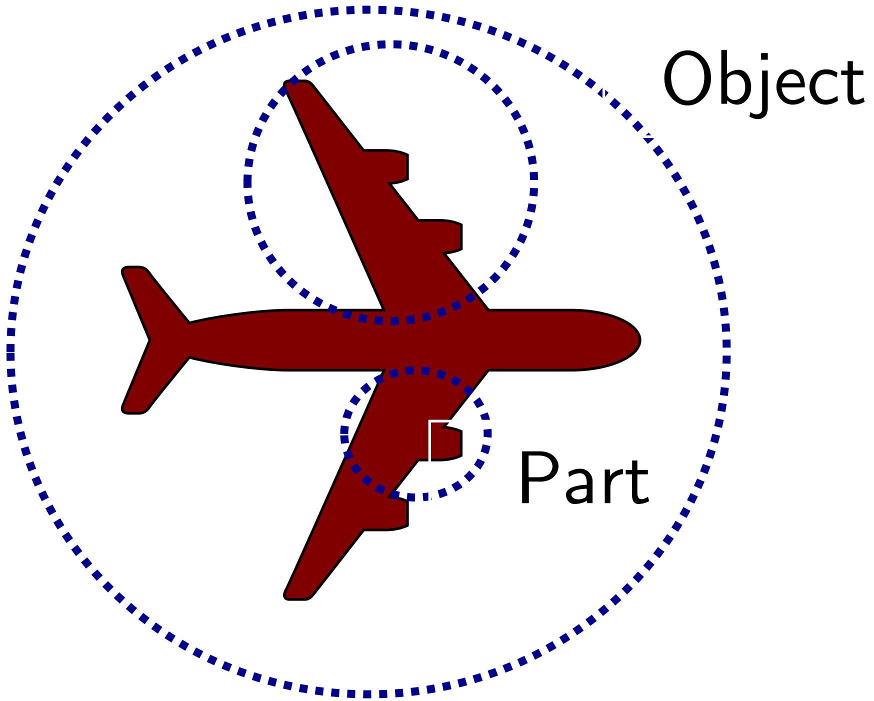
Capsules

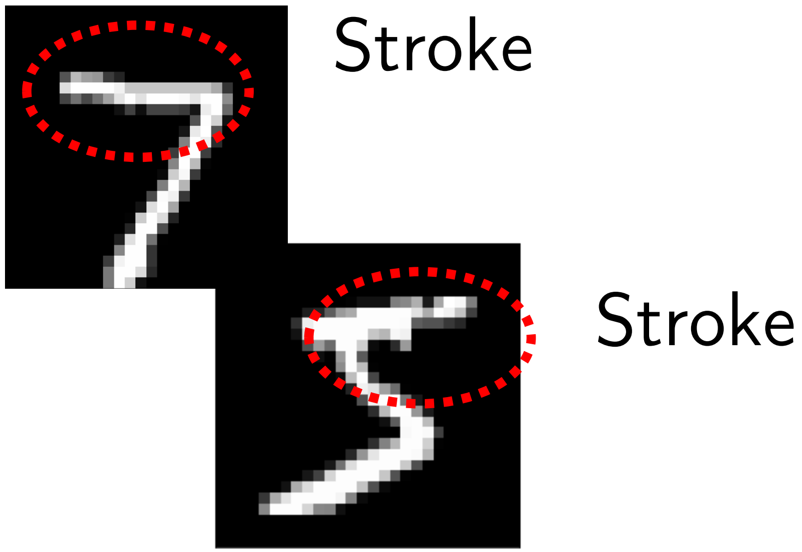
Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu

Capsules

Capsule: A group of hidden units that jointly encode one visual entity.

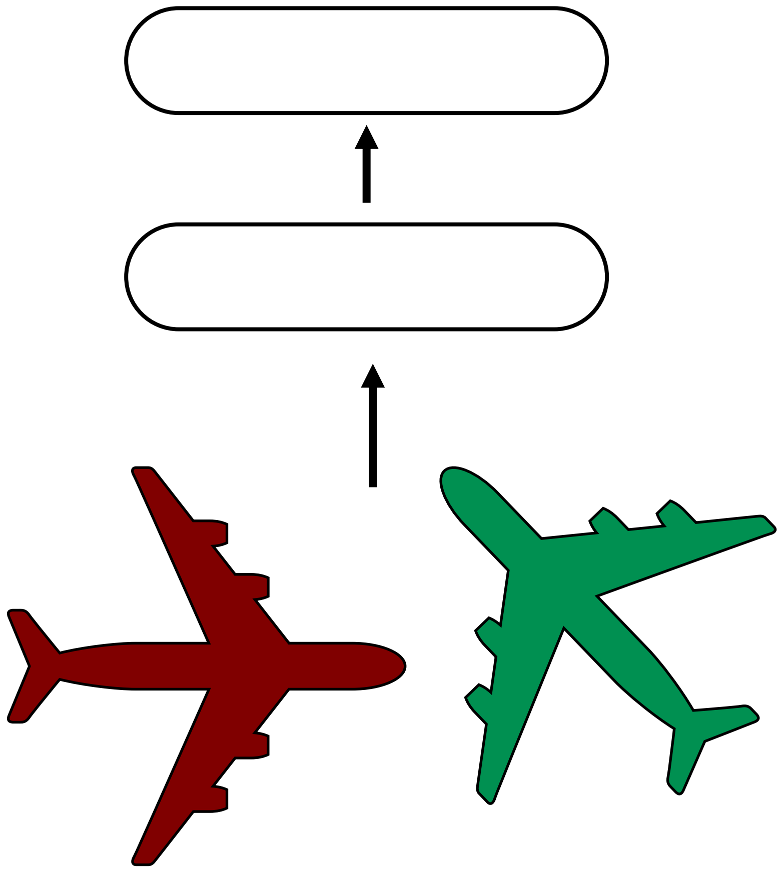




Hinton et al. ICLR 2018, Sabour et al. NeurIPS 2017

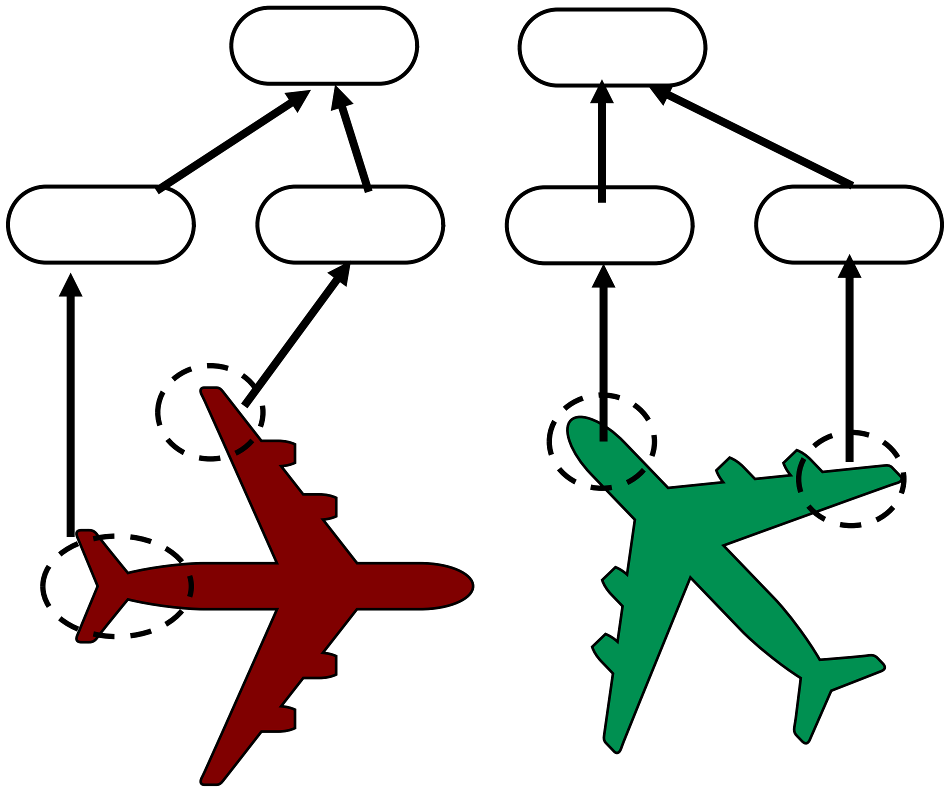
Capsules

CNN Representation



Capsule Representation

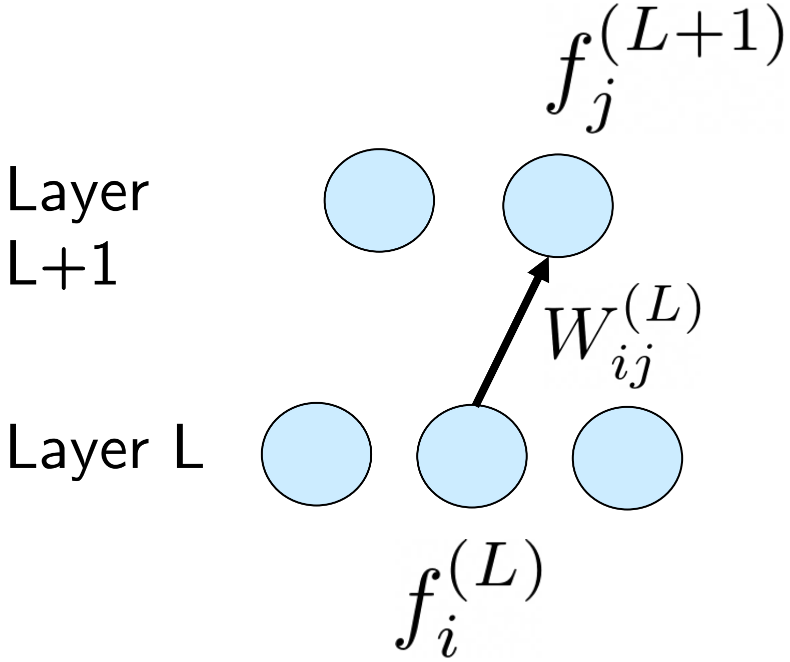
One computational entity per real-world entity



Capsules: Computing Agreement

- Transform the feature (pose) in $f_i^{(L)}$ to the vote for the features $f_j^{(L+1)}$

$$v_{ij}^{(L)} = W_{ij}^{(L)} f_i^{(L)}$$



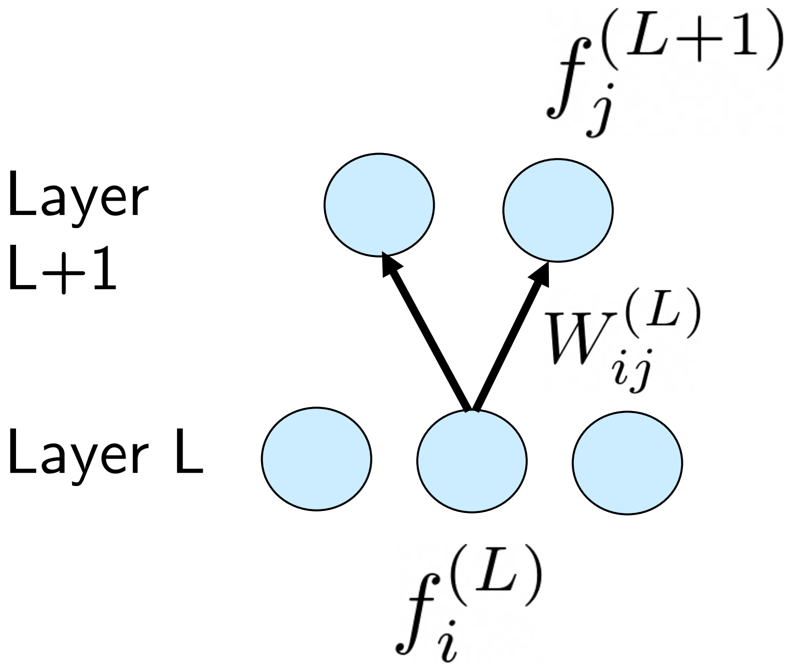
- Compute the agreement:

$$\alpha_{ij}^{(L)} = f_j^{(L+1)\top} v_{ij}^{(L)}$$

- Bi-linear relationship between capsules

Capsules: Routing

- Determine routing probabilities: $$r_{ij}^{(L)} = \frac{\exp\left(\alpha_{ij}^{(L)}\right)}{\sum_j \exp\left(\alpha_{ij}^{(L)}\right)}$$



- Inverted Attention: how high-level capsules compete with for attention of low-level capsules
- Normalization over j.
- Opposite to attention used in Transformer.

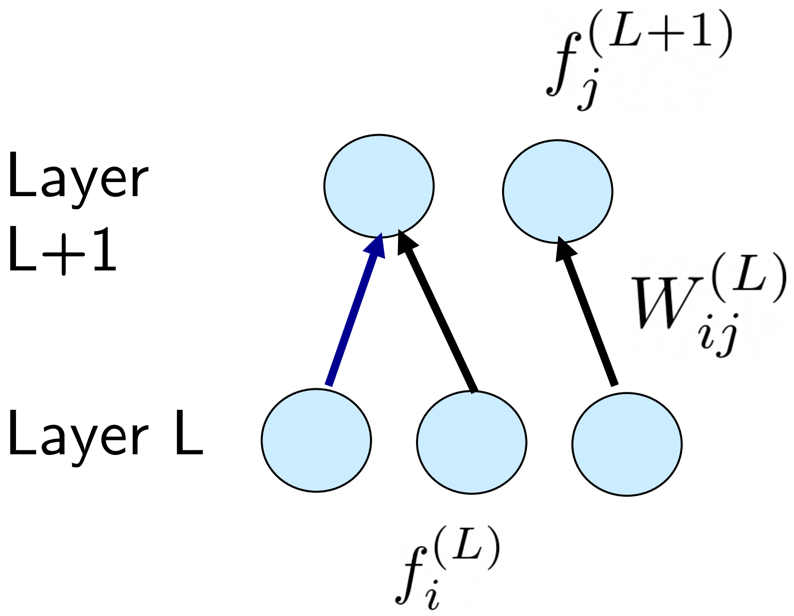
$$v_{ij}^{(L)} = W_{ij}^{(L)} f_i^{(L)}$$
$$\alpha_{ij}^{(L)} = f_j^{(L+1)\top} v_{ij}^{(L)}$$

Tsai, Srivastava, Goh, Salakhutdinov ICLR 2020

Capsules: Updates

- Update layer L+1 capsule $f_j^{(L+1)}$

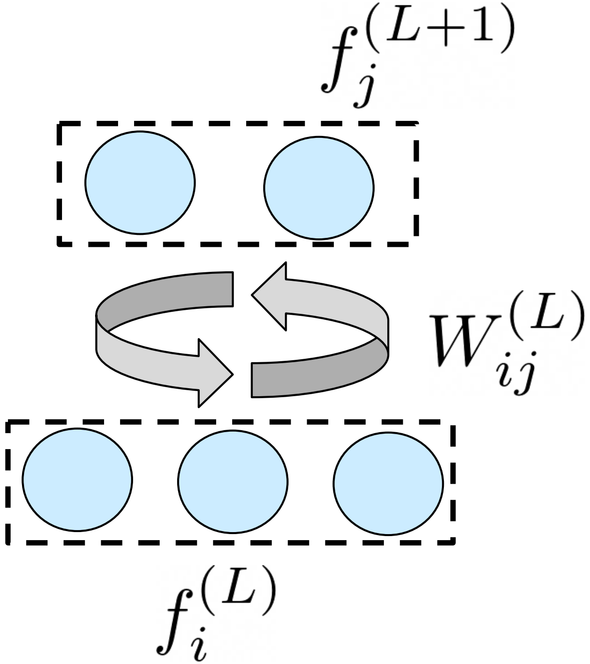
$$f_j^{(L+1)} = \sum_i r_{ij}^{(L)} v_{ij}^{(L)} = \sum_i r_{ij}^{(L)} W_{ij}^{(L)} f_i^{(L)}$$



- Note that this is a Linear Aggregation.
- Agreement depends on features (poses) at both layers → Iterative Updates.

