

# LEARNING OBJECT BOUNDING BOXES FOR 3D INSTANCE SEGMENTATION ON POINT CLOUDS

*by B. Yang et al. (2019)*

[arXiv:1906.01140v2]

NeurIPS 2019



MACIEJ MANNA

*maciejmanna@gmail.com*

Kraków, 26.11.2020 r.

# Interesujące zagadnienia

- przetwarzanie chmur punktów (*point cloud processing*) jako odpowiednik w 3D przetwarzania obrazów 2D;
- popularność i skuteczność modeli *end-to-end*;
- wykorzystanie elementów modeli, które można jednoznacznie zinterpretować jako operujące na konkretnych obiektach i strukturach związanych z rozważanym problemem.

# Plan prezentacji

## 1. ZAGADNIENIA WSTĘPNE –

- wprowadzenie do przetwarzania chmur punktów, segmentacja semantyczna i segmentacja instancji;
- przegląd dotychczasowych podejść i modeli;
- wyzwania i cele proponowanego modelu.

## 2. OPIS MODELU –

- ogólna architektura modelu;
- podsystem predykcji obwiedni (*bounding boxes*);
- predykcja masek punktowych, funkcje kosztu (model typu *end-to-end*).

## 3. PREZENTACJA WYNIKÓW I PODSUMOWANIE –

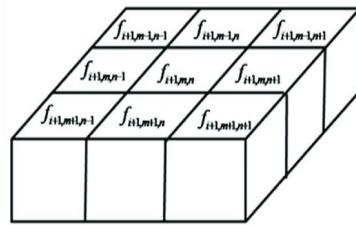
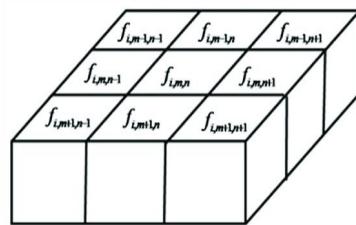
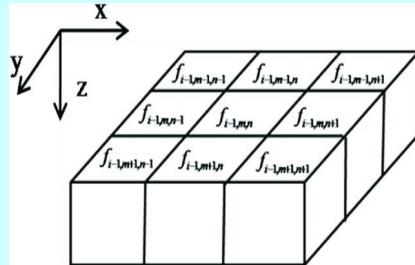
- wyniki testów na standardowych zestawach danych;
- uwagi nt. złożoności obliczeniowej;
- podsumowanie.

**- 1 -**

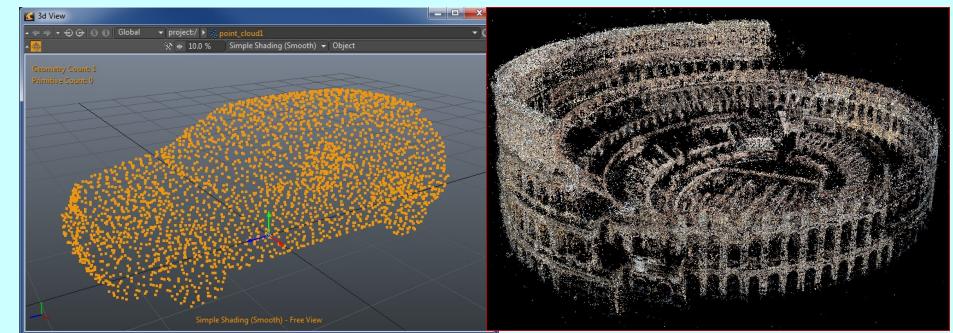
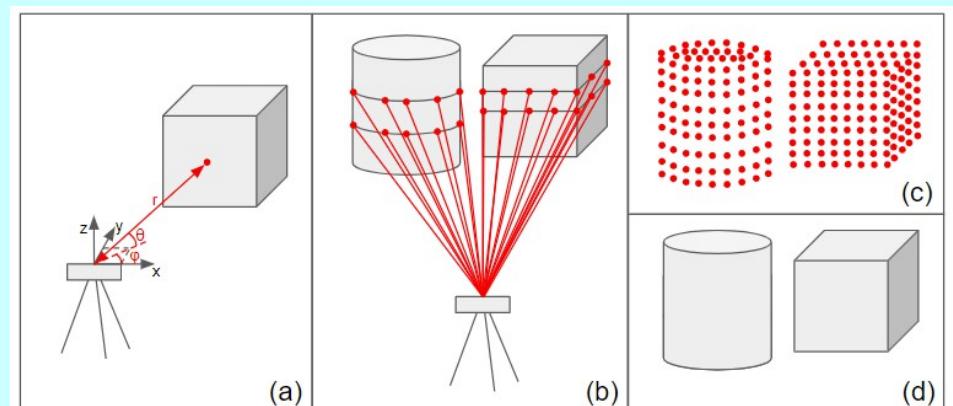
# **ZAGADNIENIA WSTĘPNE**

# Reprezentacja danych 3D

- woksele (*voxels*)



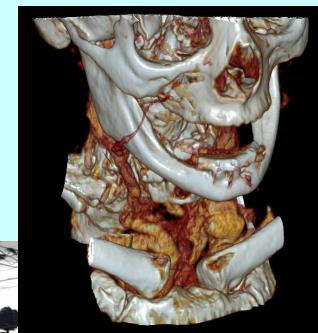
- vs    chmury punktów (*point clouds*)



- inne reprezentacje:
  - siatki (*meshes*) – trójkąty, wielokąty
  - reprezentacje parametryczne powierzchni (np. NURBS)

# Chmury punktów – zalety

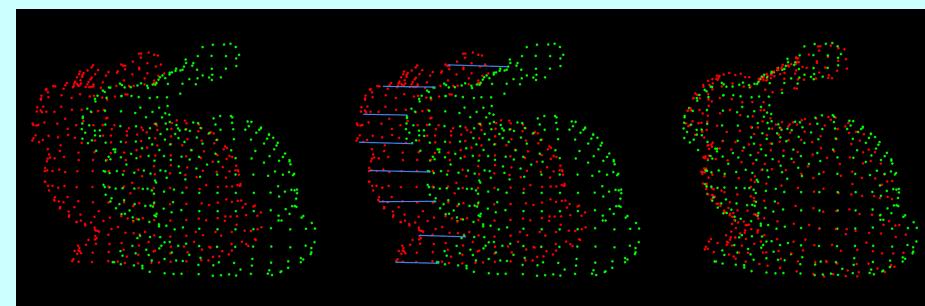
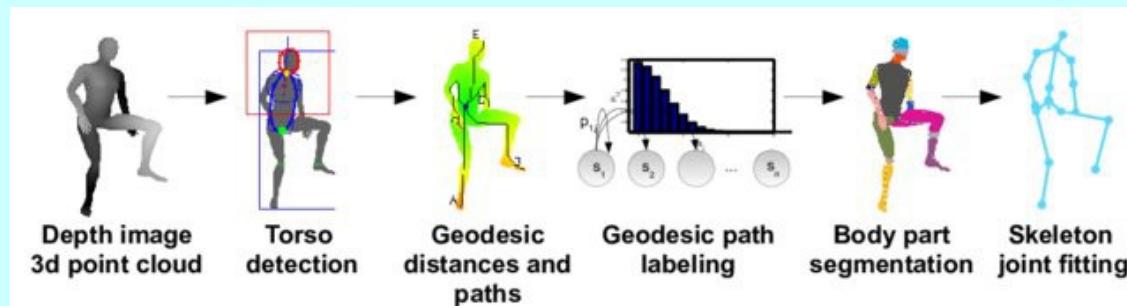
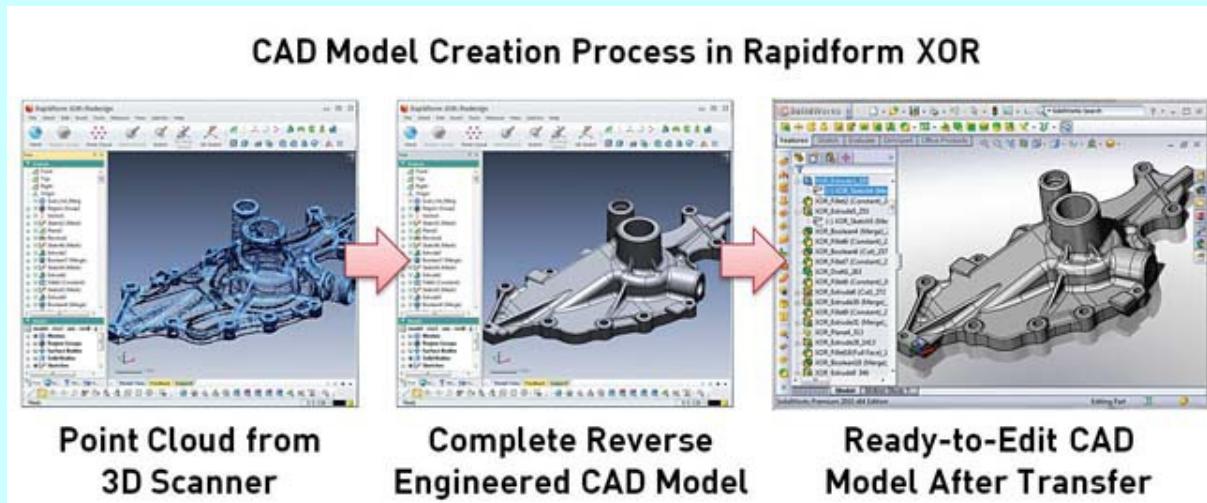
- Rzadka reprezentacja lepiej wykorzystuje pamięć i reprezentuje scenę proporcjonalnie do ilości informacji w danym miejscu:
  - ilość danych może być dopasowana do lokalnego poziomu detali (zamiast stałej, globalnej rozdzielczości);
  - omija zbędną reprezentację dla pustych przestrzeni, które często zajmują dużą część sceny.
- Naturalny, surowy format danych 3D dla wielu narzędzi i metod stosowanych w praktyce do pobierania danych, m. in.:
  - metody kontaktowe (*coordinate measuring machine, CMM*);
  - metody bezkontaktowe (wykorzystujące różne techniki optyczne, głównie triangulacja przy użyciu lasera – *lidar*);
  - fotogrametria;
  - wyjątek – metody wolumetryczne (głównie w obraz. medycznym).



