

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas

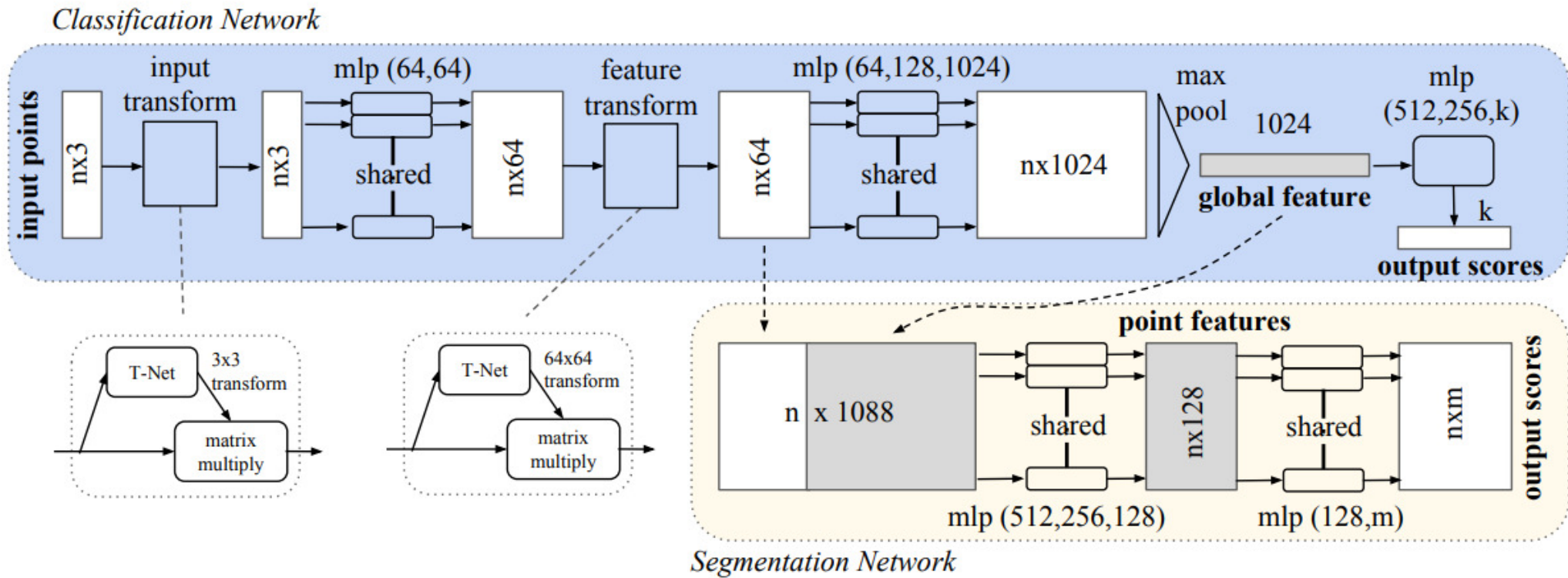
Contexte

- Article publié en 2017 (CVPR)
- Essor des CNN:
 - Segmentation / classification 2D
 - Généralisation sur d'autres tâches
- Nuages de points et ML:
 - Pas / peu de solution basé deep learning
 - Utilisation de CNN sur nuages + éléments structurant

Problème posé

- Utiliser des nuages de points
- Répondre aux contraintes:
 - Invariance aux permutations
 - Robuste à certaines transformations géométriques
 - Ensemble non structuré

Architecture



Invariance aux permutations

Trier les points

- + :
 - simple à mettre en place
- - :
 - Problématique si on est en haute dimension

