Indoor Autonomous Nevigation Drone System with Semantic-SLAM based on SD-DETR

室內自動導航無人機系統搭配語意化同時定位與地圖重建基於 SD-DETR

Cheng-Wei Huang
黃政維

Civil Engineering
National Taiwan University

Outline

- Autonomous Drone
- SLAM System
- Transformer

Indoor Autonomous Drone

- Building customized drone with following equipment

Jetson Xavier NX

onboard computer to process the sensor data

Pixhawk

flight control unit to control the basic movemont of UAV

Intel Realsense D435

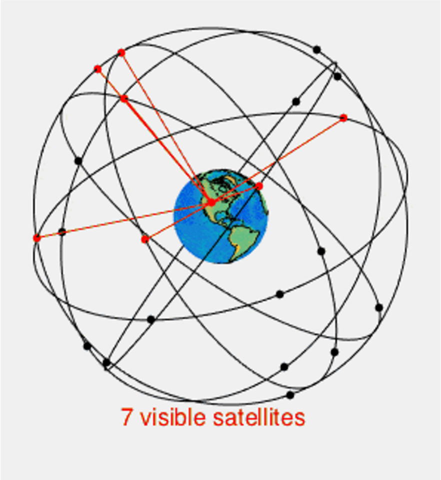
depth image and color image for mapping

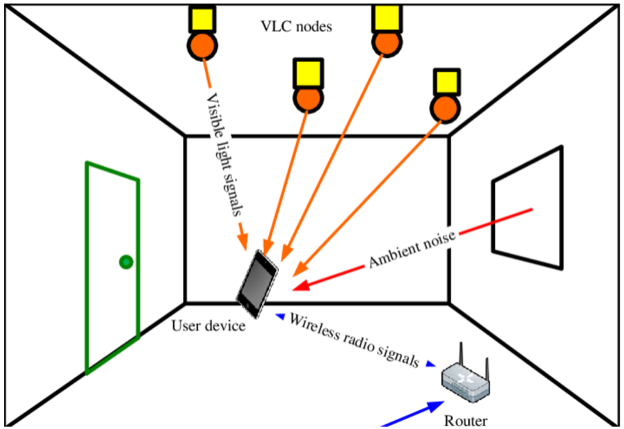
Intel Realsense T265

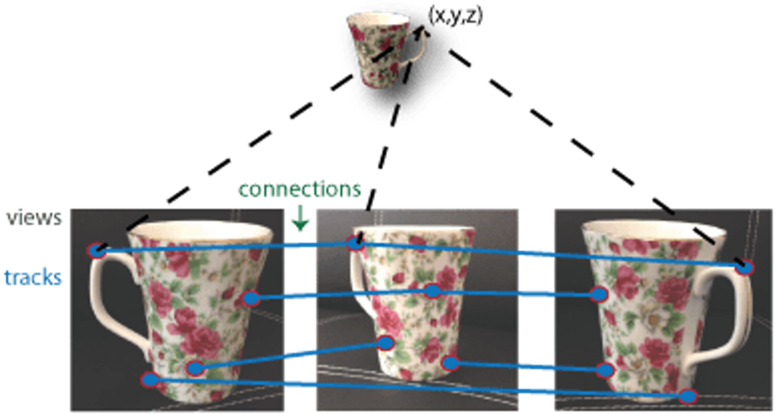
provide high precision UAV position for Indoor environment where GPS is unreachable

Robot Localization

- Outdoor Environment
  - GPS System
- Indoor Environment
  - Sensor System
  - SfM (Structure from Motion)
  - SLAM System

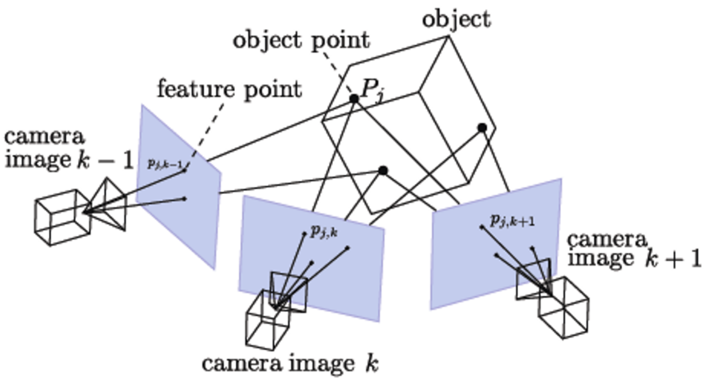




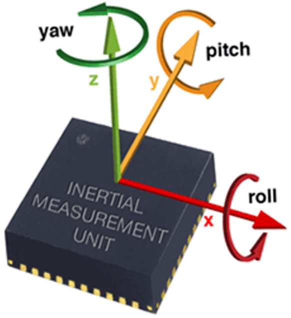


Robot Odometry (T265 Camera)

- Visual Method
  - Extract feature points in image
  - Calculate the position by geometry with feature pointes
  - Orb Feature, SIFT Feature
  - CNN model

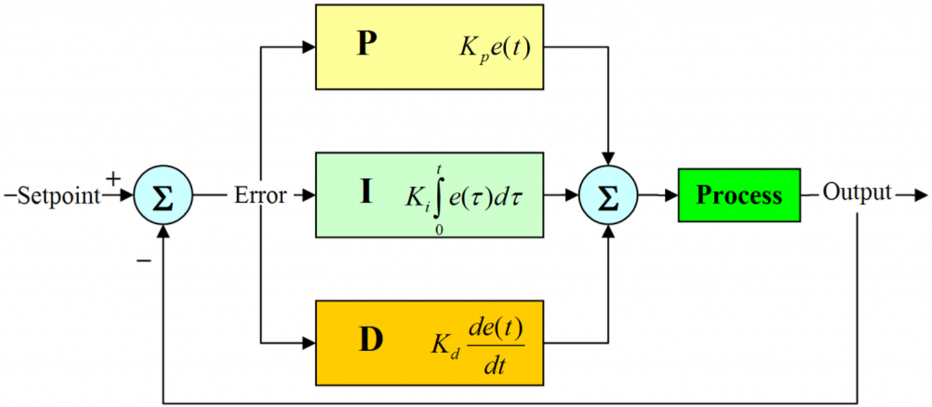


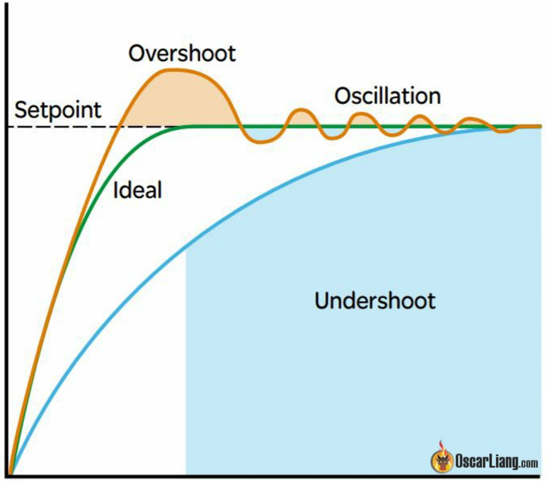
- IMU (Inertial Measurement Unit)
  - Measurement of acceleration and angular rate
  - Calculate the position by integration of acceleration and angular rate



Robot Controlling (Pixhawk)

- Due to our customized setting, the weight is imbalanced so the brush might not output stable power.
- Manual tuning the Propotional, Integral and Derivative gains to stablize the flight of drone.
  - Propotional - Sensitivity and Responsiveness
  - Integral - Stiffness
  - Derivative - Reduce the overcorrecting



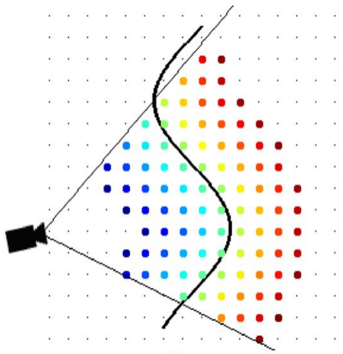


Autonomous Nevigation

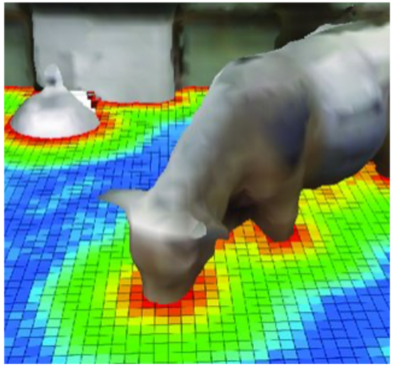
- Three essential components to fly autonomously at the indoor environment
- Drone odometry
  - T265 camera provide odometry by IMU and vision feature tracking
- Obstacle Map
  - D435 provide depth image
  - Combine depth image and odometry to detect the obstacle distance
  - Using obstacle distance to detect the nearest obstacle
- Path Planning
  - Use the path planning algorithm to avoid obstacles
  - Search Based Algorithm
    - Dijkstra's algorithm, A* algorithm
  - Sample Based Algorithm
    - Rapidly-exploring Random Tree*, Probabilistic Roadmap