# Chmury punktów – trudności

- Reprezentacja punktowa ma charakter:
  - **nieuporządkowany** (*unordered*);
  - **nieustrukturyzowany** (*unstructured*);
  - **niejednorodny** (*nonuniform*).

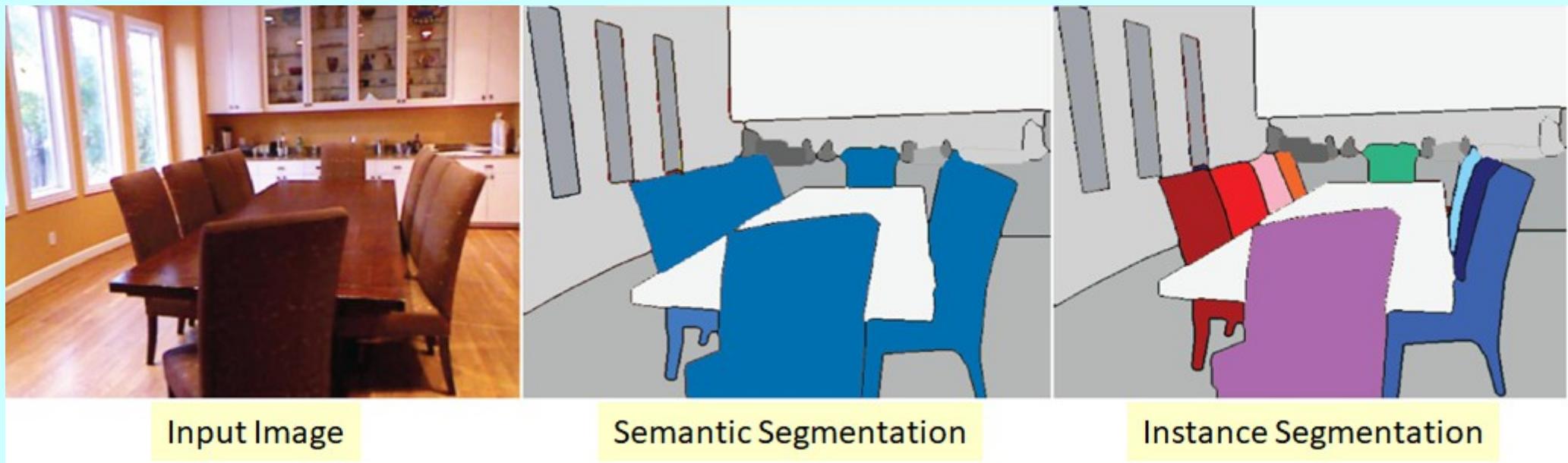
# Przetwarzanie chmur punktów - zadania



- konwersja do modeli siatkowych/CAD;
- przewidywanie pozycji ciała;
- rejestracja 3D (*3D registration*);
- segmentacja, klasyfikacja i detekcja (także w czasie rzeczywistym)  
[[https://telin.ugent.be/~ljj/ipi-topics/point\\_cloud/o7\\_tracking\\_cars.mp4](https://telin.ugent.be/~ljj/ipi-topics/point_cloud/o7_tracking_cars.mp4)]

# Segmentacja chmur punktów

- **Problem segmentacji:** przypisanie każdemu punktowi etykiety odpowiadającej poszczególnym typom lub instancjom obiektów obecnych w danej scenie.
- **Segmentacja semantyczna vs segmentacja instancji:**

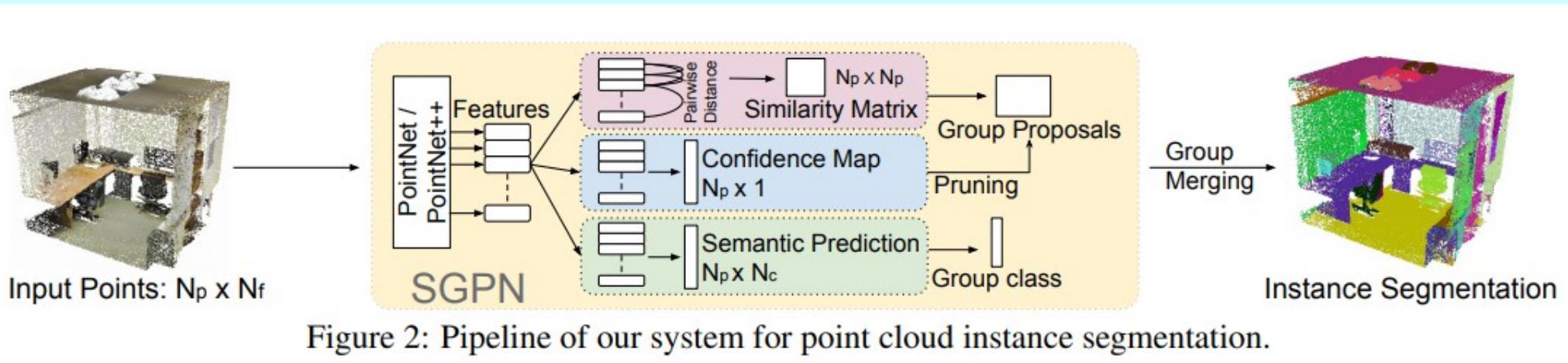


# Segmentacja instancji (1)

- Coraz większe zainteresowanie tematem (obok segmentacji semantycznej i detekcji obiektów), szczególnie w kontekście rozwoju zastosowań widzenia komputerowego w robotyce, *augmented reality* oraz *autonomous driving*.
- Dotychczasowe propozycje (*proposal-based*):
  - **SGPN** (*Similarity Group Proposal Network*; CVPR 2018) – uczy się macierzy podobieństwa, na której podstawie otrzymuje się propozycje grupowania (*group proposals*) par punktów;
  - Podobne propozycje:
    - **ASIS** (*Associatively Segmenting Instances and Semantics*, CVPR 2019),
    - **JSIS<sub>3</sub>D** (*Joint Semantic-Instance Segmentation* CVPR 2019),
    - **MASC** (*Multi-scale Affinity with Sparse Convolution*, 2019),
    - **3D-BEVIS** (*Birds-Eye-View Instance Segmentation*, GCPR 2019);
  - **PartNet** (CVPR 2019) – użycie modelu do segmentacji części do segmentacji instancji.

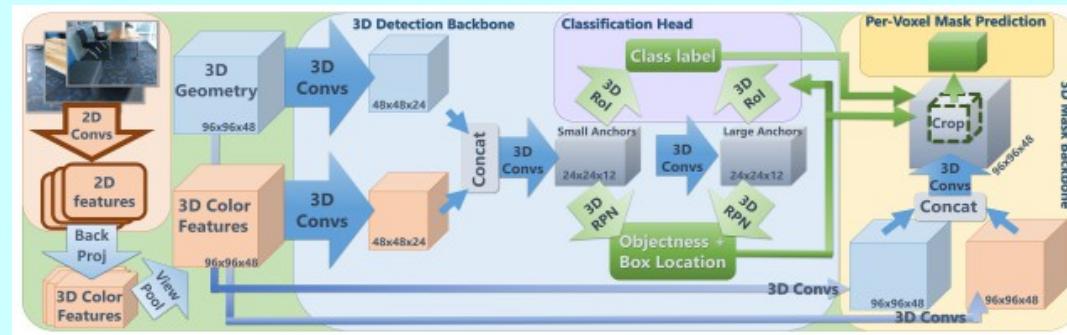
# Segmentacja instancji (2)

- Trudności dotychczasowych propozycji:
  - Metody **nie wykrywają wprost obrzeży** obiektów (instancji), co prowadzi do obniżonej skuteczności;
  - Niezbędny jest **obliczeniowo wymagający post-processing** otrzymanych propozycji grupowania (*group proposals*) punktów do ostatecznych etykiet segmentacji (np. dodatkowe klastrowanie).

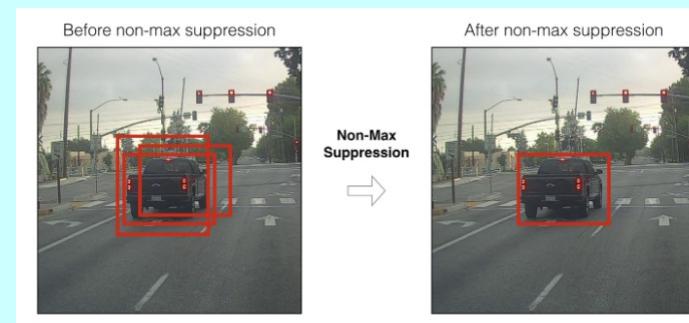


# Segmentacja instancji (3)

- Inne propozycje: **3D-SIS** (*3D Semantic Instance Segmentation*, CVPR 2019):
  - dwuetapowy model i dwuetapowy trening;



- kosztowny *post-processing*: odrzucanie zbyt gęstych propozycji instancji używając *non-maximum suppression*;



- Podobnie: **GSPN** (*Generative Shape Proposal Network*, CVPR 2019).