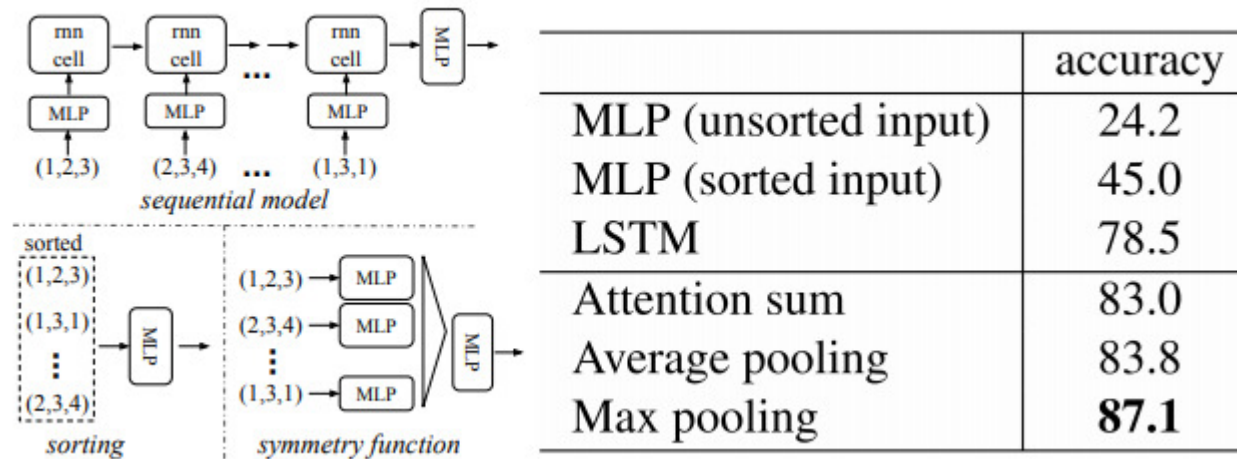
Points comme séquence

- + :
 - Utilisation d'un RNN
 - Appris par le réseau
- - :
 - Entraîner sur toutes les permutations
 - Nombre de points

Définir fonction symétrique

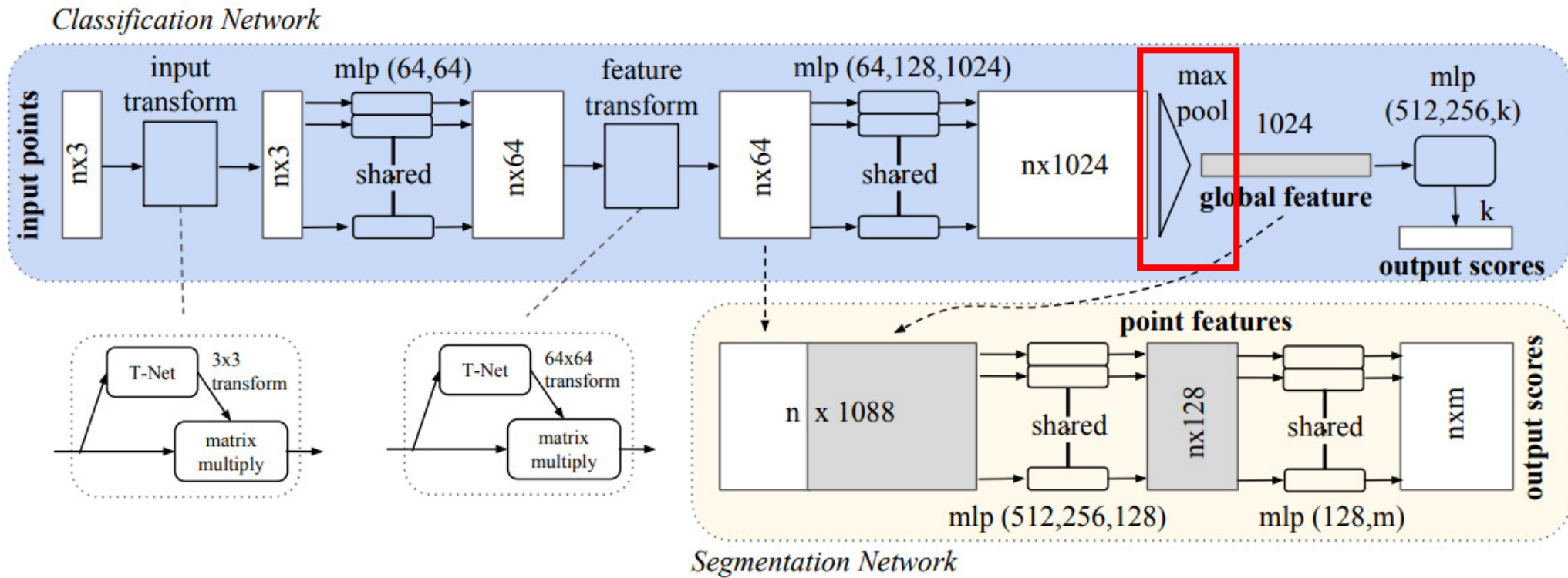
- + :
 - Apprise par le réseau
 - Ne dépend pas de la taille
- - :
 - Perte de détails ?

