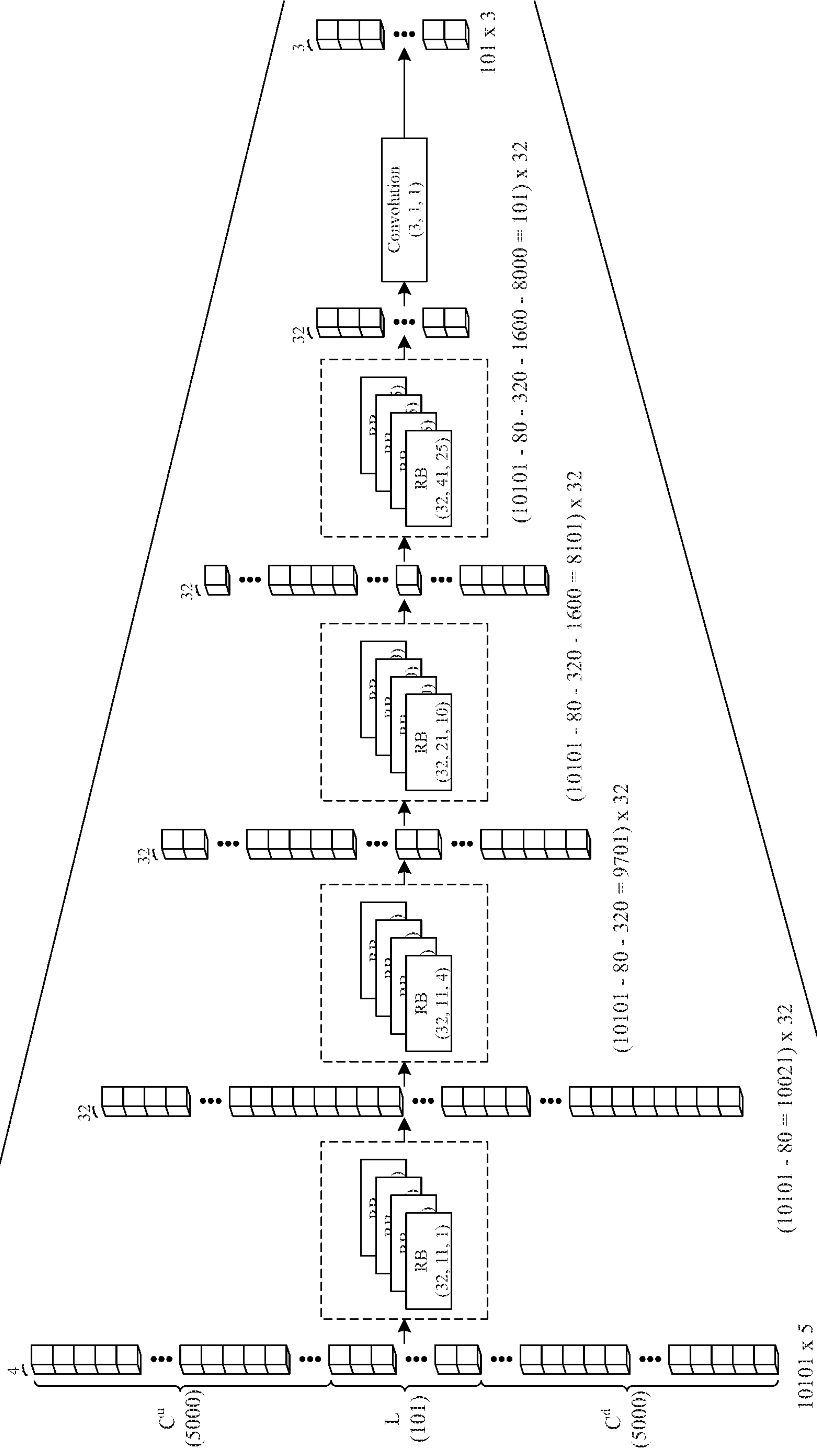
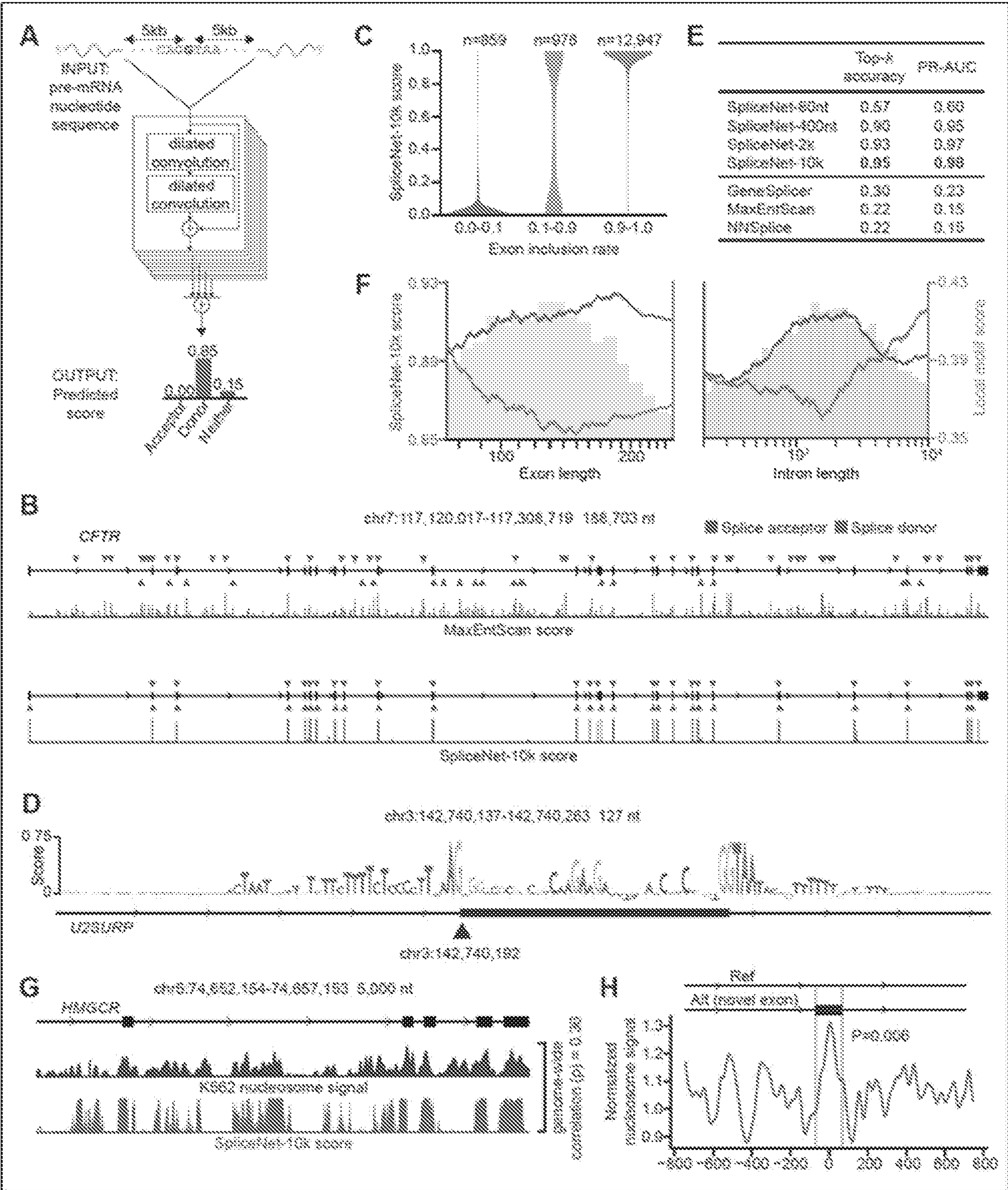
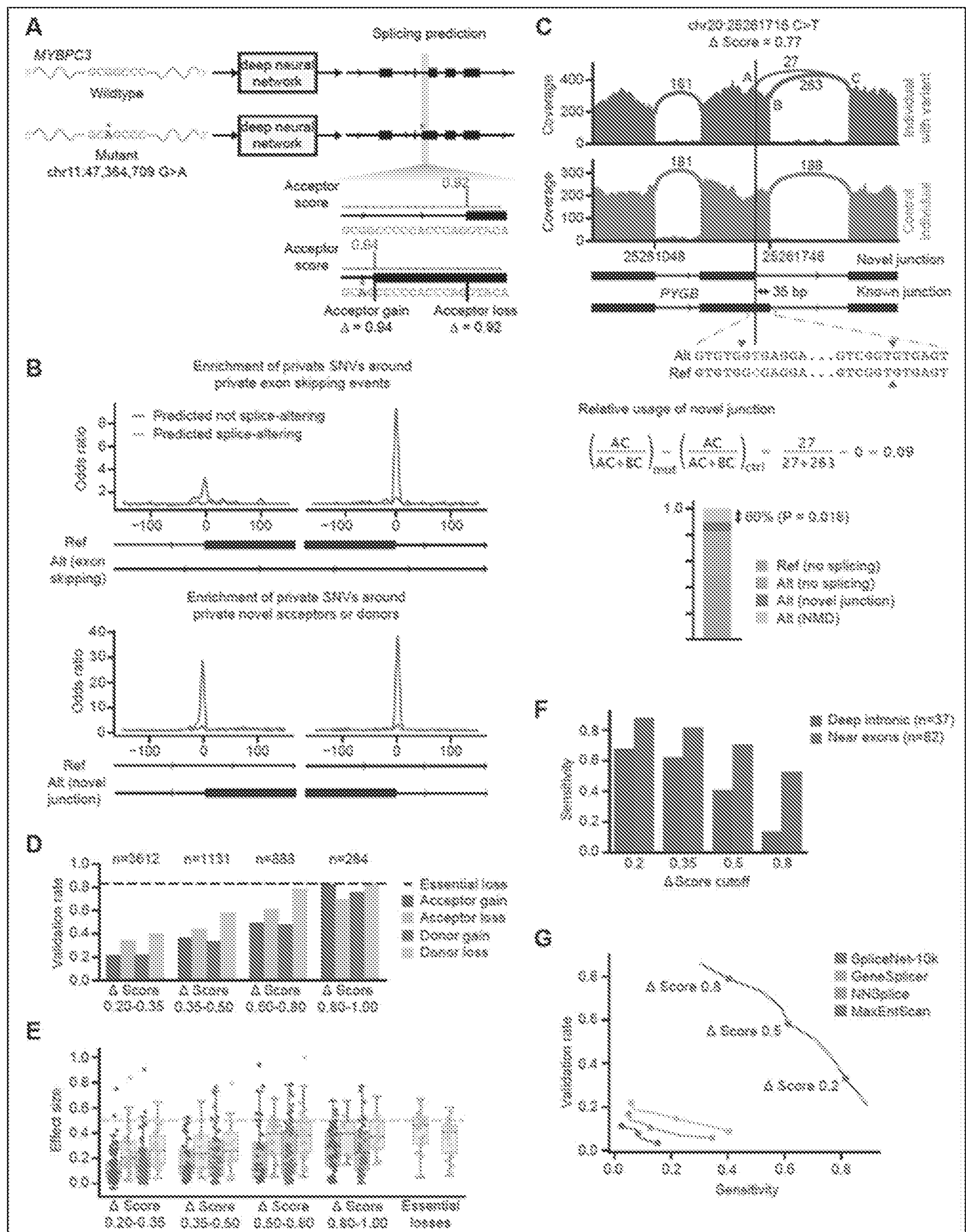