# Cele nowego modelu

- Nowy model – **BoNet** [NeurIPS 2019] – cele twórców:
  - wprowadzenie **nowego modułu uczącego się obwiedni** (*bounding boxes*) obiektów, aby wprost i skutecznie uczyć się ich obrzeży;
  - **efektywność obliczeniowa** – stosowanie prostych warstw (głównie MLP) i nadawanie etykiet segmentacyjnych przez sieć (bez potrzeby dodatkowych etapów i *post-processingu*);
  - **nie używa kosztownych i gęstych propozycji grupowania** (*grouping/object proposals*), tj. *proposal-free, anchor-free*;
  - **model jednoetapowy** – trenowany *end-to-end*.

**- 2 -**

# **OPIS MODELU**

# BoNet – Ogólna architektura (1)

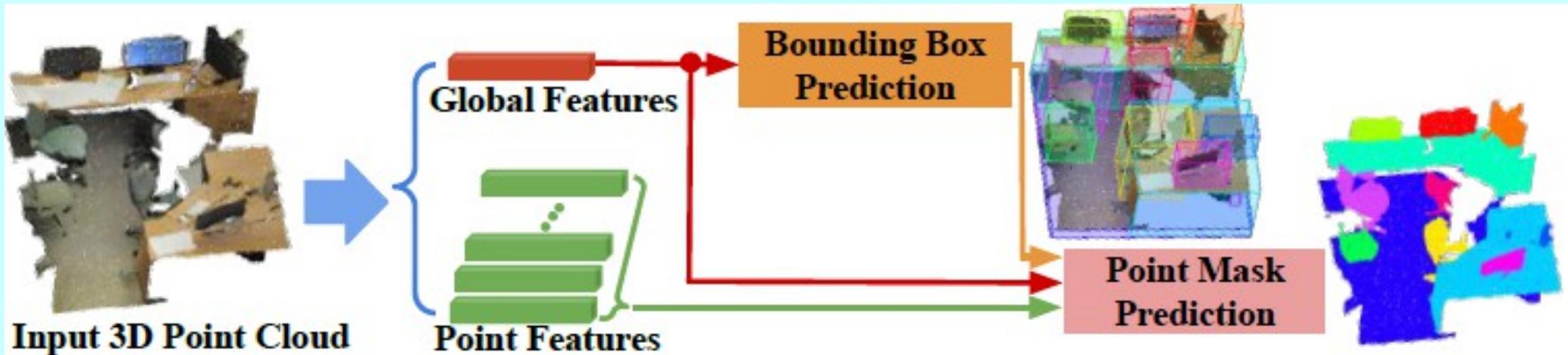


Figure 1: The 3D-BoNet framework for instance segmentation on 3D point clouds.

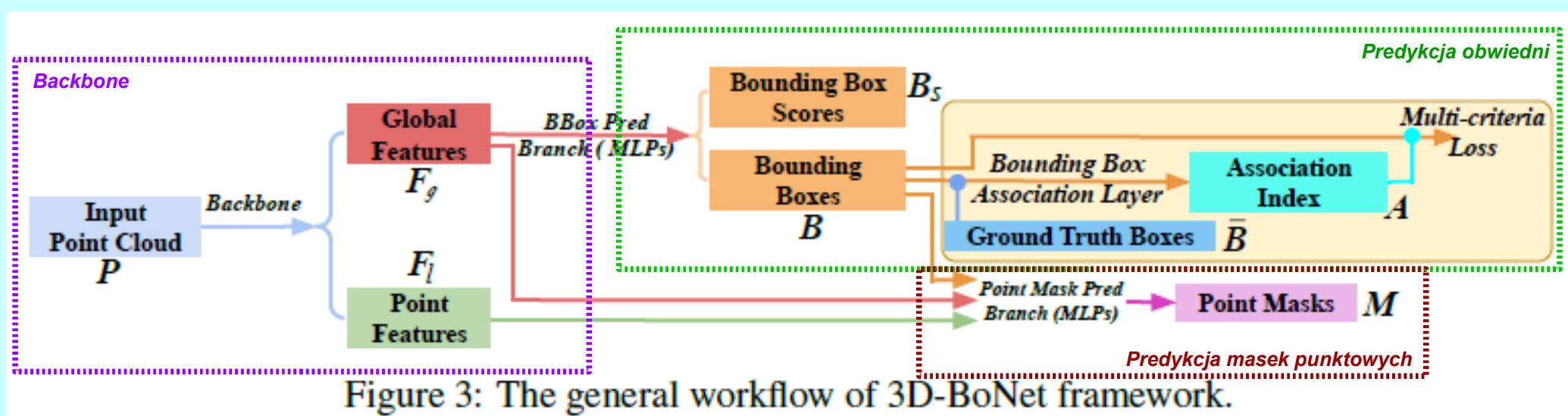


Figure 3: The general workflow of 3D-BoNet framework.

# BoNet – Ogólna architektura (2)

- Model stanowi **jedną całość** (single-stage, end-to-end), ale możemy wyróżnić jego główne **trzy elementy**:
  - **backbone** – dowolna architektura pozwalająca na skuteczne uzyskanie globalnych i lokalnych (dla każdego punktu) wektorów własności;
  - **predykcja obwiedni** (*bounding box prediction*) – centralna część architektury modelu; przewiduje obwiednie obiektów bez uprzednich danych (*anchor-free, RPN-free*); możliwe do wykonania (niekoniecznie precyzyjnie), a istotnie wpływa na skuteczność i wydajność predykcji właściwych etykiet segmentacyjnych; także zajmuje się (w czasie nadzorowanego treningu) asocjacją przewidywanych obwiedni z obwiedniami *ground truth*;
  - **predykcja masek punktowych** (*point mask prediction*) – używa własności globalnych i lokalnych, a także przewidzianych obwiedni (punkty spoza obwiedni nie sąbrane pod uwagę), aby przewidzieć maskę punktową danej instancji i przypisać punktom odp. etykiety.

# **Backbone** (1)

- **Backbone** sieci stanowi dowolna architektura pozwalająca na pozyskanie dwóch typów wektorów własności (*feature vectors*):
  - własności globalne – charakteryzujące całą chmurę punktów;
  - własności lokalne – charakteryzujące dany punkt; osobny wektor dla każdego punktu.
- **Wejściem** do sieci jest chmura punktów  $P$  (domyślnie, bez żadnego *preprocessingu*), która zawiera  $N$  punktów, z których każdy opisany jest przez  $k_o$  własności (kanałów), np. pozycja  $\{x,y,z\}$  i kolor  $\{r,g,b\}$  ( $k_o=6$ ), zatem:

$$P \in \mathbb{R}^{N \times k_o}$$

- **Wyjściem** z sieci backbone są wspomniane wektory własności  $F_g$  i  $F_l$ :

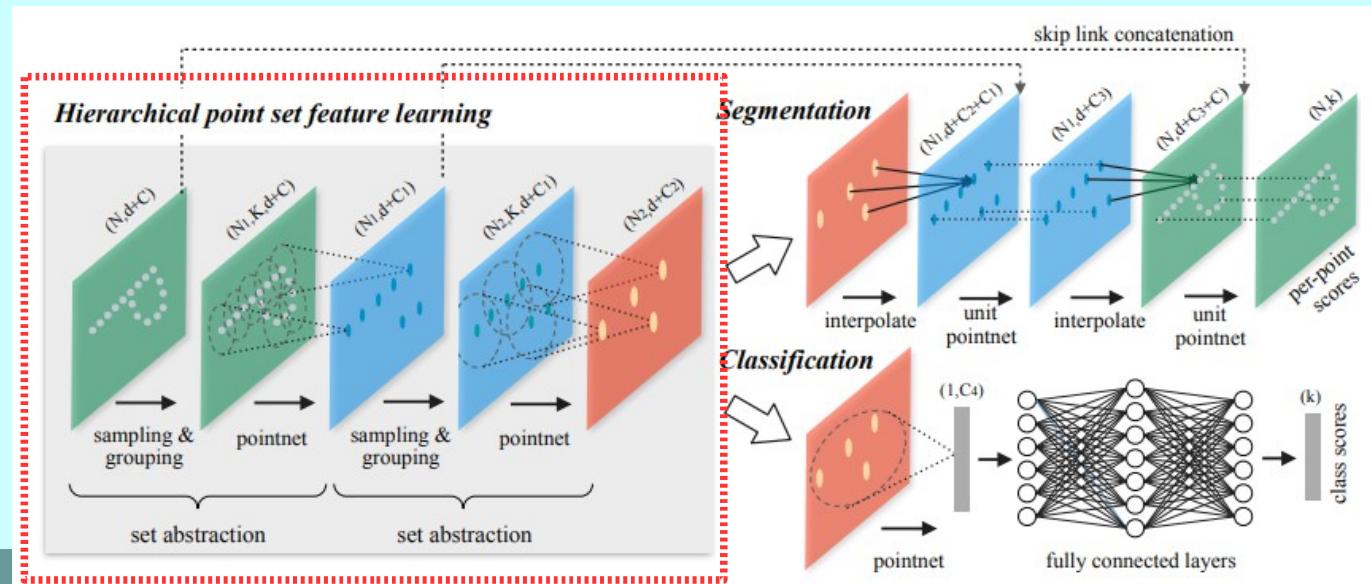
$$F_g \in \mathbb{R}^{1 \times k}$$

$$F_l \in \mathbb{R}^{N \times k}$$

gdzie  $k$  jest ustaloną długością wektorów własności (hiperparametr).