Invariance aux permutations



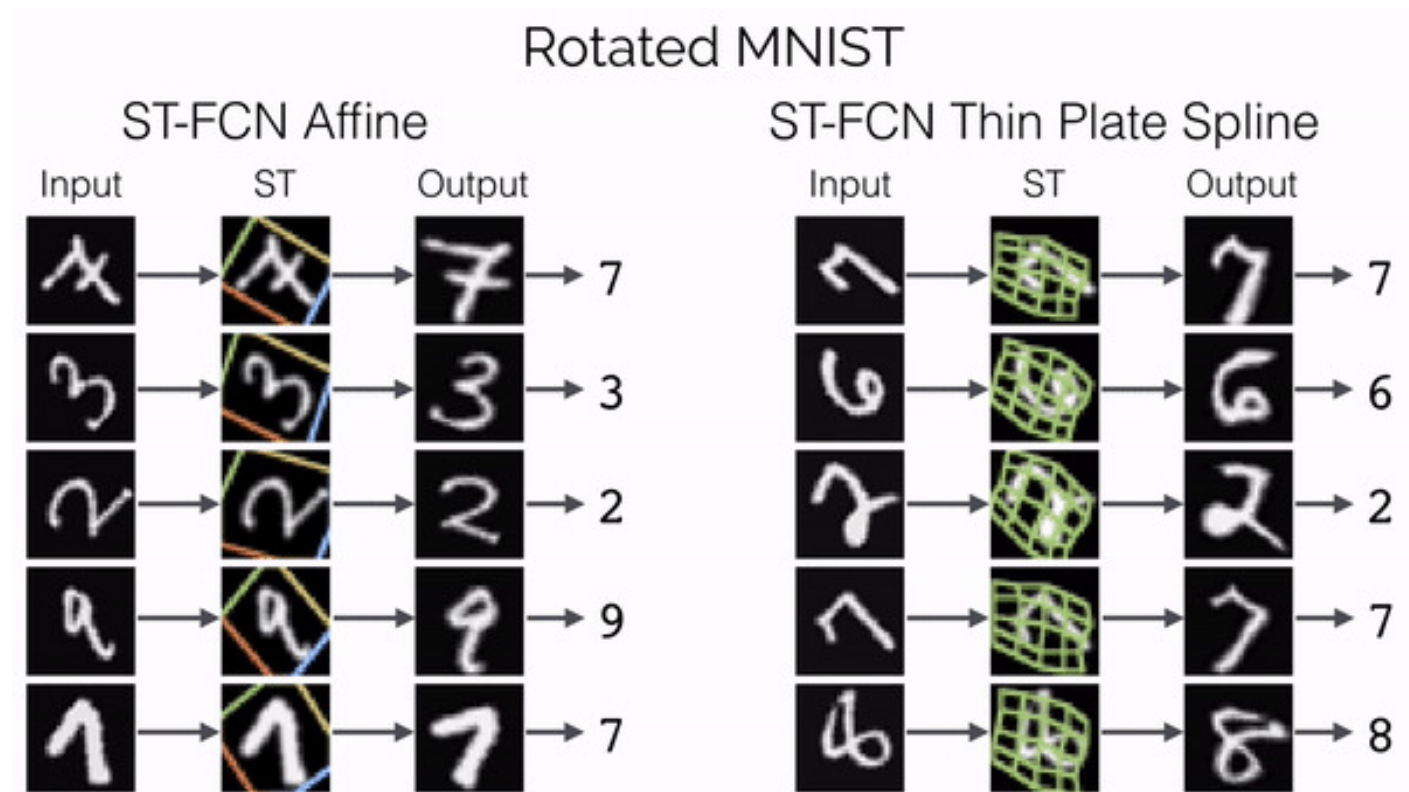
- Input trié > Input non trié
- Point en tant que séquence beaucoup plus efficace
- Fonction symétrique meilleur résultat