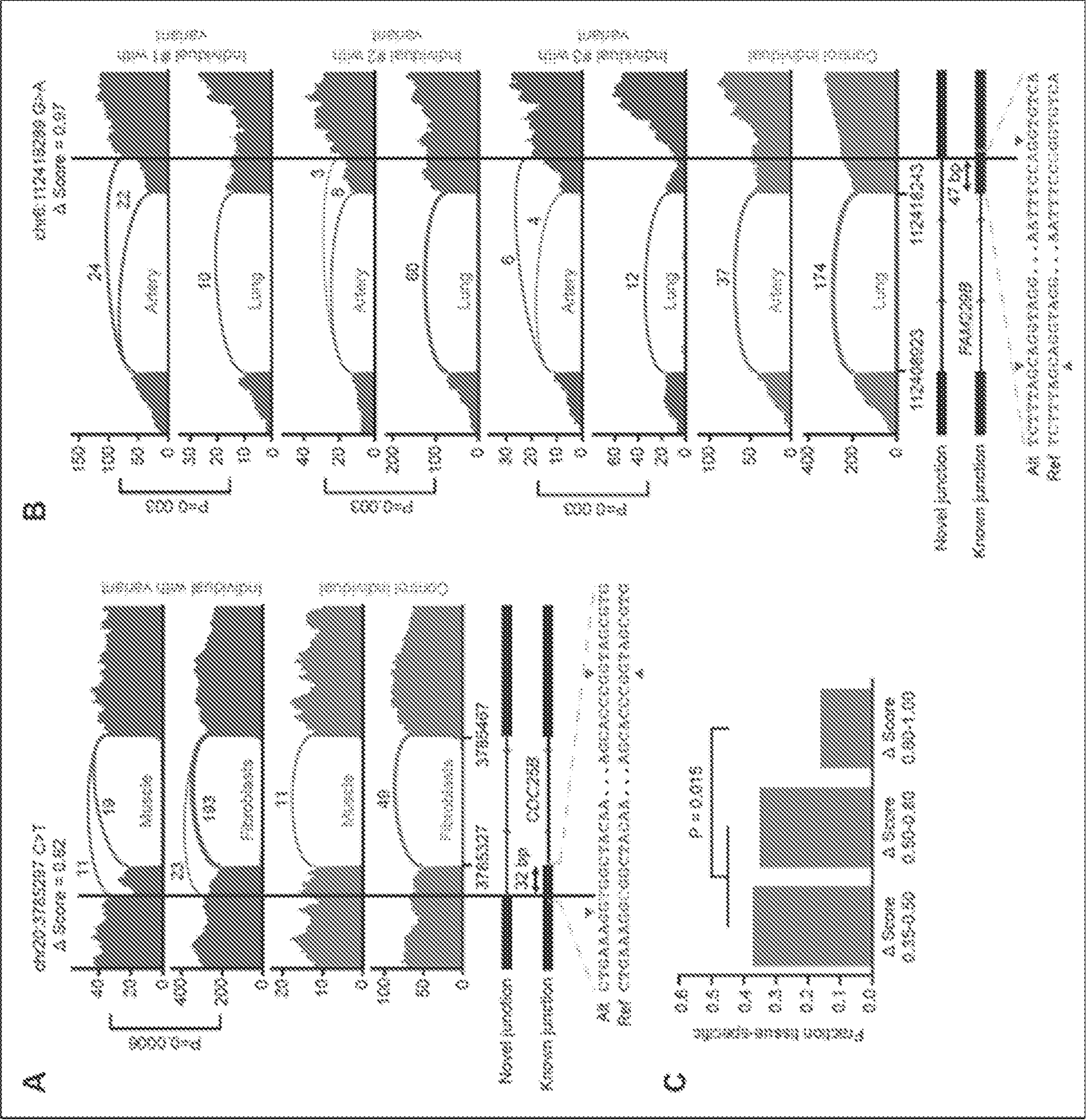
FIG. 36



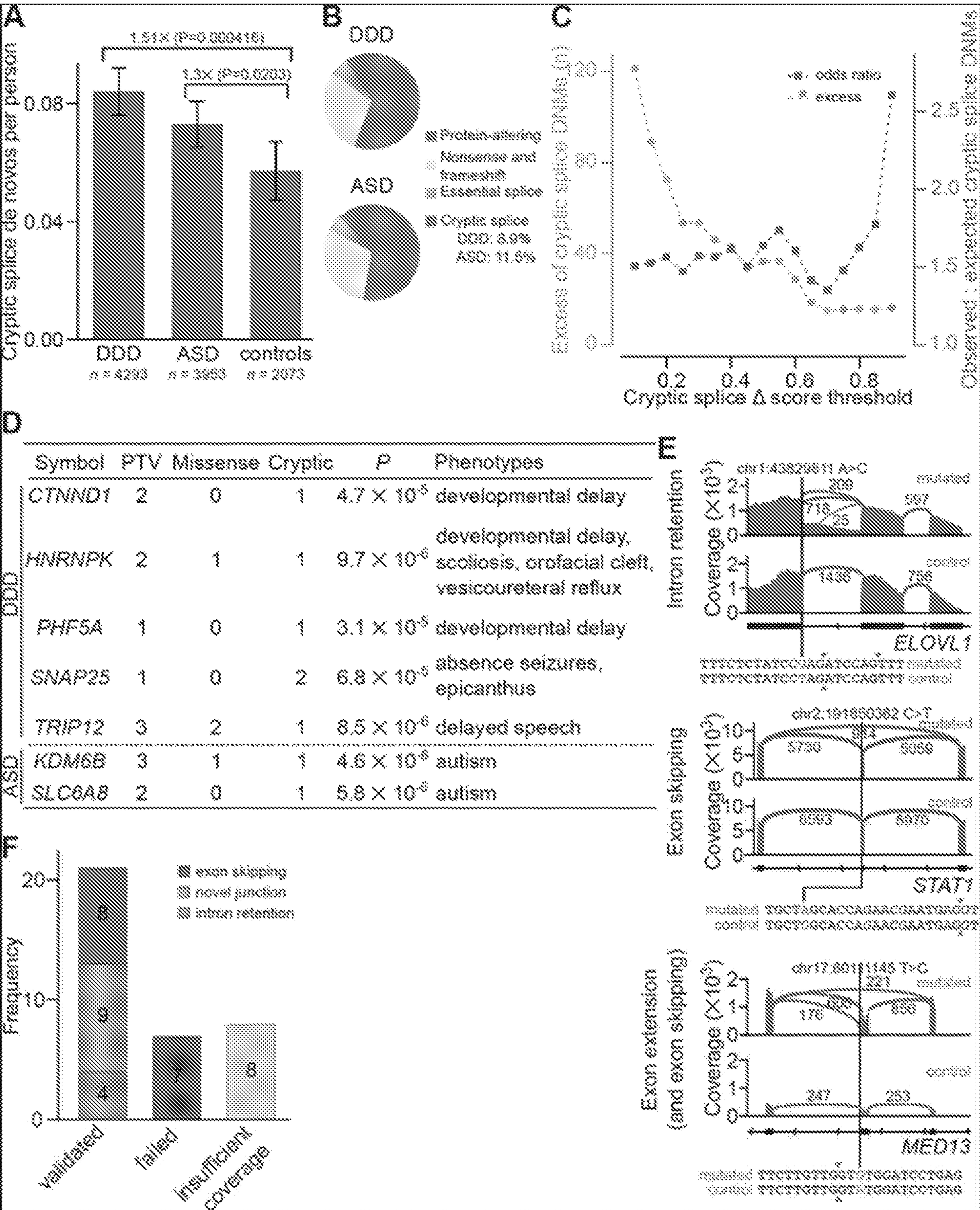
FIGs. 37A-H



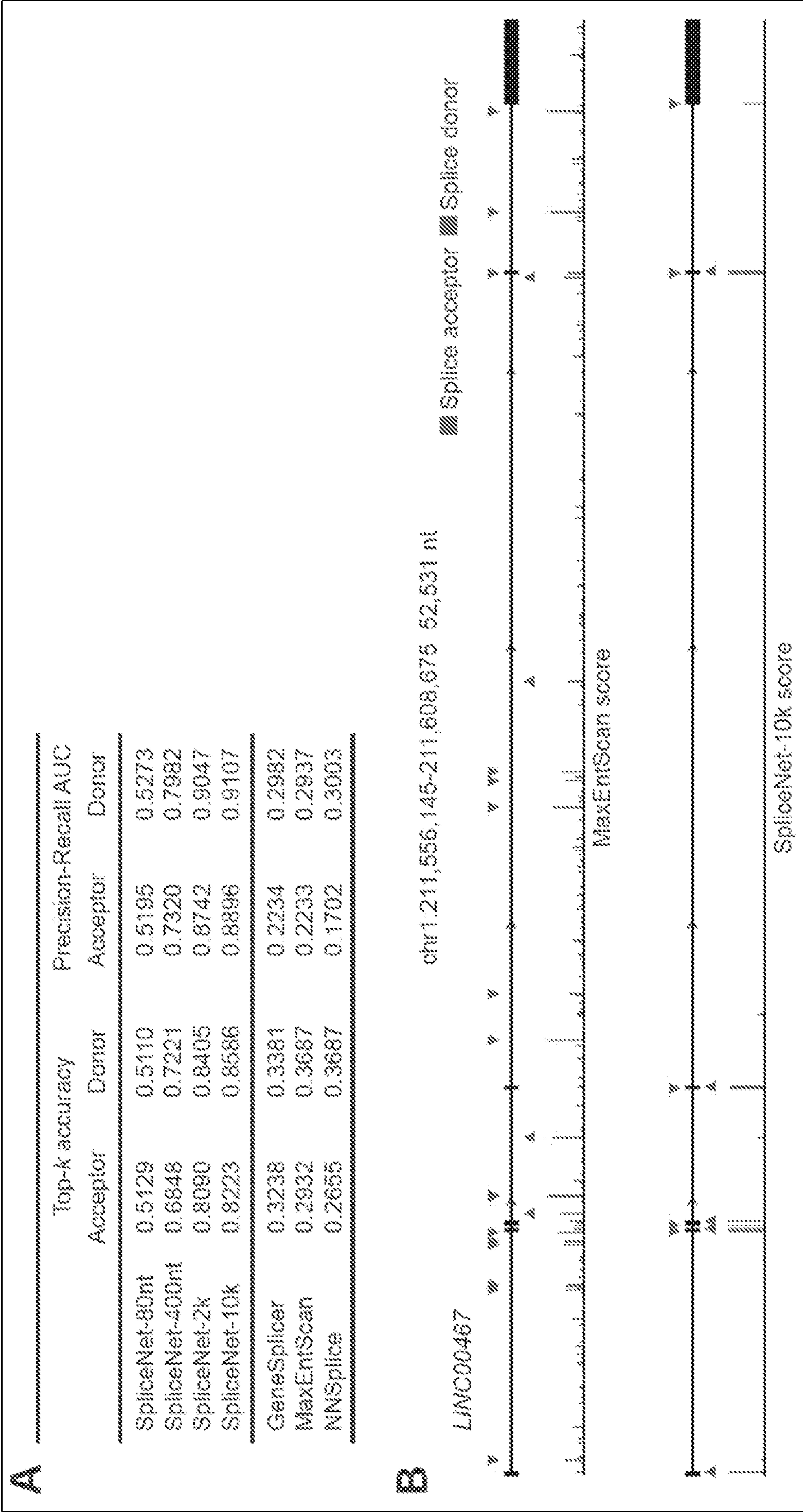