# Backbone (2)

- Implementacji *backbone'a* w sieci BoNet pozostawiona jest duża swoboda:
  - może być to dedykowana architektura (autorzy takiej nie zaproponowali);
  - można użyć istotnej części sprawdzonych i skutecznych architektur służących do segmentacji semantycznej (które również korzystają z podobnych wektorów własności).
- Autorzy w swoich eksperymentach jako *backbone'a* używają sieci **PointNet++** (NIPS 2017, *state-of-the-art* w segmentacji semantycznej).



# Predykcja obwiedni (1)

- **Reprezentacja obwiedni** (BB) – prostopadłościan o ustalonej orientacji (boki równoległe do osi ustalonego układu współrzędnych, *axis-aligned*); do jego reprezentacji wystarczą współrzędne przeciwnieległych wierzchołków:

$$\{[x_{min} \ y_{min} \ z_{min}], [x_{max} \ y_{max} \ z_{max}]\}$$

- **Wejściem** do predykcji BB jest jedynie wektor własności globalnych  $F_g$ :

$$F_g \in \mathbb{R}^{1 \times k}$$

natomiast **wyjście** stanowi wektor reprezentujący zbiór  $B$  zawierający  $H$  obwiedni:

$$B \in \mathbb{R}^{H \times 2 \times 3},$$

gdzie  $H$  jest ustalonym hiperparametrem (odpowiednio większym niż oczekiwana ilość instancji w danej scenie).

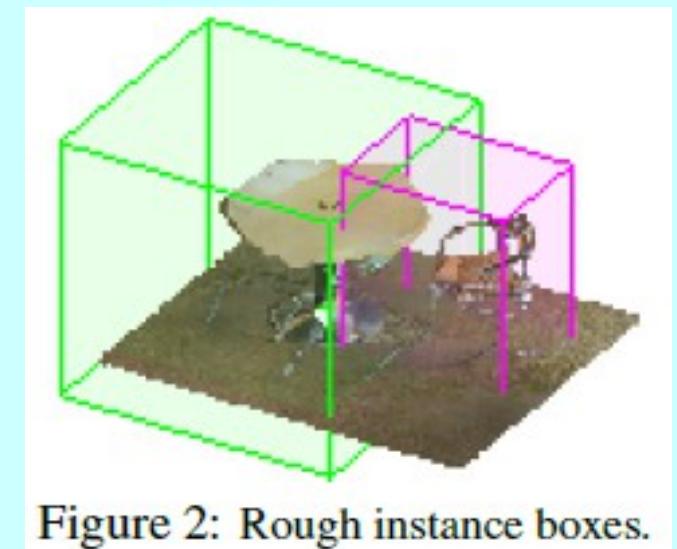


Figure 2: Rough instance boxes.

# Predykcja obwiedni (2)

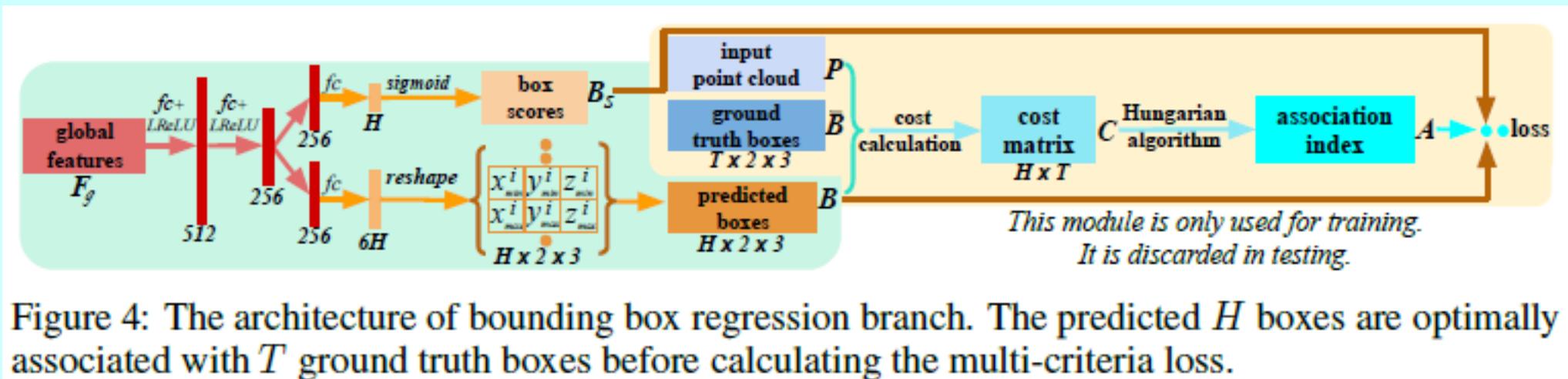
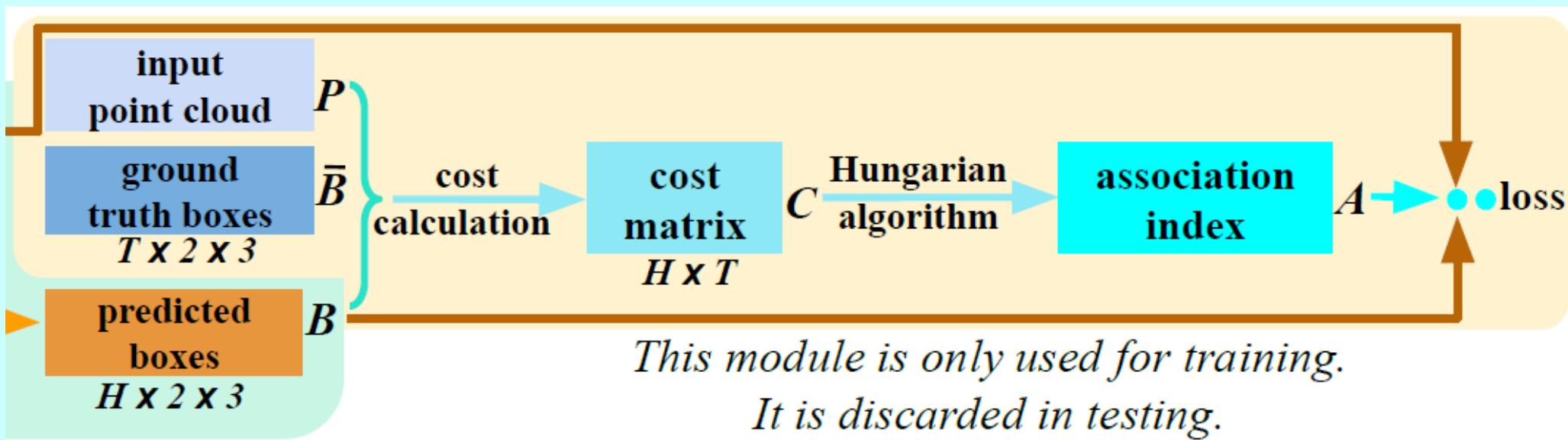


Figure 4: The architecture of bounding box regression branch. The predicted  $H$  boxes are optimally associated with  $T$  ground truth boxes before calculating the multi-criteria loss.

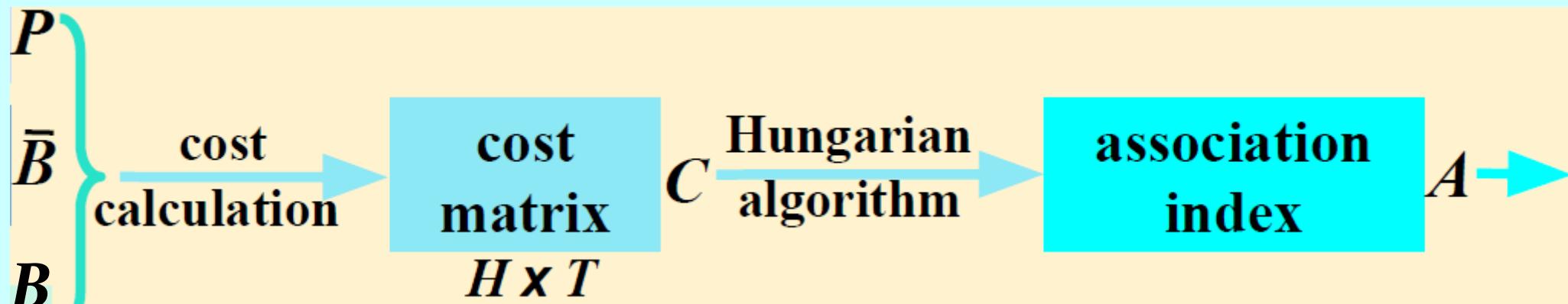
- Struktura sieci do predykcji BB:
  - (1) warstwa *fully-connected* ( $k \rightarrow 512$ ) z aktywacją nieliniową *Leaky ReLU*;
  - (2) warstwa *fully-connected* ( $512 \rightarrow 256$ ) z aktywacją nieliniową *Leaky ReLU*;
  - (3.1) gałąź predykcji –
    - warstwa *fully-connected* ( $256 \rightarrow 6H$ ), a następnie przekształcenie w tensor ( $H \times 2 \times 3$ ) reprezentujący przewidziane obwiednie  $B$ ;
  - (3.2) gałąź treningowa (używana jedynie podczas treningu) –
    - osobna warstwa *fully-connected* ( $256 \rightarrow H$ ), a następnie *sigmoida*, aby sprowadzić wyniki w zakres  $[0,1]$  – interpretowane jako wektor ocen jakości odpowiednich obwiedni  $B_s$ .