Inverted Dot-Product Attention Routing Algorithm

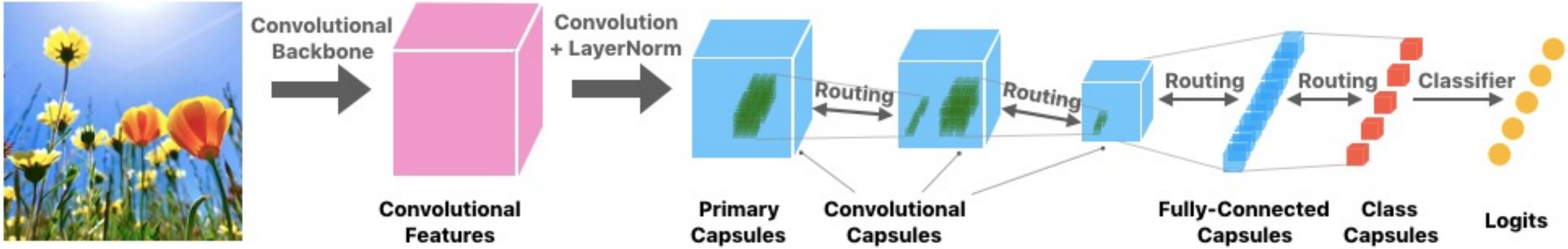
- Vote: $v_{ij}^{(L)} = W_{ij}^{(L)} f_i^{(L)}$
- Agreement: $\alpha_{ij}^{(L)} = f_j^{(L+1)\top} v_{ij}^{(L)}$
- Routing Probabilities: $r_{ij}^{(L)} = \frac{\exp\left(\alpha_{ij}^{(L)}\right)}{\sum_j \exp\left(\alpha_{ij}^{(L)}\right)}$
- Update: $f_j^{(L+1)} = \sum_i r_{ij}^{(L)} v_{ij}^{(L)}$
- Repeat



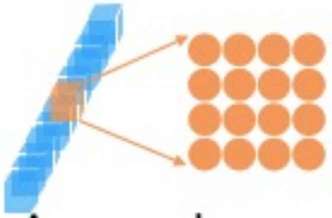
Tsai, Srivastava, Goh, Salakhutdinov ICLR 2020

Capsule Net

A capsule network = a backbone CNN block + convolutional capsule layers + fully-connected capsule layers

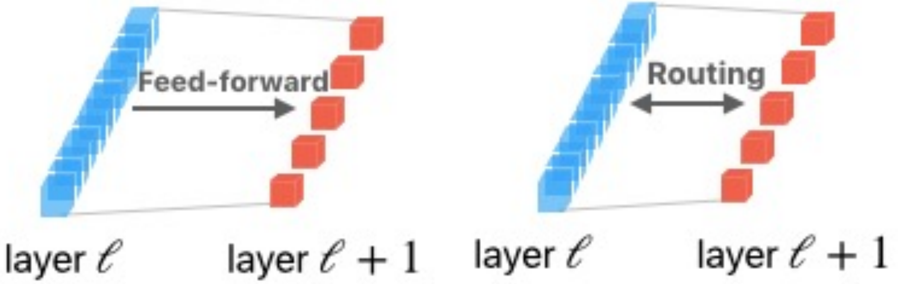


A capsule layer
= group of capsules



A capsule
= group of neurons (pose)

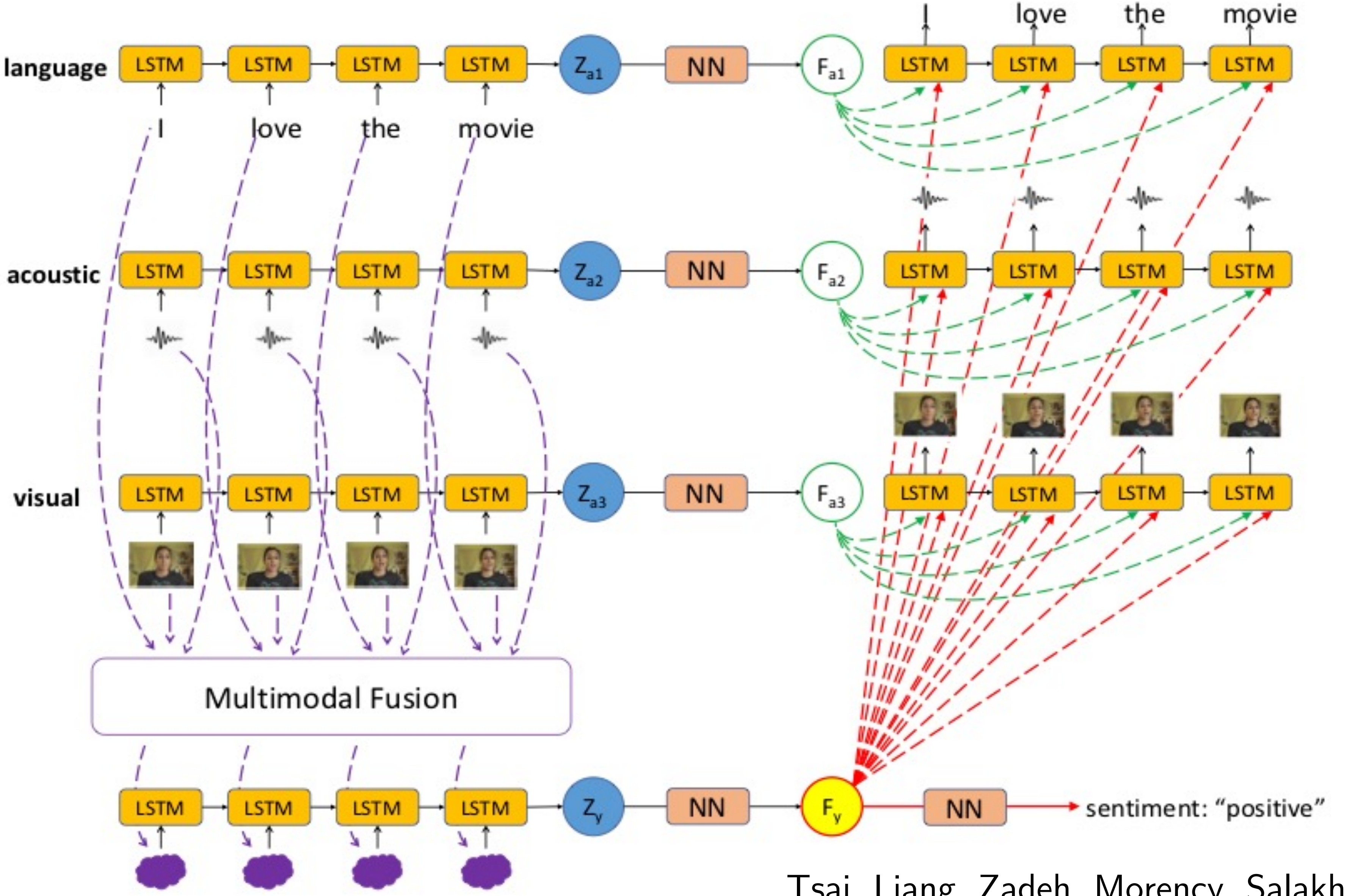
Traditional v.s. CapsNet Inference



Routing

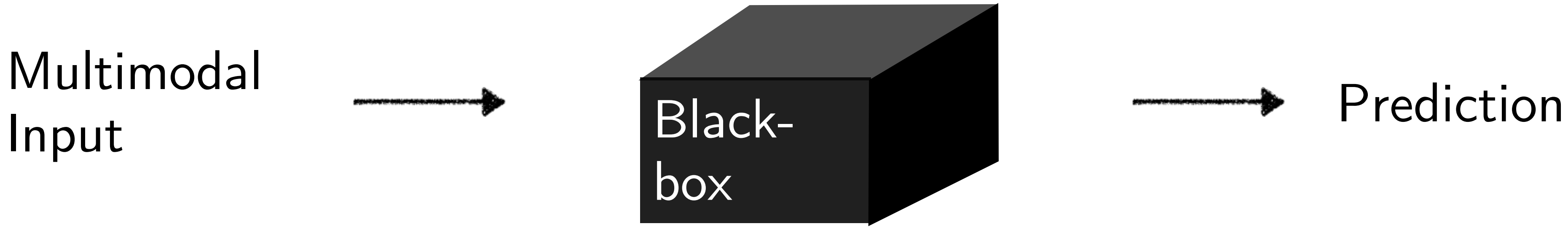
Multimodal Language

- Human Language is naturally multimodal.
- It contains textual (e.g. spoken/written words), visual (e.g. body gestures) and acoustic (e.g. voice tones) modalities.
- It is important to understand both single modality and interactions between modalities in modeling multimodal language.



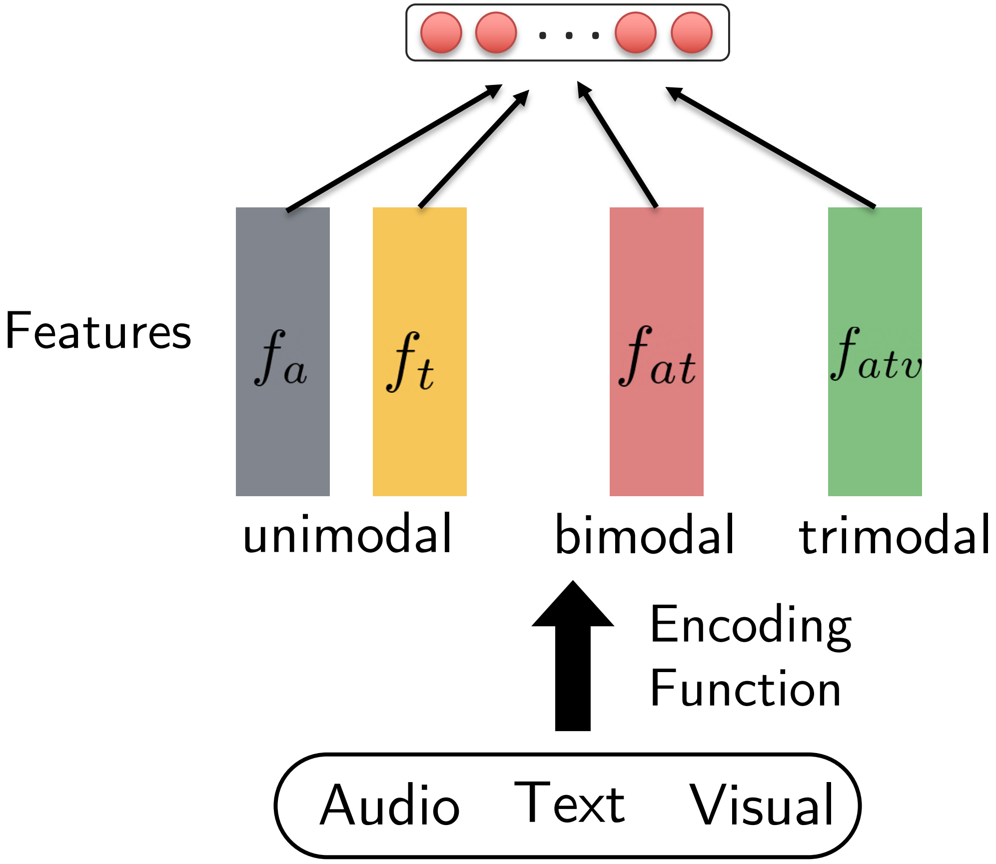
Tsai, Liang, Zadeh, Morency, Salakhutdinov, ICLR 2019

Interpretability



- Interpretability allows to identify crucial explanatory features for model prediction.
- Provide further insight into multimodal learning, improving model design or dataset debugging.

Multimodal Models

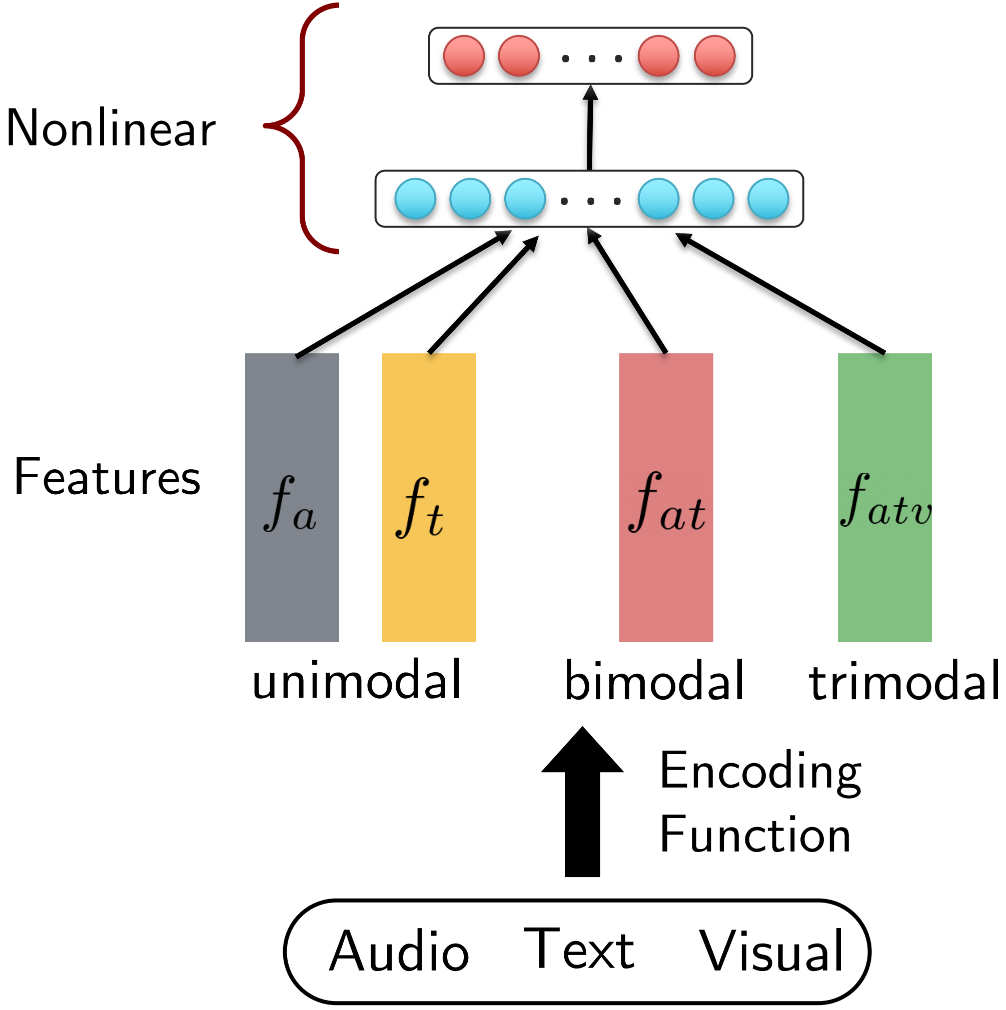


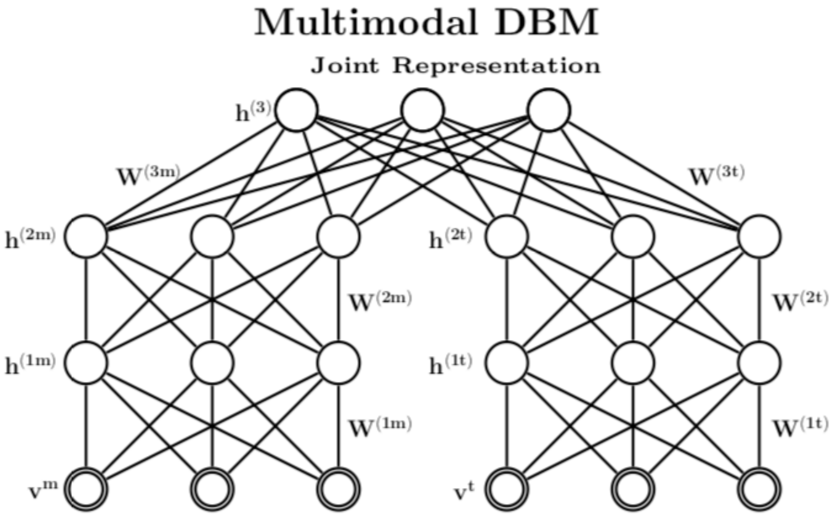
- We can use a linear prediction:

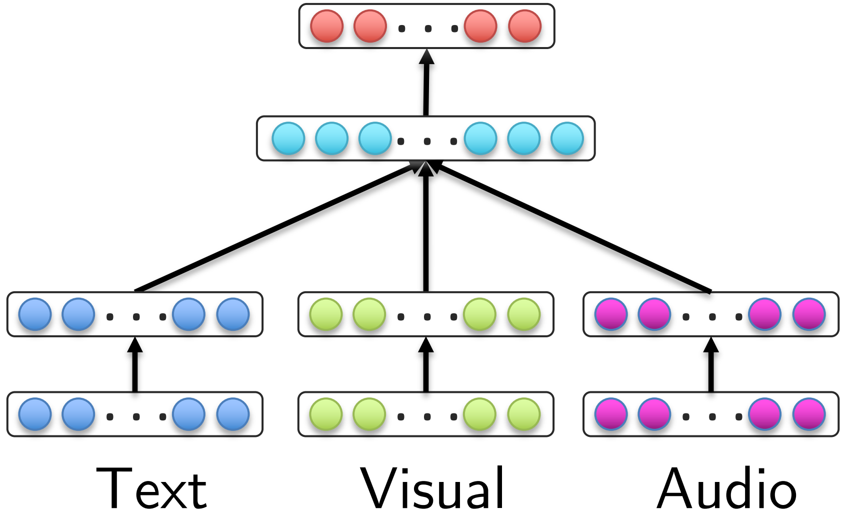
$$\begin{aligned}\hat{y} \quad = \quad & \beta_1 f_a + \beta_2 f_t + \beta_3 f_v + \\ & \beta_{12} f_{at} + \beta_{13} f_{av} + \beta_{23} f_{tv} + \\ & \beta_{123} f_{atv}\end{aligned}$$

- Betas can provide global interpretability: the general insight of the importance of explanatory features over the whole dataset
- Local interpretability: the high-resolution insight of feature importance specifically depending on each *individual* sample during training/inference

Multimodal Models: In Practice



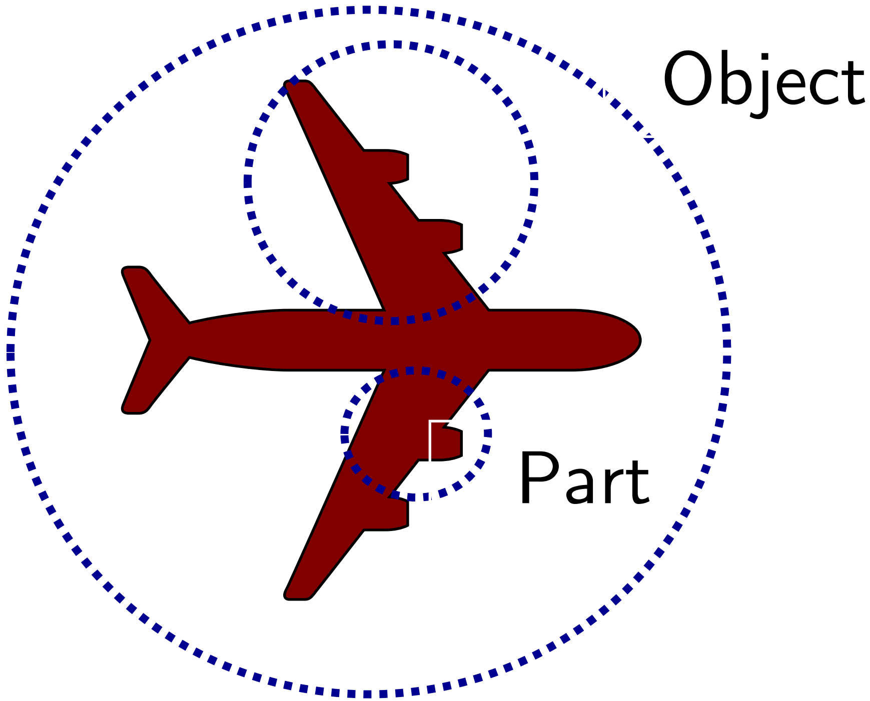


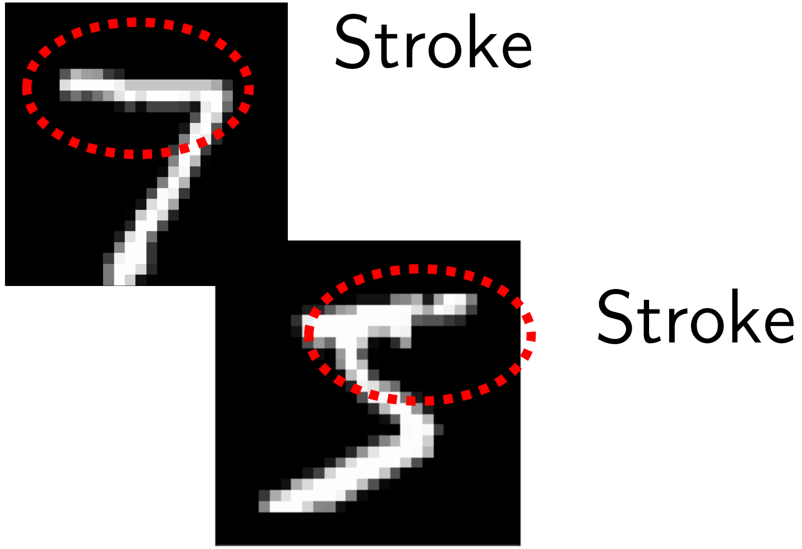


Snoek et al. 2005, Srivastava et al 2014, Tsai et. al ICLR, 2019

Capsules

Capsule: A group of hidden units that jointly encode one visual entity.

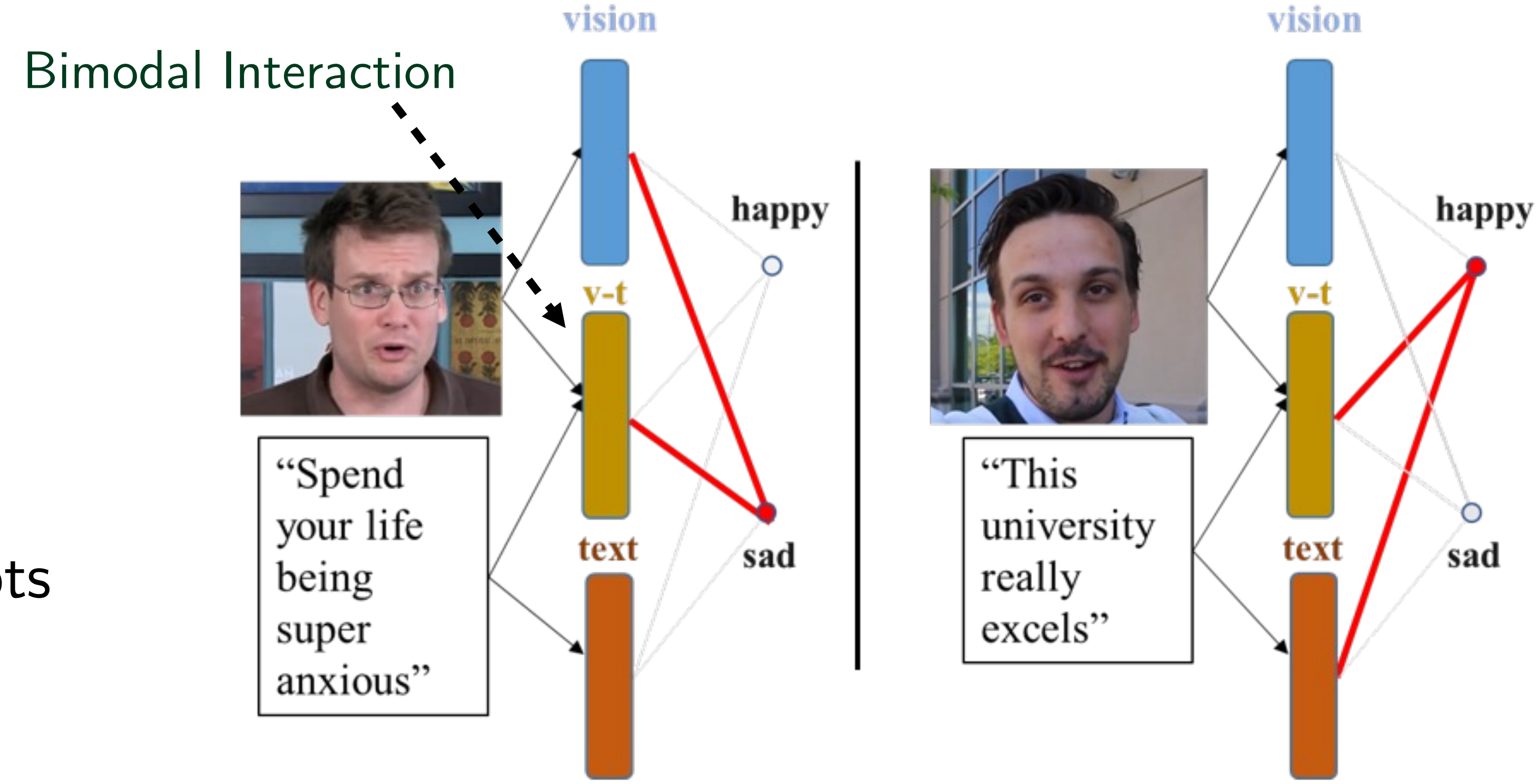




Hinton et al. ICLR 2018, Sabour et al. NeurIPS 2017

Multimodal Routing

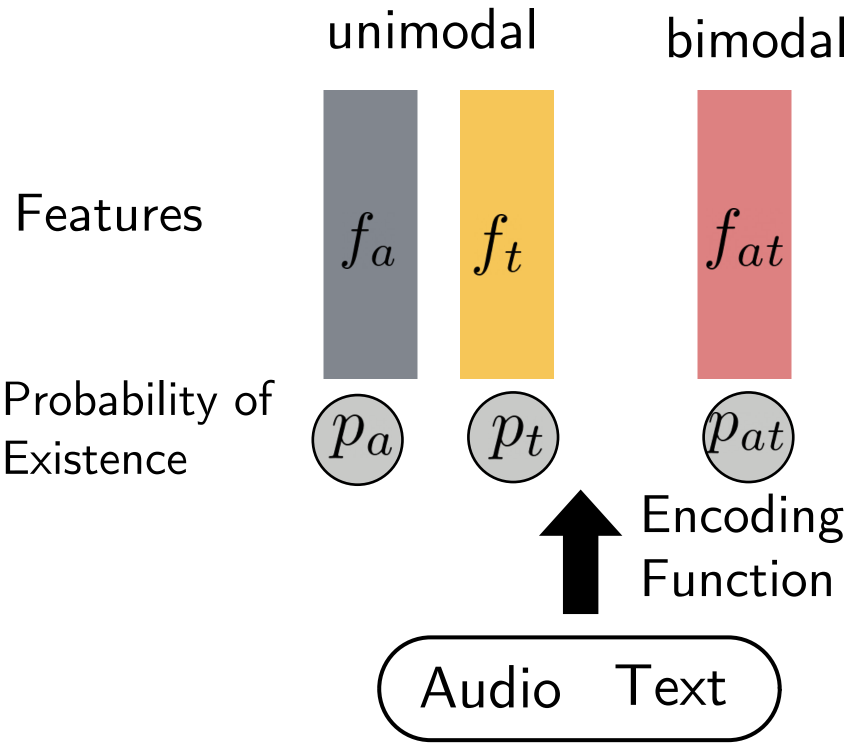
- Dynamically adjust local weights of unimodal/multimodal features
- Iteratively update concepts and routing coefficients
- Use the updated concepts for prediction

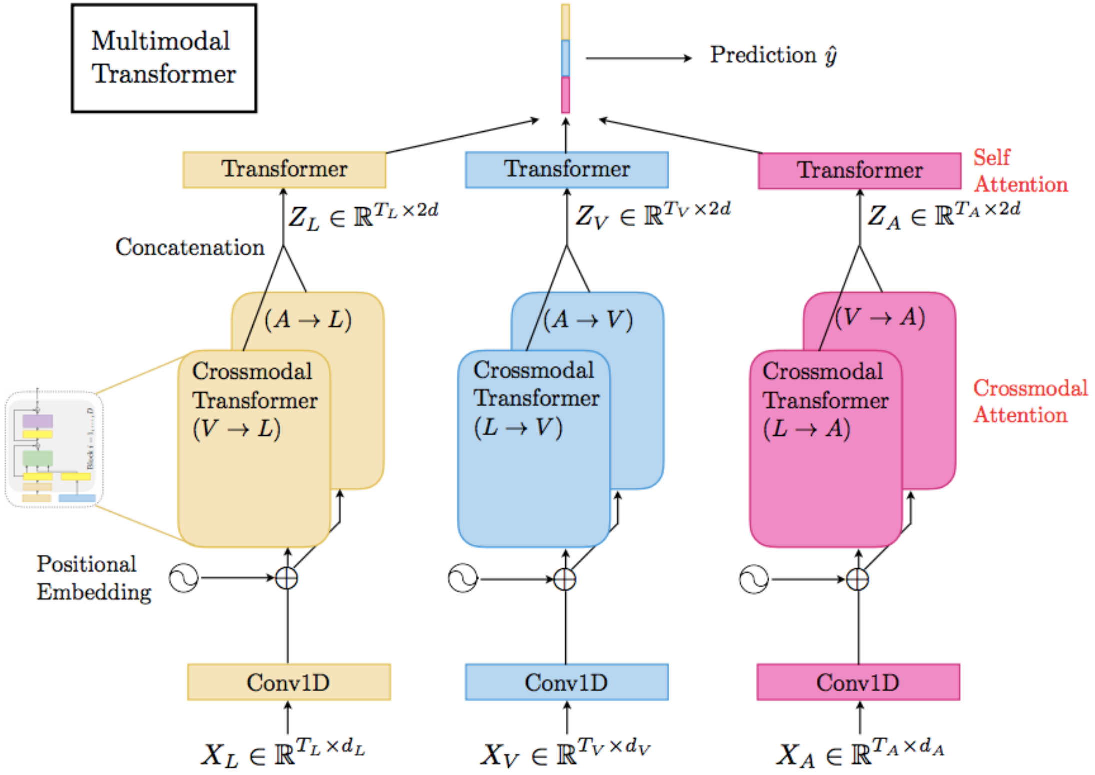


Dynamic Weight Assignment

Input Representation

- For example, let x_a, x_t represent raw audio/textual features
- Trough encoding, we obtain feature vectors f_a, f_t , and bimondal f_{at}