Architecture



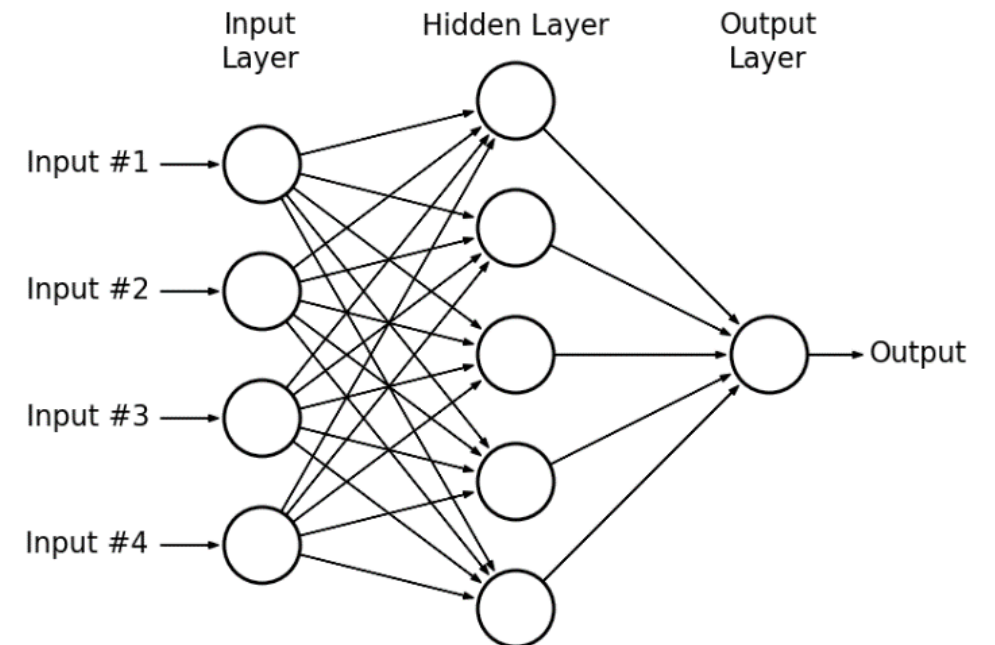
Robustesse transformations

- Solution classique : aligner les éléments dans une pose canonique
- Cas 2D:
 - Spatial Transform Network [1]

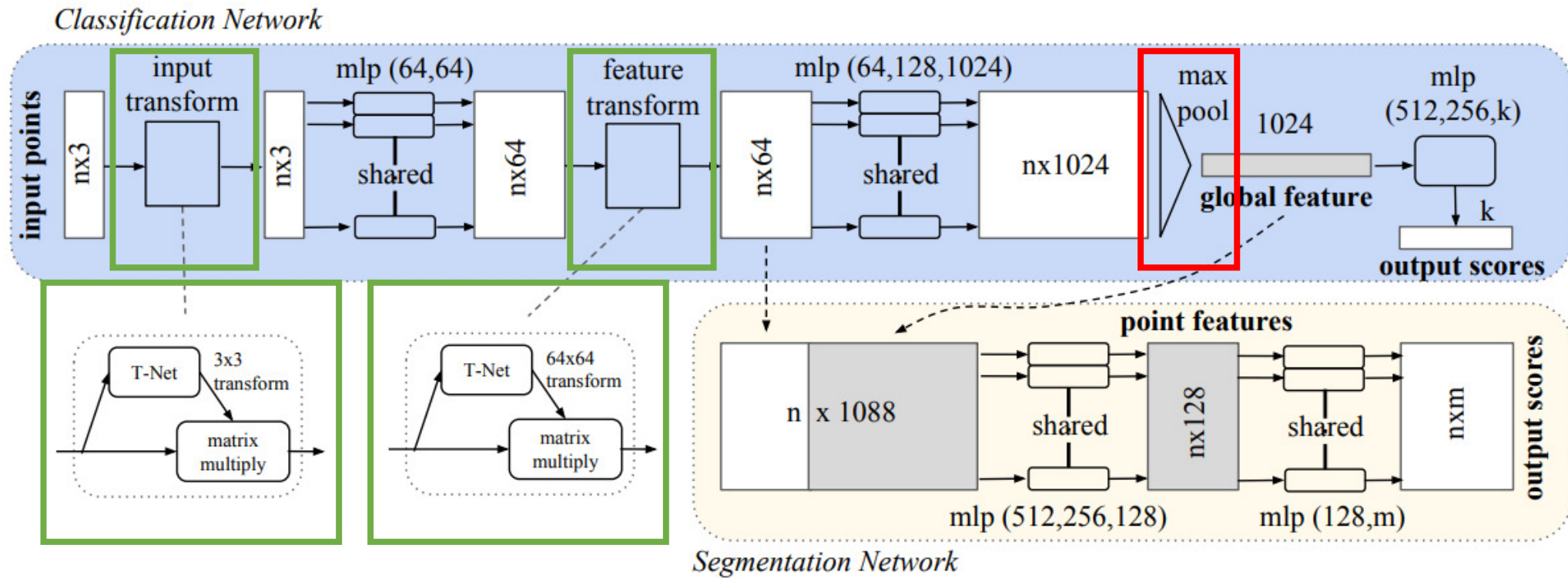


Robustesse transformations

- Proposition :
 - Utiliser un MLP pour apprendre à aligner des éléments
- Aligner dans une pose les points
+ les features
 - + = même traitement sur tout les points
 - - = features peuvent être de grande dimension
=> forcer la matrice de transfo vers une matrice orthogonale