# Predykcja obwiedni – trening (1)



- Otrzymujemy wejściową chmurę punktów  $P$ , przewidziane obwiednie  $B$  oraz obwiednie *ground truth*  $\bar{B}$  – celem jest stworzenie odpowiedniej funkcji kosztu (biorącej pod uwagę wiele czynników, *multi-criteria loss*) oceniającej jak dobrze przewidziane obwiednie zawierają odpowiednie instancje.
- Trudności:
  - mamy więcej przewidywanych BB niż BB *ground truth* (tj.  $H > T$ ), gdzie  $T$  jest ilością instancji (a zarazem BB *ground truth*) – niektóre trzeba odrzucić,
  - przewidziane BB nie są uporządkowane i nie ma *anchorów* pomagających w ich kojarzeniu – trzeba je skojarzyć z BB *ground truth*.

# Preidykcja obwiedni – trening (2)



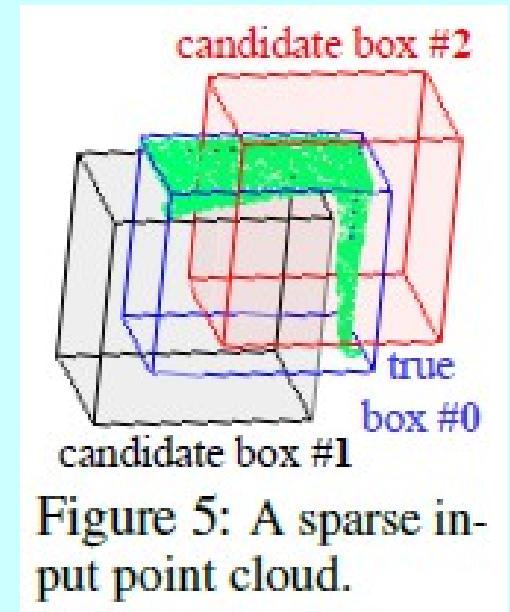
- Proces kojarzenia przewidzianych BB z *ground truth* BB:
  - chcemy otrzymać  $A$  – macierz skojarzenia ( $H \times T$ ), gdzie 1 oznacza, że (dokładnie jeden)  $i$ -ty przewidziany BB jest skojarzony (z co najwyżej jednym)  $j$ -tym *ground truth* BB;
  - tworzymy  $C$  – macierz kosztu ( $H \times T$ ), gdzie  $C_{ij}$  oznacza koszt przypisania  $i$ -tego przewidzianego BB do  $j$ -tego *ground truth* BB (im mniejszy koszt, tym bardziej dwa BB są podobne do siebie).
  - konieczne rozwiązanie problemu optymalizacyjnego:

$$A = \arg \min_A \sum_{i=1}^H \sum_{j=1}^T C_{i,j} A_{i,j} \quad \text{subject to } \sum_{i=1}^H A_{i,j} = 1, \sum_{j=1}^T A_{i,j} \leq 1, j \in \{1..T\}, i \in \{1..H\}$$

# Preidykcja obwiedni – trening (3)



- Rozwiązywanie problemu optymalizacyjnego – **algorytm węgierski** (Kuhn-Munkres) do rozwiązywania problemu skojarzenia na grafach (tj. dopasowanie o minimalnej/maksymalnej wadze w grafie dwudzielonym);
- Obliczanie kosztu:
  - naiwne rozwiązanie – odległości euklidesowe między dwoma wierzchołkami reprezentującymi BB;
  - nieoptymalne w naszej sytuacji – chcemy, aby przewidziany BB zawierał jak najwięcej prawidłowych punktów (lepiej wiecej nieprawidłowych, niż mniej prawidłowych);
  - konieczne opracowanie złożonej funkcji kosztu respektującej powyższe wymaganie.



# Predykcja obwiedni – trening (4)

- Właściwa funkcja kosztu składa się z trzech składników:

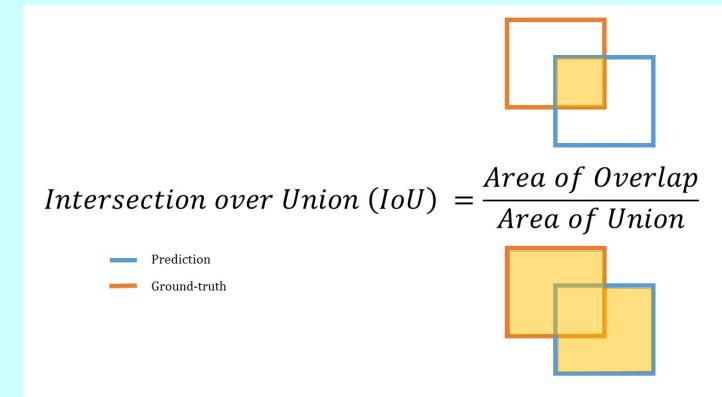
$$C_{i,j} = C_{i,j}^{ed} + C_{i,j}^{sIoU} + C_{i,j}^{ces}$$

(1) **ed** – odległość euklidesowa wierzchołków:

(2) **sIoU** (*soft Intersection-over-Union*) – różniczkowalny odpowiednik IoU, który operuje na gładkich wartościach binarnych (tj. z przedziału (0,1)) zamiast „twardych“ (tj. ze zbioru {0,1}) do określania przynależności do poszczególnych BB; w praktyce:

$$C_{i,j}^{sIoU} = \frac{-\sum_{n=1}^N (q_i^n * \bar{q}_j^n)}{\sum_{n=1}^N q_i^n + \sum_{n=1}^N \bar{q}_j^n - \sum_{n=1}^N (q_i^n * \bar{q}_j^n)}$$

$$C_{i,j}^{ed} = \frac{1}{6} \sum (B_i - \bar{B}_j)^2$$



(3) **ces** (*cross-entropy score*) – entropia krzyżowa między „gładkimi“ wektorami przynależności do obu BB (w przeciwieństwie do **sIoU**, która preferuje mniejsze BB, tutaj entropia faworyzuje BB, które zawierają jak najwięcej poprawnych pkt., a więc większe, bardziej inkluzywne BB):

$$C_{i,j}^{ces} = -\frac{1}{N} \sum_{n=1}^N [\bar{q}_j^n \log q_i^n + (1 - \bar{q}_j^n) \log(1 - q_i^n)]$$

# Predykcja obwiedni – trening (5)

- Algorytm obliczania „gładkich“ wektorów przynależności do BB:

---

**Algorithm 1** An algorithm to calculate point-in-pred-box-probability.  $H$  is the number of predicted bounding boxes  $B$ ,  $N$  is the number of points in point cloud  $P$ ,  $\theta_1$  and  $\theta_2$  are hyperparameters for numerical stability. We use  $\theta_1 = 100$ ,  $\theta_2 = 20$  in all our implementation.

---

for  $i \leftarrow 1$  to  $H$  do

- the  $i^{th}$  box min-vertex  $B_{min}^i = [x_{min}^i \ y_{min}^i \ z_{min}^i]$ .
- the  $i^{th}$  box max-vertex  $B_{max}^i = [x_{max}^i \ y_{max}^i \ z_{max}^i]$ .

for  $n \leftarrow 1$  to  $N$  do

- the  $n^{th}$  point location  $P^n = [x^n \ y^n \ z^n]$ .
- step 1:  $\Delta_{xyz} \leftarrow (B_{min}^i - P^n)(P^n - B_{max}^i)$ .
- step 2:  $\Delta_{xyz} \leftarrow \max[\min(\theta_1 \Delta_{xyz}, \theta_2), -\theta_2]$ .
- step 3: probability  $p_{xyz} = \frac{1}{1 + \exp(-\Delta_{xyz})}$ .
- step 4: point probability  $q_i^n = \min(p_{xyz})$ .
- obtain the soft-binary vector  $q_i = [q_i^1 \cdots q_i^N]$ .

The above two loops are only for illustration. They are easily replaced by standard and efficient matrix operations.

---

# Predykcja obwiedni – trening (6)

- Otrzymana macierz skojarzenia  $\mathbf{A}$  pozwala na takie uporządkowanie BB i ich ocen ( $B$ ,  $B_s$ ), że pierwsze  $T$  odpowiadają dokładnie skojarzonym BB *ground truth*, a ostatnie ( $H - T$ ) można traktować jako odrzucone.
- Na podstawie otrzymanych kosztów i skojarzenia możemy zdefiniować odpowiednie, właściwe funkcje kosztu dla predykcji  $B$  i  $B_s$ :
  - Dla predykcji **BB** naturalne jest użycie kosztu, który wykorzystany był w procesie ich kojarzenia – wybieramy sumę kosztów przewidzianych BB w odniesieniu do dopasowanych im BB *ground truth*, które faktycznie zostały wybrane, a wyniki odrzuconych pomijamy; mamy więc:

$$\ell_{bbox} = \frac{1}{T} \sum_{t=1}^T (C_{t,t}^{ed} + C_{t,t}^{sIoU} + C_{t,t}^{ces})$$

- Przewidziane **oceny BB** mają na celu stwierdzić na ile poprawne są przewidziane BB; jako oceny ground truth bierzemy **1** dla dopasowanych BB oraz **0** dla odrzuconych i liczymy entropię krzyżową (klasyfikacja binarna):

$$\ell_{bbs} = -\frac{1}{H} \left[ \sum_{t=1}^T \log B_s^t + \sum_{t=T+1}^H \log(1 - B_s^t) \right]$$

(taka funkcja kosztu utwierdza ocenę dobrze przewidzianych BB, a penalizuje odrzucone – sytuacja regesji więcej niż jednego BB do BB *gr. tr.*).