FIGs. 38A-G



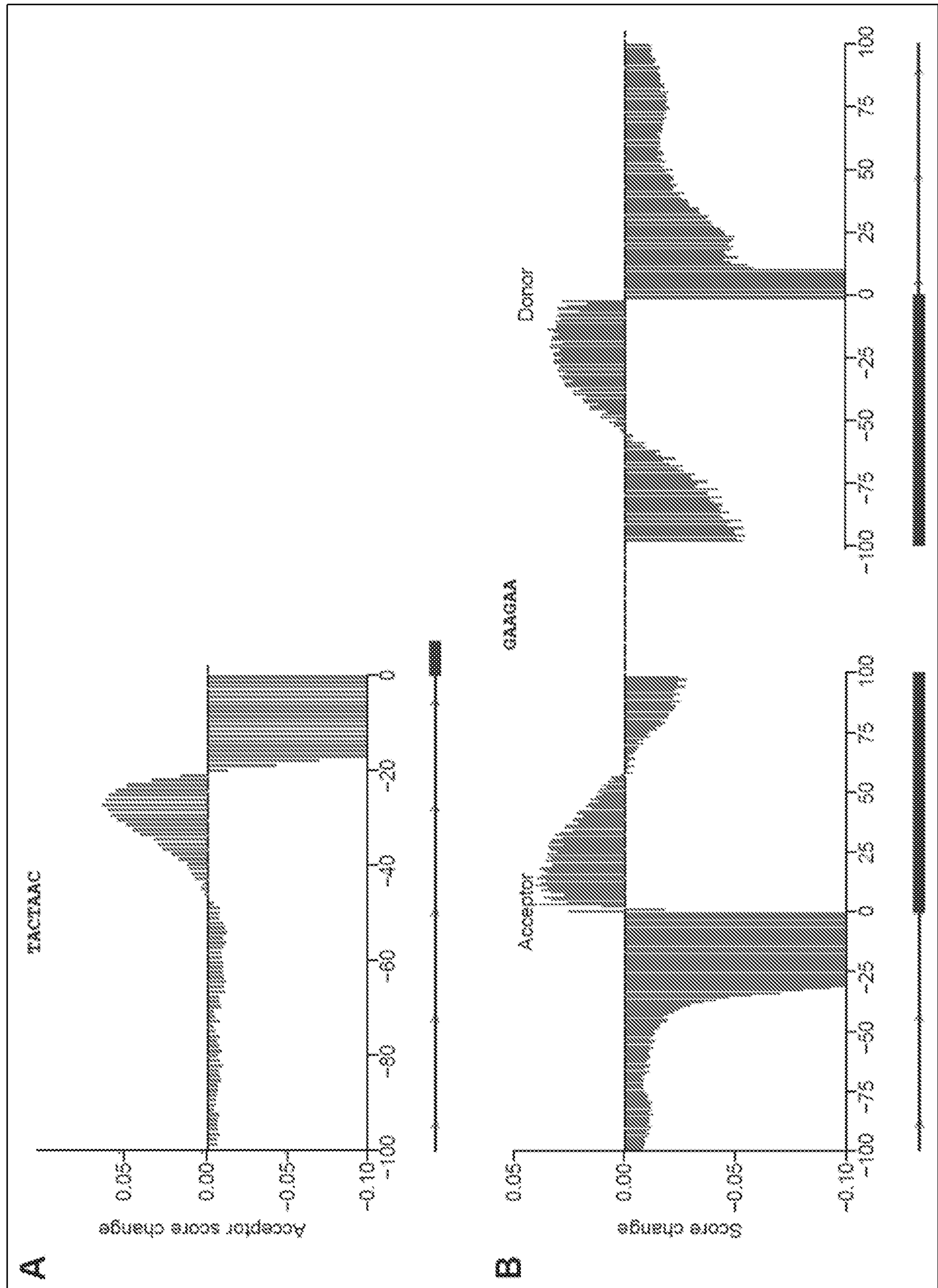
FIGs. 39A-C

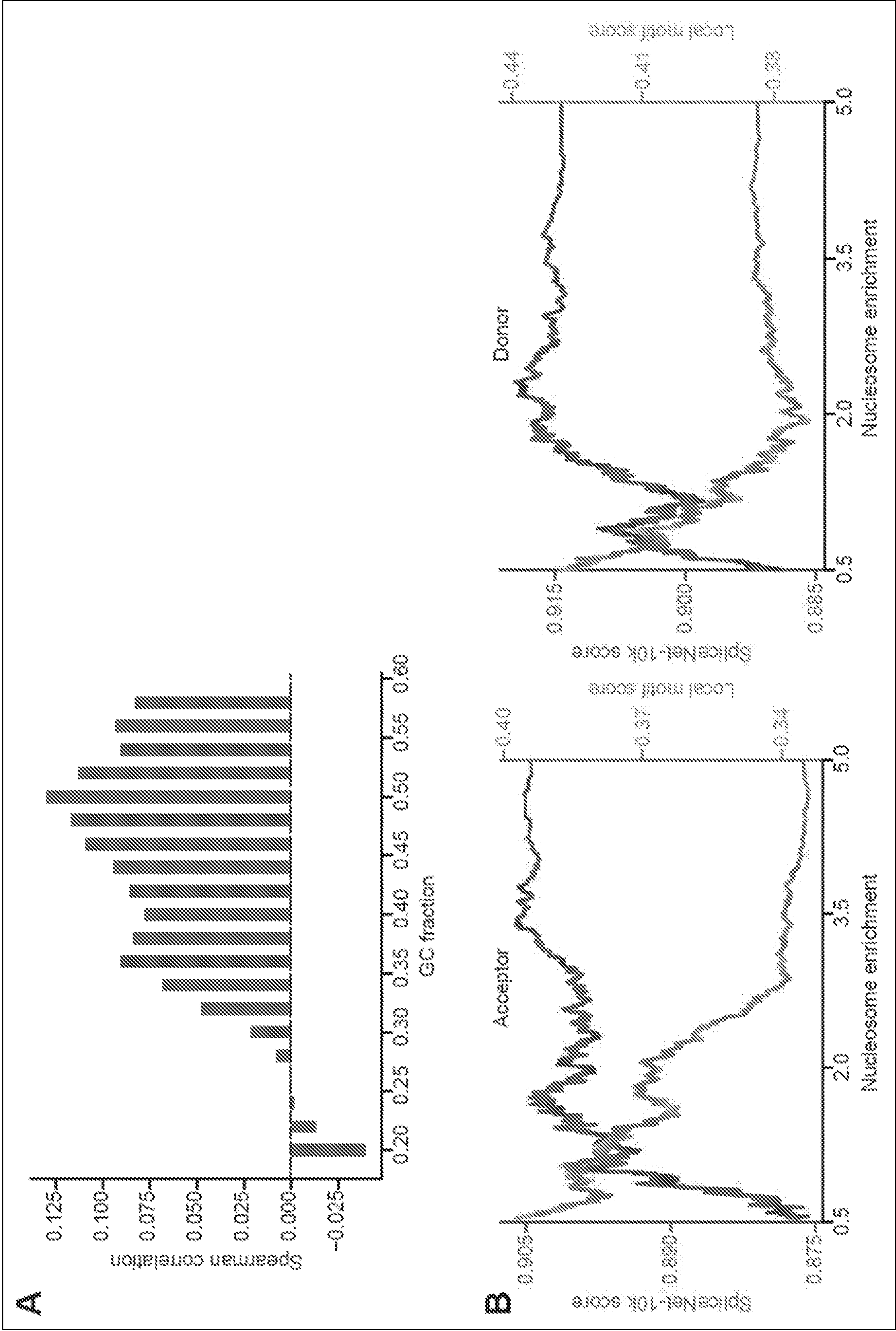


FIGs. 41A-F



FIGs. 42A-B





