



# Handling Noise and Metric Issue in Few-Shot Learning Tasks with In-Memory Search

---

Speaker : B11901027 王仁軒

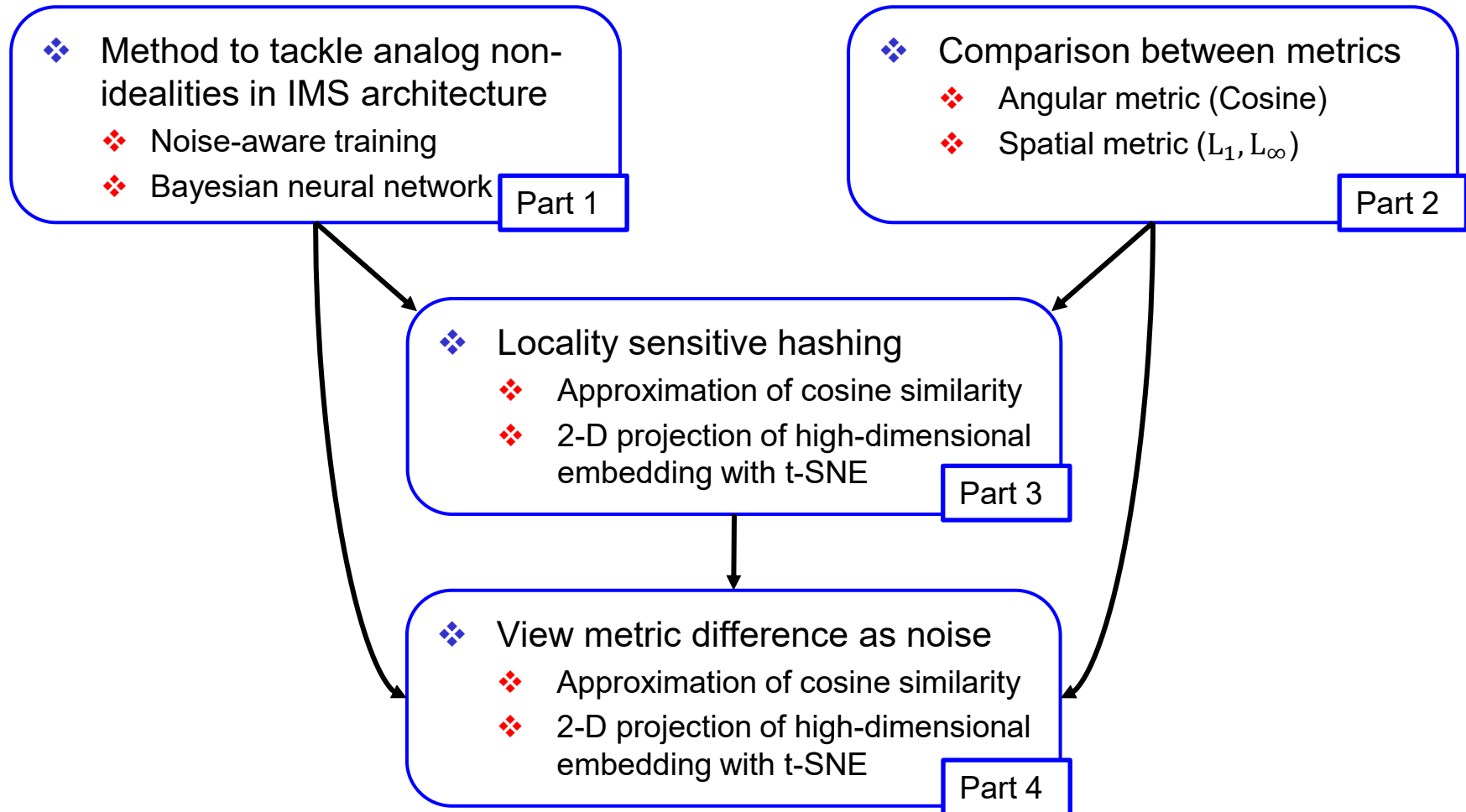
Mentor : Rick Huang

Advisor: Prof. An-Yeu (Andy) Wu

Date : 2025/06/??



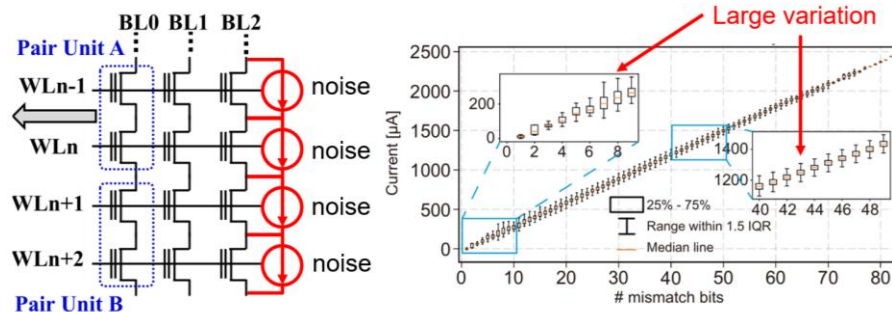
# Outline





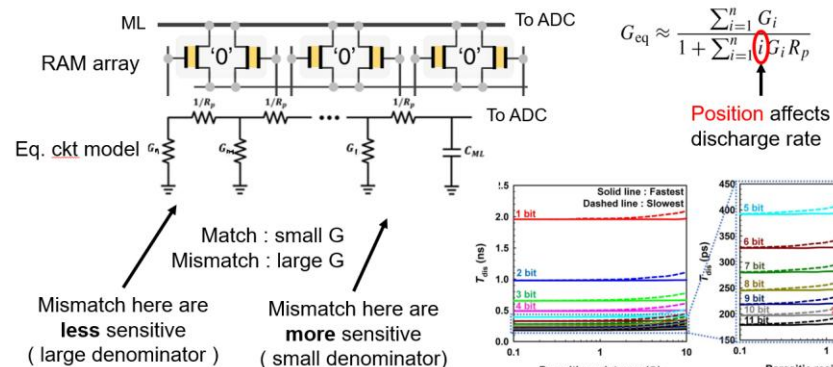
# Analog Non-Ideal Effects of TCAM

- ❖ TCAM : Ternary content addressable memory
- ❖ Analog non-ideal effects of in-memory-search