# Predykcja masek punktowych (1)

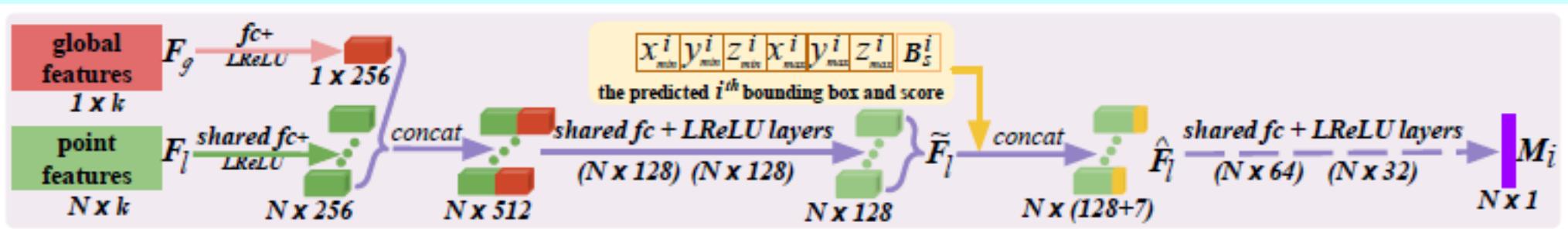


Figure 6: The architecture of point mask prediction branch. The point features are fused with each bounding box and score, after which a point-level binary mask is predicted for each instance.

- Ostatnia część sieci ma na celu bezpośrednią predykcję etykiet segmentacyjnych dla wszystkich punktów.
- **Wejście** stanowią wektory cech globalnych i lokalnych ( $F_g$  i  $F_l$ ) oraz przewidziane obwiednie  $B$  i ich oceny  $B_s$  (są one używane na późniejszym etapie, więc mogą być one obliczane równolegle ze wstępnią fazą predykcji masek punktowych).
- **Wyjście** stanowią wektory masek punktowych  $M_i$  (o długości  $N$ ), dla poszczególnych obwiedni.

# Predykcja masek punktowych (2)

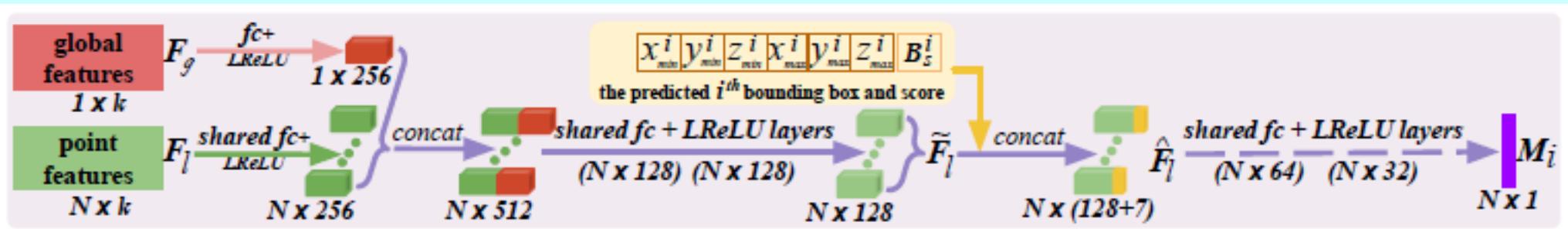


Figure 6: The architecture of point mask prediction branch. The point features are fused with each bounding box and score, after which a point-level binary mask is predicted for each instance.

- Struktura sieci do predykcji masek punktowych:
  - (1) warstwy *fully-connected* z aktywacją nieliniową *Leaky ReLU* dla własności globalnych ( $k \rightarrow 512$ ) i lokalnych ( $N^*k \rightarrow N^*512$ , wspólne wagi);
  - (2) konkatenacja wektora własności globalnych do każdego z lokalnych ( $N$  razy:  $256 \rightarrow 512$ );
  - (3) dwie warstwy *fully-connected* ( $N^*256 \rightarrow N^*128 \rightarrow N^*128$ ) z aktywacją nieliniową *Leaky ReLU* (wynik: ***mixed point features***);
  - (4) dla każdego BB: konkatenacja reprezentacji i oceny BB do *MPF* ( $N$  razy:  $128 \rightarrow 128 + 7$ ; wynik: ***box-aware features***);
  - (5) trzy warstwy *fully-connected* ( $N^*(128+7) \rightarrow N^*64 \rightarrow N^*32 \rightarrow N$ ) z aktywacją nieliniową *Leaky ReLU*, a następnie sigmoidą, aby otrzymać maskę.

# Predykcja masek punktowych (3)

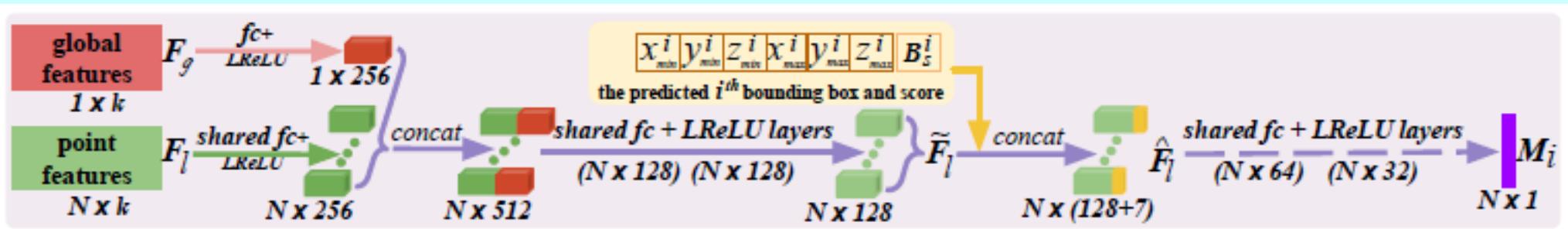


Figure 6: The architecture of point mask prediction branch. The point features are fused with each bounding box and score, after which a point-level binary mask is predicted for each instance.

- **Funkcja kosztu dla predykcji masek punktowych:**
  - przewidziane maski są utożsamiane z odpowiadającymi im BB *ground truth* używając macierzy  $A$ ;
  - Problem – duża przewaga punktów należących do tła w porównaniu z tymi, które należą do danej instancji;
  - rozwiążanie – użycie funkcji kosztu ***focal loss*** (ważona wersja entropii krzyżowej, dająca znacznie większe znaczenie punktom o pozytywnej wartości *ground truth*):

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t).$$

- do ostatecznego kosztubrane są tylko maski poprawnie skojarzonych BB.

# Model *end-to-end*

- Istotną charakterystyką proponowanego rozwiązania jest możliwość użycia go jako jednej całości w treningu.
- Wymaga to zdefiniowania wspólnej funkcji kosztu dla całego modelu – autorzy używają sumy odpowiednich kosztów poszczególnych elementów modelu, a więc:

$$\ell_{all} = \ell_{sem} + \ell_{bbox} + \ell_{bbs} + \ell_{pmask}$$

gdzie:

- $\ell_{bbox}$  i  $\ell_{bbs}$  – funkcje kosztu dla przewidywanych BB i ich ocen,
- $\ell_{pmask}$  – funkcja kosztu dla przewidywanych masek,
- $\ell_{sem}$  – wynik standardowej entropii krzyżowej softmax dla równolegle przeprowadzonej segmentacji semantycznej przy użyciu sieci będącej *backbonem*.