FIGs. 44A-B

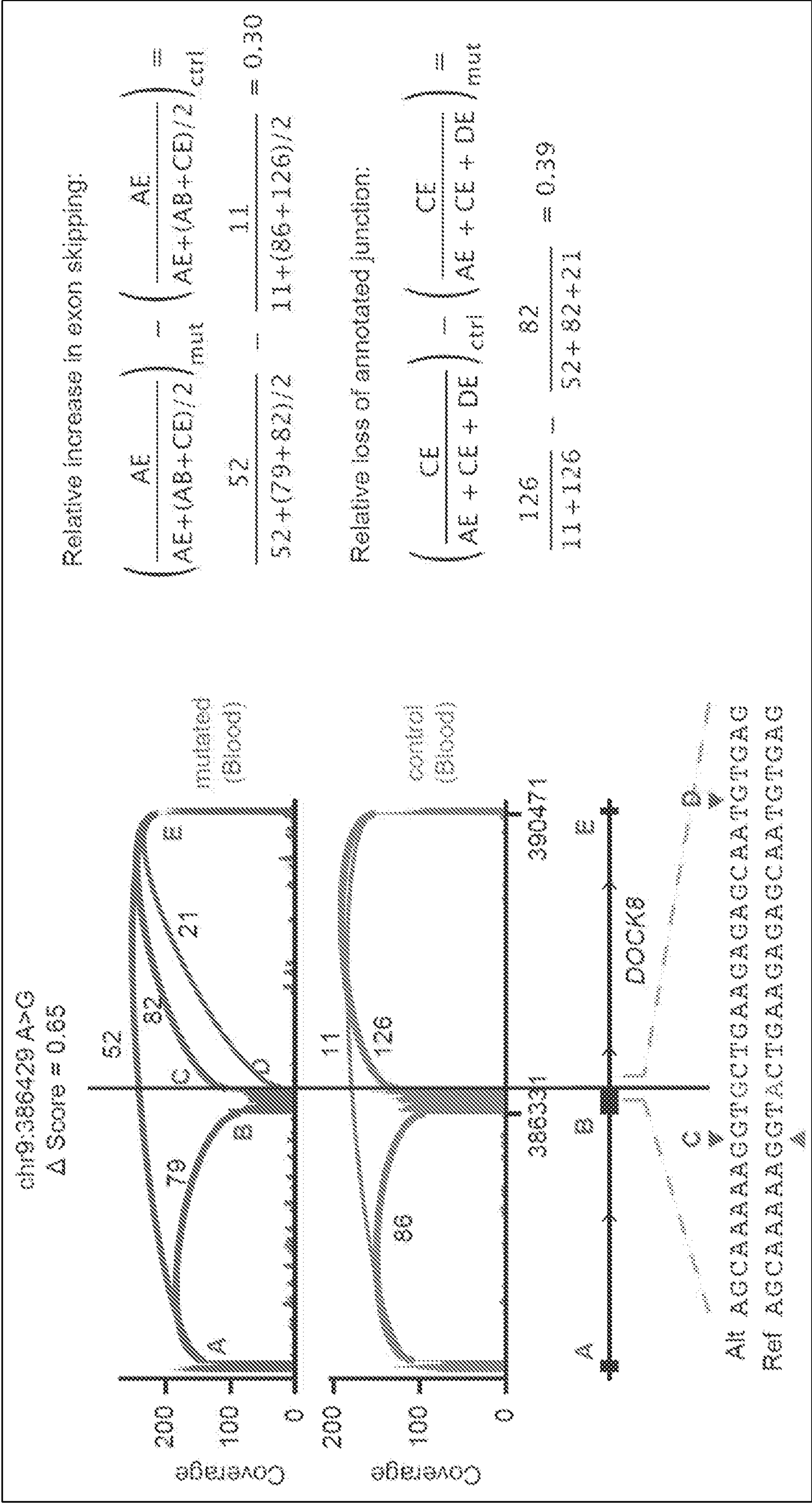
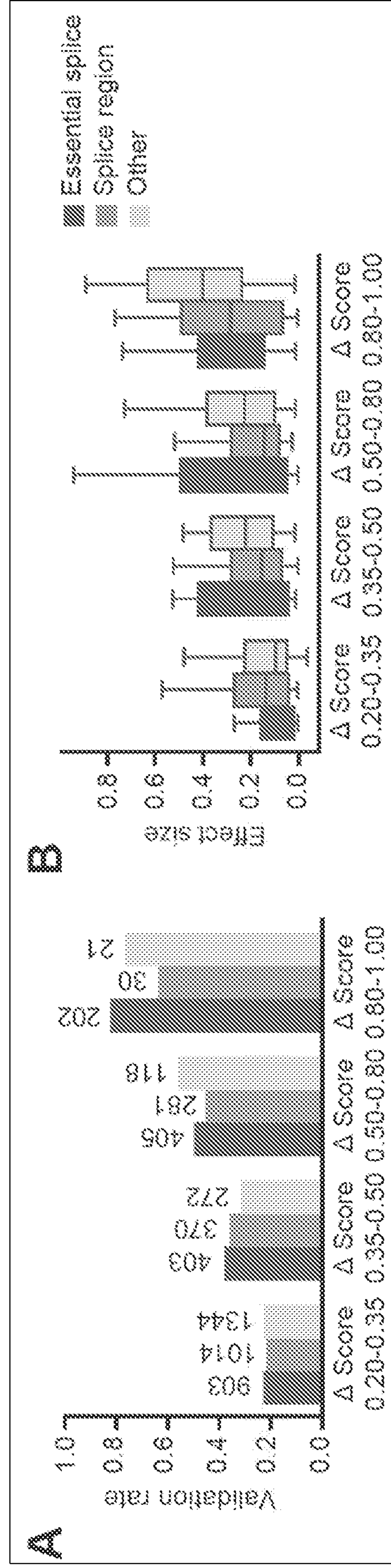
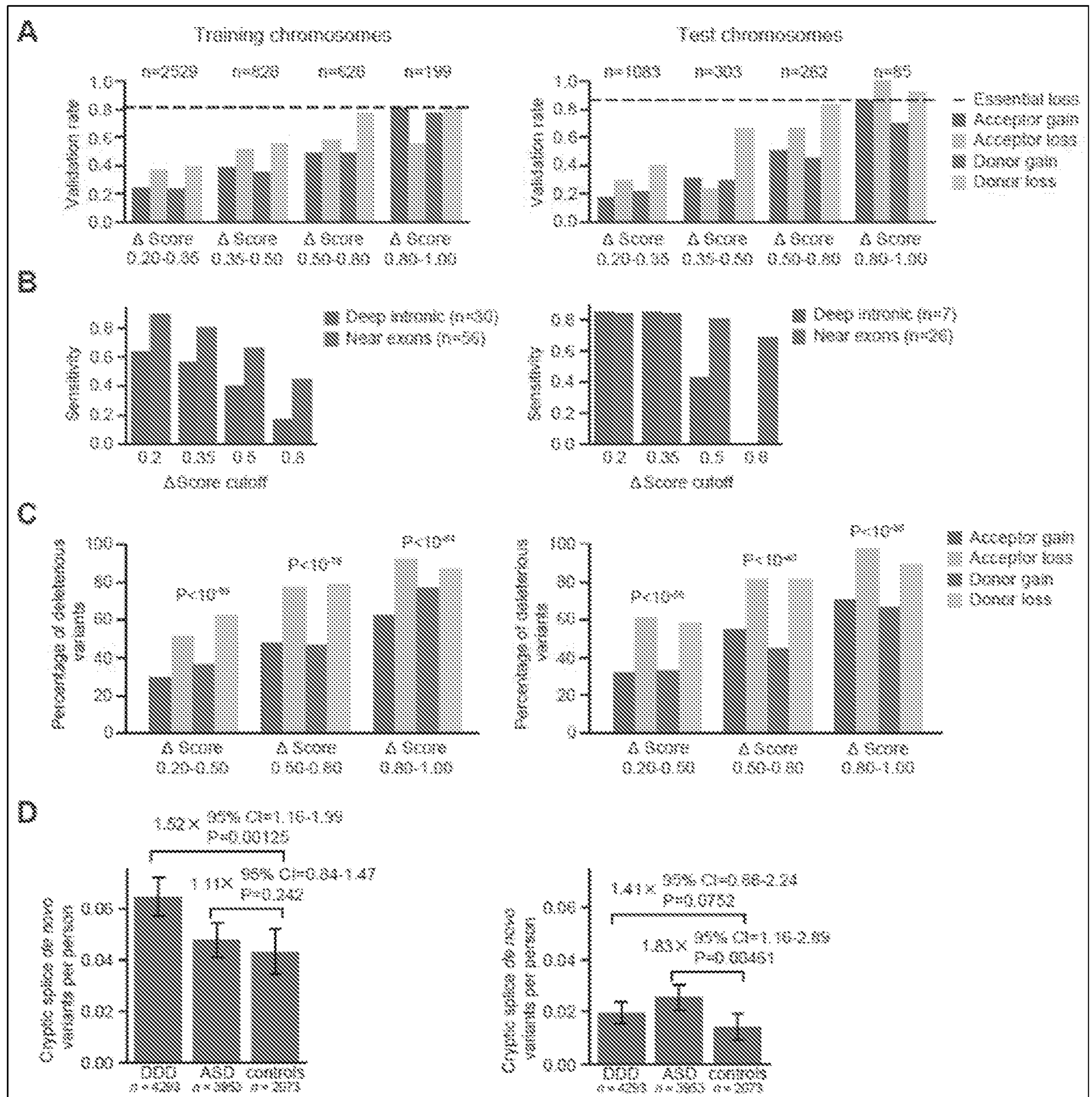