Obstacle Map

- TSDF (Truncated Signed Distance Function)
  - Use the depth image to reconstruct the object surface
  - Truncated the point distance that’s far away from the obstacle



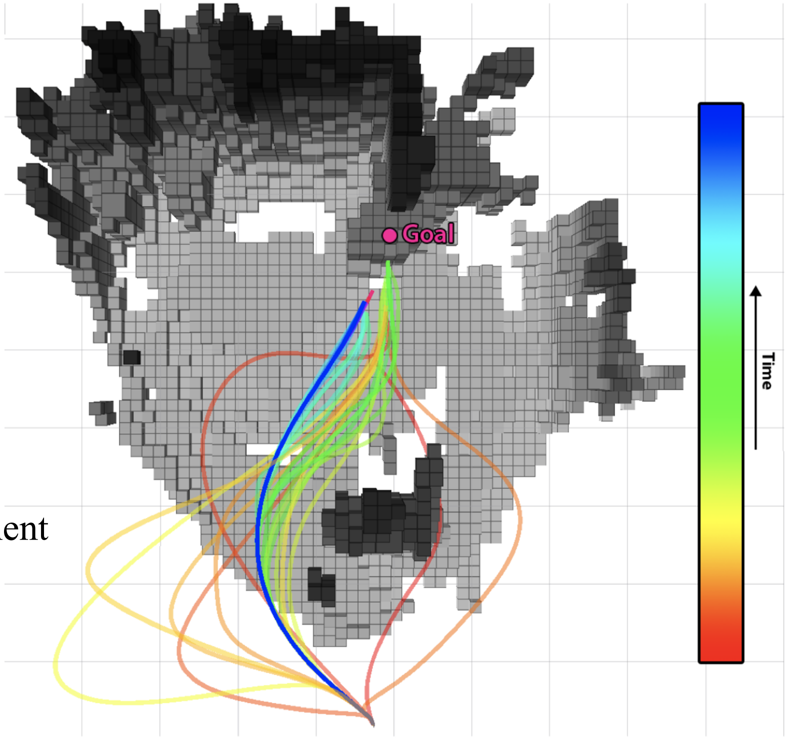
- ESDF (Euclidean Signed Distance Function)
  - Integrate the TSDF for each frame and frame position
  - A map that indicates the nearest obstacle for given point



Path Planning

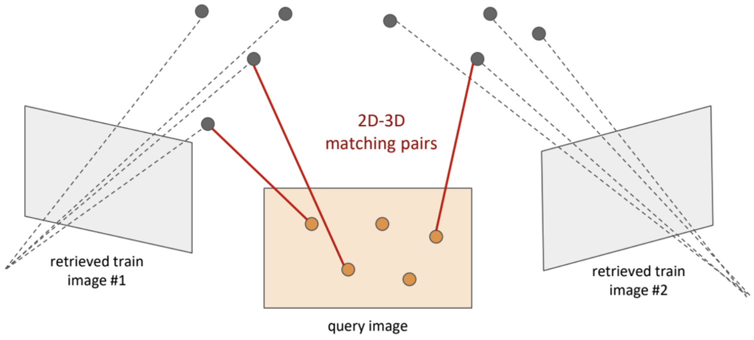
- Global Path Planning
  - Path Planning on known environment
  - RRT* Planning

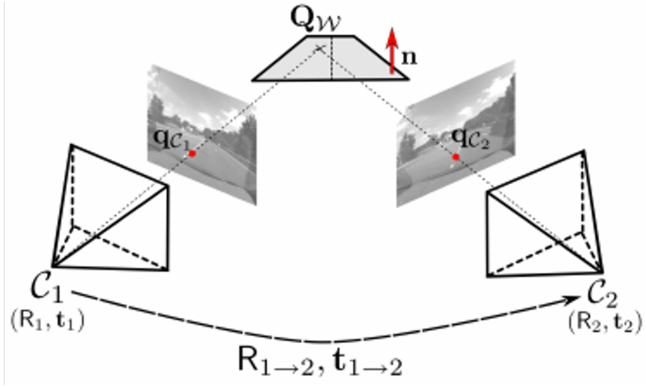
- Local Path Planner
  - Planner keep updating new path in unknown environment
  - Similar to RRT* Planning



Structure from Motion

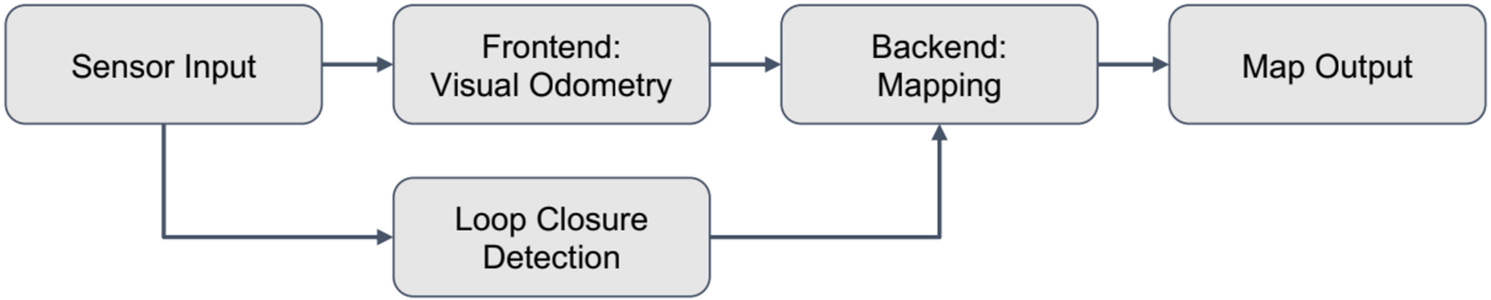
- Feature Detection
  - Orb, SIFT, SURF feature detection
- Pose Estimation
  - P3P, PnP
- Project Image Point using Pose
  - Reprojection Matrix
  - Point Cloud





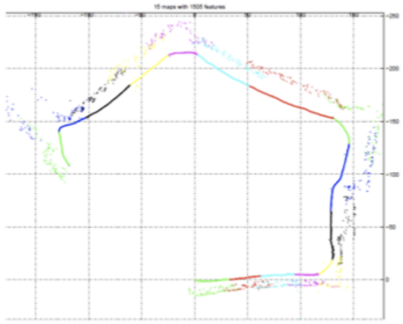
SLAM

- Construct egomotion and map with input camera frames.
    - Input: monocular, stereo (binocular), or RGBD camera frames
    - Frontend (VO): estimate relative motion & construct local map
    - Loop Closure Detection: determine whether the place is visited
    - Backend (Mapping): construct and maintain global map to reduce drift
    - Output: global map and consistent camera trajectory



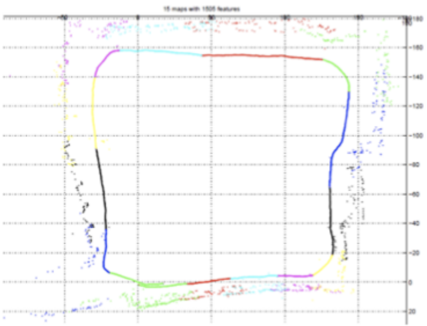
Comparison

- Visual Odometry:
  Incrementally estimate the pose of the agent by examining the changes that motion induces on the images of its onboard cameras.
  ① Incremental estimation (relative pose)
  ② Local consistency (drift)



Visual odometry

- Visual SLAM:
  ① Visual odometry + Loop detection & closure
  ② Global consistency (drift recovery)



Visual SLAM

Image courtesy of [Clemente et al., RSS'07]