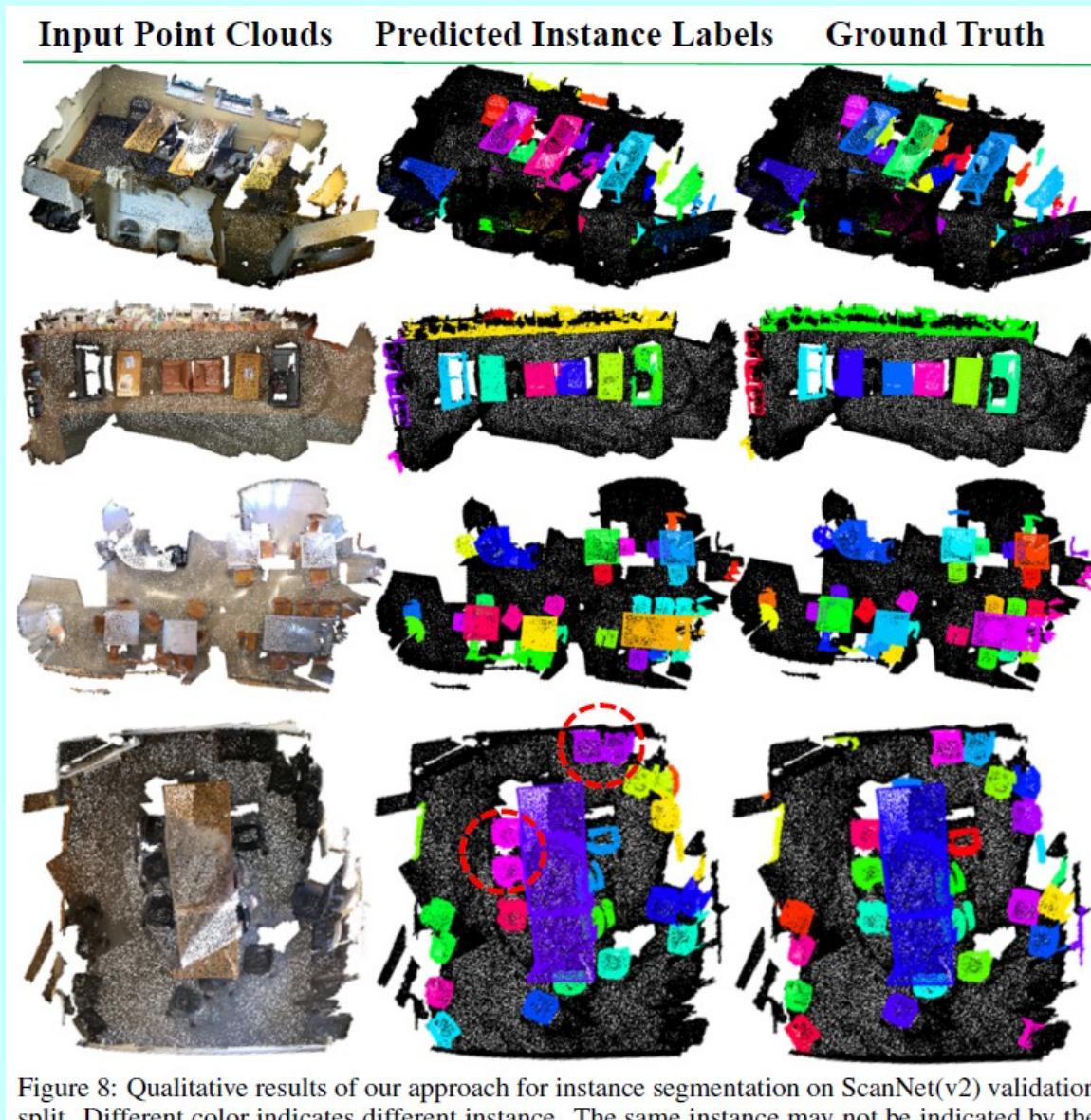
**- 3 -**

# **PREZENTACJA WYNIKÓW I PODSUMOWANIE**

# Parametry i zbiory danych

- **Parametry trenowania sieci:**
  - *backbone*: PointNet++ z domyślnymi hiperparametrami;
  - *optimizer*: ADAM z domyślnymi parametrami;
  - *learning rate*: początkowa: 0.0005 (dzielona przez 2 co 20 epok).
- **Zbiory danych:**
  - **ScanNet(v2)** – benchmark do semantycznej segmentacji instancji, 1513 zeskanowane sceny wewnętrz budynków (1201/312 split); podobnie jak SGPN, trenowanie na blokach wielkości 1m x 1m (1.5m x 1.5m ???) z wybranymi 4096 punktami i testowanie na wszystkich punktach (*BlockMerging algorithm*);
  - **S3DIS** – benchmark do semantycznej segmentacji instancji, 271 skanów pokoi z 6 dużych obszarów; przygotowanie danych testowych jak dla innych modeli;  $H = 24$ ;

# Wyniki – ScanNet(v2) (1)



# Wyniki – ScanNet(v2) (2)

- Porównanie wyników z innymi modelami (metryka – precyza w % na *Intersection-over-Union*):

	mean	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	other	picture	refrig	showerCur	sink	sofa	table	toilet	window
MaskRCNN [13]	5.8	33.3	0.2	0.0	5.3	0.2	0.2	2.1	0.0	4.5	2.4	23.8	6.5	0.0	1.4	10.7	2.0	11.0	0.6
SGPN [50]	14.3	20.8	39.0	16.9	6.5	27.5	2.9	6.9	0.0	8.7	4.3	1.4	2.7	0.0	11.2	35.1	16.8	43.8	13.8
3D-BEVIS [8]	24.8	66.7	56.6	7.6	3.5	39.4	2.7	3.5	9.8	9.9	3.0	2.5	9.8	37.5	12.6	60.4	18.1	85.4	17.1
R-PointNet [58]	30.6	50.0	40.5	31.1	34.8	58.9	5.4	6.8	12.6	28.3	29.0	2.8	21.9	21.4	33.1	39.6	27.5	82.1	24.5
UNet-Backbone [28]	31.9	66.7	71.5	23.3	18.9	47.9	0.8	21.8	6.7	20.1	17.3	10.7	12.3	43.8	15.0	61.5	35.5	91.6	9.3
3D-SIS (5 views) [15]	38.2	<b>100.0</b>	43.2	24.5	19.0	57.7	1.3	26.3	3.3	32.0	24.0	7.5	42.2	85.7	11.7	<b>69.9</b>	27.1	88.3	23.5
MASC [30]	44.7	52.8	55.5	38.1	<b>38.2</b>	63.3	0.2	50.9	26.0	36.1	43.2	32.7	<b>45.1</b>	57.1	36.7	63.9	38.6	<b>98.0</b>	27.6
ResNet-Backbone [28]	45.9	<b>100.0</b>	<b>73.7</b>	15.9	25.9	58.7	<b>13.8</b>	47.5	21.7	<b>41.6</b>	40.8	12.8	31.5	71.4	41.1	53.6	<b>59.0</b>	87.3	30.4
PanopticFusion [33]	47.8	66.7	71.2	<b>59.5</b>	25.9	55.0	0.0	61.3	17.5	25.0	<b>43.4</b>	<b>43.7</b>	41.1	85.7	<b>48.5</b>	59.1	26.7	94.4	35.9
MTML	48.1	<b>100.0</b>	66.6	37.7	27.2	<b>70.9</b>	0.1	57.9	25.4	36.1	31.8	9.5	43.2	<b>100.0</b>	18.4	60.1	48.7	93.8	38.4
<b>3D-BoNet(Ours)</b>	<b>48.8</b>	<b>100.0</b>	67.2	59.0	30.1	48.4	9.8	<b>62.0</b>	<b>30.6</b>	34.1	25.9	12.5	43.4	79.6	40.2	49.9	51.3	90.9	<b>43.9</b>

- Wnioski:
  - 3D-BoNet osiąga najwyższy średni wynik, choć używa jedynie surowych danych (chmury punktów), podczas gdy inne modele używają różnych metod i danych pomocniczych (klastrowanie własności punktów – SGPN, 3D-BEVIS, MASC; gęste propozycje obiektów – R-PointNet; używanie dodatkowych kolorowych zdjęć 2D – 3D-SIS);
  - 3D-BoNet nie faworyzuje specyficznych kategorii semantycznych i jest względnie równomiernie skuteczny w nich wszystkich.

# Wyniki – S3DIS (1)

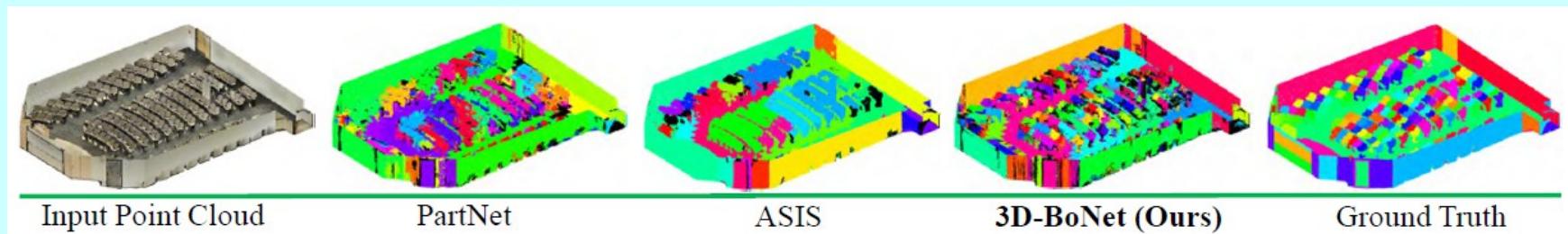


Figure 7: This shows a lecture room with hundreds of objects (e.g., chairs, tables), highlighting the challenge of instance segmentation. Different color indicates different instance. The same instance may not have the same color. Our framework predicts more precise instance labels than others.