Tsai, Ba, Liang, Kolter, Morency, Salakhutdinov, ACL2019

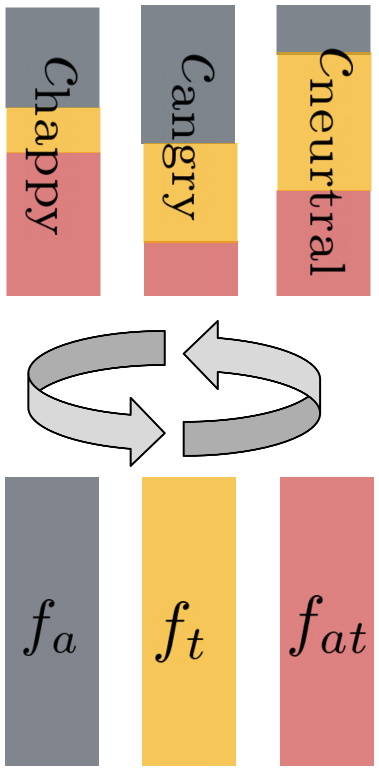
Model

- Concepts: 1-d vectors, where $c_j \in \mathbb{R}^{d_c}$ representing the concept for the jth class

- Features → Parts, Concepts → Objects

Concepts

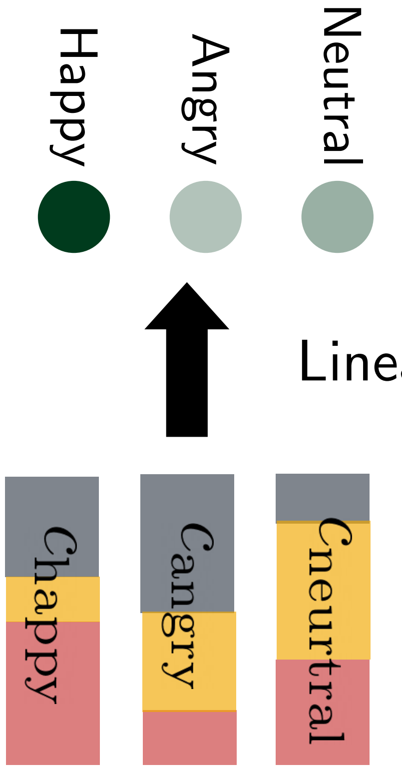
Features



(Prediction)

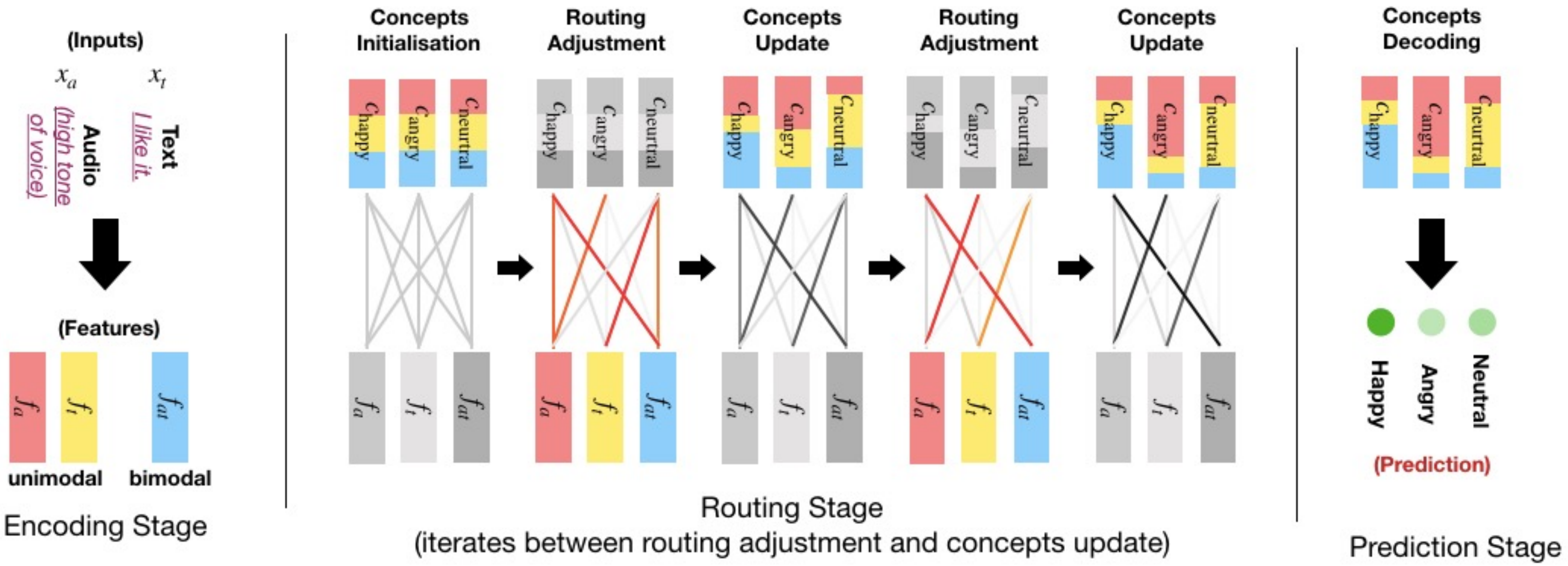
Linear transform

Concept Decoding



Tsai, Ma, Yang, Salakhutdinov, Morency, 2020

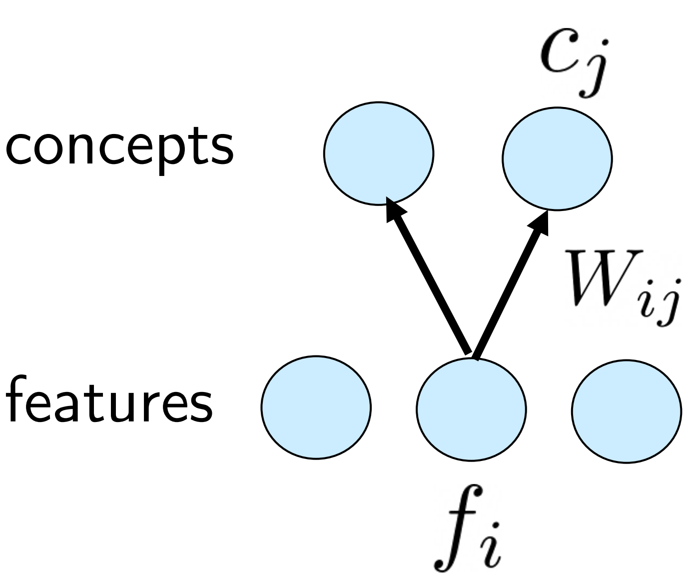
Model



Tsai, Ma, Yang, Salakhutdinov, Morency, 2020

Dynamic Routing

- Agreement: bilinear model: $\alpha_{ij} = c_j^\top W_{ij} f_i$



- Routing coefficients:

$$r_{ij} = \frac{\exp(\alpha_{ij})}{\sum_j \exp(\alpha_{ij})}$$

- Concept update:

$$c_j = \sum_i p_i r_{ij} (W_{ij} f_i)$$

Dynamic Routing

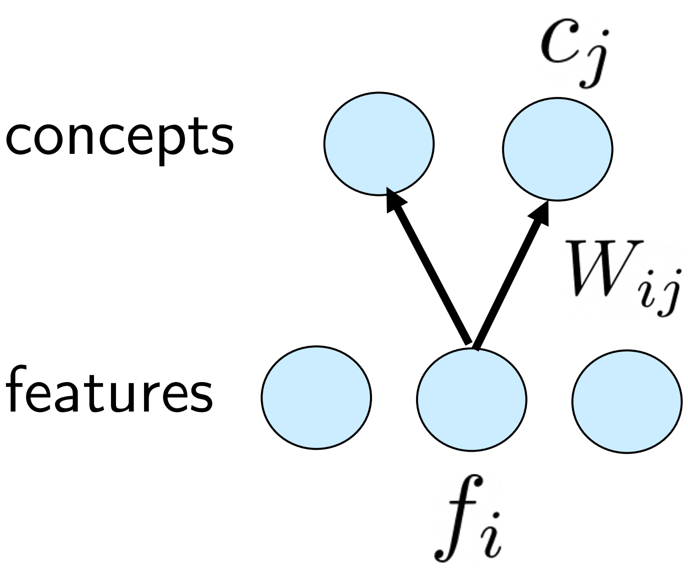
- Concept Update

$$c_j = \sum_i p_i r_{ij} (W_{ij} f_i)$$

Probability of existence of feature f_i.

Probability of routing feature i to concept j.

Linear transform of feature i into concept space

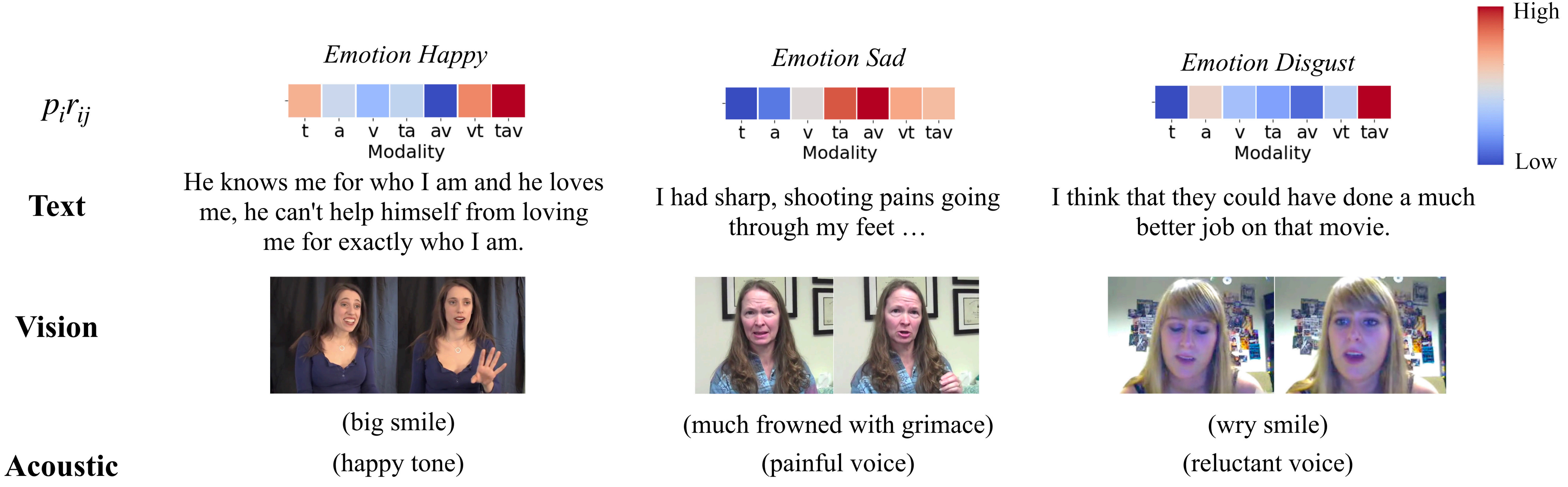


- Prediction:

$$\begin{aligned} \text{logit}_j &= \beta_j^\top c_j \\ &= \sum_i p_i r_{ij} \beta_j^\top W_{ij} f_i \end{aligned}$$

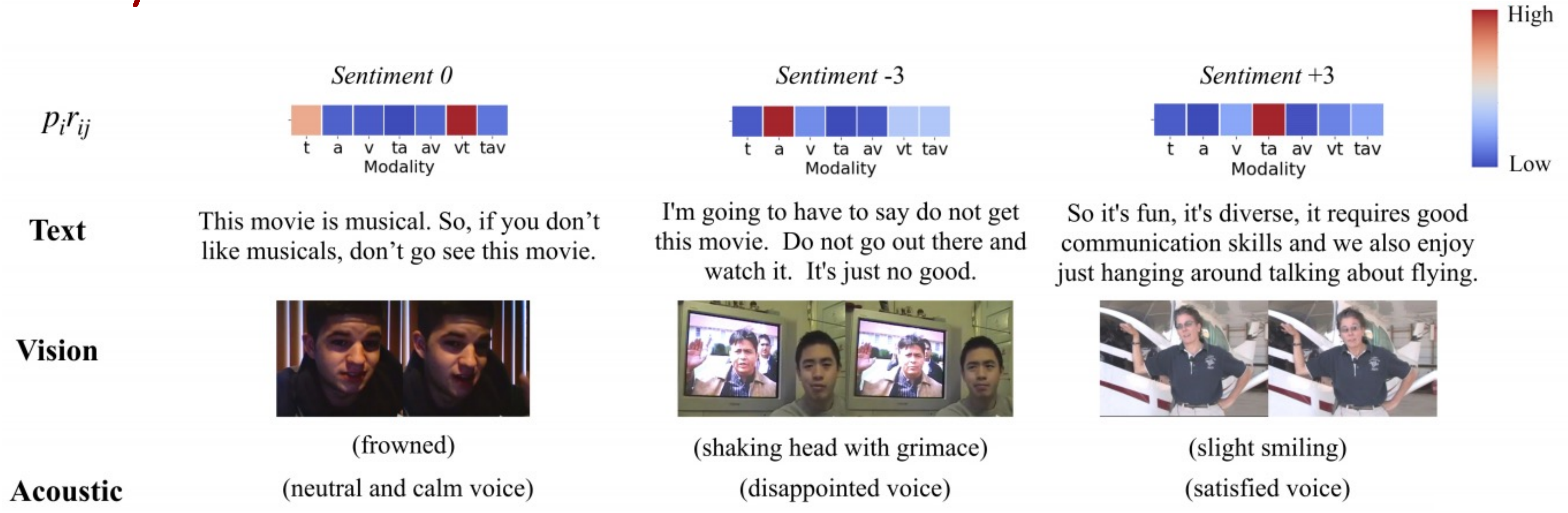
- Softmax/Sigmoid is then applied on the logits to obtain the final prediction