Noise from memory device

- Thermal noise
- Flicker noise
- Leakage current



$$G_{eq} \approx \frac{\sum_{i=1}^n G_i}{1 + \sum_{i=1}^n G_i R_p}$$

Position affects discharge rate

Parasitic effects of lump elements

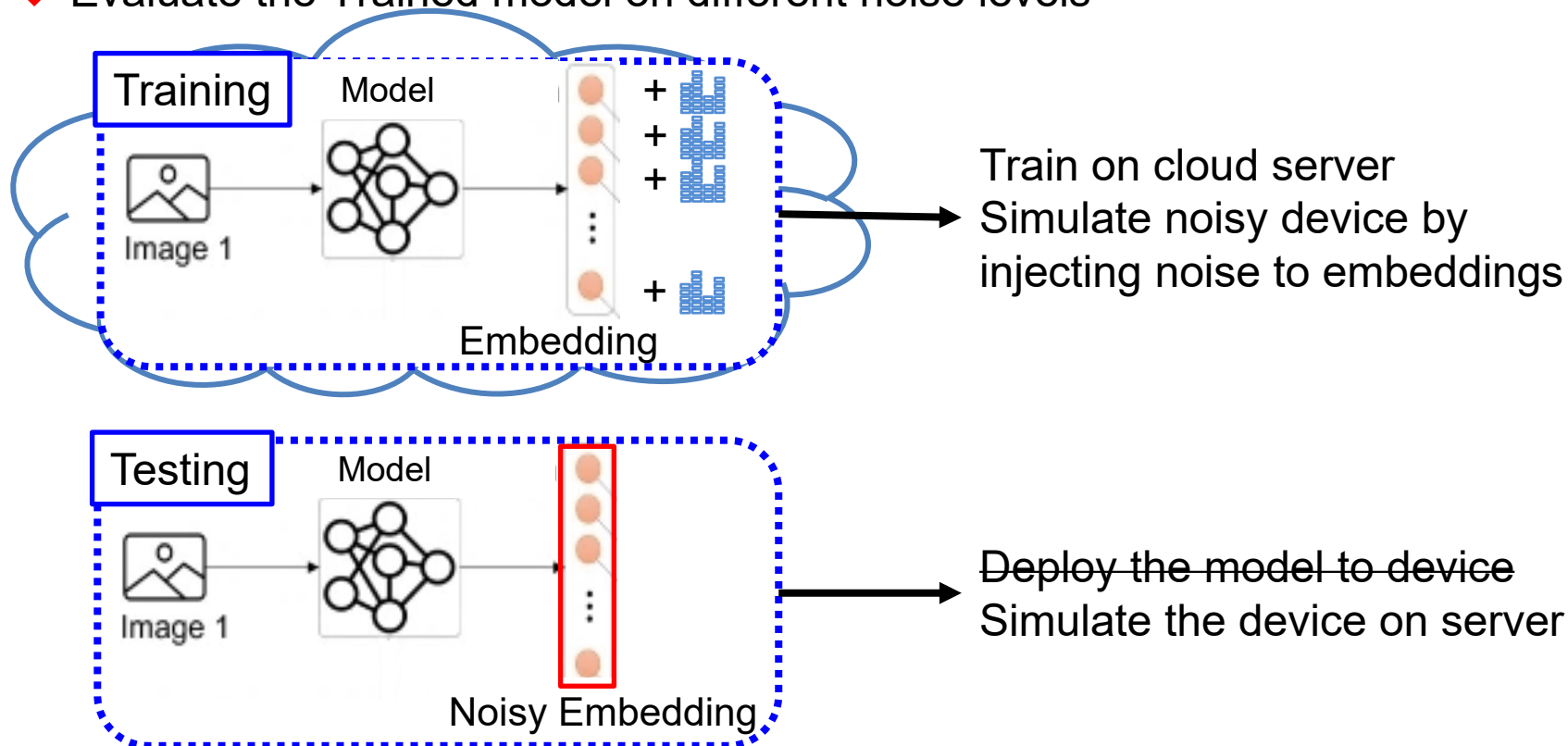
- Resistance
- Capacitance



# Method 1 : Noise-Aware Training

## ❖ Noise-Aware Training

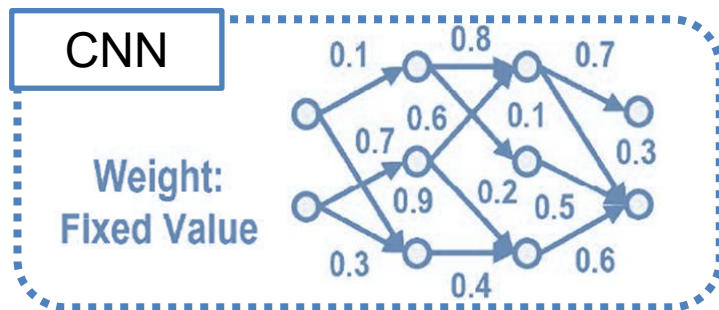
- ❖ Inject simulated noise into embeddings while training the model
- ❖ Evaluate the Trained model on different noise levels





# Method 2 : Bayesian Neural Network

- ❖ Bayesian Neural Network (BNN)
  - ❖ Train a robust model that **embraces noise**
  - ❖ BNN minimizes KL-divergence (maximize Evidence Lower Bound, ELBO)



Loss : Cross Entropy

$$\sum -P(D) \log P(W)$$



Loss : KL-divergence

$$\frac{1}{K} \sum_{k=1}^K \sum -f(D) \log f(W) + \beta \cdot KL(P(W)|\text{Normal})$$

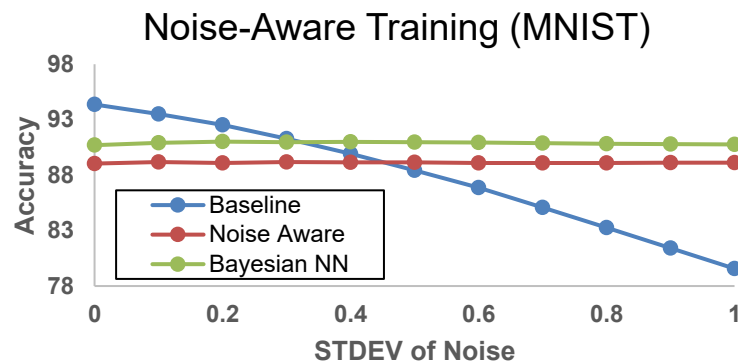
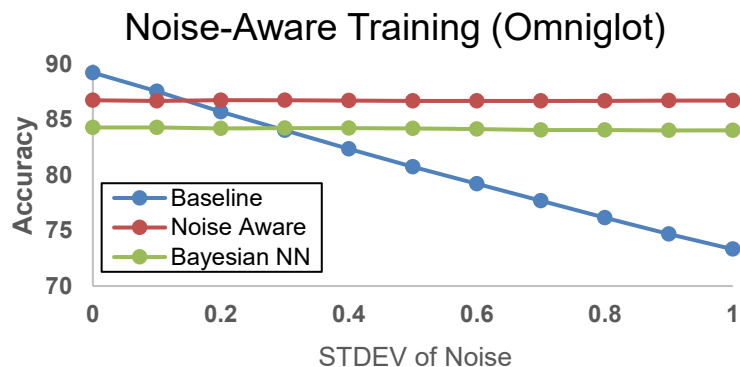
Mean of cross entropy  
loss across samples

Ensure robustness  
against noise

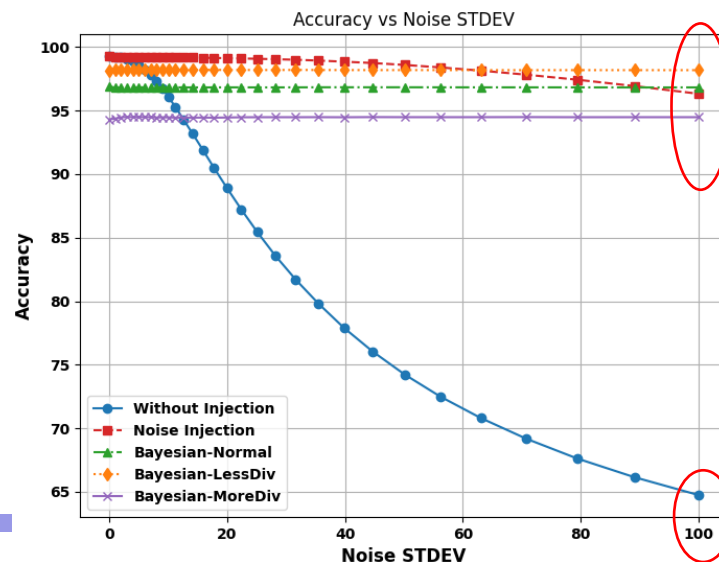
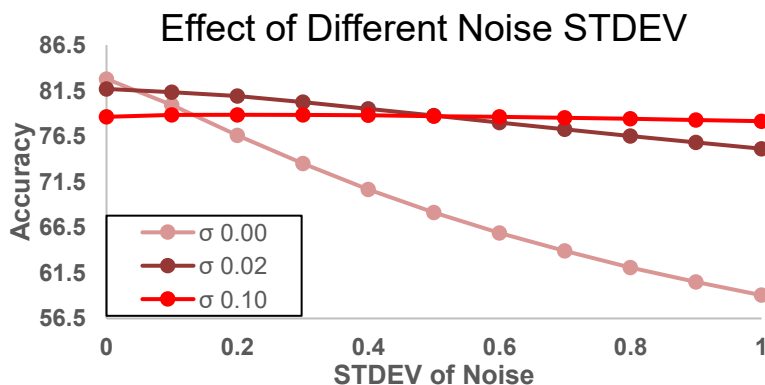