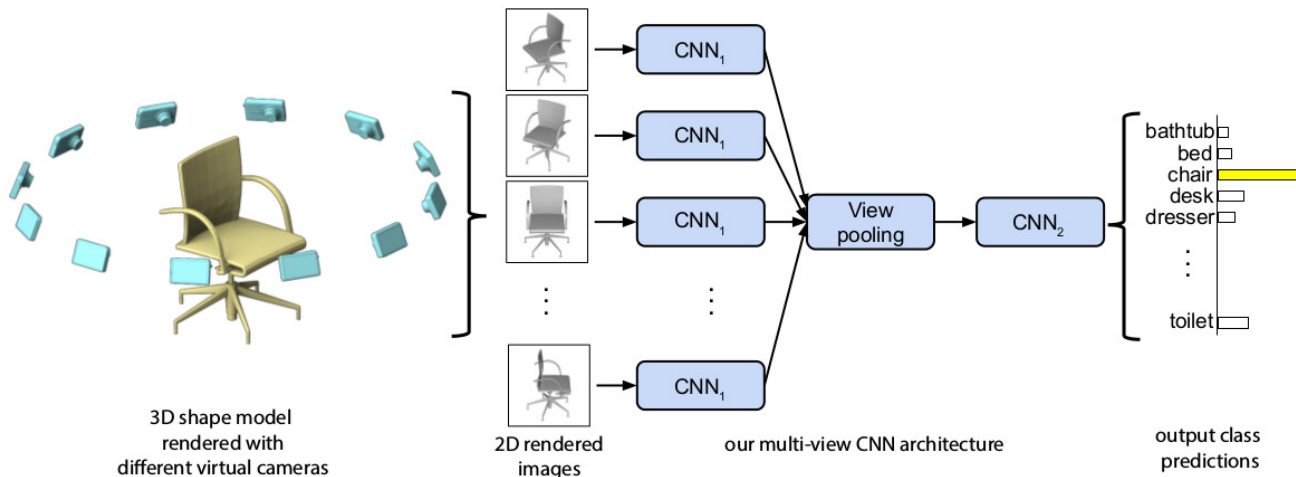


Architecture



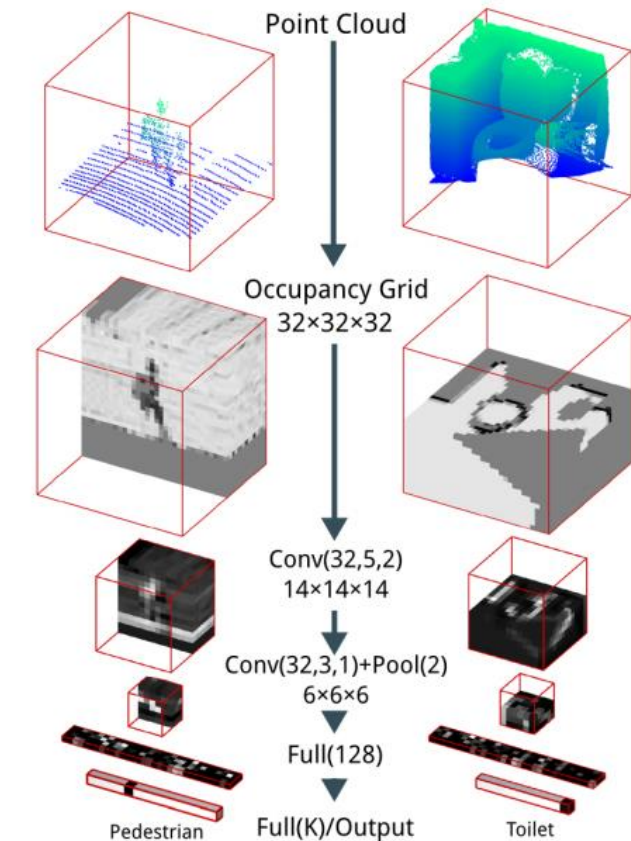
Ensemble non structuré

- Comment extraire des features ?
- Méthode précédente:
 - Features extraite « à la main »
 - Utilisation d'élément structurant



MVCNN [3]

