Deep Model Transition

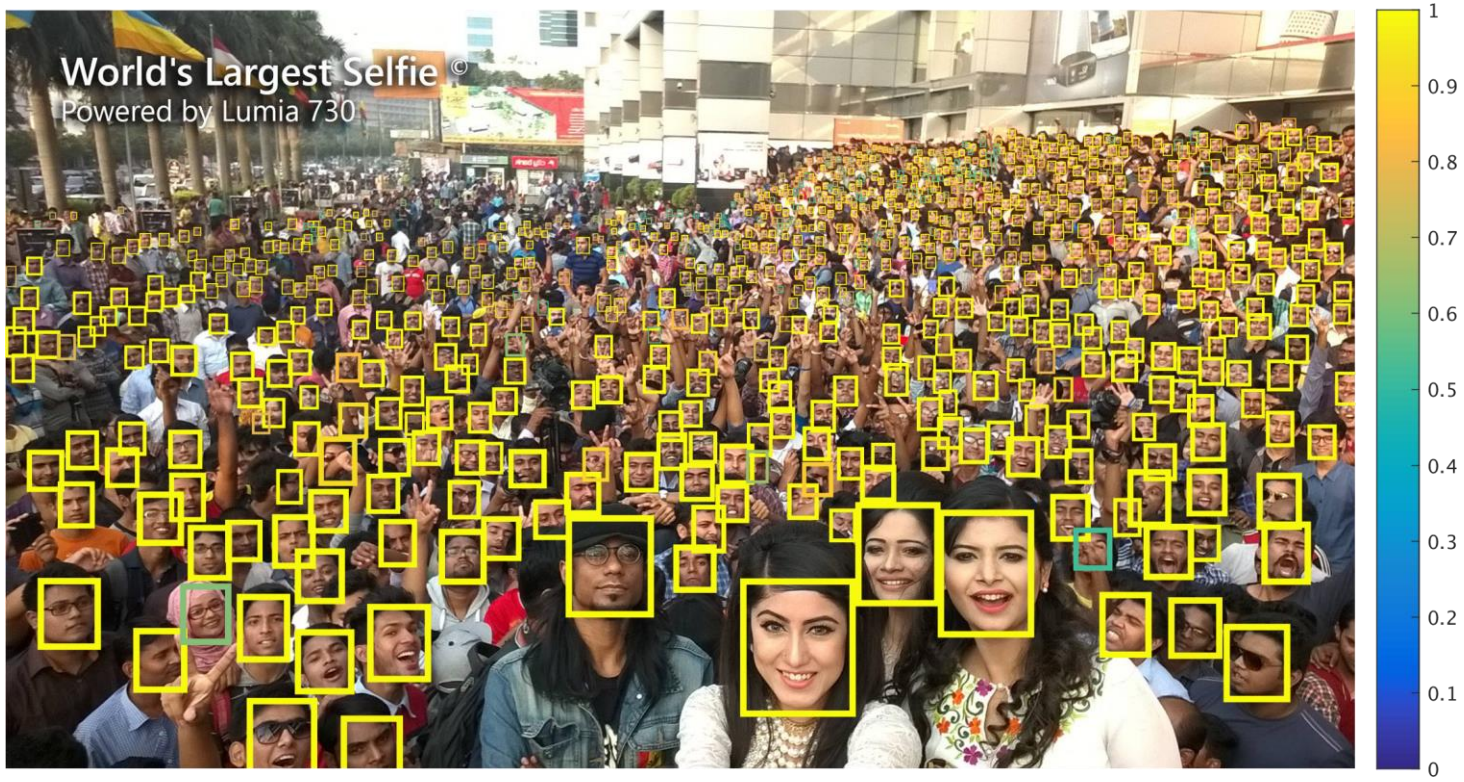
FOR ONE SHOT OBJECT CLASSIFICATION

JEDRZEJ J. KOZERAWSKI

General outline

- Problem overview
- Related work
- Proposed solution
- Results
- Future work

Problem description