# Robustness Against Noise

- ❖ Both method works well on different datasets
- ❖ Trade-off between accuracy on clean data & noise tolerance



- ❖ Tolerance against large noise
- ❖ Little noise has great effect



High acc.

large noise

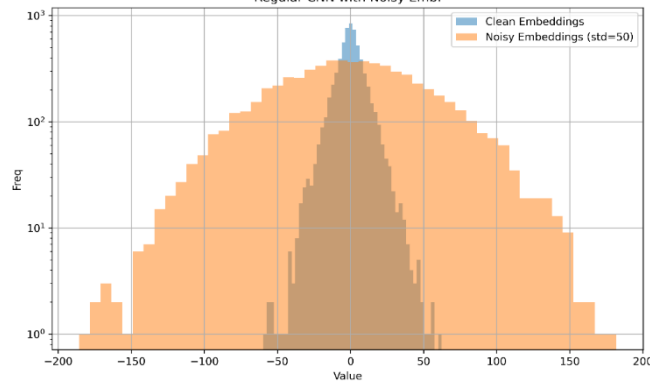


# Origin of Noise Resilience in NN

- ❖ Collect the value of every embeddings
  - ❖ Blue : Original embedding value distribution
  - ❖ Orange : New distribution on simulated noisy device
  - ❖ Model learns to against noise by amplifying magnitude of embeddings

## Baseline

Regular CNN with Noisy Emb.

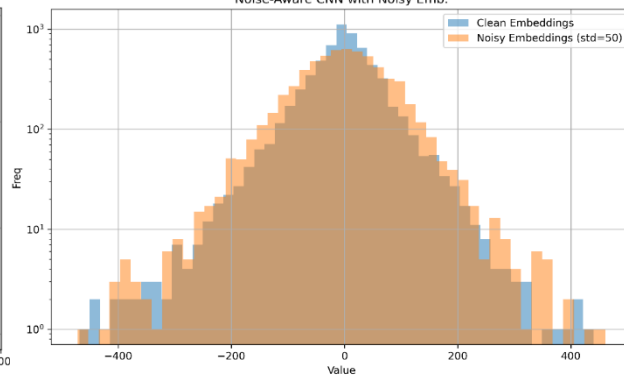


27 % accuracy

Noise dominates the embedding

## Noise-Aware Training

Noise-Aware CNN with Noisy Emb.

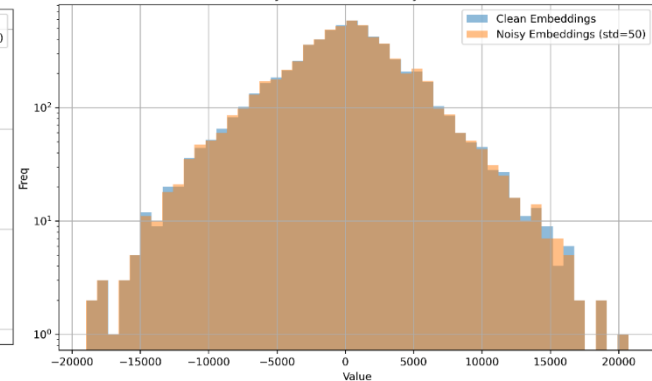


92 % accuracy

Noise has little impact on embedding

## Bayesian NN

Bayesian-CNN with Noisy Emb.



88 % accuracy

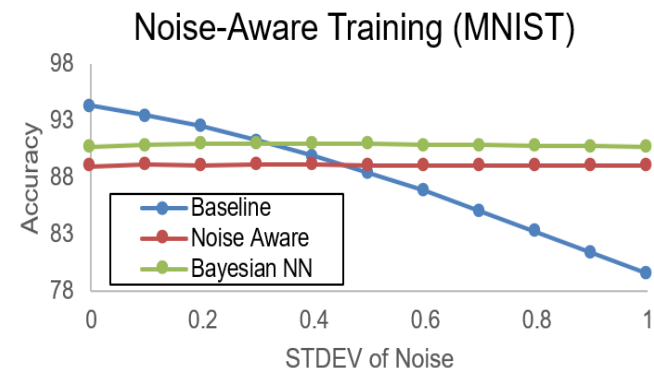
Noise has almost no impact on embedding



# Conclusion 1

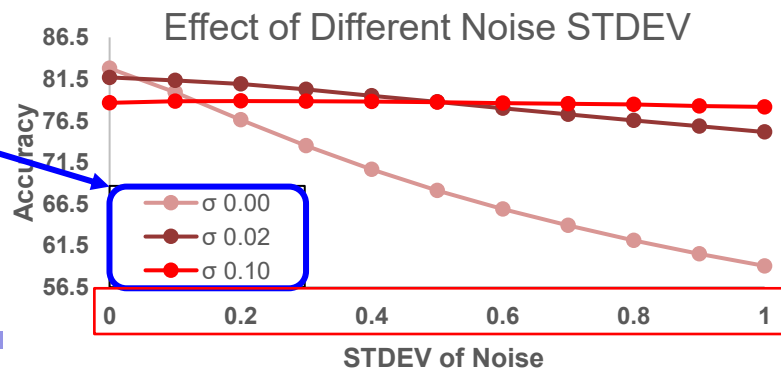
## ❖ Accuracy drop

- ❖ Original CNN model has higher acc. on clean device, but accuracy drops on noisy device
- ❖ Both noise-aware model and Bayesian NN resists noise by amplifying the mag. of embed.



- ❖ Trade off between model-robustness and accuracy on ideal device
- ❖ One small noise for training model, one giant leap for noise-tolerance

Little perturbation in training  
Great effect in testing



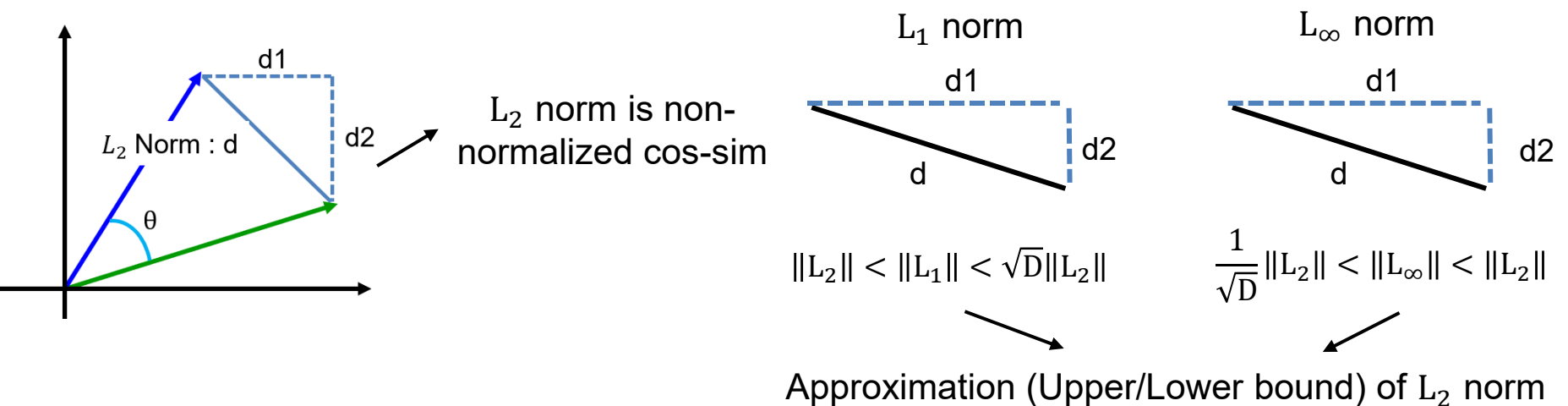


# Impact of Metric Selection on Accuracy

- ❖ Cosine similarity is too complicated to implement in memory cell

$$\text{sim}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} * \mathbf{B}}{\|\mathbf{A}\| * \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

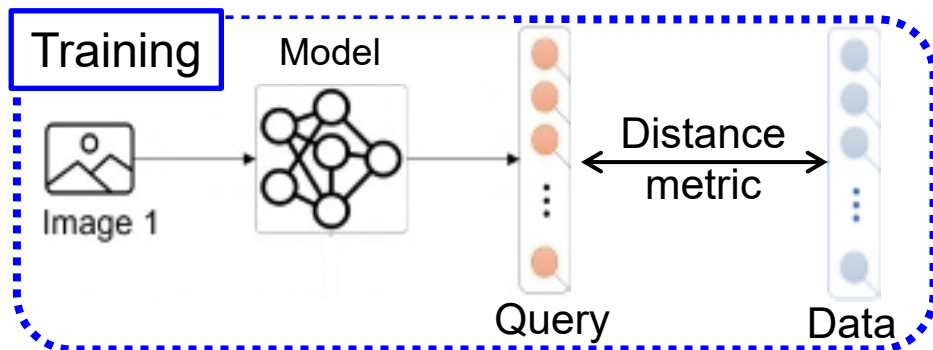
- ❖ Spatial metric is used to calculate similarity in memory
  - ❖ Simple hardware, but at what cost?
  - ❖ The performance may vary slightly between different metrics.