Semantic SLAM

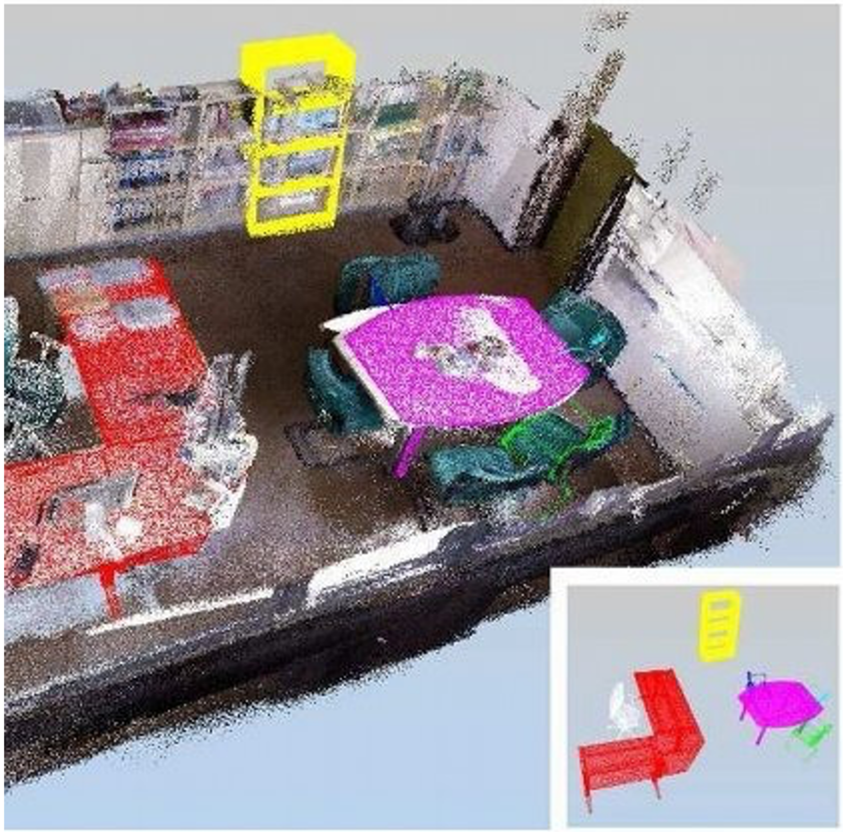
- **SLAM++: Simultaneous Localisation and Mapping at the Level of Objects** (Renato et al, Imperial College London)
    - Build SLAM that can be aware of individual object by using ICP and Point- Pair Features algorithm

- **SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks** (John et al, Dyson Robotics Lab, arXiv, 2016)
    - Combined the CNN and ICP to build semantic map and use CRF to merge different frame result.

- **DA-RNN: Semantic Mapping with Data Associated Recurrent Neural Networks** (Yu et al, Paul G. Allen School of Computer Science & Engineering, arXiv, 2017)
    - Use RNN model to further process multi-frame data with RGB and Depth frame as input.

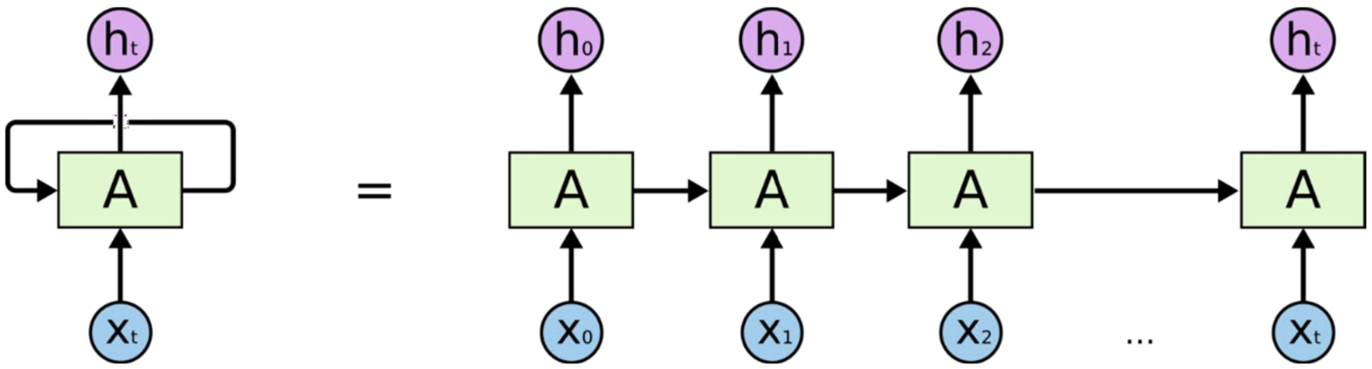
Semantic SLAM

- Project segmentation outcome into 3d Map
- Enhance the Bundle Adjustment
  - Object close to other should be same semantic label
- Detect moving object and filter out
- Invarient of light, season, photo position



RNN

- How to process sequential data
- Memory old time data
- Gradient Vanishing or Gradient Exploring