Classification

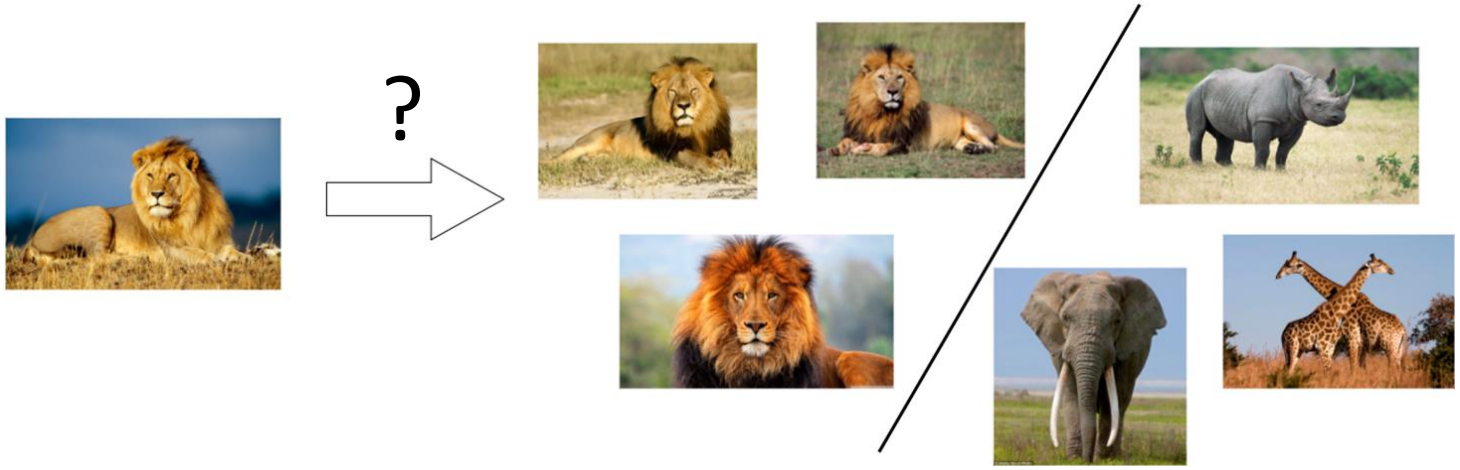
Visual object classification

DATA is the KEY

Tiny Faces – finds 685 faces out of 1000 reported

Source: Hu, Peiyun, and Deva Ramanan. "Finding Tiny Faces." *arXiv preprint arXiv:1612.04402* (2016).