# Experiment Setup

- ❖ Evaluate accuracy of a model trained with different distance metric
  - ❖ Training : Approximated metric ( $L_\infty$  norm has no gradient for back prop.)
  - ❖ Testing : Regular metric on quantized embeddings

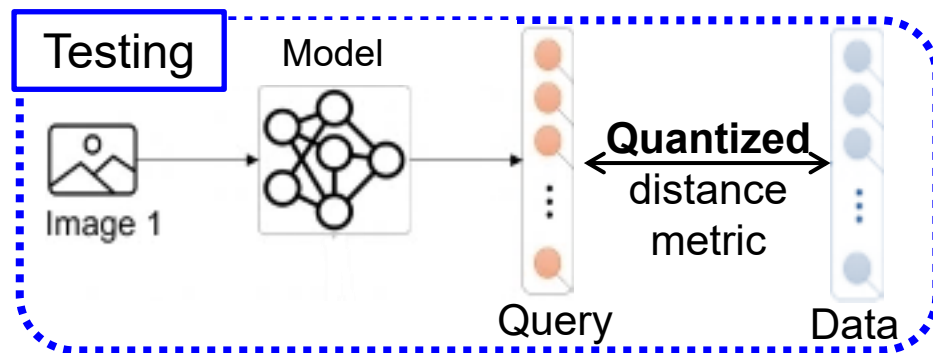


Metric (FP32) :

Cosine : regular cosine distance

$L_1$  : regular  $L_1$  norm

$L_\infty$  :  $\sum(\text{emb} \odot \text{softmax}(\kappa \text{ emb}))_i$



Metric (INT8) :

Cosine : cosine distance

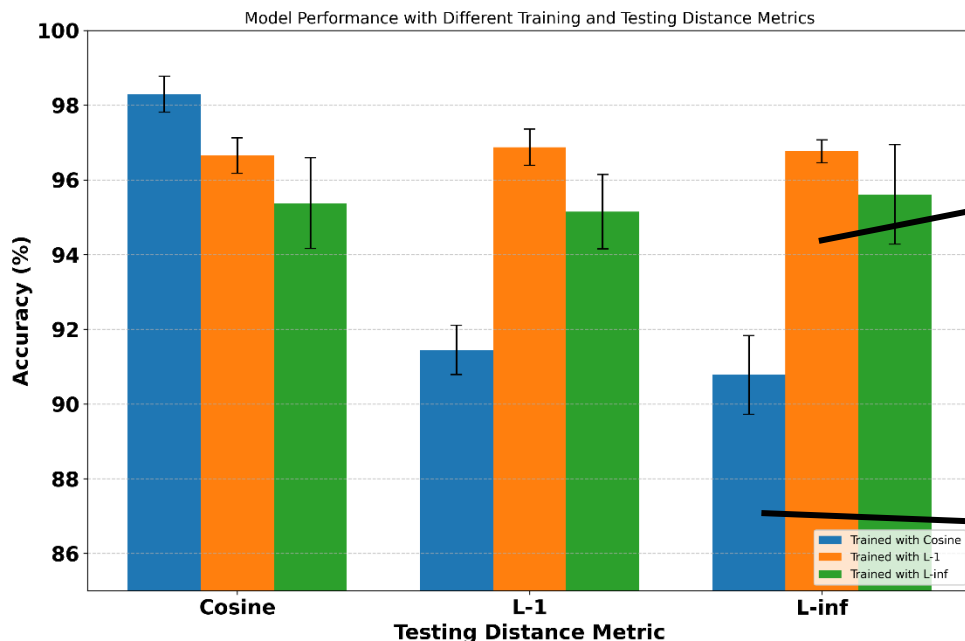
$L_1$  :  $L_1$  norm (  $\sum \text{emb}_i$  )

$L_\infty$  :  $L_\infty$  norm (  $\max(\text{emb}_i)$  )



# Evaluate Accuracy with Different Metrics

- ❖ Compare accuracies across models trained with different metrics
  - ❖ Dataset Omniglot and MNIST is used in the experiment
    - **Blue** : Trained with cosine distance (angular metric)
    - **Orange** : Trained with  $L_1$  norm (spatial metric)
    - **Green** : Trained with  $L_\infty$  norm (spatial metric)



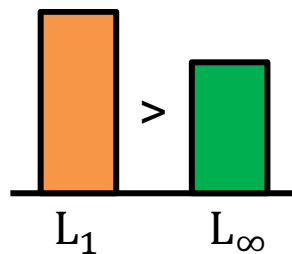
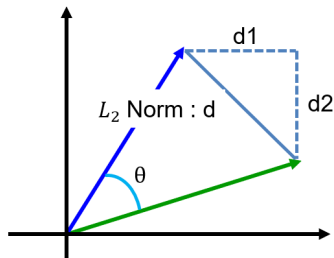
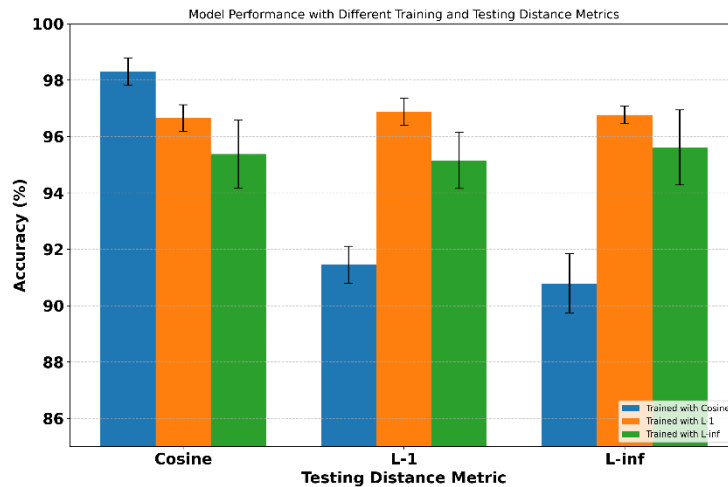
Approximate training metric of  $L_1$  norm performs well on evaluation

Using angular metric for training results in lower accuracy under spatial metric evaluation



# Observation Across Distance Metrics

- ❖ Model trained with  $L_1$  norm outperforms model trained with  $L_\infty$  norm in evaluation under all metrics



$L_1$  norm is a better metric?