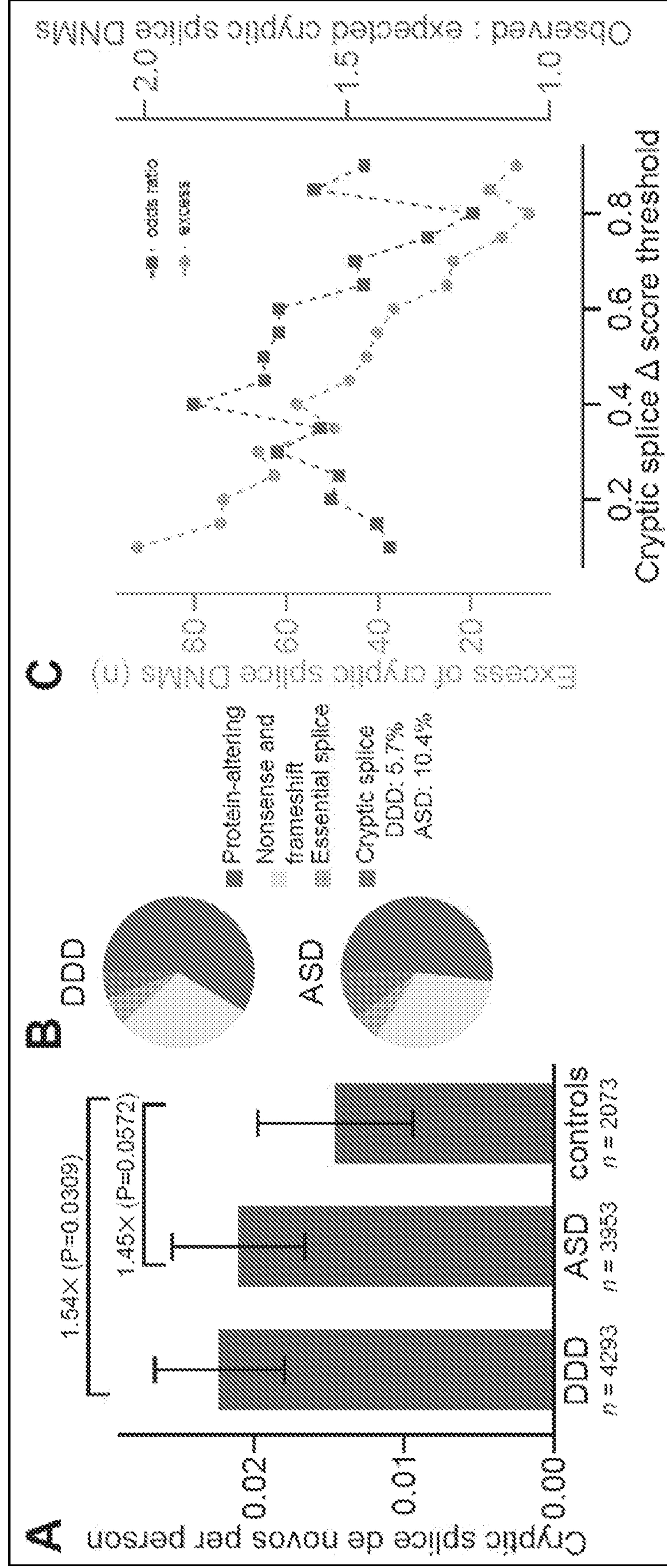
FIG. 45



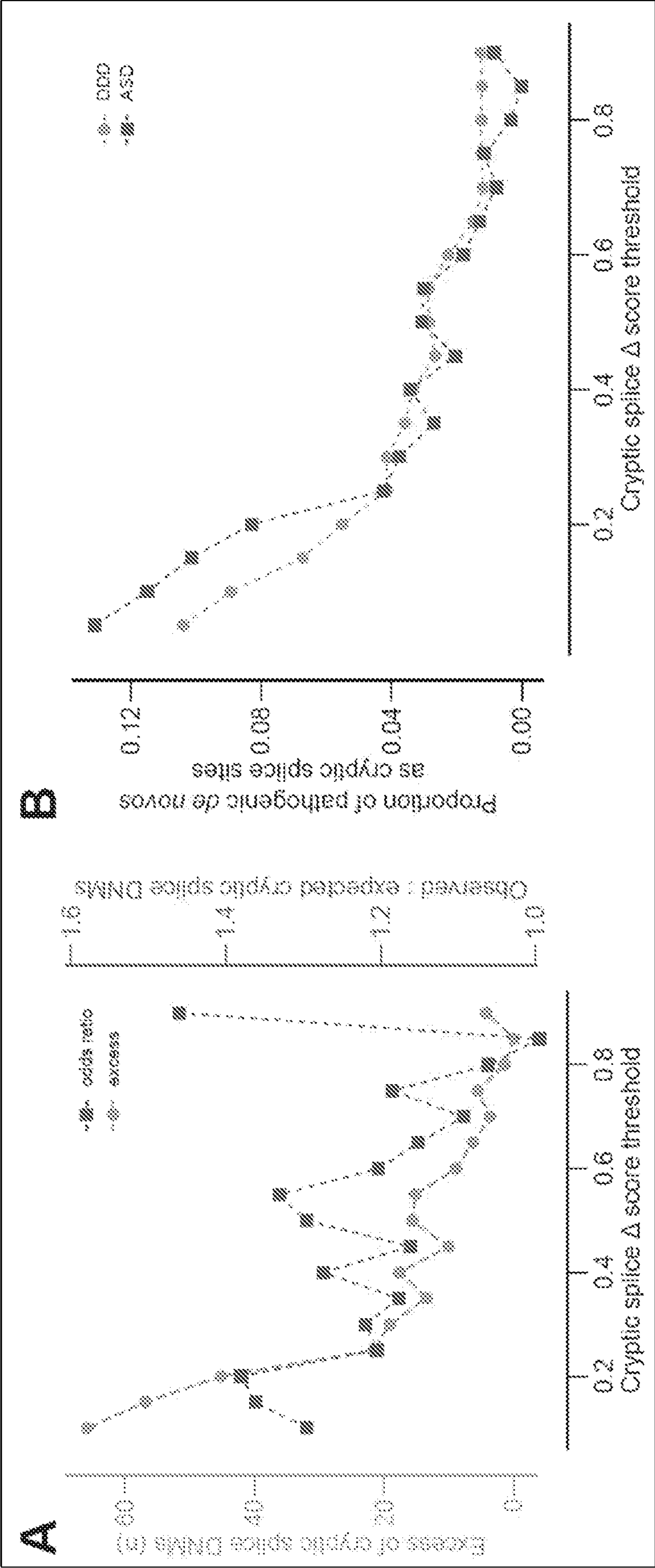
FIGs. 47A-B



FIGs. 48A-D



FIGs. 49A-C