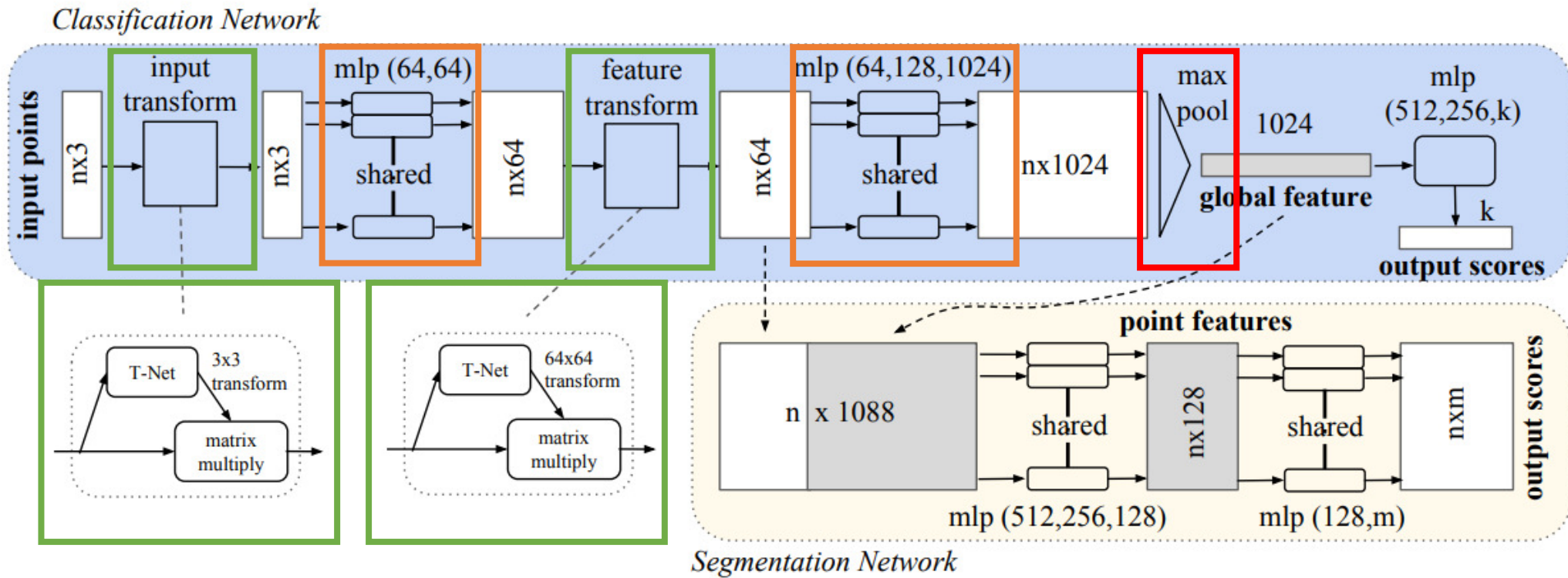
VoxNet [2]