$$L_2 \text{ norm} : \sqrt{\sum (\text{emb}_i - q_i)^2}$$

$$\nabla L_2 = 2 \cdot (\text{emb}_i - q_i)$$

$$L_1 \text{ norm} : \sum |\text{emb}_i - q_i|$$

$$\nabla L_1 = \text{sign}(\text{emb}_i - q_i)$$

Quantize  
to 1 bit

$$L_\infty \text{ norm} : \max(\text{emb}_i)$$

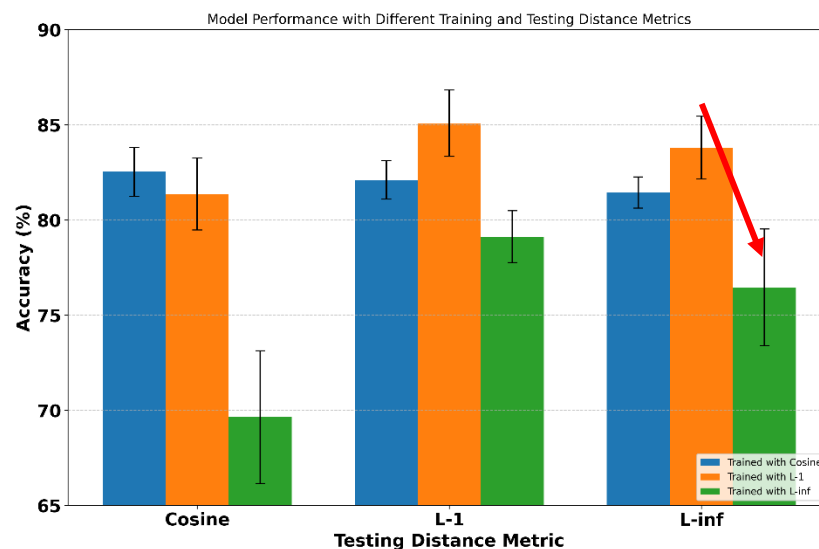
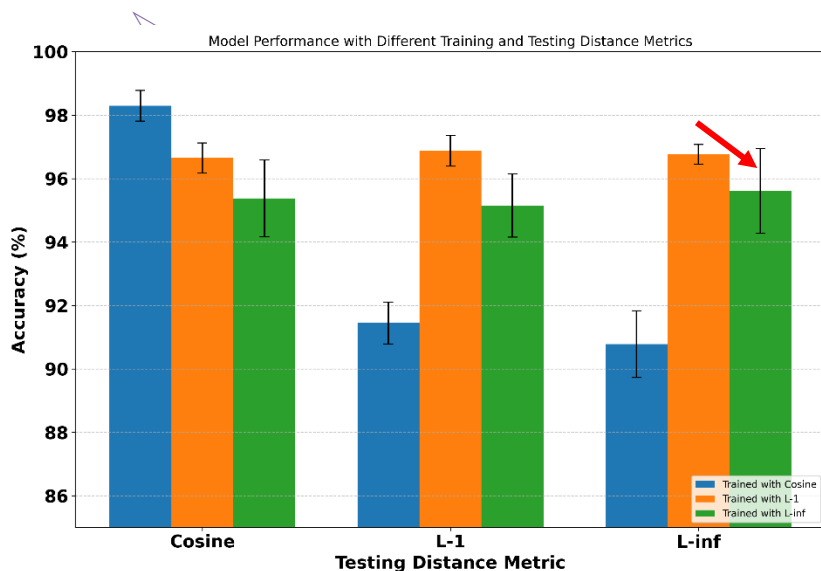
$$\nabla L_\infty = \begin{cases} \text{sign}(\text{emb}_i - q_i), & \text{emb}_i = \max(\text{emb}_i - q_i) \\ \text{None}, & \text{else} \end{cases}$$

$L_1$  norm can be view as a comprehensive version of  $L_\infty$  norm in training phase



## Conclusion 2

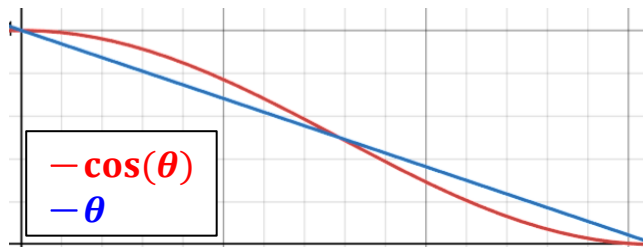
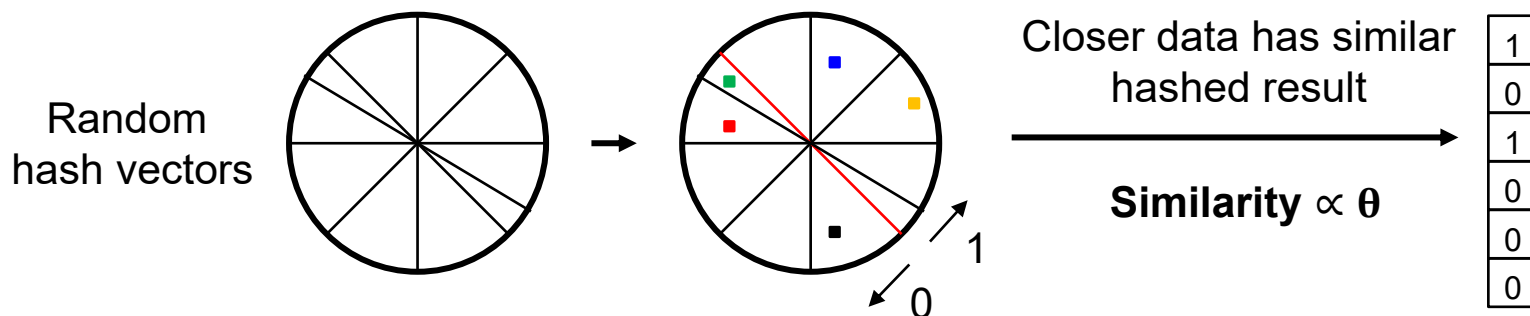
- ❖ Search function implemented in memory uses distance metric
  - ❖ Train the model with distance metric can achieve better performance
    - Use differentiable approximated distance function that allows backpropagation
  - ❖ Using  $L_1$  norm for training may have higher accuracy than using  $L_\infty$  norm
    - Model Trained with  $L_1$  norm may achieve higher accuracy than model trained with  $L_\infty$  norm when evaluating the performance with  $L_\infty$  norm





# Approximation of Cosine Similarity

- ❖ Sometimes we can only use a pretrained model
  - ❖ Cannot customize distance metric used in training
- ❖ Locality-Sensitive Hashing
  - ❖ A stochastic technique for finding neighbor with highest cosine similarity
  - ❖ Similar items map to the same buckets with high probability.