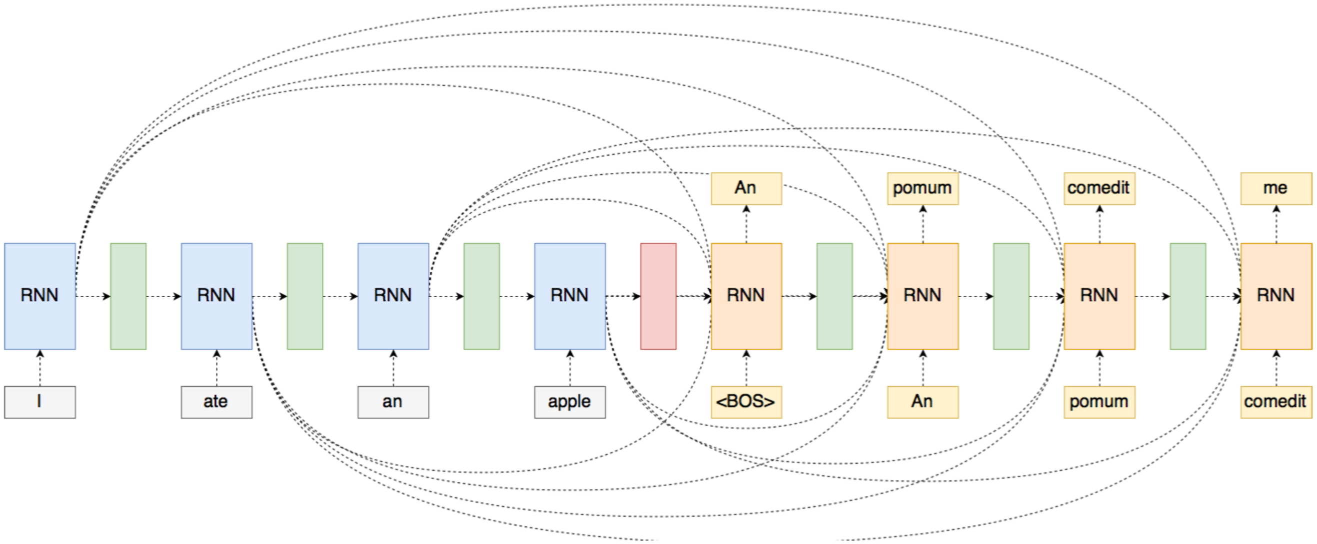


Different Version of RNN

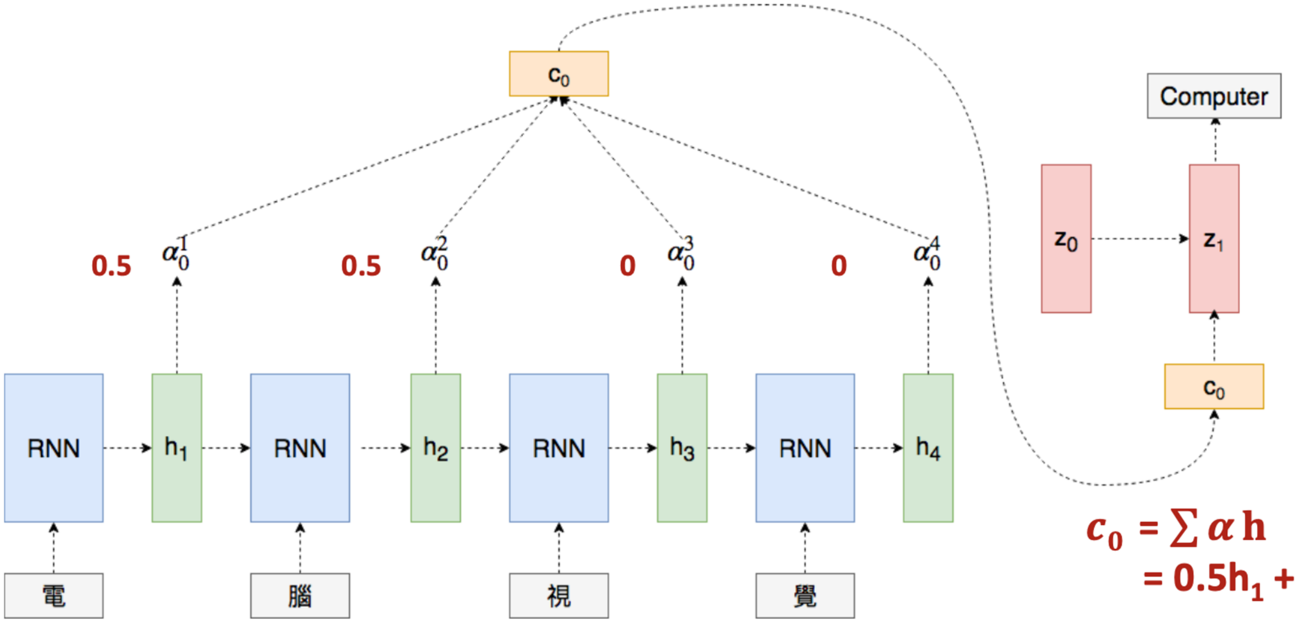
- Long Short-term Memory (LSTM) [Hochreiter et al., 1997] • Additional memory cell
  - Input/Forget/Output Gates
  - Handle gradient vanishing
  - Learn long-term dependencies
- Gated Recurrent Unit (GRU) [*Cho et al., EMNLP 2014*] • Similar to LSTM
  - Noadditionalmemorycell
  - Reset / Update Gates
  - Fewer parameters than LSTM
  - Comparable performance to LSTM [*Chung et al., NIPS Workshop 2014*]

Attention

- Sequential data extract feature with given time series
- What if different time data have different meaning
- OverParameterized