Analysis



Red to Blue: Most High to Most Low importance features

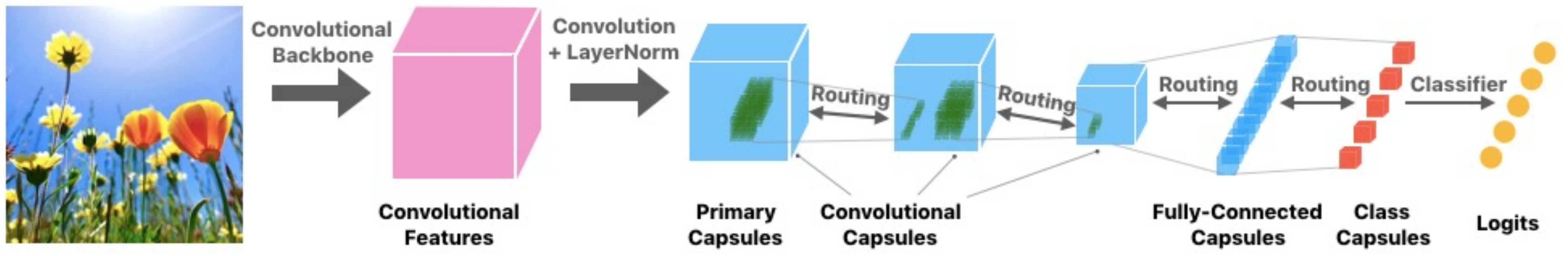
Analysis

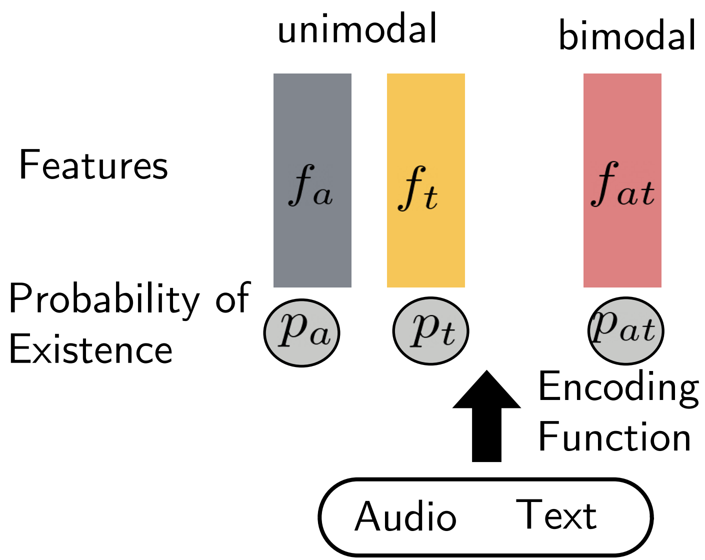


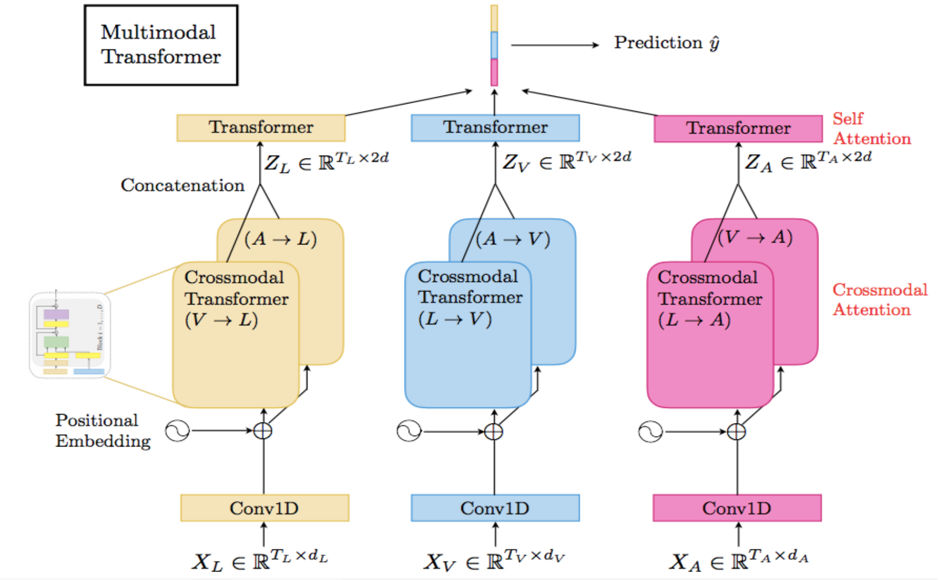
Red to Blue: Most High to Most Low importance features

Applications in Vision

- We already start with nonlinear representations:

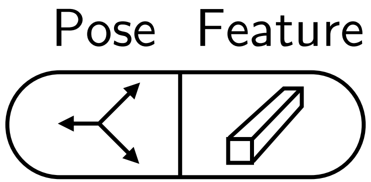




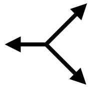


Geometric Capsules

- Each capsule contains pose and feature



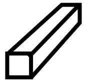
$$c = (c_q, c_f)$$



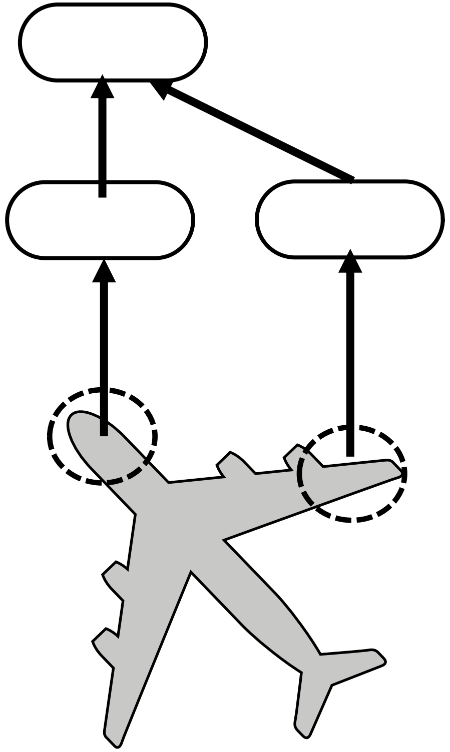
Translation | Rotation

$t \in \mathbb{R}^3$ | $q \in \mathbb{R}^4$

$$||q|| = 1$$

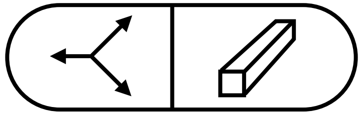


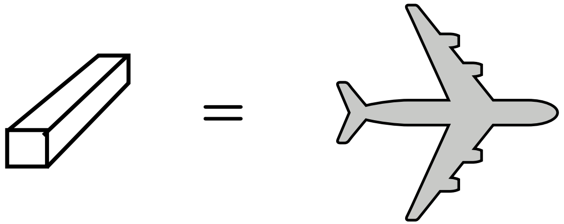
$f \in \mathbb{R}^D$

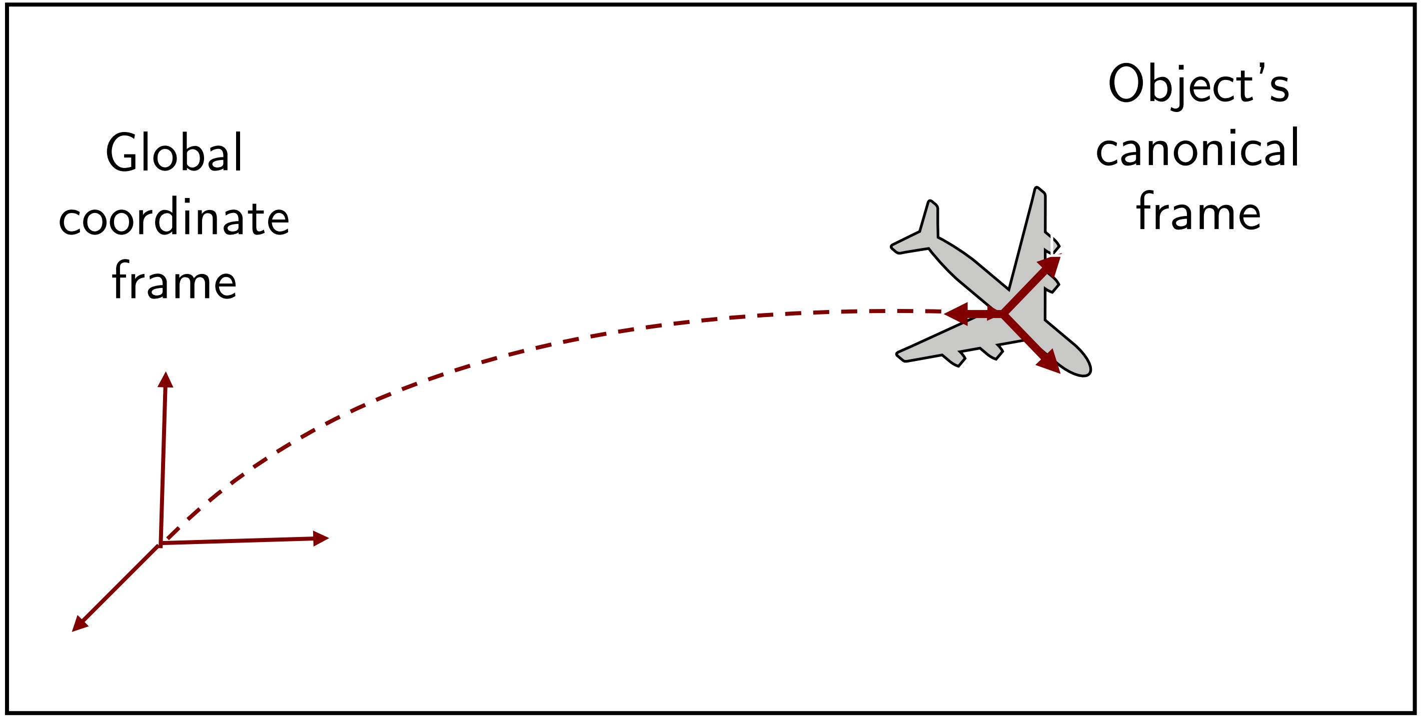


Srivastava, Goh, Salakhutdinov, 2020

Geometric Capsules

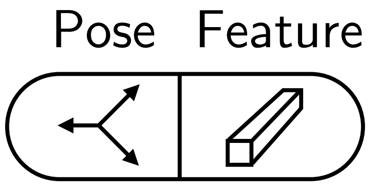




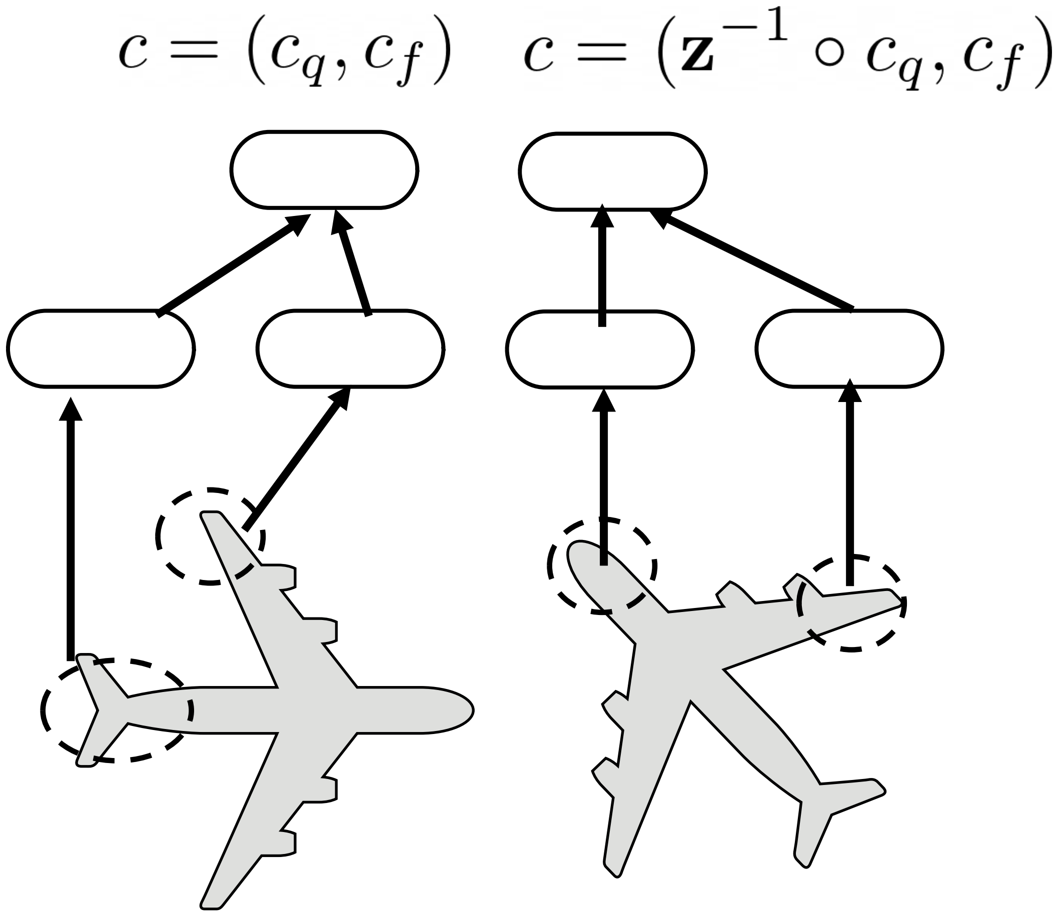


Geometric Capsules

- Each capsule can be viewed from any viewpoint z:



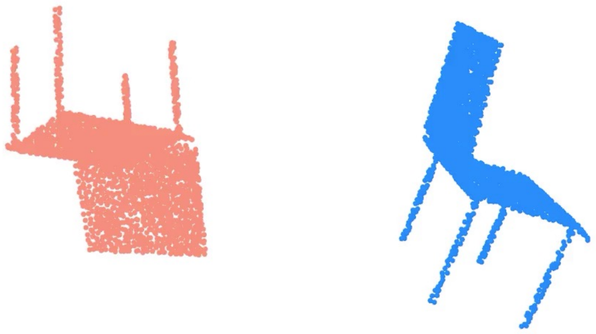
$c = (c_q, c_f)$



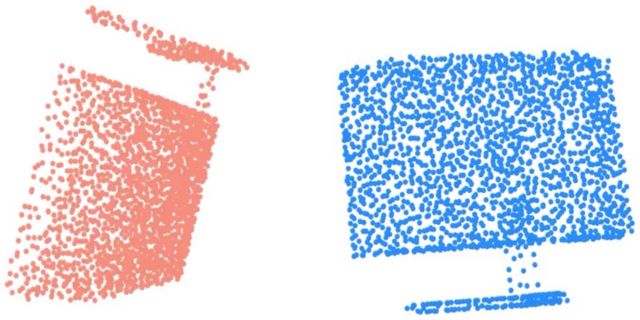
- Feature c_f is pose invariant

Pose Equivariance Results

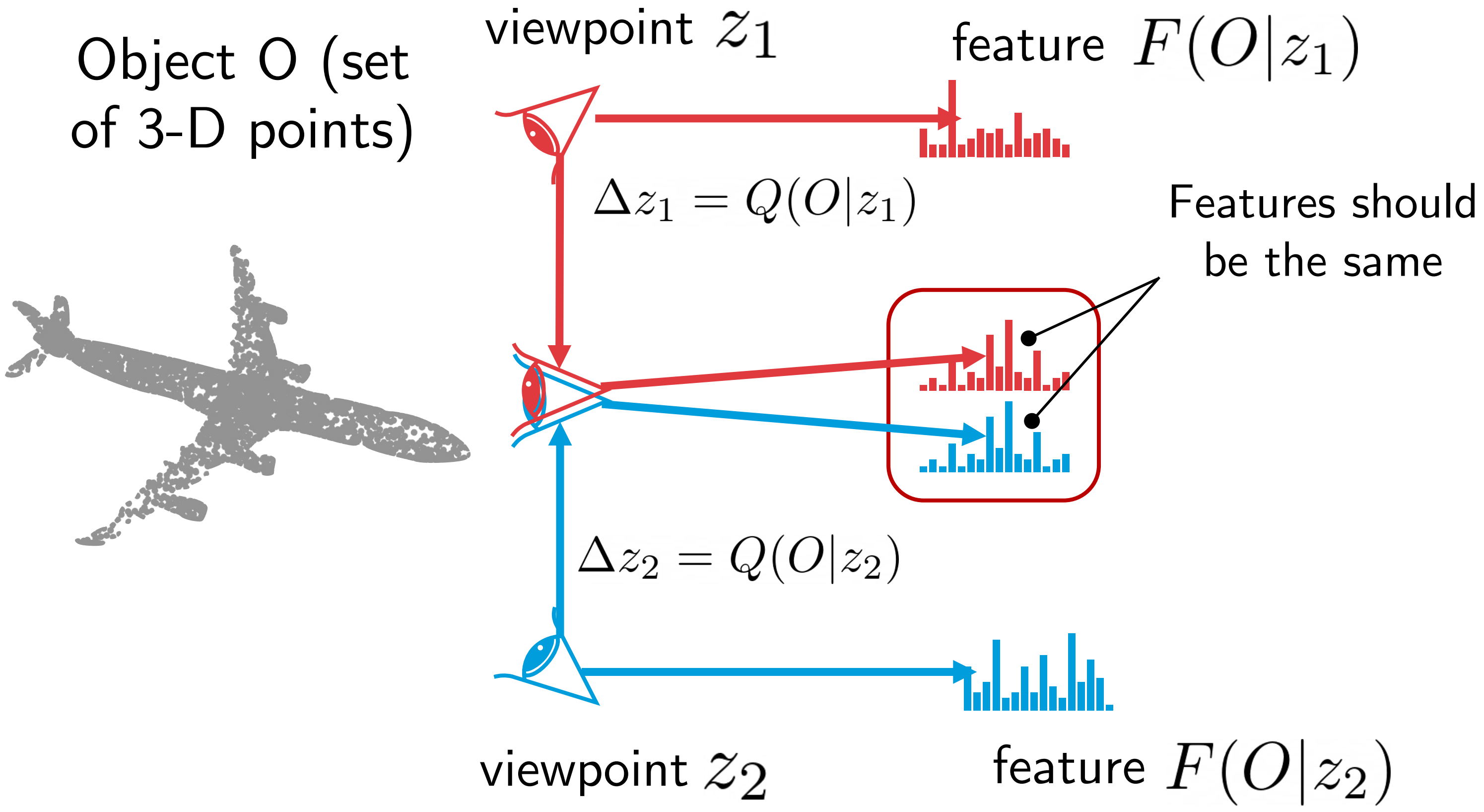
Input | Object as viewed from inferred pose



Input | Object as viewed from inferred pose

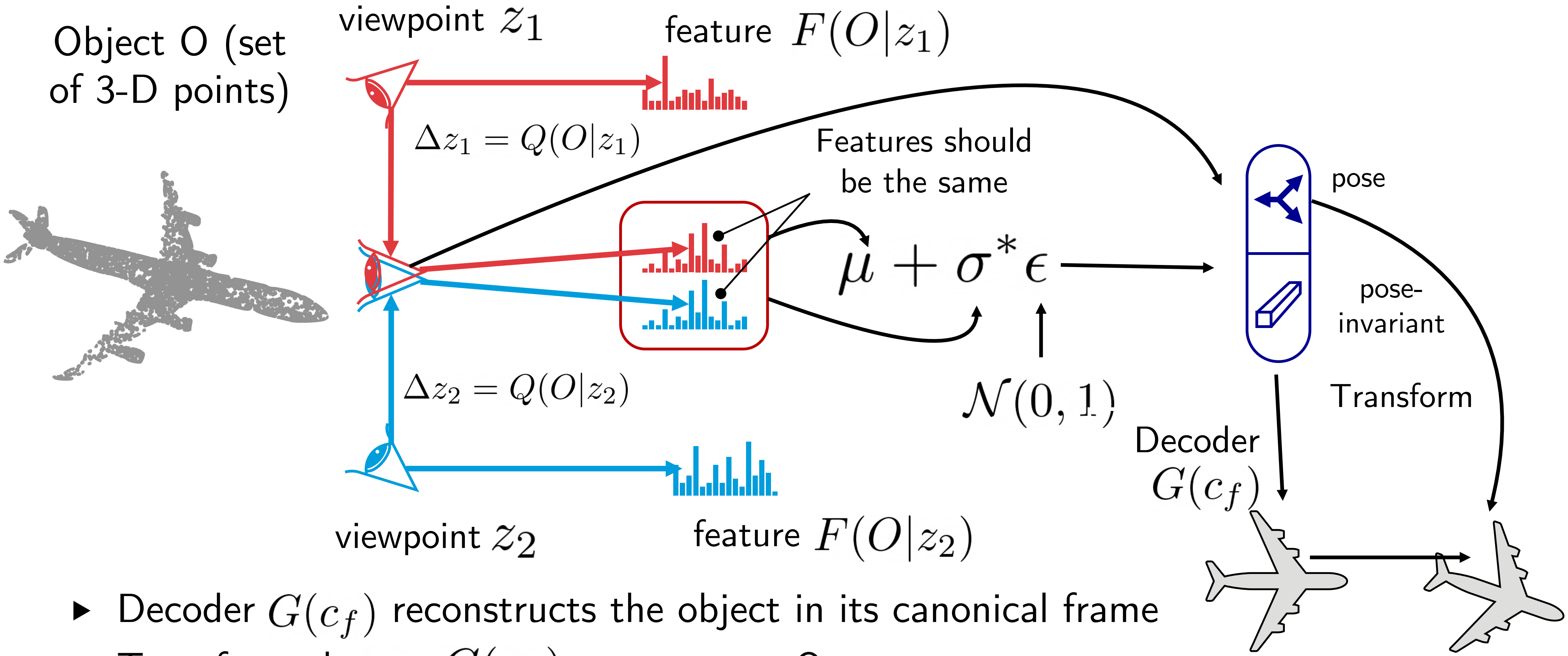


Multi-View Agreement: Unsupervised Learning



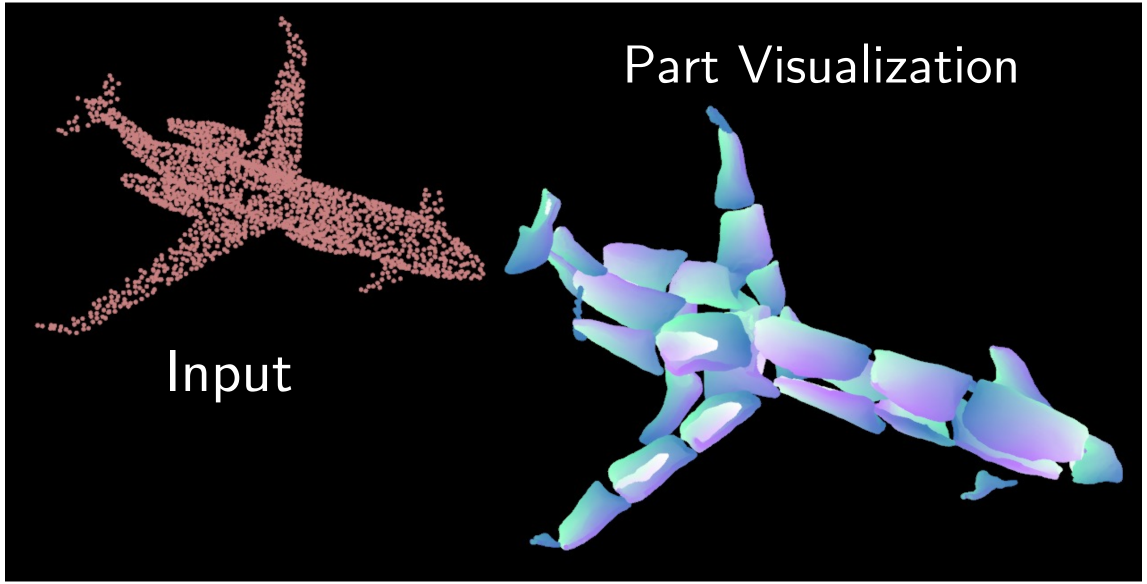
- $F(O|z_k)$: parameterized set-to-value function
- $\Delta z_k = Q(O|z_k)$ such that . $z_k \circ \Delta z_k$ is a canonical pose of the object.
- $F(O|z_k \circ \Delta z_k)$ will be the same for all k.

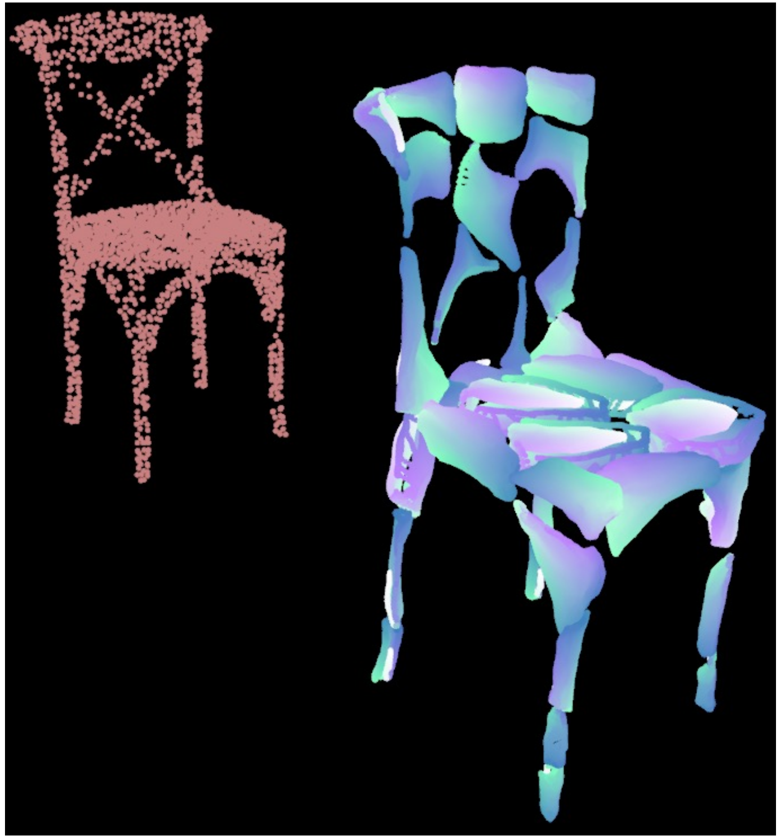
Multi-View Agreement



- Decoder $G(c_f)$ reconstructs the object in its canonical frame
- Transformed: $c_q \circ G(c_f)$ reconstructs O.
- $F(O|z), Q(O|z), G(c_f)$ can be learned jointly.

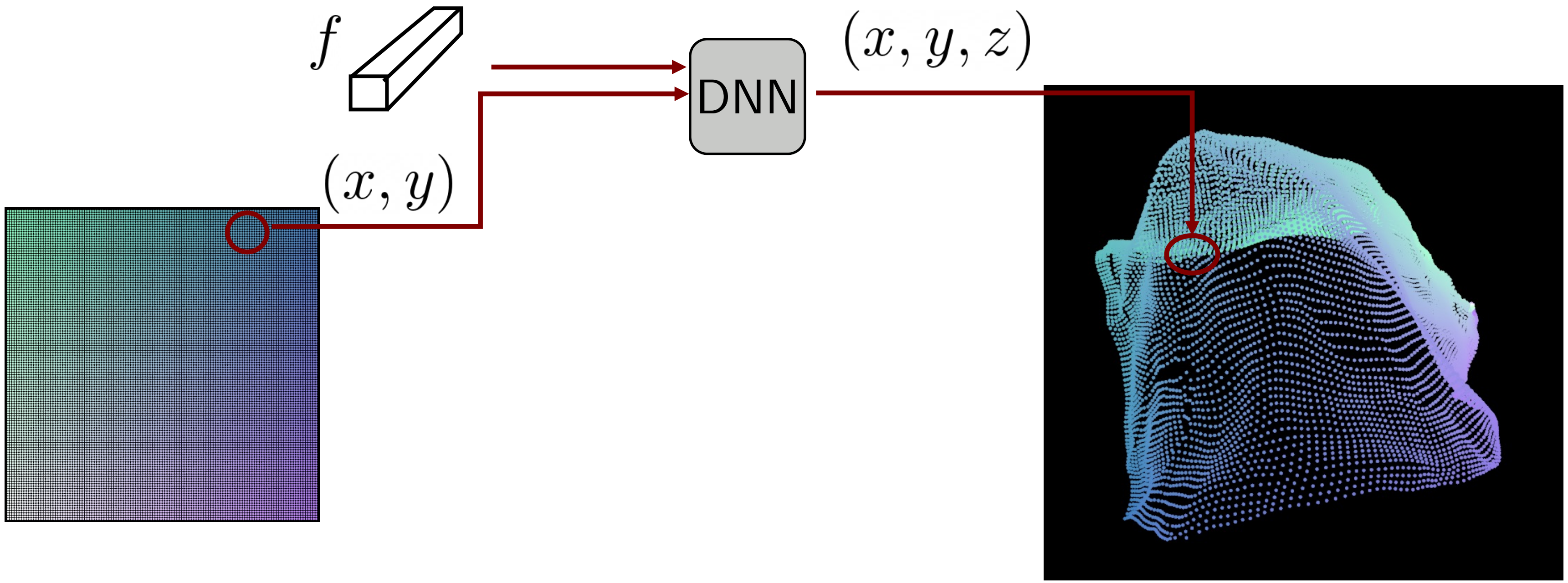
Points to Parts



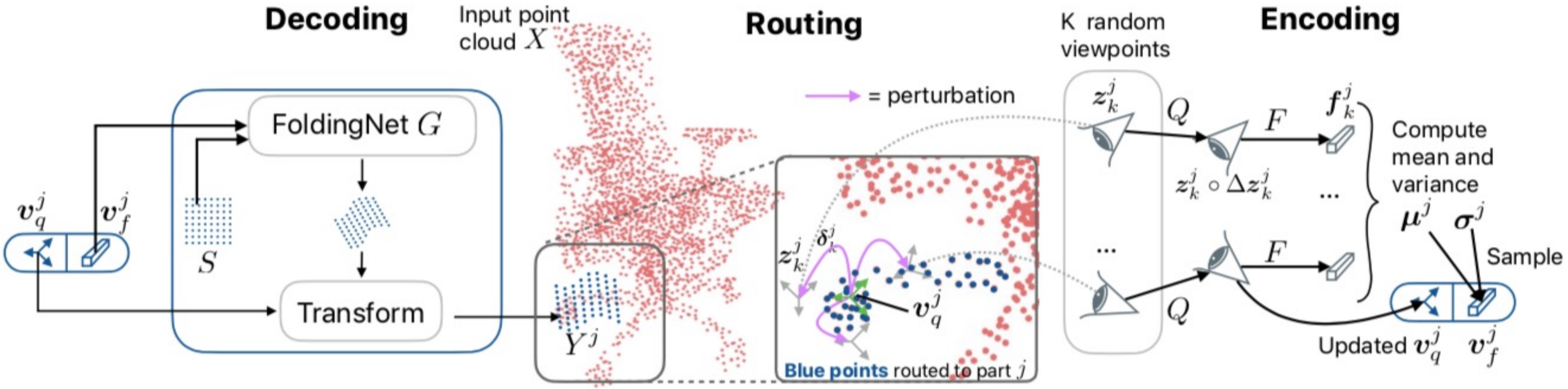


Representing Parts using Folding Net

- FoldingNet (Yang et al., CVPR 2018) is a way to parametrize folded surfaces.