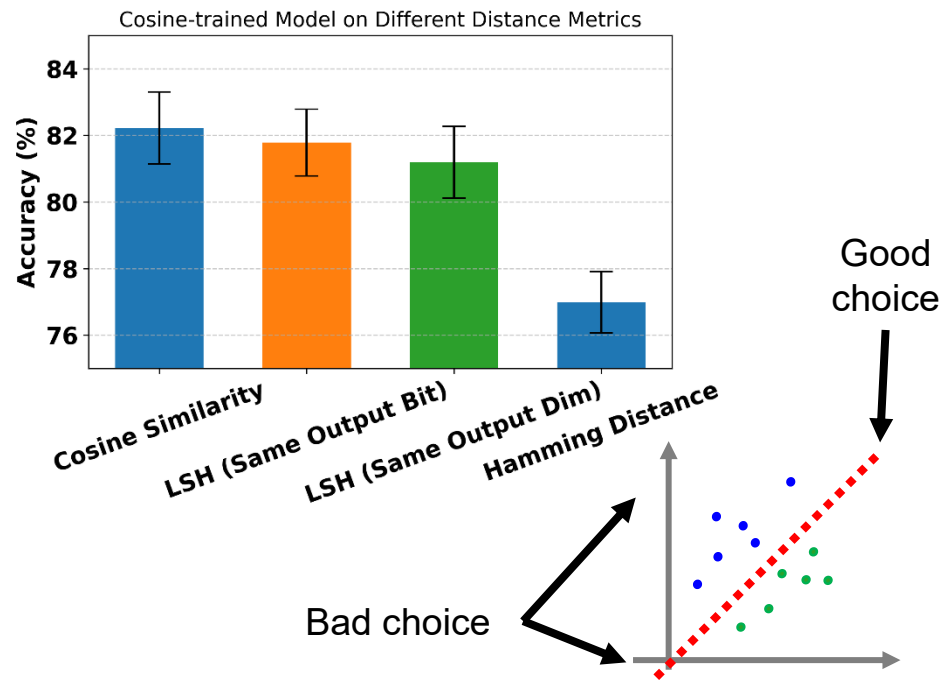
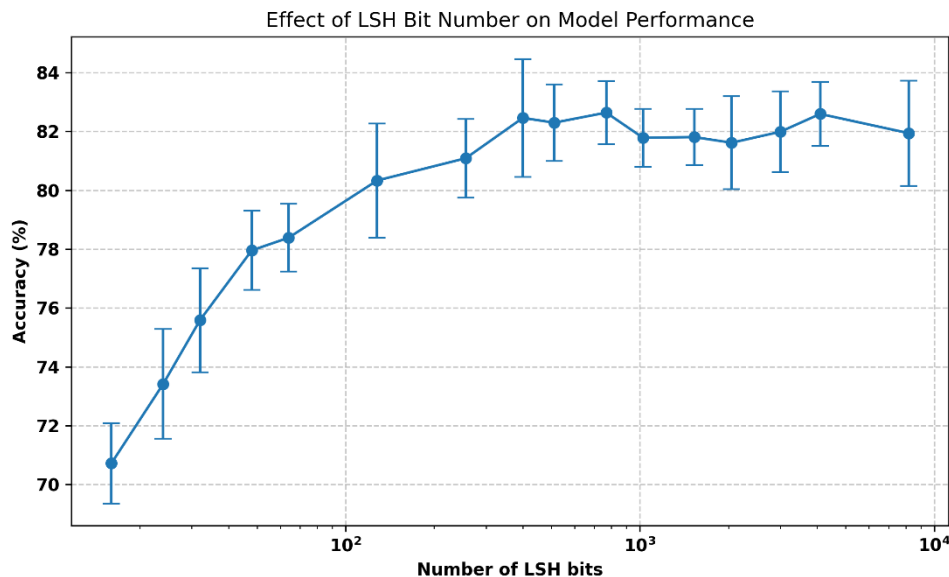


$\theta$  and  $\cos(\theta)$  are monotonic functions  
 $\Rightarrow$  Should have same nearest neighbor structure



# Effect of Locality-Sensitive Hashing

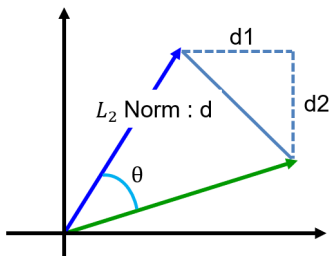
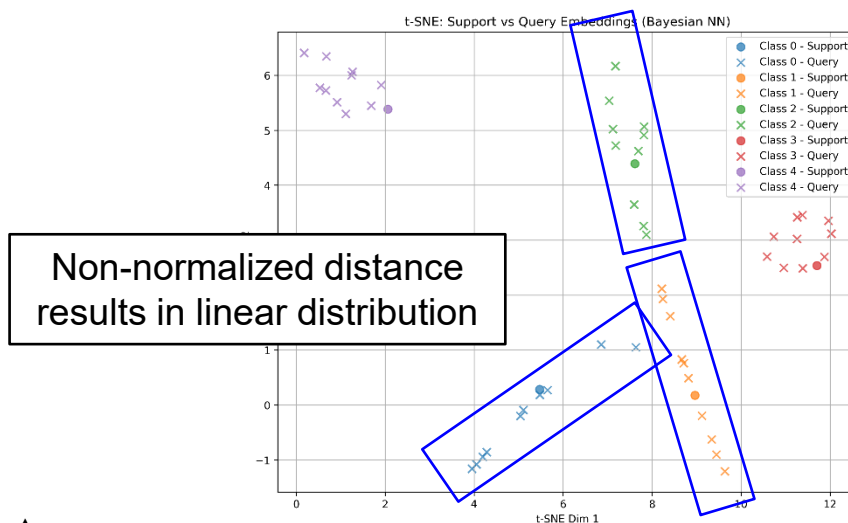
- ❖ Hashing vectors of LSH perform partitioning in Hilbert space.
- ❖ Hamming distance is a special case of LSH  
(hashing vectors are normal vector of coordinate planes)
- ❖ LSH can generally performs better than Hamming distance



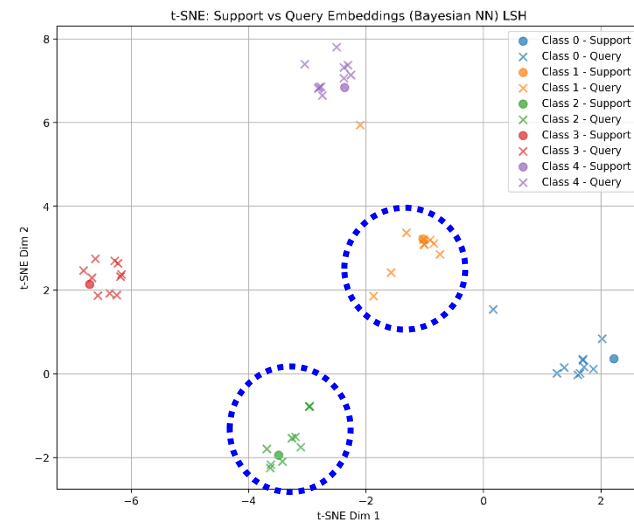


# What Else Can LSH Do

- ❖ 2-D data visualization using t-SNE method
- ❖ Visualization method that maintains distance in Hilbert space
- ❖ Locality-Sensitive Hashing maps angular distance to spatial distance