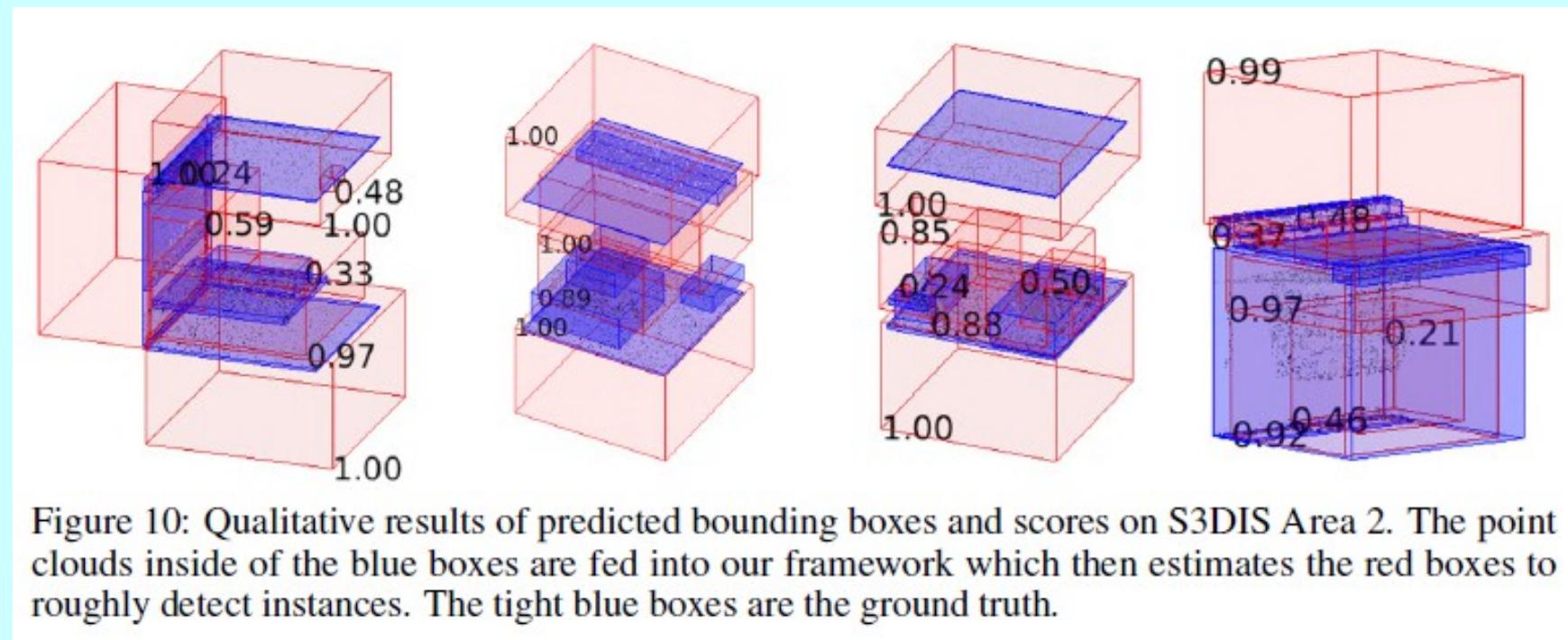
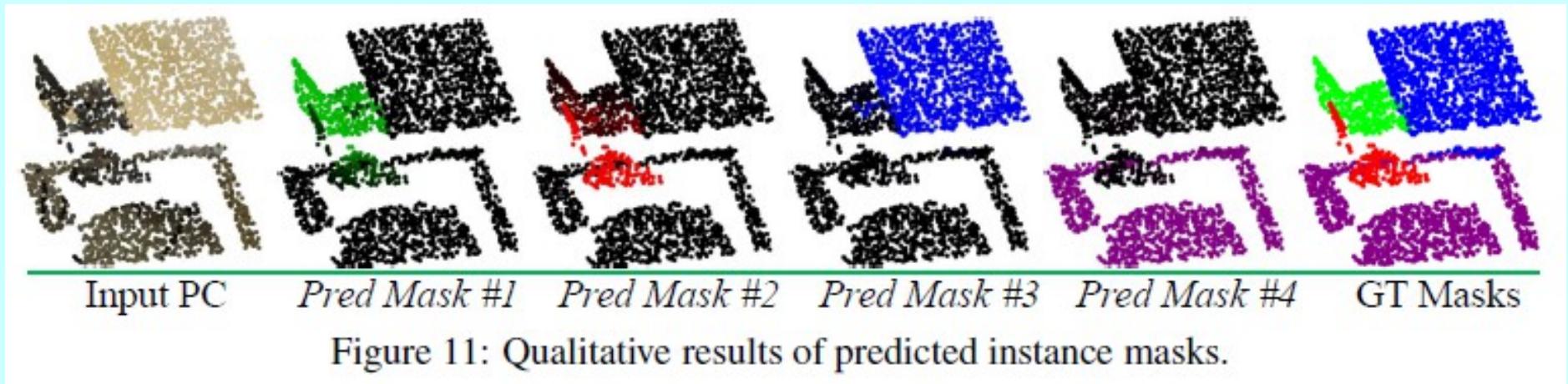


Figure 10: Qualitative results of predicted bounding boxes and scores on S3DIS Area 2. The point clouds inside of the blue boxes are fed into our framework which then estimates the red boxes to roughly detect instances. The tight blue boxes are the ground truth.

# Wyniki – S3DIS (2)



- Wyniki – porównanie z ASIS oraz PartNet (traktowane jako *baseline*; trenowany z takim samym *backbonem* – PointNet++); metryki – *mean precision* i *mean recall* (*6-fold cross-validation*) na *IoC*; średnie wyniki z 13 kat.
- Wnioski:
  - znacznie lepsze wyniki niż baseline (oparte na tym samym *backbonie*);
  - Lepsze wyniki niż ASIS, ale niewiele – część do predykcji semantycznej (PointNet++) jest słabsza niż dedykowany predyktor w ASIS, który jednocześnie uczy się własności sementycznych i instancyjnych, które wzajemnie sprzyjają swojej optymizacji (cel na przyszłość dla 3D-BoNet).

Table 2: Instance segmentation results on S3DIS dataset.

	mPrec	mRec
PartNet [32]	56.4	43.4
ASIS [51]	63.6	47.5
<b>3D-BoNet (Ours)</b>	<b>65.6</b>	<b>47.6</b>

# Ablation study

Table 3: Instance segmentation results of all ablation experiments on Area 5 of S3DIS.

	mPrec	mRec
(1) Remove Box Score Sub-branch	50.9	40.9
(2) Euclidean Distance Only	53.8	<b>41.1</b>
(3) Soft IoU Cost Only	55.2	40.6
(4) Cross-Entropy Score Only	51.8	37.8
(5) Do Not Supervise Box Prediction	37.3	28.5
(6) Remove Focal Loss	50.8	39.2
<b>(7) The Full Framework</b>	<b>57.5</b>	40.2

- Wnioski:
  - ad (1) – oceny BB penalizują regresję do tego samego BB *ground truth*;
  - ad (2-4) – *soft IoU* jest nalepszą samodzielną miarą podobieństwa BB;
  - ad (5) – bez nadzorowanego uczenia, wyniki znaczaco sie pogarszają – sieć nie potrafi samodzielnie rozpoznać, co jest osobna instancją, a co nie;
  - ad (6) – zastąpienie *focal lossa* zwykłą entropią krzyżową przewartościowuje znaczenie punktów tła i prowadzi do pogorszenia wyników;

# Uwagi o złożoności obliczeniowej

- Proponowany model również stanowi poprawę efektywności obliczeniowej ze względu na:
  - Prostą architekturę (MLP);
  - Brak konieczności dodatkowych kroków *post-processingu* (np. algorytm *mean shift* – zob. SGPN, ASIS, JSIS<sub>3</sub>D, 3d-BEVIS, MASC – ma złożoność:  $O(TN^2)$ ; podobnie *non-maximum suppression* – GSPN, 3D-SIS)
- Empirycznie – na takim samym sprzęcie przetwarzanie 4K punktów zabiera około **20ms**, podczas gdy dla większości z powyższych czas ten osiąga **200ms**.

Table 4: Time consumption of different approaches on the validation split (312 scenes) of ScanNet(v2) (seconds).

	SGPN [50]	ASIS [51]	GSPN [58]	3D-SIS [15]	3D-BoNet(Ours)
network(GPU):	650	650	500	voxelization, projection, network, etc. (GPU+CPU):	650
group merging(CPU):	46562	53886	2995	38841	SCN (GPU parallel): 208
block merging(CPU):	2221	2221	neighbour search(CPU): 468	block merging(CPU): 2221	
total	49433	56757	3963	38841	<b>2871</b>

# Wnioski końcowe

- **Podsumowanie:**
  - prosty, efektywny i wydajny obliczeniowo model do segmentacji instancji dla chmur punktów w 3D;
  - model jednoetapowy, pozwalający na trening *end-to-end*;
  - nie wymaga dodatkowych etapów *post-processingu* (często kosztownych obliczeniowo i wykonywanych na CPU);
  - nowy, oryginalny model pozwalający na dalsze badania i udoskonalenia.
- **Potencjalne kierunki dalszego rozwoju 3D-BoNet:**
  - ulepszenie funkcji kosztu przy kojarzeniu BB – zamiast prostej sumy uczenie się wag dających optymalną mieszankę trzech kryteriów;
  - zamiast trenowania osobnej, przejętej z zewnątrz gałęzi do segmentacji semantycznej (tj. *backbone*), opracowanie dedykowanej architektury, pozwalającej na wspólne uczenie własności semantycznych i instancyjnych;
  - prosta struktura (MLP) sprawia, że model nie jest wrażliwy na ilość wejściowych punktów, więc konieczne jest dzielenie sceny na mniejsze bloki, a korzystna byłaby możliwość trenowania na dużych zbiorach danych za jednym razem.

# Dziękuję za uwagę!

[kontakt: **maciejmanna@gmail.com**]

[slajdy: **github.com/xann16/talks/blob/master/ml/3d-bonet/3d-bonet.pdf**]