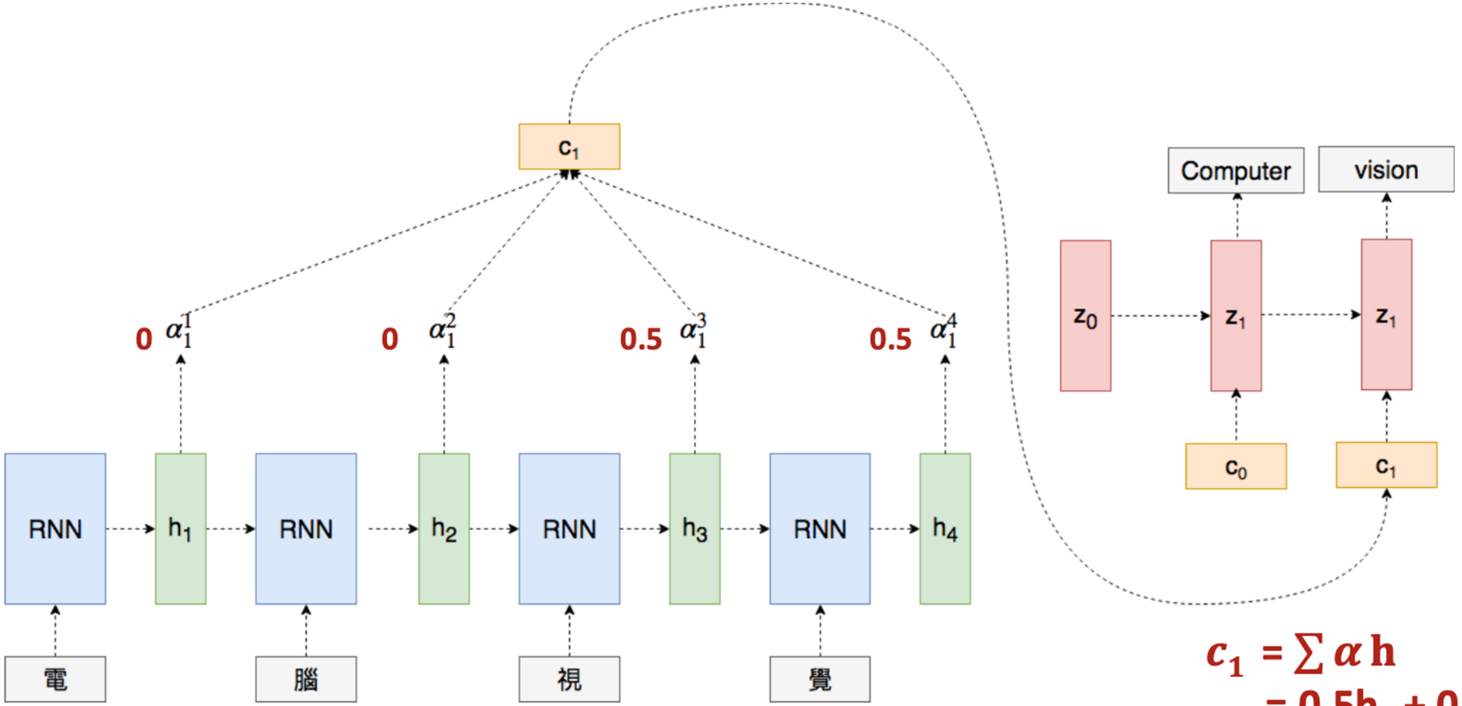


Attention



$$c_0 = \sum \alpha h$$
$$= 0.5h_1 + 0.5h_2$$

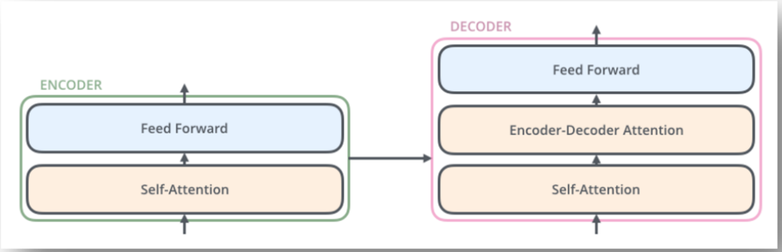
Attention

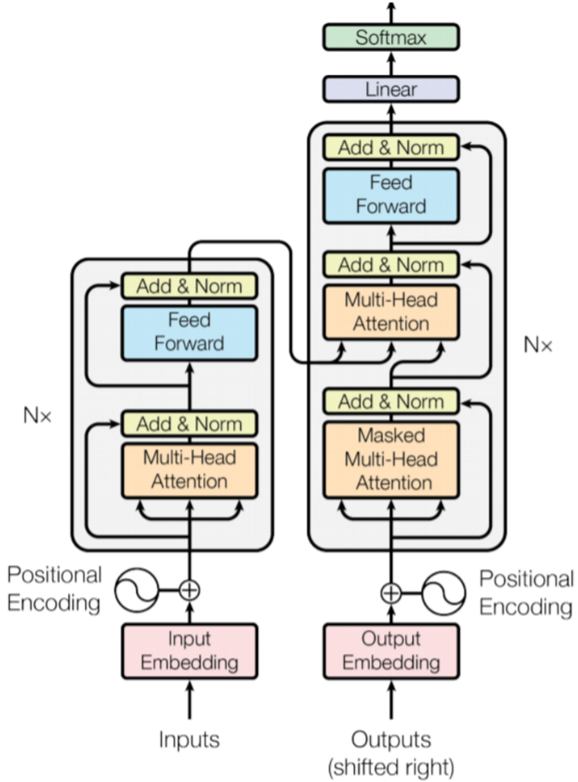


$$c_1 = \sum \alpha \, \mathbf{h}$$
$$= 0.5\mathbf{h}_3 + 0.5\mathbf{h}_4$$

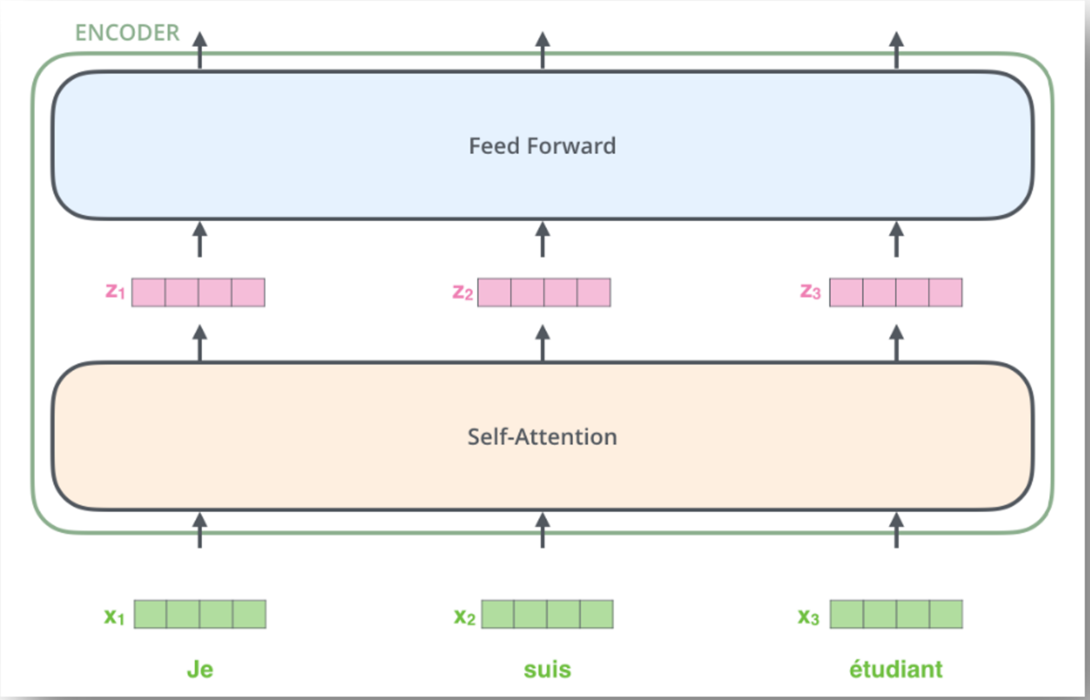
Transformer

- Attention is all you need





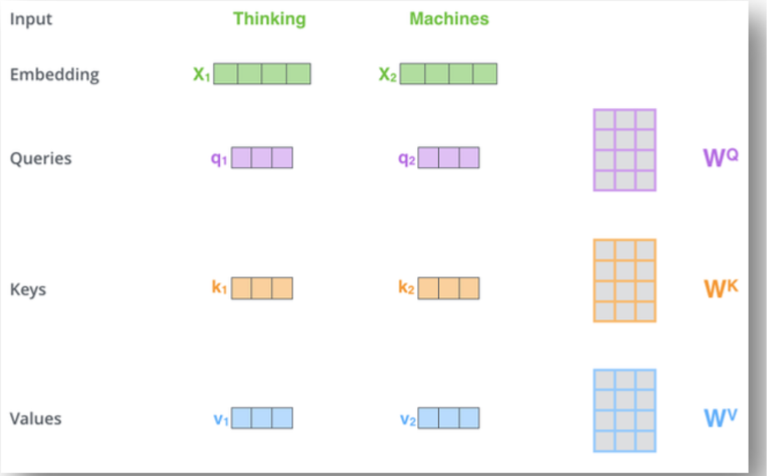
Self-Attention

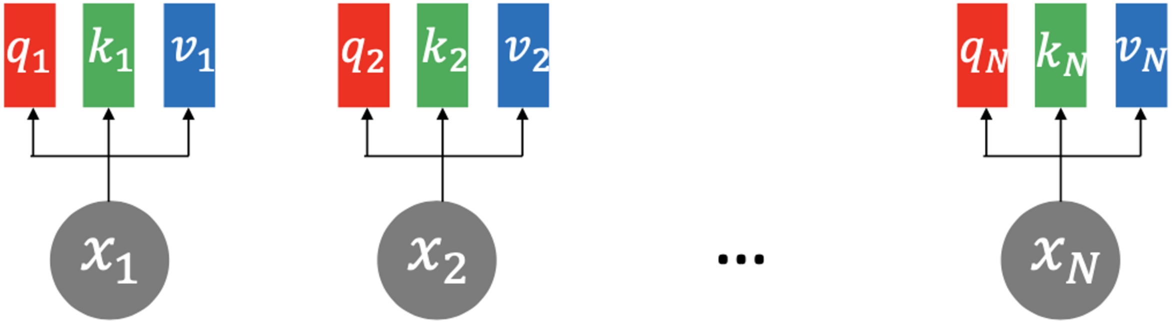


Self-Attention

- Query ***q***, key ***k***, value ***v*** vectors are learned from each input ***x***

$$q_i = W^Q x_i$$
$$k_i = W^K x_i$$
$$v_i = W^V x_i$$

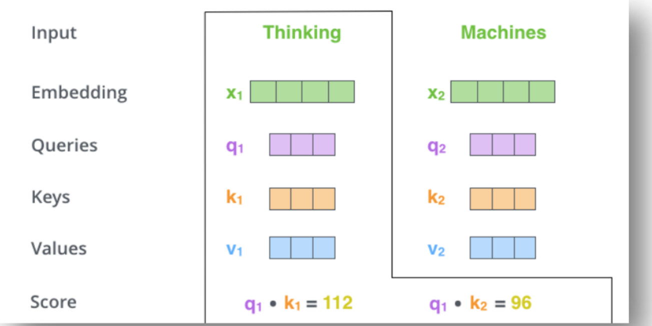


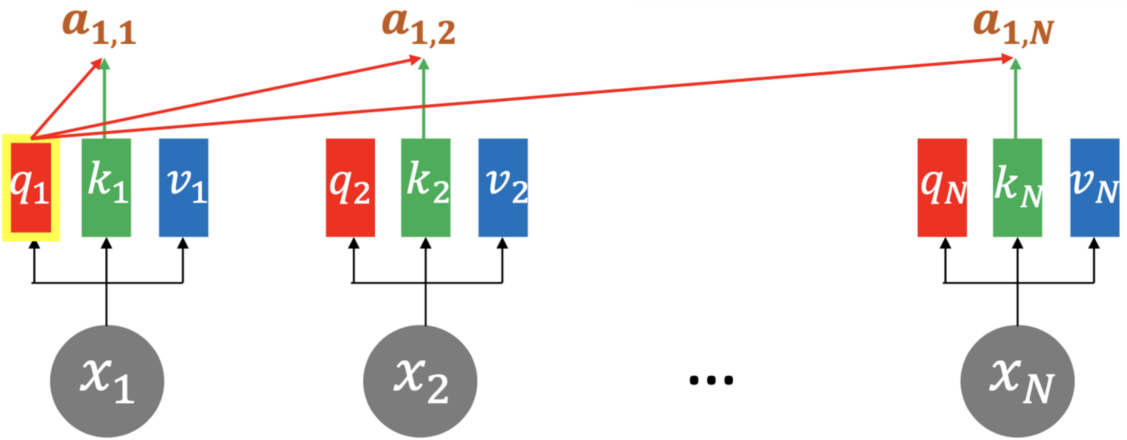


Self-Attention

- Relation between each input is modeled by inner-product of query $\boldsymbol{q}$ and key $\boldsymbol{k}$.

$$a_{1,i} = \frac{q_1 \cdot k_i}{\sqrt{d}}, \text{ where } a \in R, q, k \in R^d$$