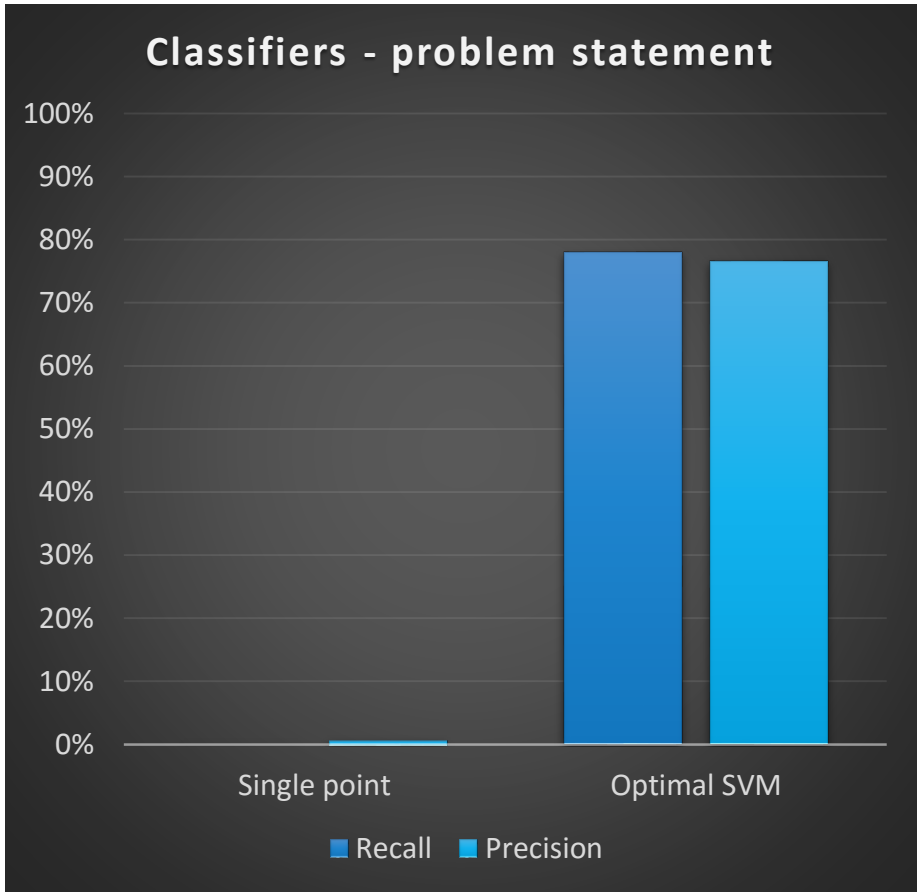
Data problem



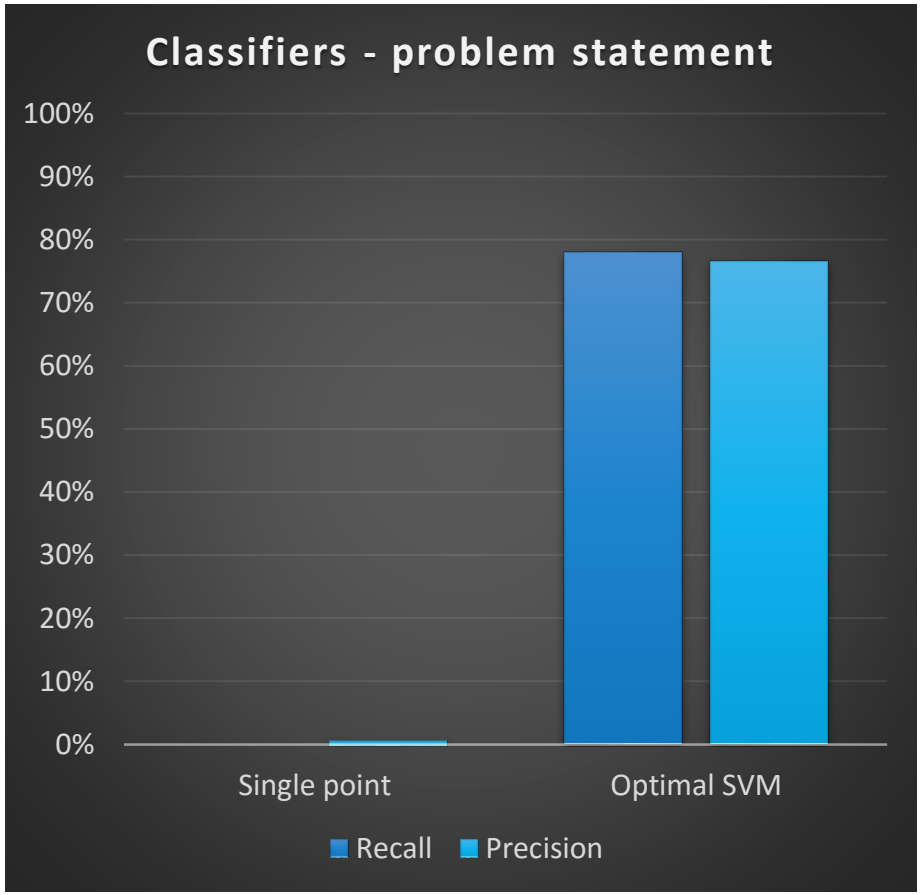
One image is not sufficient to properly train a classifier.