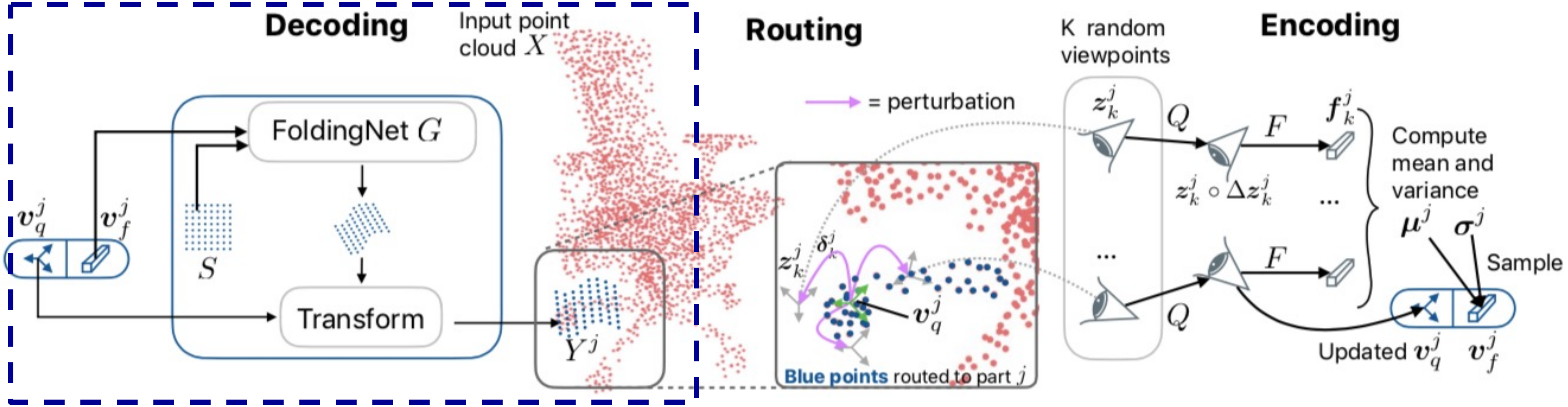


Points to Parts Autoencoder



- Let $X = \{\mathbf{x}^i\}_{i=1}^{I}, \mathbf{x}^i \in \mathbb{R}^3$ be the set of 3-D points
- Let $V = \{(\mathbf{v}_q^j, \mathbf{v}_f^j)\}_{j=1}^{J}$ be the set of J part capsules
- Let $R_{ij} \in [0, 1]$ probability of point i belonging to part capsule j
- Iteratively update V and R.

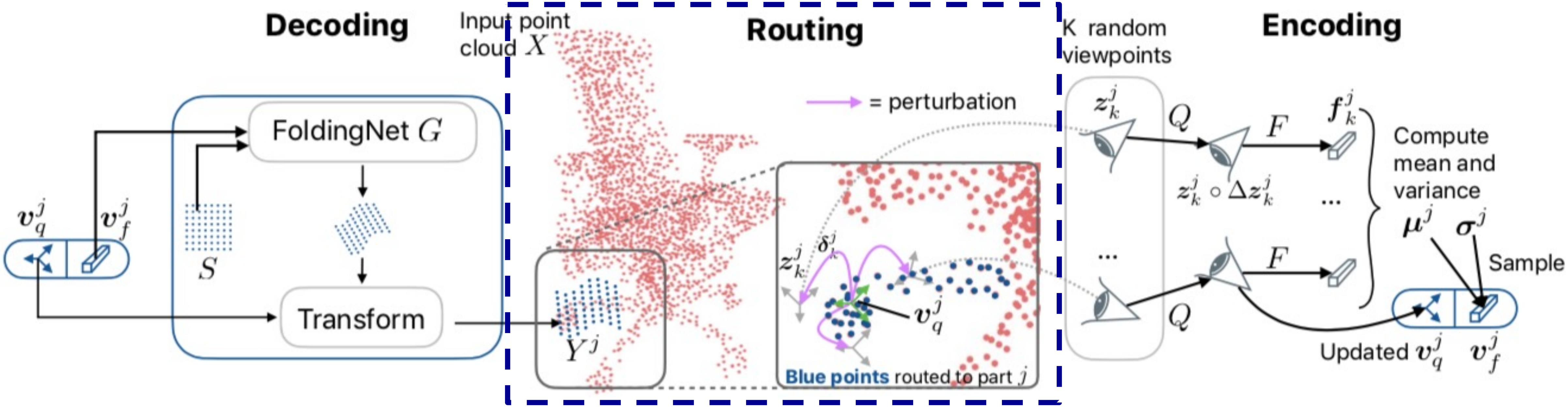
Points to Parts Decoder



- Decoder: $G : (\mathbb{R}^D \times \mathbb{R}^2) \rightarrow \mathbb{R}^3$ maps capsule's feature $\mathbf{v}_f^j$ concatenated with a 2D point sampled from a unit square to a 3D point
- The pose $\mathbf{v}_q^j$ transform the generated 3D surface to the global frame:

$$Y^j = \{\mathbf{v}_q^j \circ G(\mathbf{v}_f^j, s) | s \in S\}$$

Points to Parts Routing

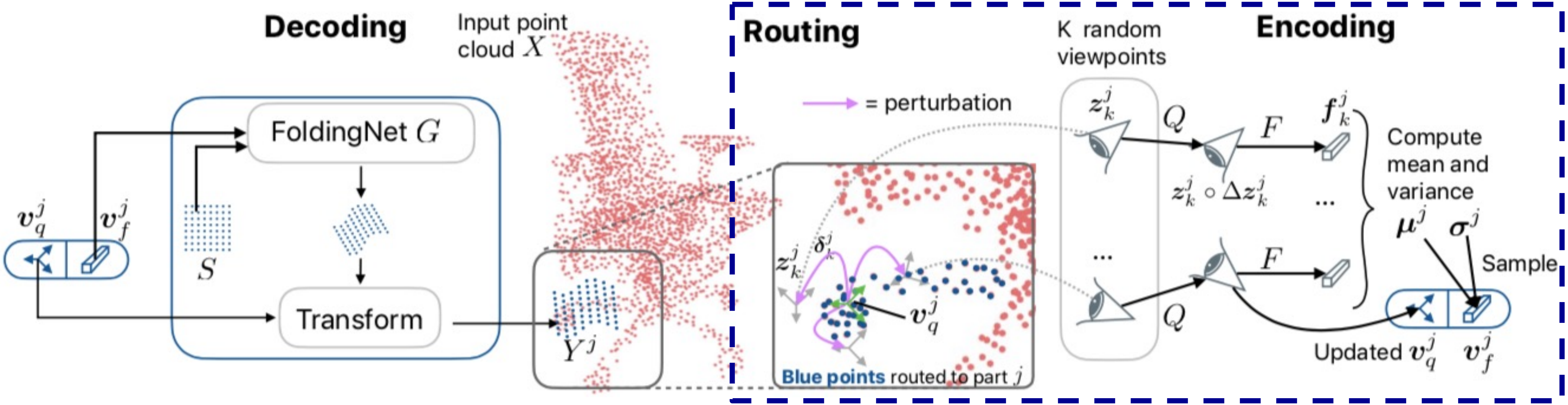


- Point $\mathbf{x}^i$ should be routed to capsule j if it is well explained by some point in Y^j

$$b_{ij} \equiv \log P(\boldsymbol{x}^i | Y^j) \propto - \min_{\boldsymbol{y} \in Y^j} \left(\frac{||\boldsymbol{x}^i - \boldsymbol{y}||^2}{\sigma_j^2} + \log(\sigma_j) \right)$$

- The log probs are used to compute $R_{ij} \in [0, 1]$ using softmax over J capsules.

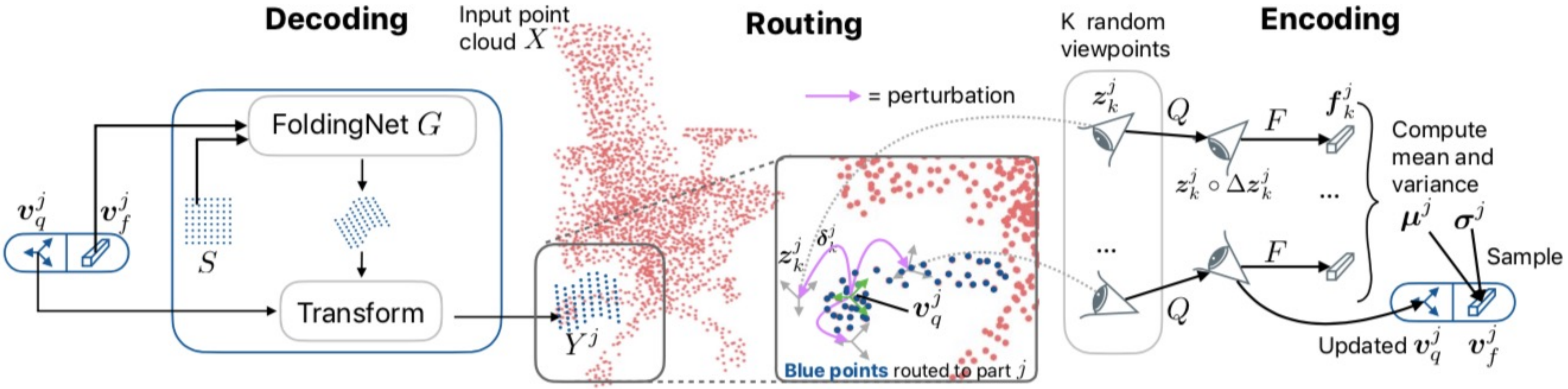
Points to Parts Encoder



- Given R, Multi-View Agreement is used to infer each capsule $(\mathbf{v}_q^j, \mathbf{v}_f^j)$
- Generate K random viewpoints. 3D points are then embedded, pooled and projected

$$\Delta \boldsymbol{z}_k^j = Q(X|\boldsymbol{z}_k^j, R) = Q_{\text{project}}(\text{maxpool}_i R_{ij} Q_{\text{embed}}((\boldsymbol{z}_k^j)^{-1} \odot \boldsymbol{x}^i))$$
$$\boldsymbol{f}_k^j = F(X|\boldsymbol{z}_k^j \circ \Delta \boldsymbol{z}_k^j, R) = F_{\text{project}}(\text{maxpool}_i R_{ij} F_{\text{embed}}((\boldsymbol{z}_k^j \circ \Delta \boldsymbol{z}_k^j)^{-1} \odot \boldsymbol{x}^i))$$

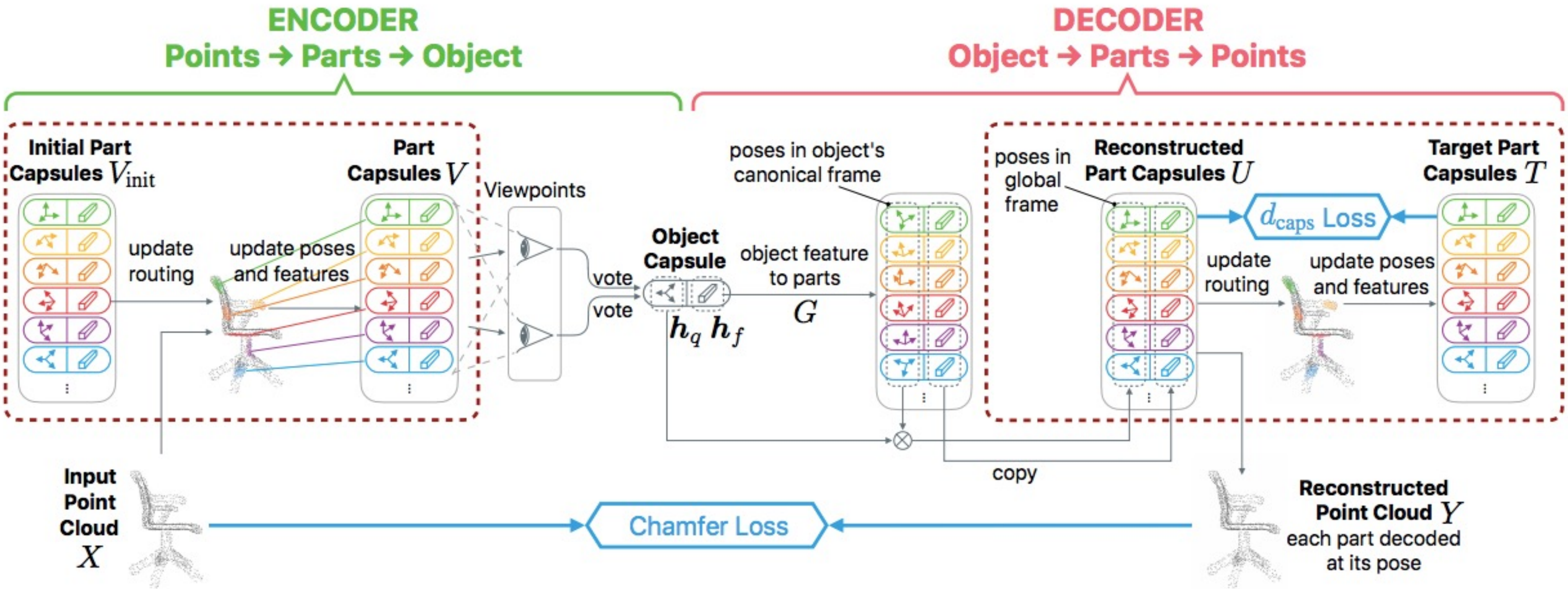
Points to Parts Autoencoder Loss



- Chamfer Loss:

$$\mathcal{L} = d_{\text{Chamfer}}(X, Y) = \frac{1}{|Y|}\sum_{\boldsymbol{y}\in Y} \min_{\boldsymbol{x}\in X} ||\boldsymbol{x}-\boldsymbol{y}||^2 + \frac{1}{|X|}\sum_{\boldsymbol{x}\in X} \min_{\boldsymbol{y}\in Y} ||\boldsymbol{x}-\boldsymbol{y}||^2$$

Parts to Object Autoencoder



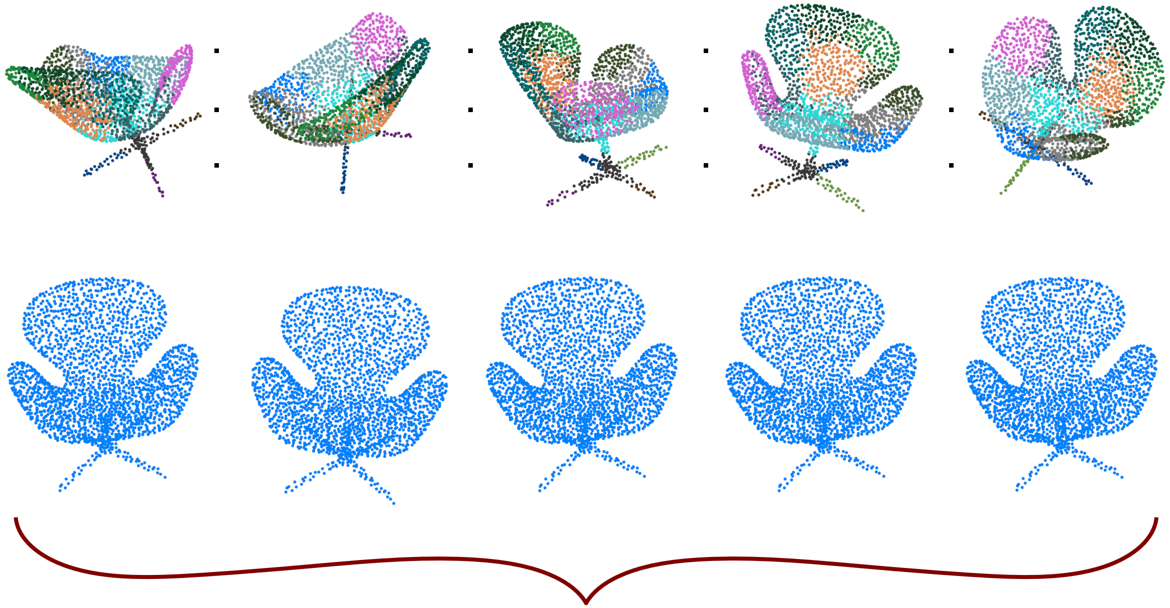
Results:

- Datasets : Training on ShapeNet Core 55, Testing on ModelNet40.
  - ShapeNet: 55-object-category, with 57,448 CAD models, each uniformly sampled to 2048 3D points. We used 2,468 test objects
  - Use 16 part capsules, each with 16-D feature
  - Entire object is modeled with 1024-D feature
- Two things to evaluate:
- **Pose-invariance of the feature component**: Object retrieval
  - Can the inferred feature be used to query and find an object, if it is present in a different view in the database?
  - Metric: Top-k retrieval accuracy.
- **Pose-equivariance of the pose component**: Alignment
  - Given two rotated views of the object, can the inferred pose be used to align them?
  - Metric: Relative Rotation Error.