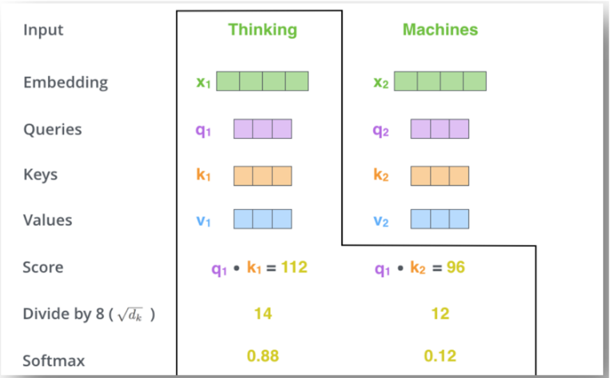




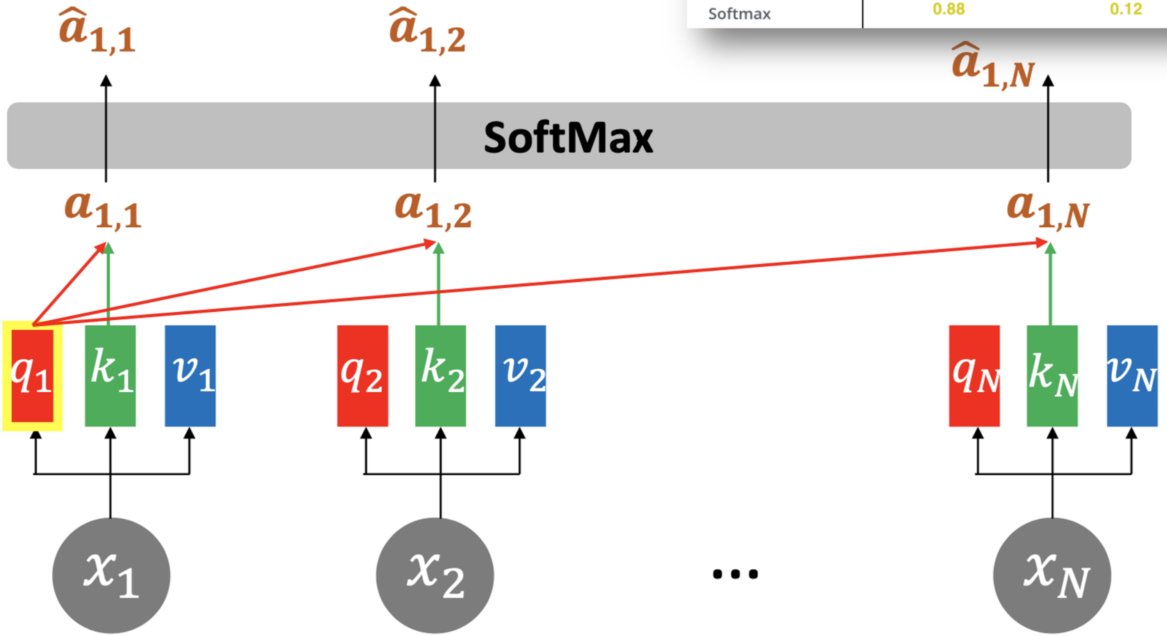
Self-Attention

- SoftMax is applied:

$$0 \le \hat{a}_i = e^{a_i} / \sum_j^N e^{a_j} \le 1 \text{ , for I =1, ..., N}$$

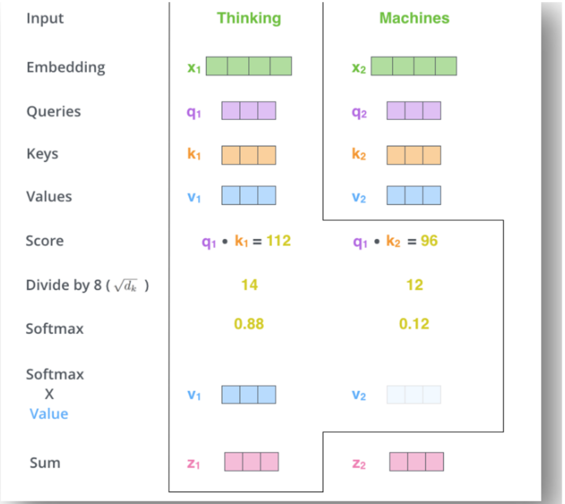


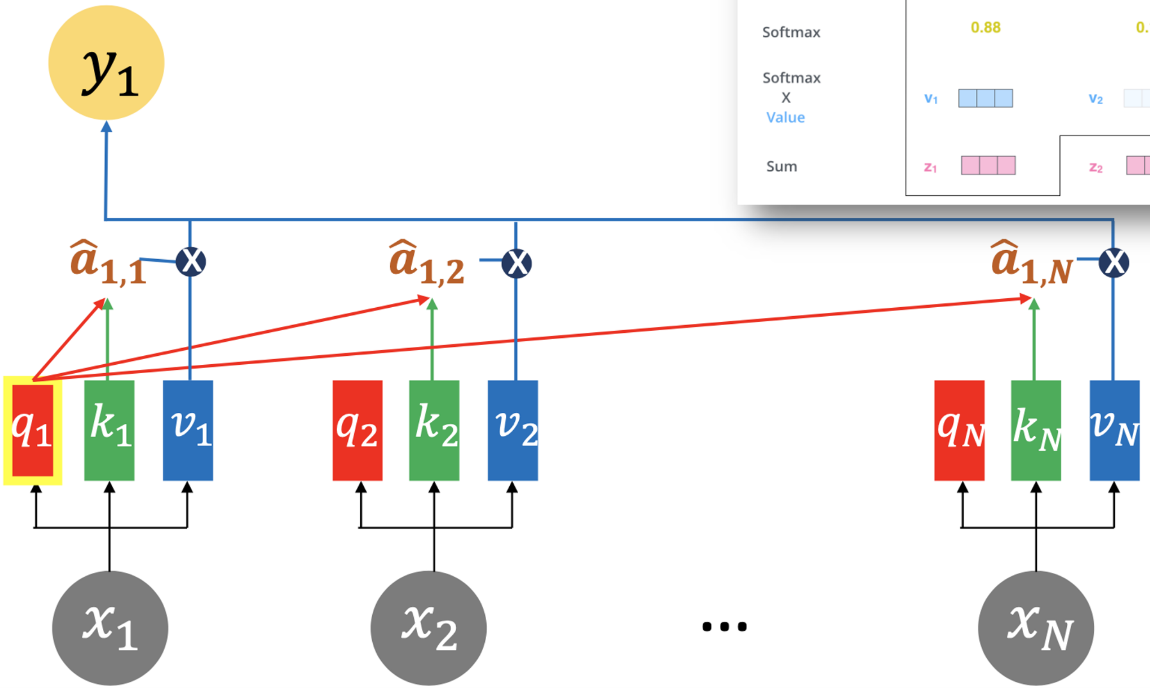
Input	Thinking	Machines
Embedding	x_1	x_2
Queries	q_1	q_2
Keys	k_1	k_2
Values	v_1	v_2
Score	$q_1 \cdot k_1 = 112$	$q_1 \cdot k_2 = 96$
Divide by 8 ($\sqrt{d_k}$)	14	12
Softmax	0.88	0.12



Self-Attention

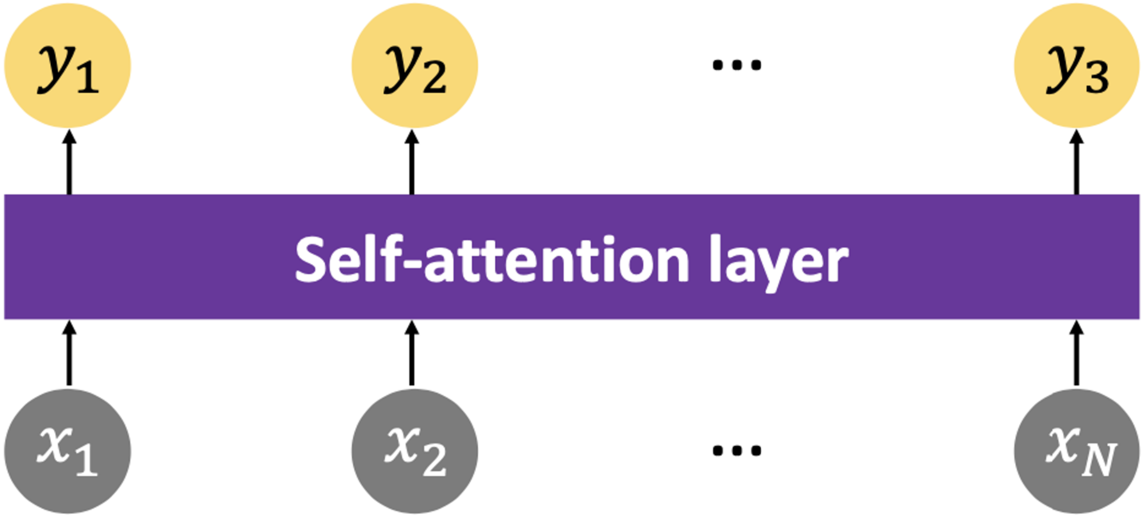
- Value vectors $\boldsymbol{v}$ are aggregated with attention weight $\hat{a}$, i.e., $y_1 = \sum_i^N \hat{a}_i \cdot v_i$