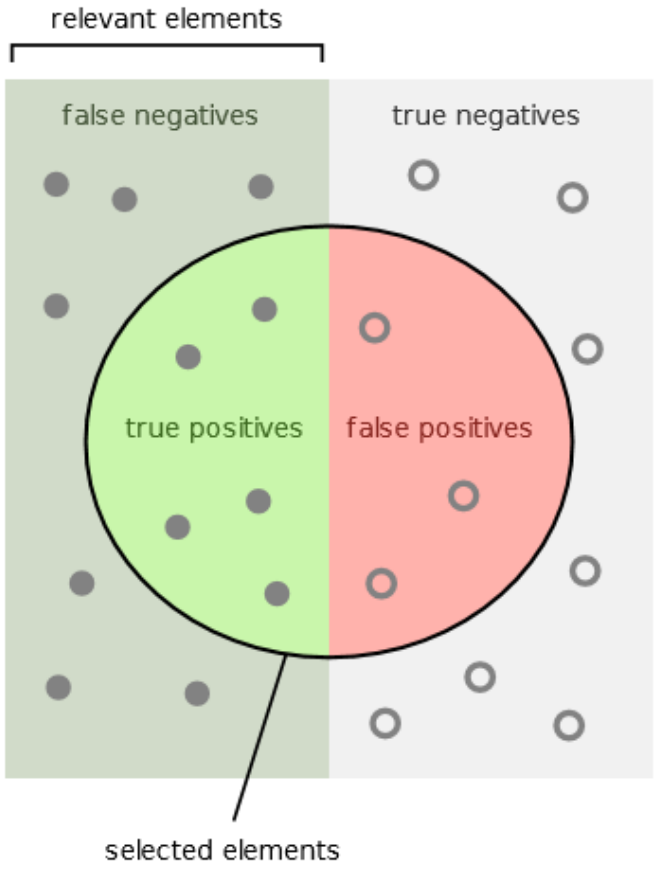
Task:
Given unlabeled data set and a query image representing previously unseen object build a classifier for this novel category.

No data - how bad can it be?



No data - how bad can it be?





How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

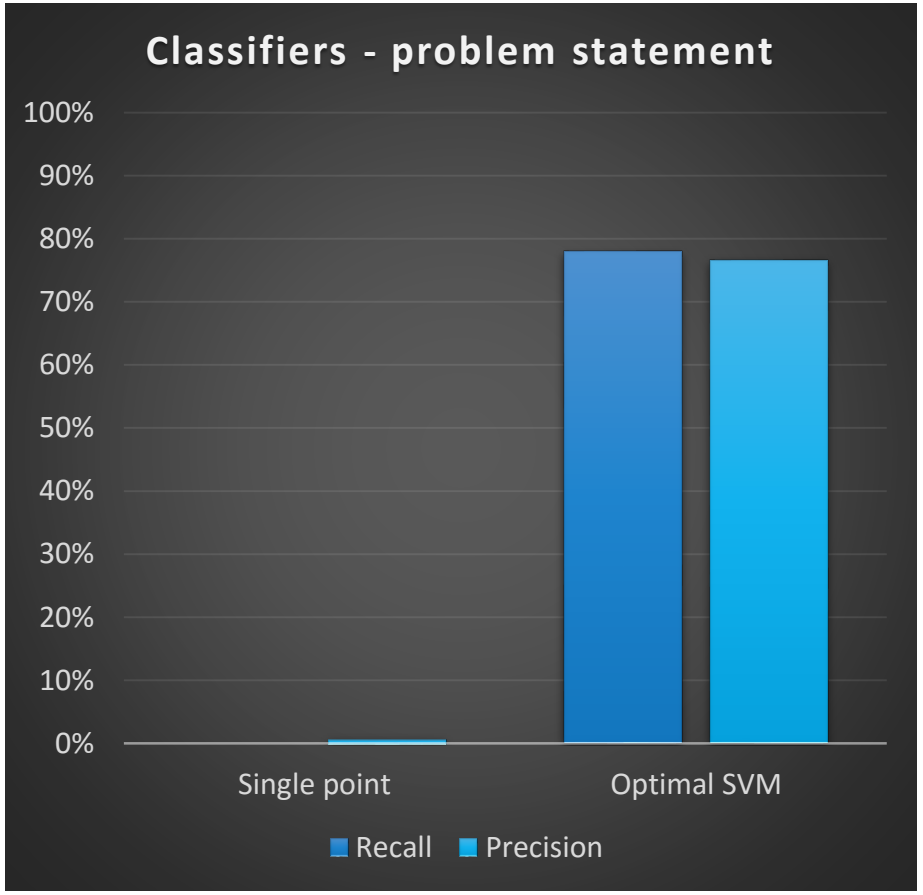
$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Source: en.wikipedia.org/wiki/Precision_and_recall