Pose Equivariance Results

Transformed part capsules

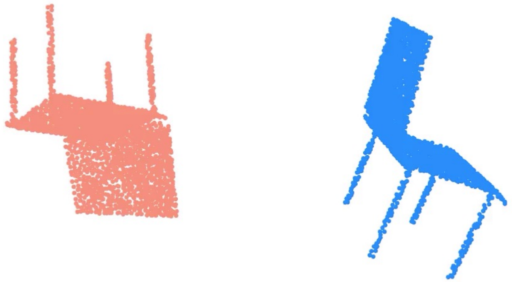
Point cloud in recovered canonical pose



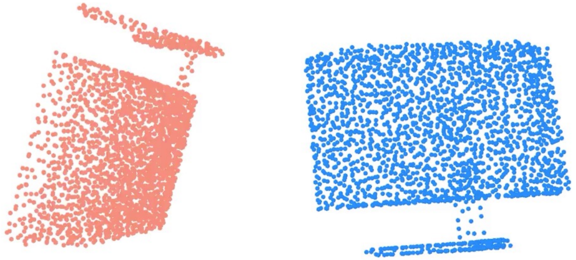
Superimposed point cloud in recovered canonical pose

Pose Equivariance Results

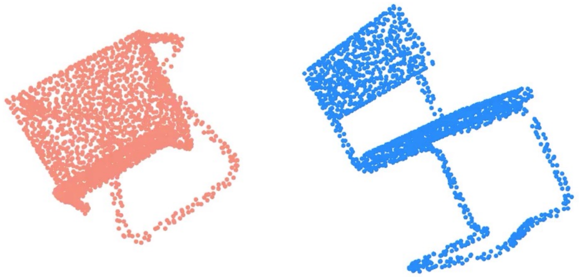
Input | Object as viewed from inferred pose



Input | Object as viewed from inferred pose

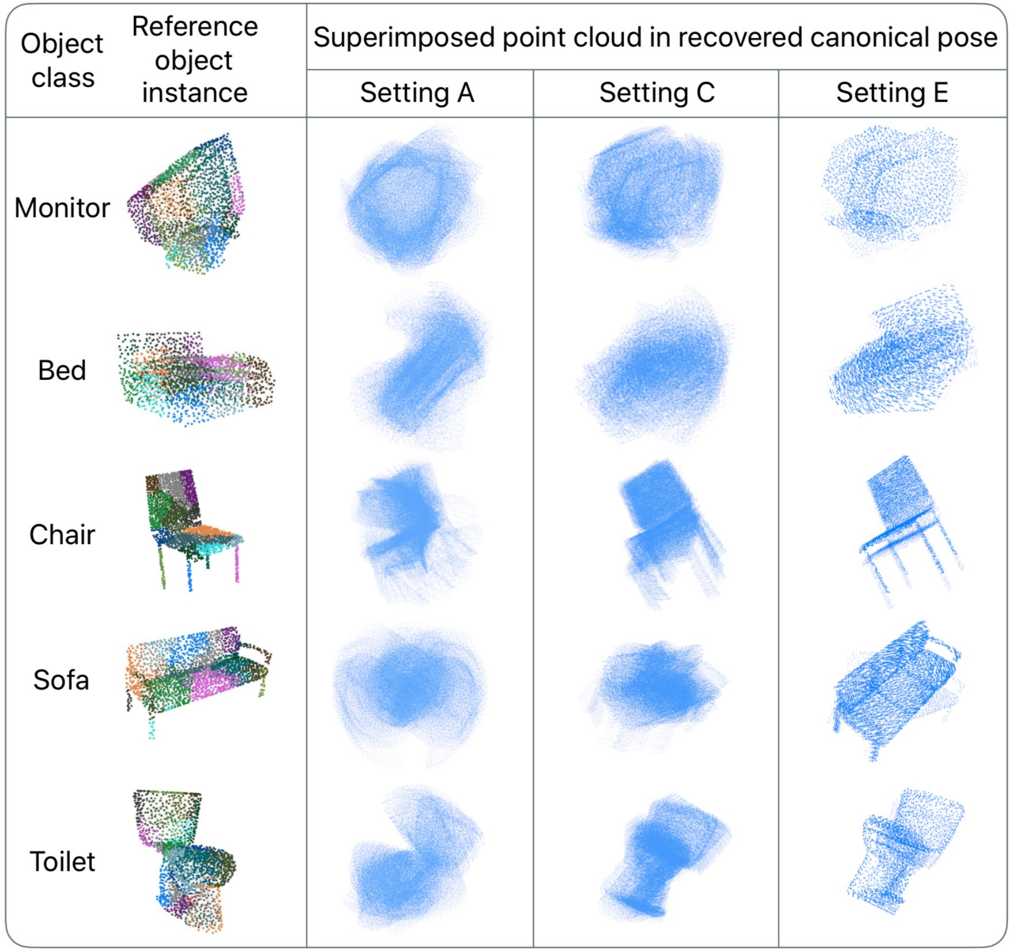


Input | Object as viewed from inferred pose



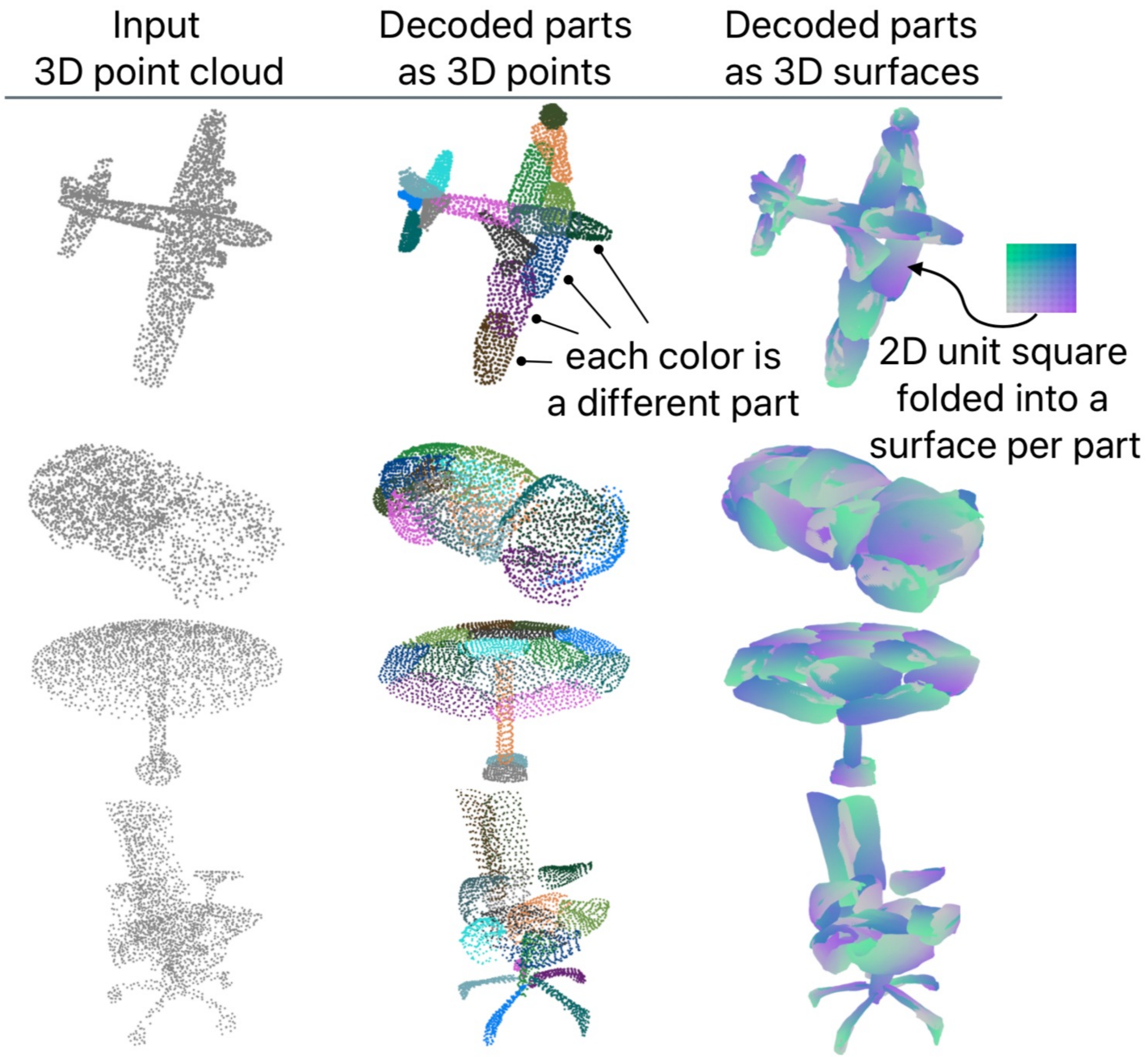
Pose Equivariance Results

- If the inferred pose is rotation equivariant, the superposed point clouds should align.

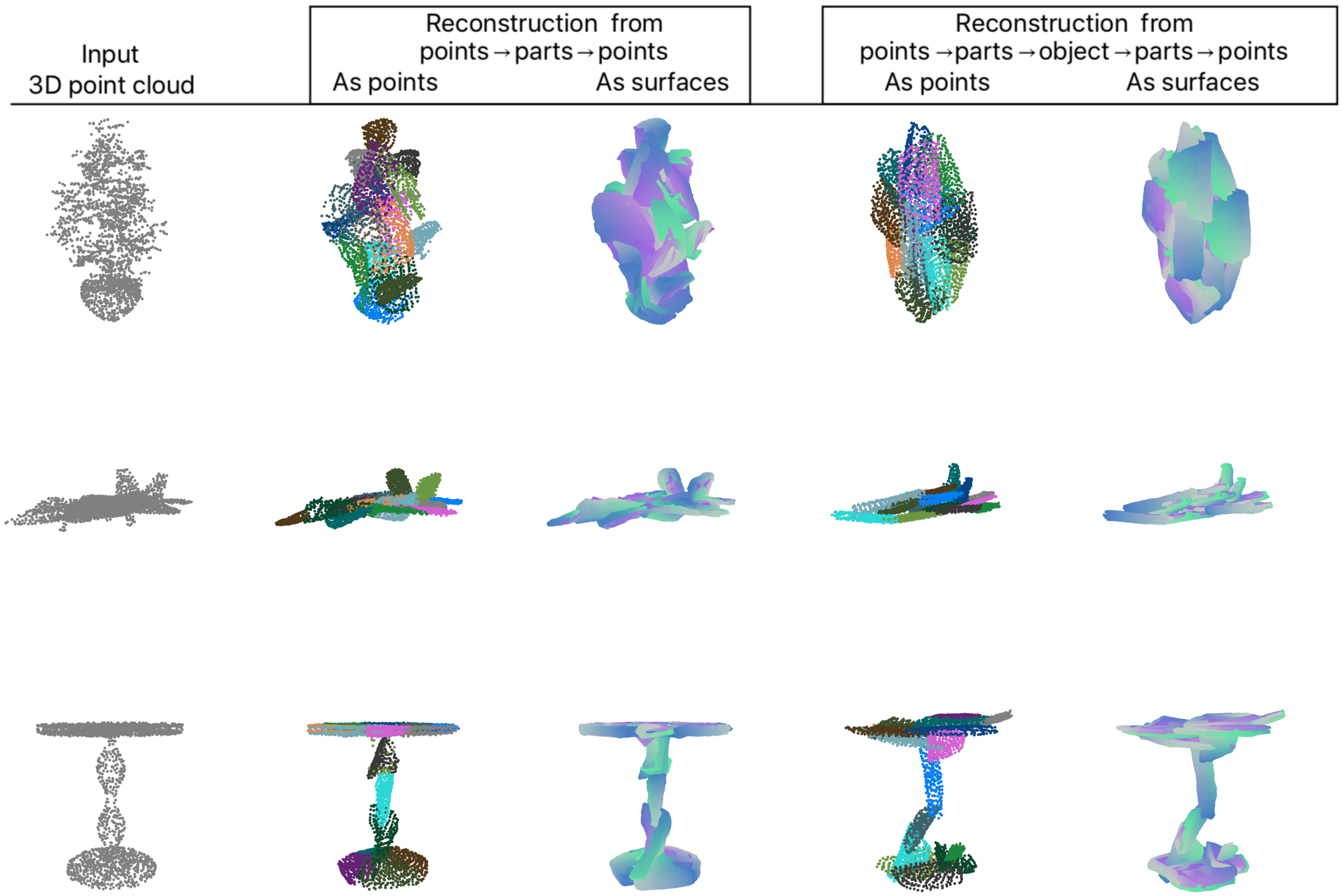


	Setting			Average	Instance
	K	θ_δ	steps	E_R	Retrieval
A	1	45	1	0.184 ± 0.022	0.42 ± 0.09
B	2	45	1	0.106 ± 0.020	0.78 ± 0.07
C	4	45	1	0.099 ± 0.018	0.82 ± 0.05
D	4	180	1	0.056 ± 0.021	0.99 ± 0.01
E	4	180	3	0.021 ± 0.003	0.95 ± 0.01

Reconstructions



Reconstructions



Conclusion

- Capsules allow us to potentially learn more interpretable object representations via learning "parts – to – whole" representations compared to black-box deep neural nets, such as CNNs.
  - Multi-Modal Routing can provide interpretability without sacrificing performance
  - Geometric Capsules can provide interpretable parts based representation

- Adaptively set the number of part capsules.
- 3D Scene flow using consistency of part-whole relationships over time.
- Need better inference (routing) algorithms
- How can we apply capsules to the raw input data (raw audio, visual, textual input).