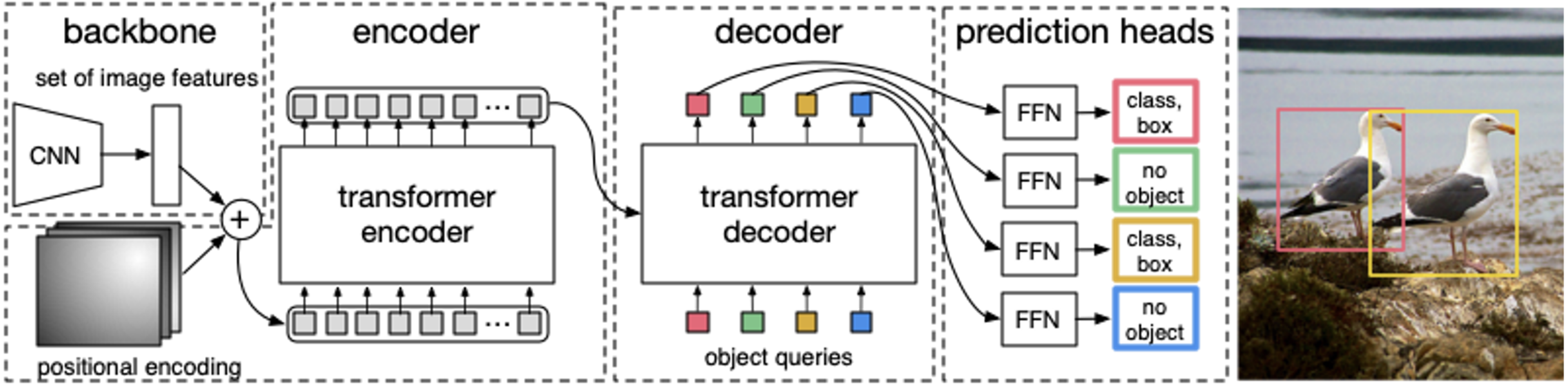
Self-Attention

- All y_i can be computed in parallel
- y_i considers $x_1 \sim x_N$, modeling long-distance dependencies.
- Global feature can be obtained by average-pooling over $y_1 \sim y_N$

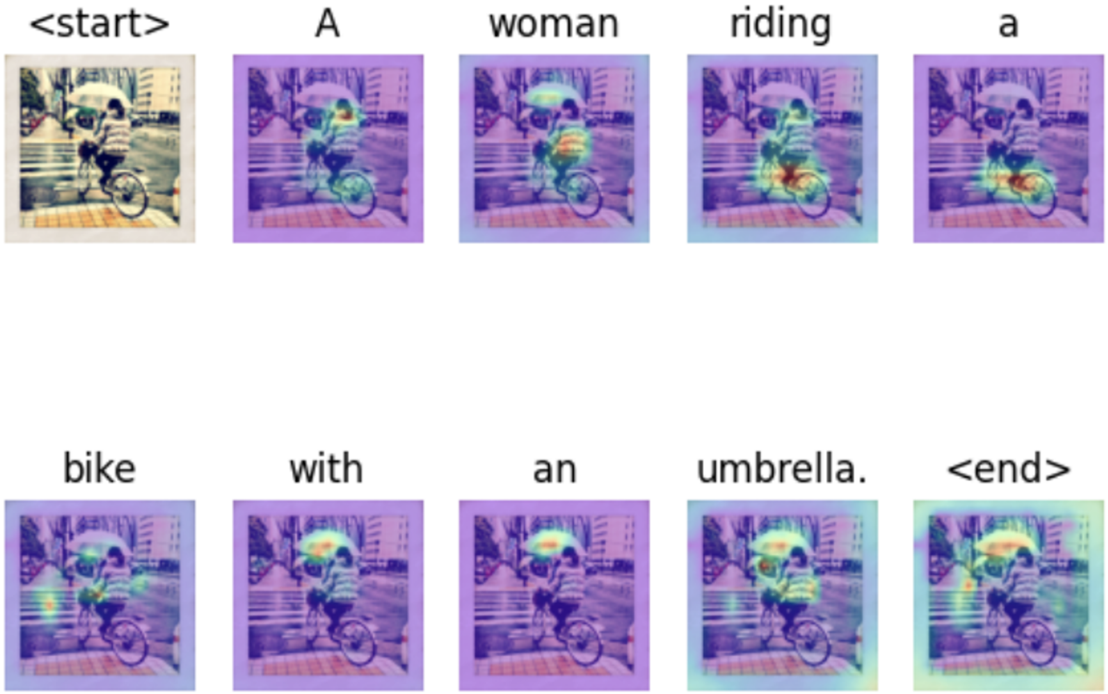


DETR (End-to-End Object Detection with Transformers)

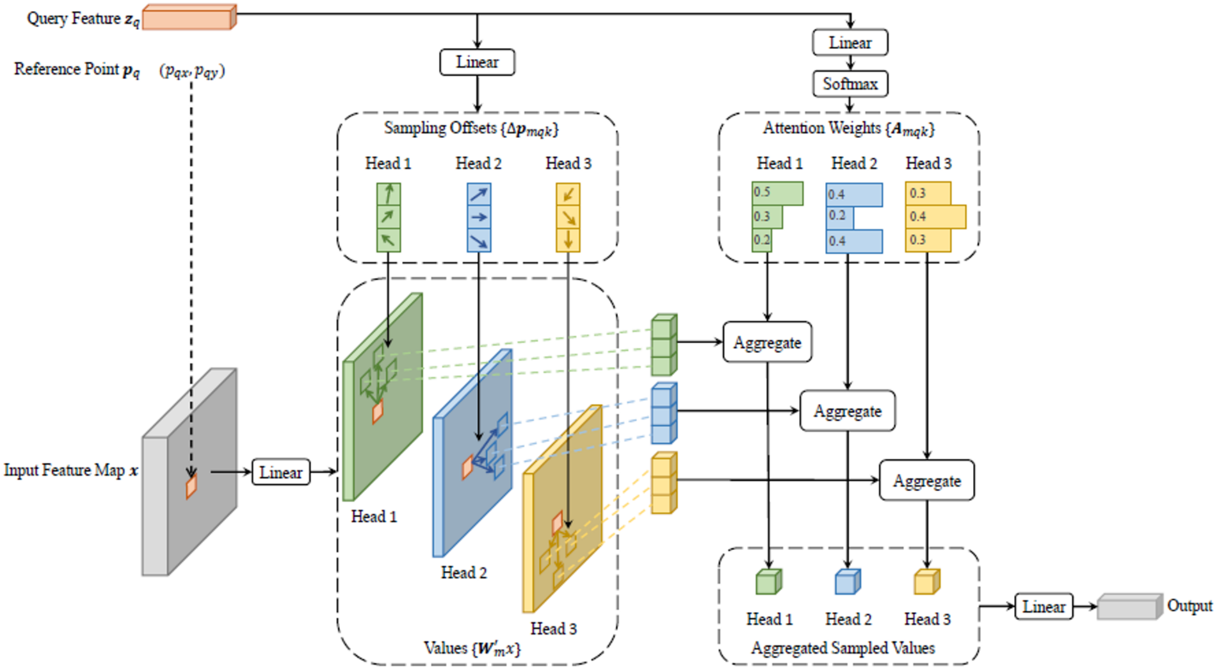
- Without region proposal
- More easy to visualize what our model is paying attention to