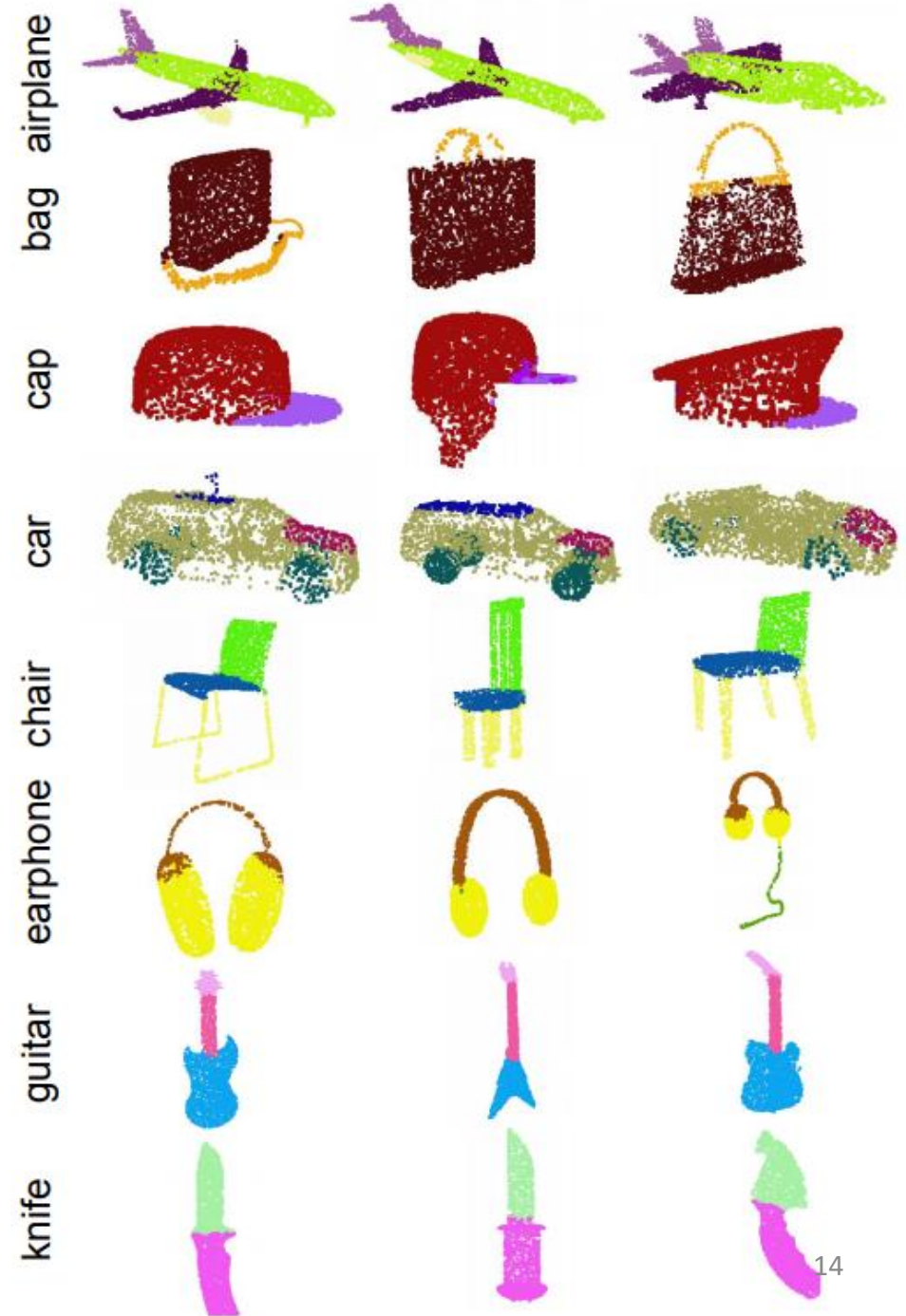
Ensemble non structuré

- Proposition :
 - Traiter les points indépendamment et agréger les features
- 2 contraintes :
 - Traiter les points de la même manière
 - Pouvoir agréger l'information des points
- => solution : utiliser un MLP (encore)

Architecture

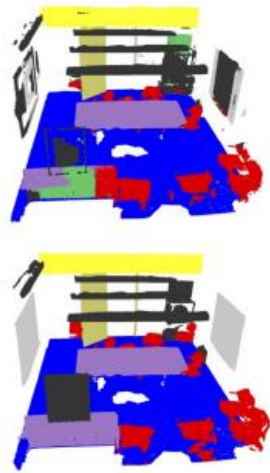
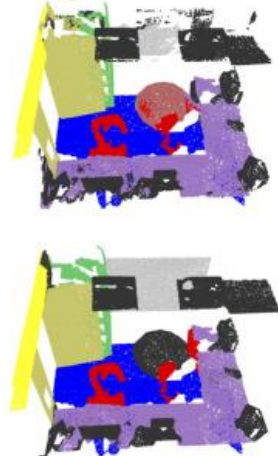
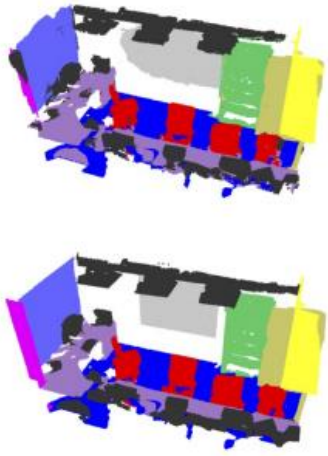


Résultat



Semantic Segmentation

GT pred

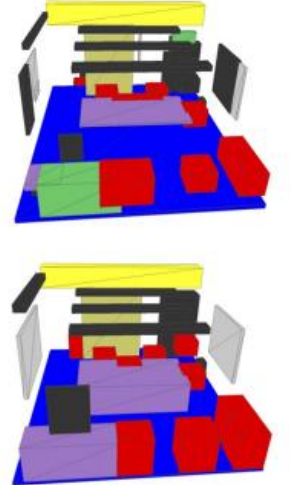
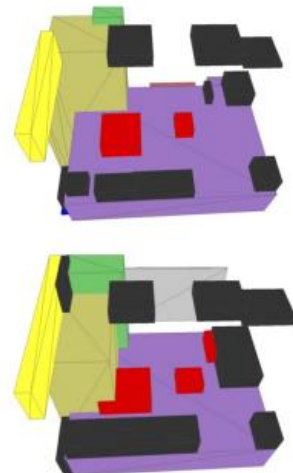
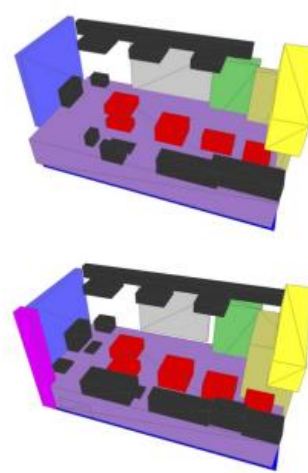