Use embeddings generated by model trained with cosine distance directly



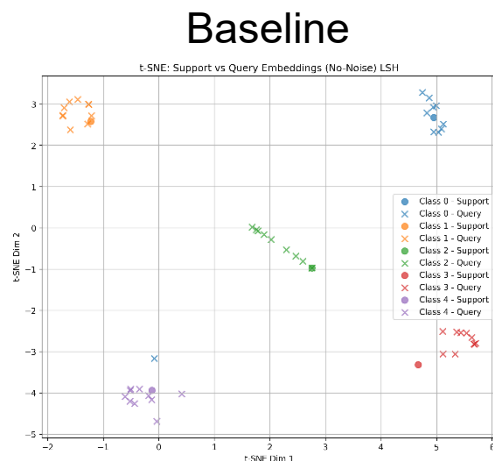
Do LSH on embeddings



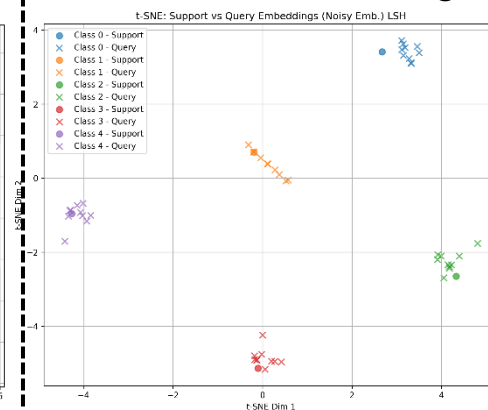
# 2-D Data Visualization with LSH

❖ Visualize clean and noisy data in experiment 1 using LSH & t-SNE

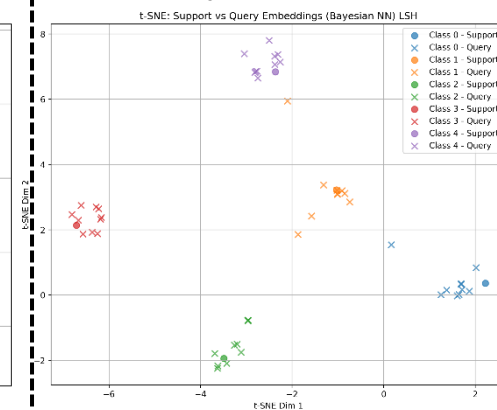
Zero  
Noise



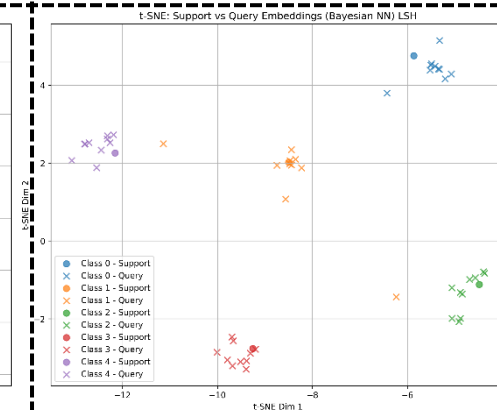
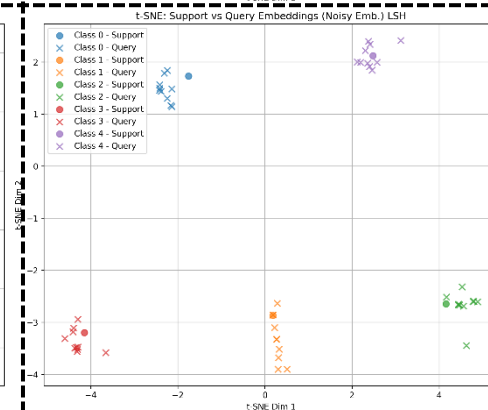
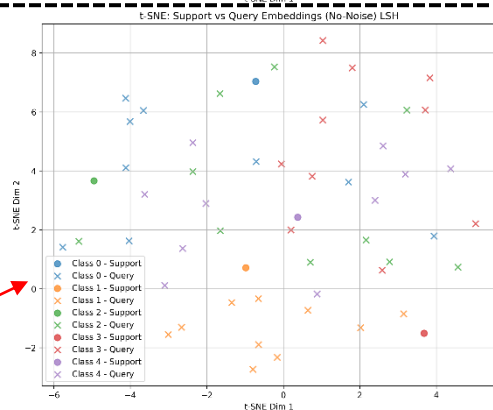
Noise-Aware Training



Bayesian NN



Large  
Noise

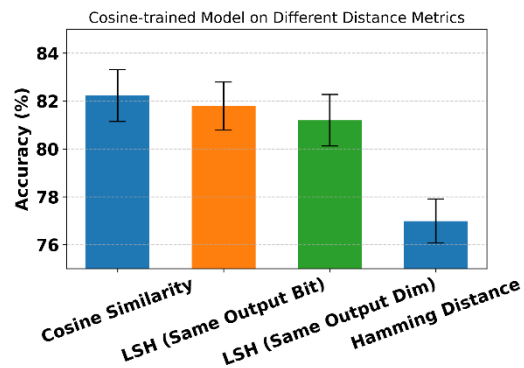


Low accuracy

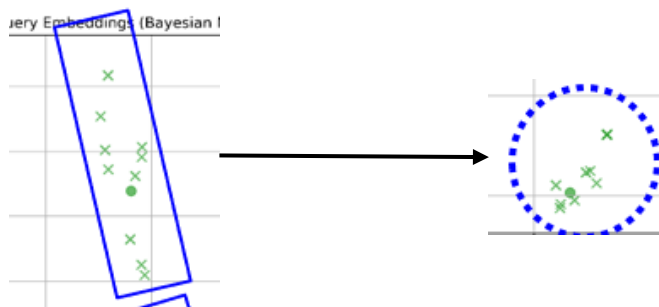


## Conclusion 3

- ❖ Approximation of cosine similarity
  - ❖ Locality sensitive hashing is an alternative method if we can only get a model trained with cosine similarity which IMS does not support



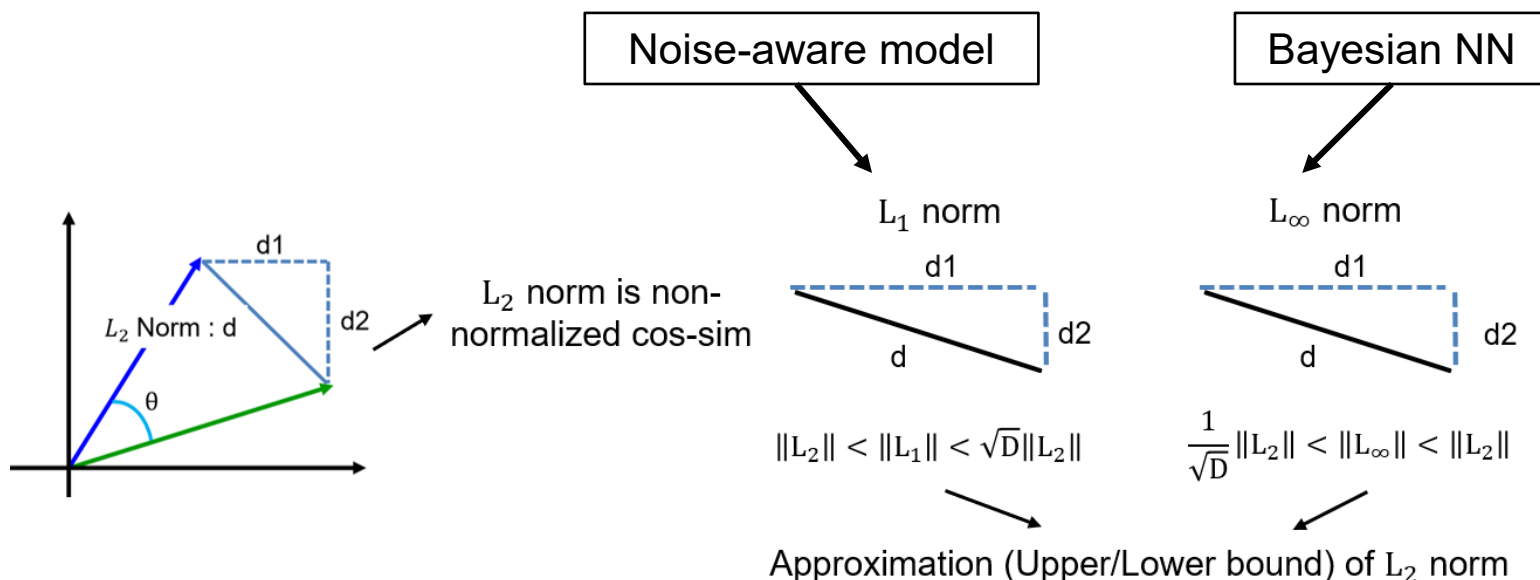
- ❖ 2-D data visualization
  - ❖ LSH converts angular metric to spatial metric which is better for t-SNE