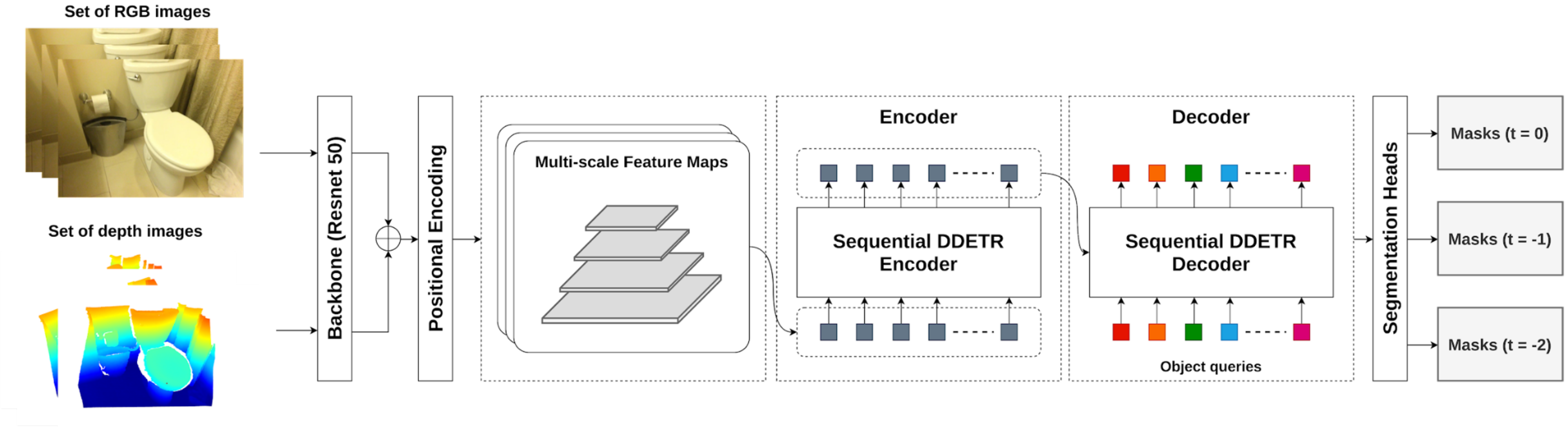
Attention Example



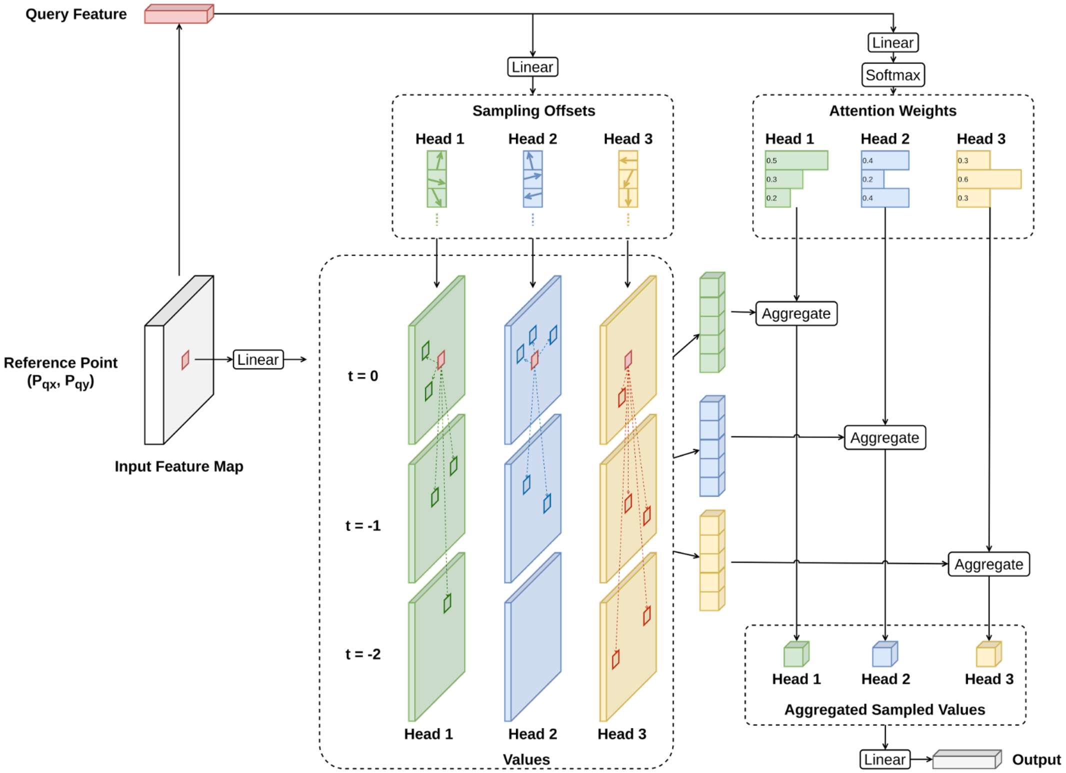
Deformable DETR



Sequential Deformable DETR

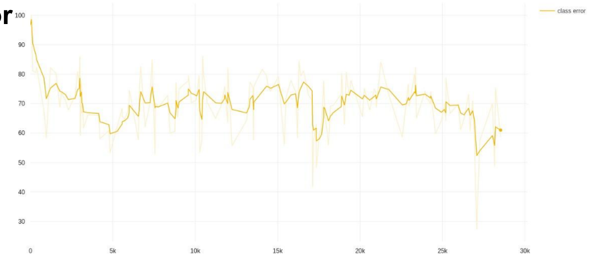


Sequential Deformable DETR

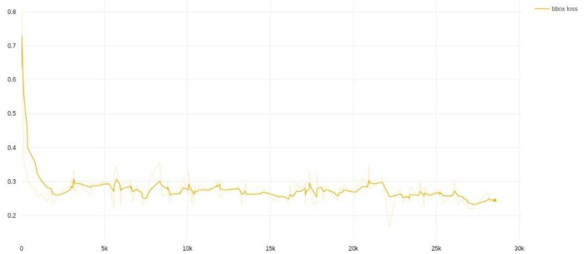


Training Result (ongoing …)

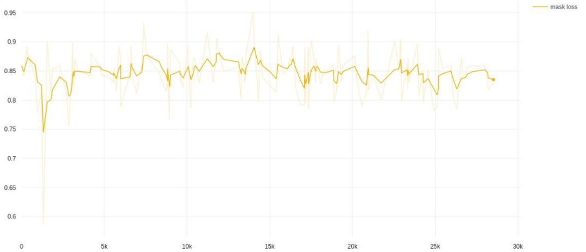
class error



bbox error

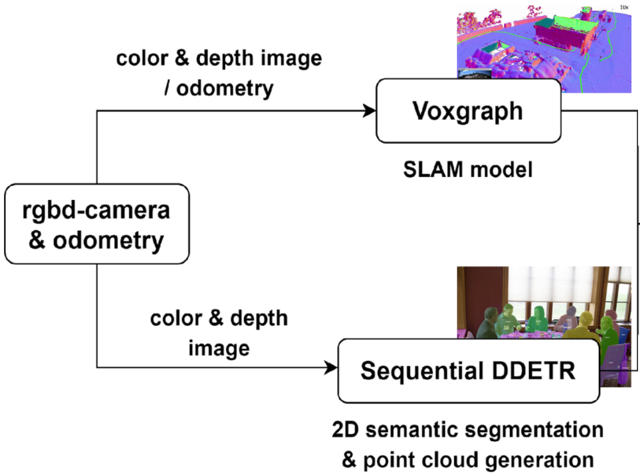


mask error



Method - Semantic SLAM with SD-DETR

- We use Voxgraph as our backbone SLAM model
- Fuse the predictions of SD-DETR to generate map with sematic meaning
- Use the Semantic SLAM to help the UAV find desired object



Reference

- Helen Oleynikova, Zachary Taylor, Marius Fehr, Juan Nieto, and Roland Siegwart, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017.
- Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Andrew Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking∗", in IEEE International Symposium on Mixed and Augmented Reality ISMAR, 2011.
- Helen Oleynikova, Michael Burri, Zachary Taylor, Juan Nieto, Roland Siegwart, and Enric Galceran, "Continuous-Time Trajectory Optimization for Online UAV Replanning". In IEEE Int. Conf. on Intelligent Robots and Systems (IROS), October 2016.
- Helen Oleynikova, Zachary Taylor, Roland Siegwart, and Juan Nieto, "Safe Local Exploration for Replanning in Cluttered Unknown Environments for Micro-Aerial Vehicles". IEEE Robotics and Automation Letters, 2018.
- Helen Oleynikova, Zachary Taylor, Roland Siegwart, and Juan Nieto, "Sparse 3D Topological Graphs for Micro-Aerial Vehicle Planning". In IEEE Int. Conf. on Intelligent Robots and Systems (IROS), October 2018.
- Helen Oleynikova, Christian Lanegger, Zachary Taylor, Michael Pantic, Alexander Millane, Roland Siegwart, Juan Nieto, "An Open-Source System for Vision-Based Micro-Aerial Vehicle Mapping, Planning, and Flight in Cluttered Environments", in arXiv, 2018.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, "Mask R-CNN", in arXiv, 2018.
- Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", in arXiv, 2018.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", in arXiv, 2017.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko, End-to-End Object Detection with Transformers, in arXiv, 2020.
- Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, Jifeng Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection", in arXiv, 2020.
- Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, Andrew J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects", in Conference on Computer Vision and Pattern Recognition (CVPR), 2013

Reference

- John McCormac, Ankur Handa, Andrew Davison, Stefan Leutenegger, “SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks”, in arXiv, 2016
- Yu Xiang, Dieter Fox, DA-RNN: Semantic Mapping with Data Associated Recurrent Neural Networks, in arXiv, 2017
- Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, Matthias Nießner, “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”, in arXiv, 2017
- Victor Reijgwart, Alexander Millane, Helen Oleynikova, Roland Siegwart, Cesar Cadena, Juan Nieto, “Voxgraph: Globally Consistent, Volumetric Mapping using Signed Distance Function Submaps”, in arXiv, 2020
- Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng, “ROS: an open-source Robot Operating System”, 2009