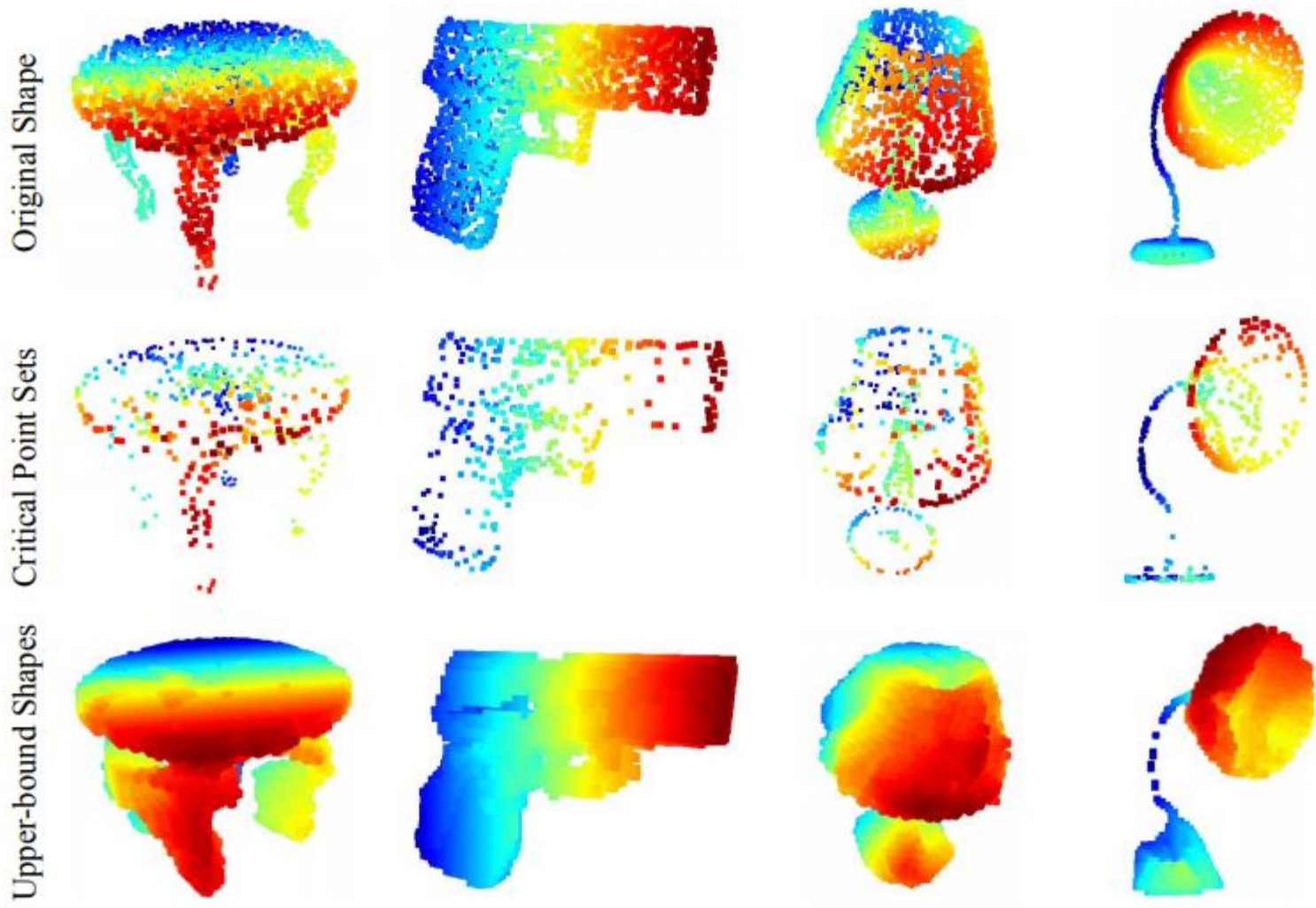
Input
Point Cloud



Object Detection

GT pred





Critical Point set : Points ayant contribué au descripteur global
=> Squelette de l'objet

Upper bound shape : Points dans un cube ayant des features égales ou moins grandes que le descripteur global

Références

R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, « PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation », 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 77-85, doi: 10.1109/CVPR.2017.16.

[1] *Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. 2015. Spatial transformer networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15). MIT Press, Cambridge, MA, USA, 2017–2025.*

[2] *D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 922-928, doi: 10.1109/IROS.2015.7353481.*

[3] *H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 945-953, doi: 10.1109/ICCV.2015.114.*