FIGs. 50A-B

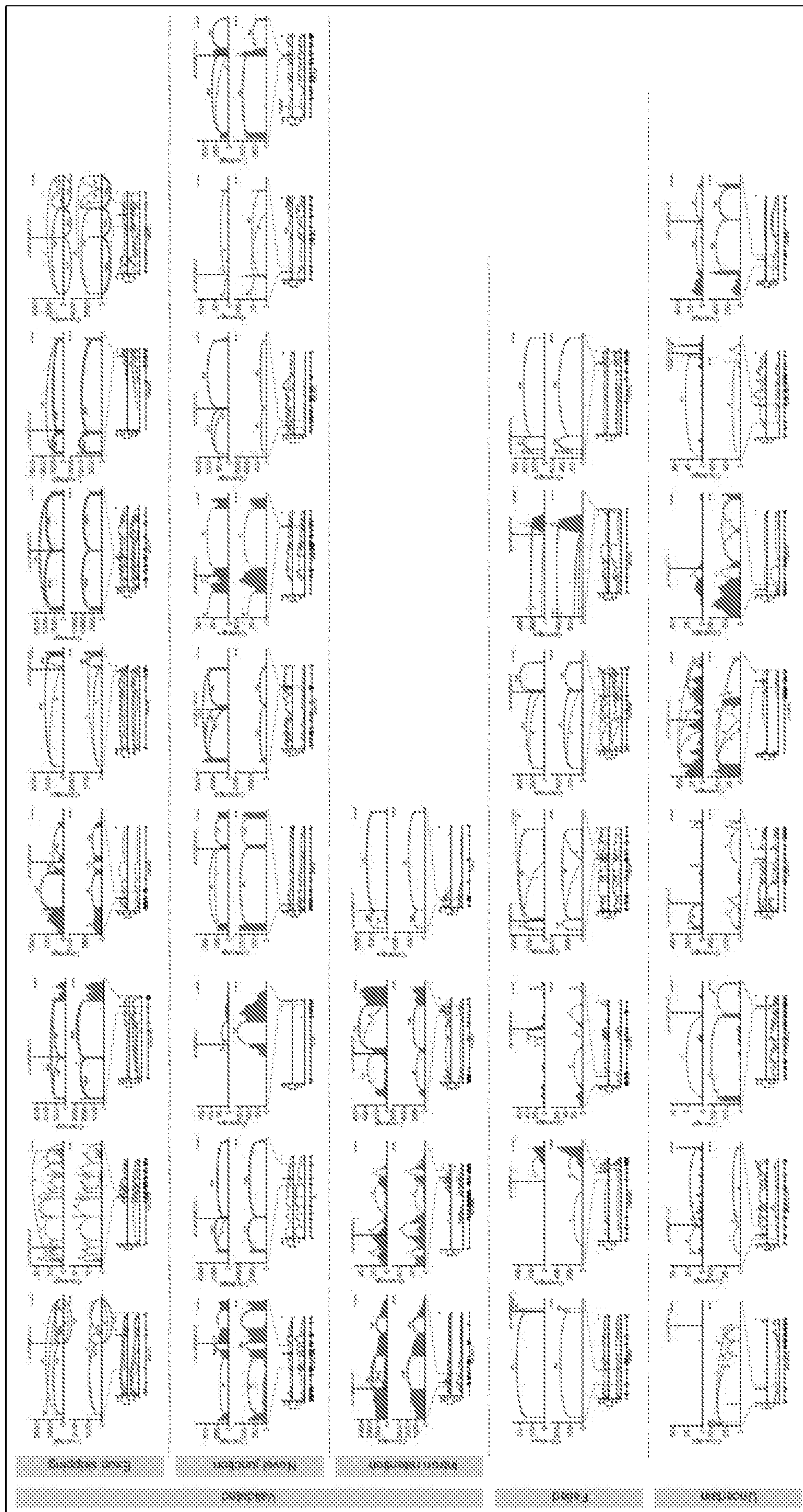
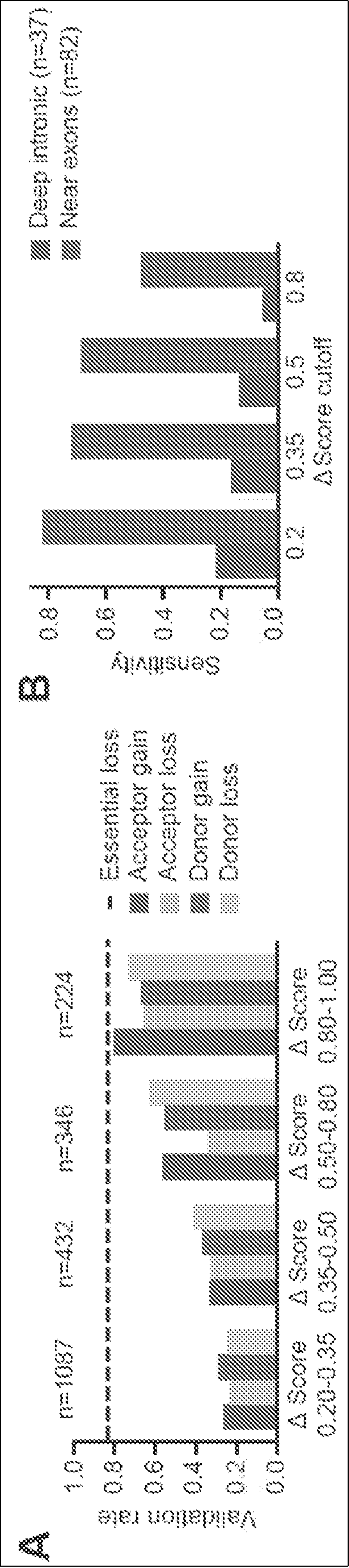
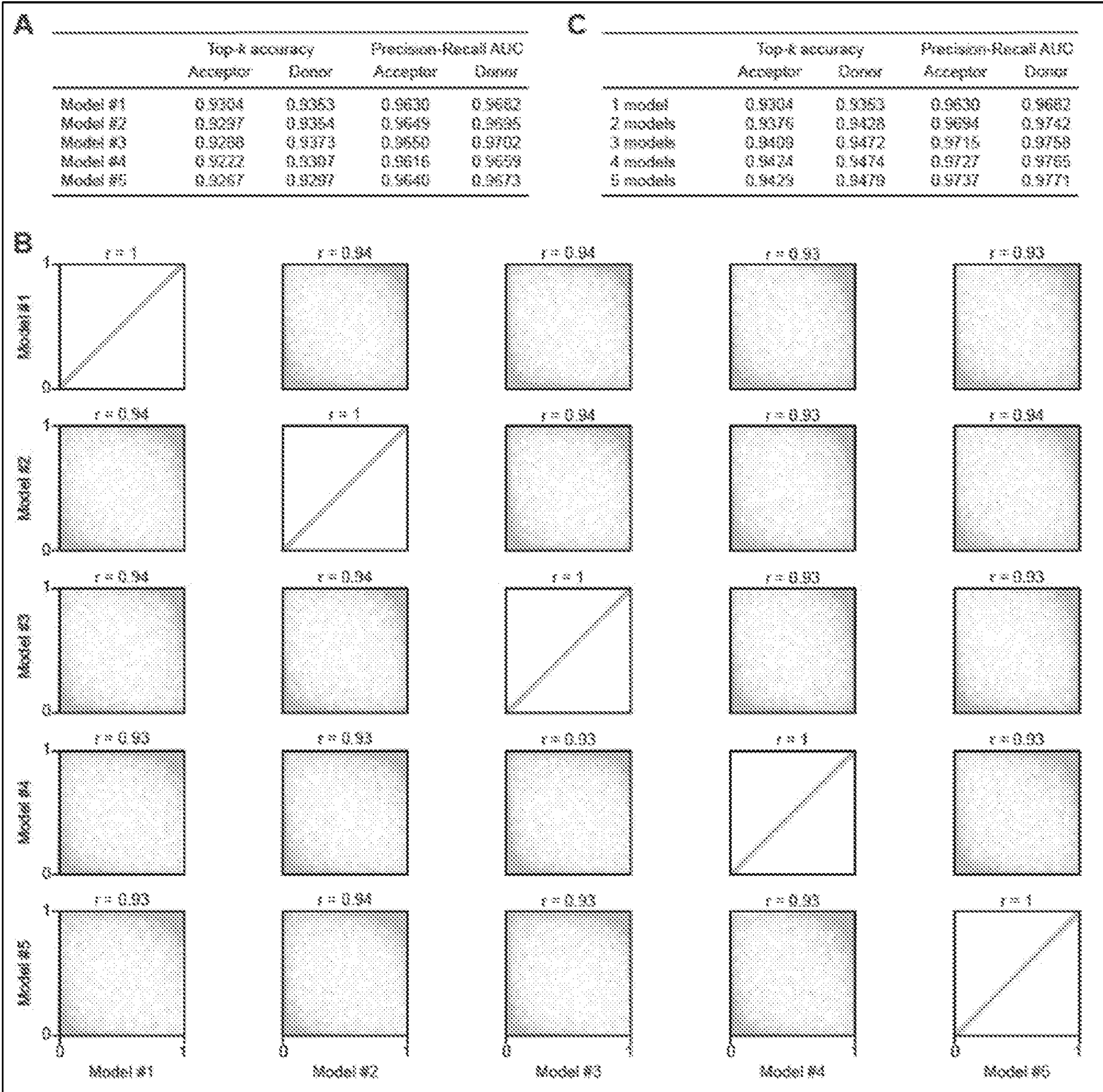