# A General Way to Resolve Metric Issue

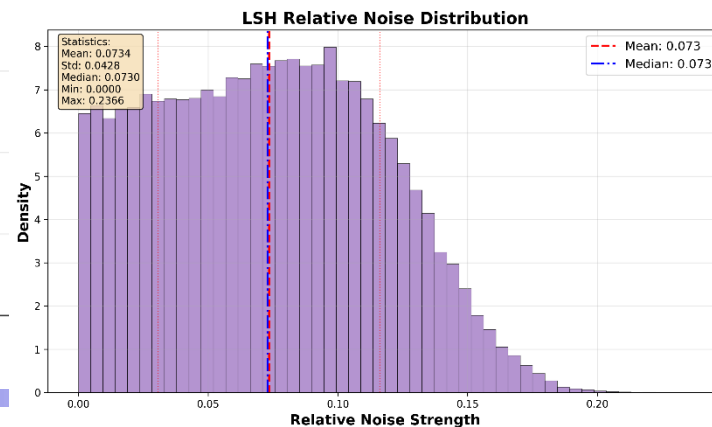
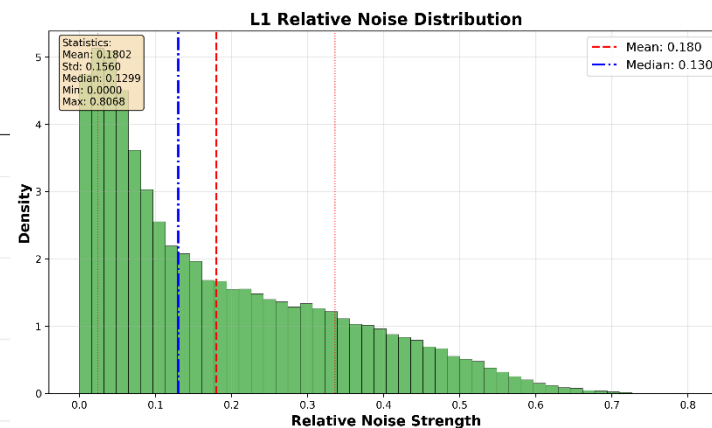
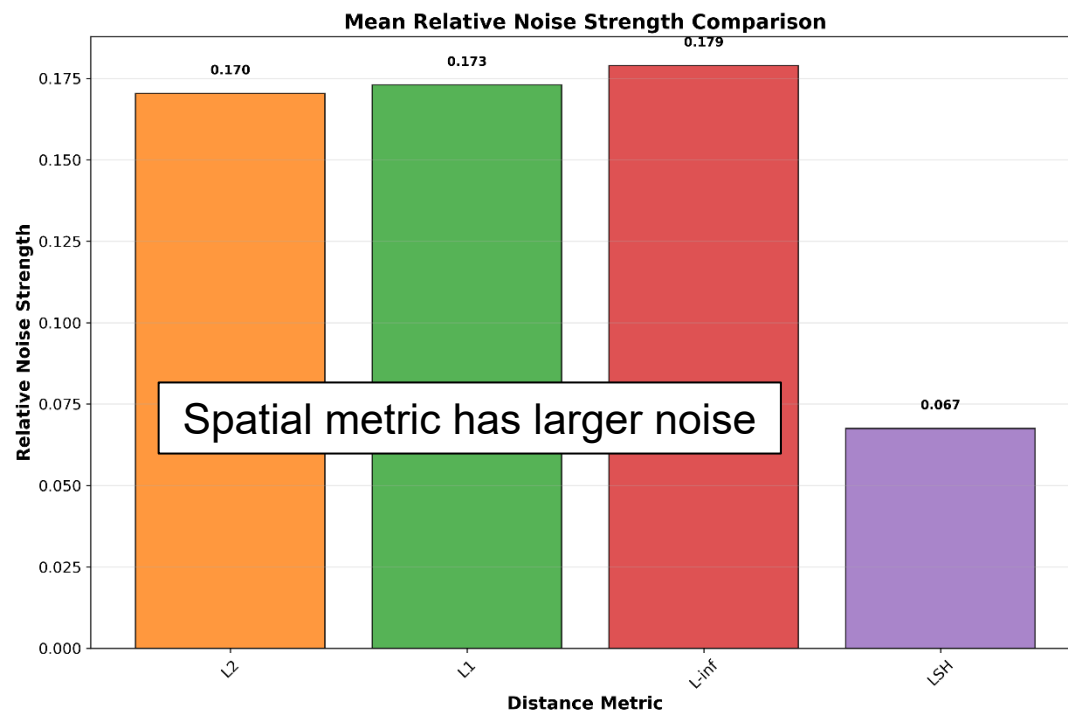
- ❖ Can we train a general model that has acceptable accuracy on each distance metric?
  - ❖ View different metric as a noisy version of cosine similarity
  - ❖ Train a noise-resilient model using cosine similarity





# Noise Strength of Metrics

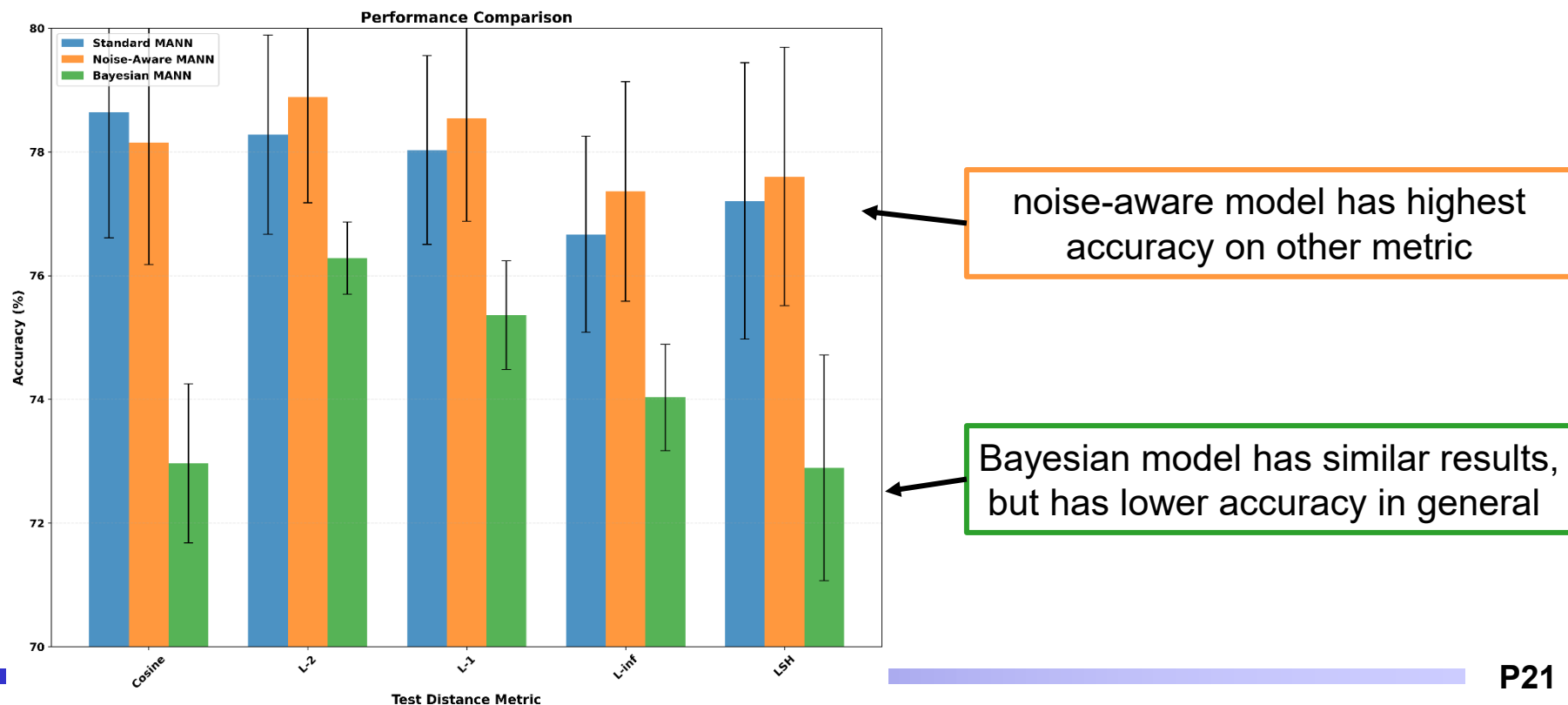
- ❖ Observation of the magnitude of each distance metric if we regard them as the source of noise
  - ❖ Clean signal : cosine similarity
  - ❖ Relative noise :  $LSH < L_2 < L_1 < L_\infty$





# Noise-Resilient Model Resolves Metric Issue

- ❖ Noise-Resilient Model achieves higher accuracy when distance metric is not cosine distance
- ❖ Original CNN model has highest accuracy on cosine similarity





## Conclusion 4

- ❖ View different metrics as inaccurate versions of cosine similarity

- ❖  $L_2$  norm : Non-normalized cosine distance
- ❖  $L_1$  norm : Upper bound of  $L_2$  norm
- ❖  $L_\infty$  norm : Lower bound of  $L_2$  norm

$$\|L_\infty\| < \|L_2\| < \|L_1\|$$

- ❖ Train a model that performs well on general distance metrics
  - ❖ Noise-aware model can achieve better performance in general cases
  - ❖ Bayesian model has similar behavior



# Conclusion of All Experiments