FIG. 51



FIGs. 52A-B



FIGs. 53A-C

A**SpliceNet-10k**

Exon density	Top-k accuracy		Precision-Recall AUC	
	Acceptor	Donor	Acceptor	Donor
1 exon	0.9165	0.9292	0.9592	0.9659
2 exons	0.9307	0.9357	0.9645	0.9674
3 exons	0.9519	0.9512	0.9816	0.9822
4 exons	0.9556	0.9540	0.9833	0.9845
≥5 exons	0.9579	0.9665	0.9826	0.9872

B**MaxEntScan**

Exon density	Top-k accuracy		Precision-Recall AUC	
	Acceptor	Donor	Acceptor	Donor
1 exon	0.1834	0.2487	0.1140	0.1660
2 exons	0.2180	0.3124	0.1512	0.2375
3 exons	0.2616	0.3348	0.1964	0.2773
4 exons	0.2950	0.3820	0.2307	0.3347
≥5 exons	0.3583	0.4652	0.3103	0.4314

FIGs. 54A-B

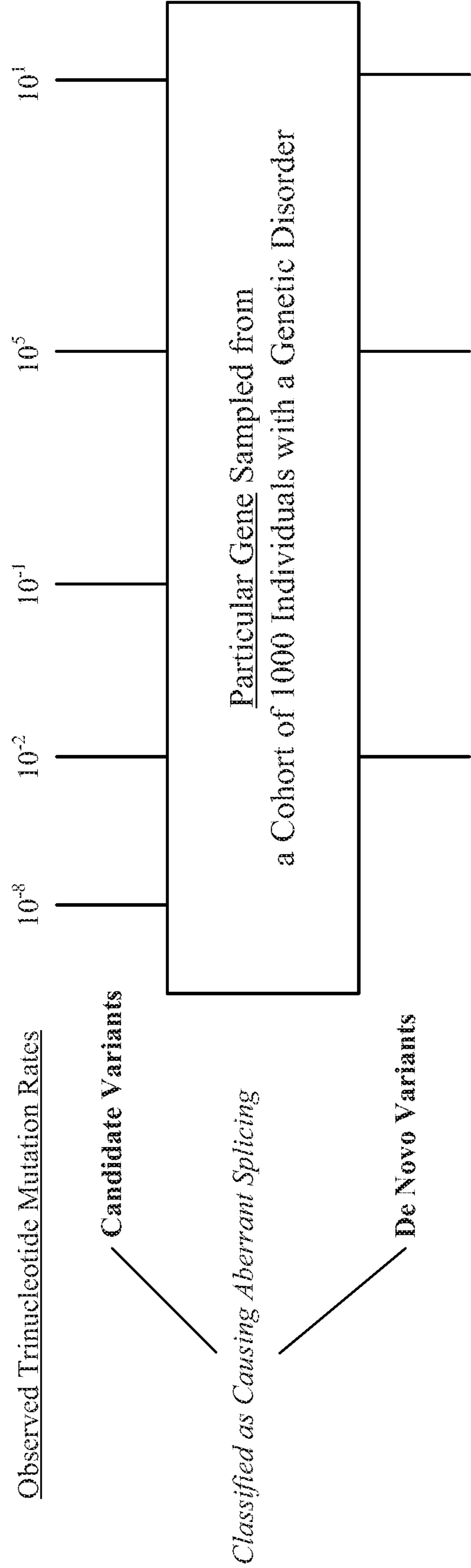
Tissues	Variant individuals	Control individuals
Skin - Sun Exposed (Lower leg)	T6MO	OXRK
Muscle - Skeletal, Transformed fibroblasts	WHSB	WK11
Artery - Tibial, Lung	XOT4, S341, POMQ	WFON
Whole Blood	TMMY	T8EM

FIG. 55

Gains				Losses			
SpliceNet-10k	GeneSplicer	NNSplice	MaxEntScan	SpliceNet-10k	GeneSplicer	NNSplice	MaxEntScan
0.1	103.5145	0.9024	0.901	0.1	1.5183	0.1083	0
0.2	105.6838	0.9288	0.9171	0.2	2.6295	0.2315	0.0068
0.3	107.1061	0.9542	0.9419	0.3	3.4707	0.3554	0.0312
0.4	108.0655	0.967	0.9582	0.4	4.0856	0.4625	0.0646
0.5	108.9011	0.9739	0.9694	0.5	4.6085	0.5349	0.1003
0.6	109.6138	0.9781	0.9759	0.6	5.2084	0.6345	0.1624
0.7	110.388	0.9817	0.9826	0.7	6.4456	0.7573	0.2838
0.8	111.2775	0.9839	0.9875	0.8	8.1053	0.8202	0.4271
0.9	112.6826	0.9858	0.9915	0.9	10.2586	0.8529	0.641

FIG. 56

Per-Gene Enrichment Analysis



57 / 59

$$\sum (observed\ trinucleotide\ mutation\ rates) = 10^{-5}$$

Baseline Number of Mutations = $2 \times 1000 \times 10^{-5} = 0.2$

Transmission (Chromosome) Count

Cohort Size

De Novo Variant Count = 3

FIG. 57

Genome-Wide Enrichment Analysis

Cohort of 2000 Healthy Individuals
Classified as Causing Aberrant Splicing

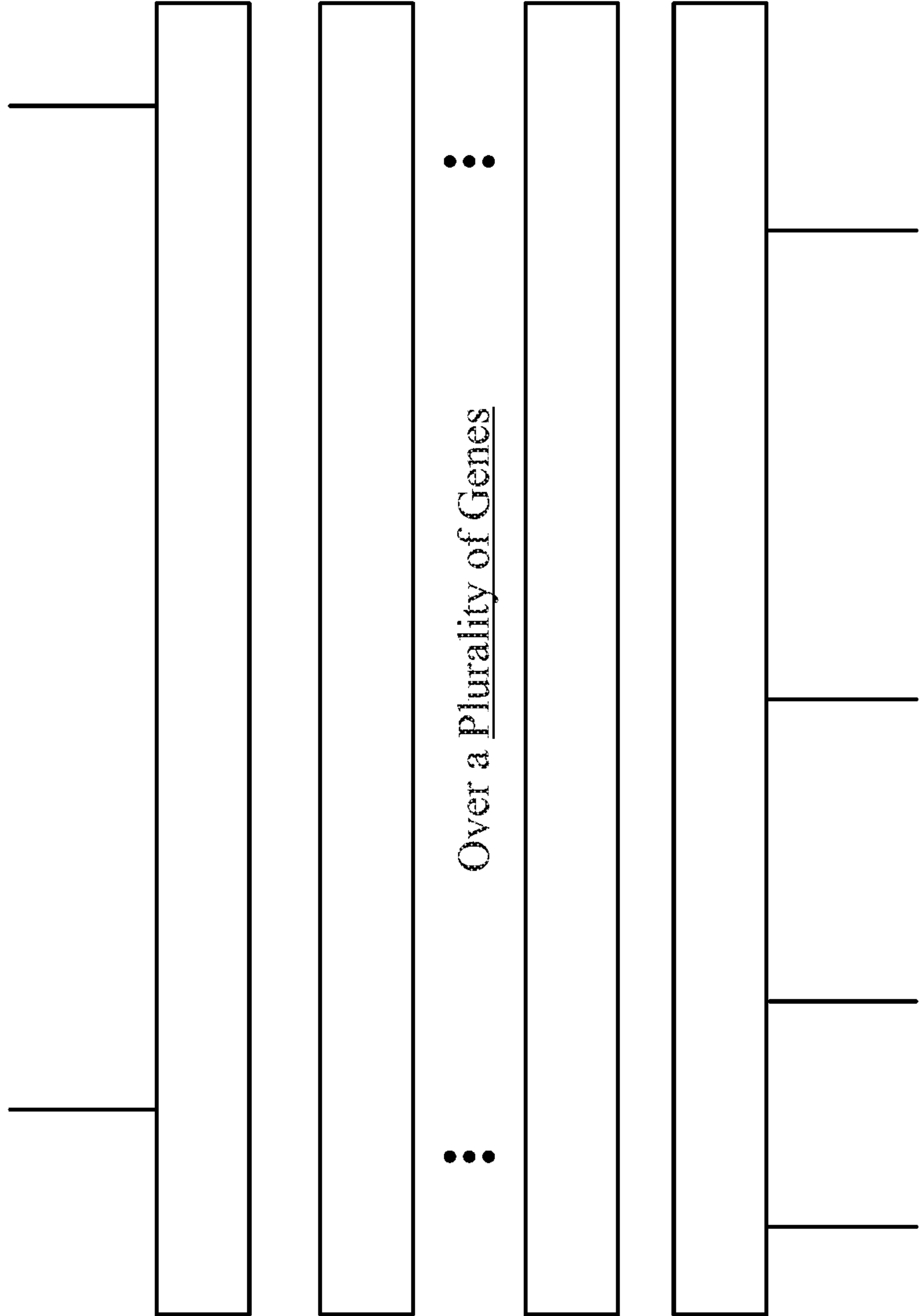


FIG. 58

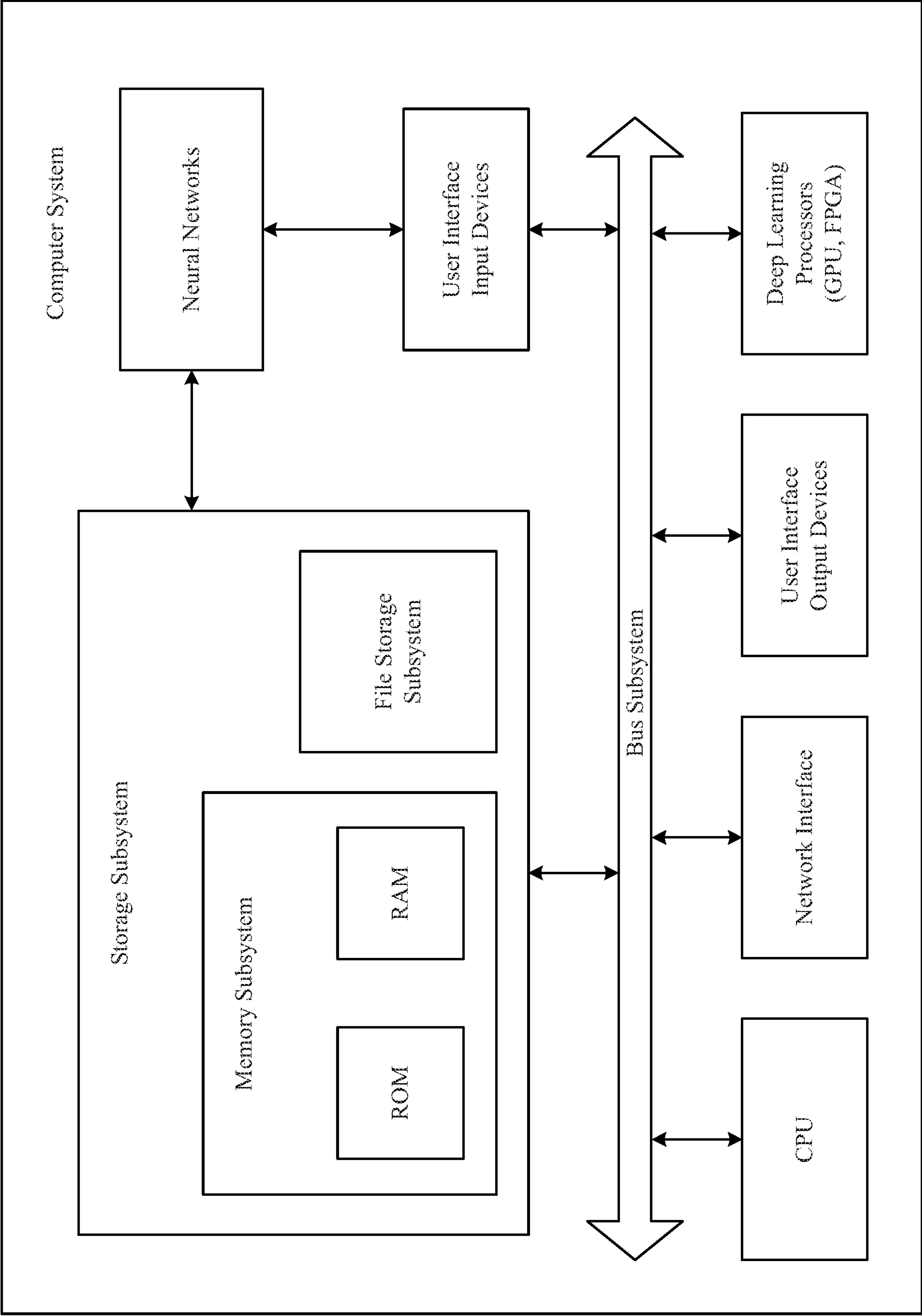


FIG. 59