No data - why that can be?

- SVM classification results on Pascal 2007 VOC dataset

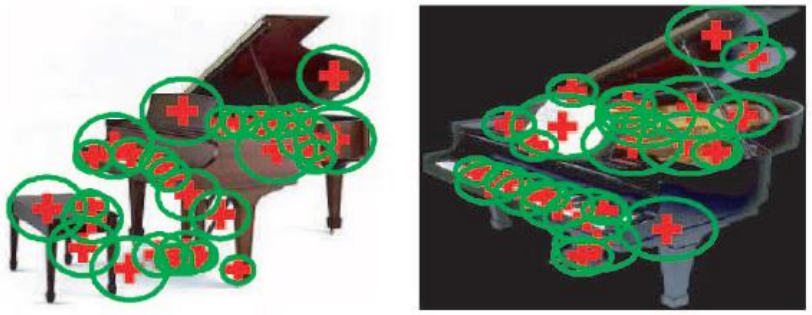
- But seriously, why? - Possible reasons:
  - Data:
    - Not available
    - Too expensive
    - Time consuming
  - User may lack technical know-how
  - Simple classifier creation ability for end-users:
    - Smartphone
    - AR/VR
  - Easier dataset creation – automated labeling



Related work

Bayesian One Shot learning

- Using Bayesian Inference to find key descriptors of a novel category



- Find interesting feature points
- Calculate Priors using auxiliary data categories
  - Which descriptors are most likely to help creating a good classifier
- Calculate Posterior from the Prior

- Great contribution - they created CALTECH-101 dataset for this paper

Source: Li, Fei-Fei, Rob Fergus, and Pietro Perona. "One-shot learning of object categories.“ (2006)

Adaptive SVM

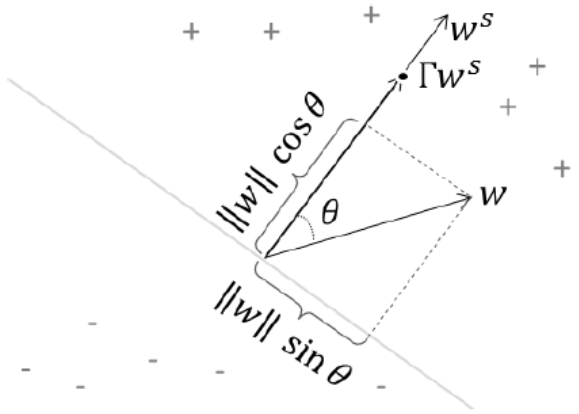
Find most similar category – w^s

Tweak SVM formula to include w^s:

$$L_A = \min_{w,b} \|w - \Gamma w^s\|^2 + C \sum_{i}^{N} l(x_i, y_i; w, b)$$

Where Γ is learning parameter (0-1) – how much knowledge you want to transfer

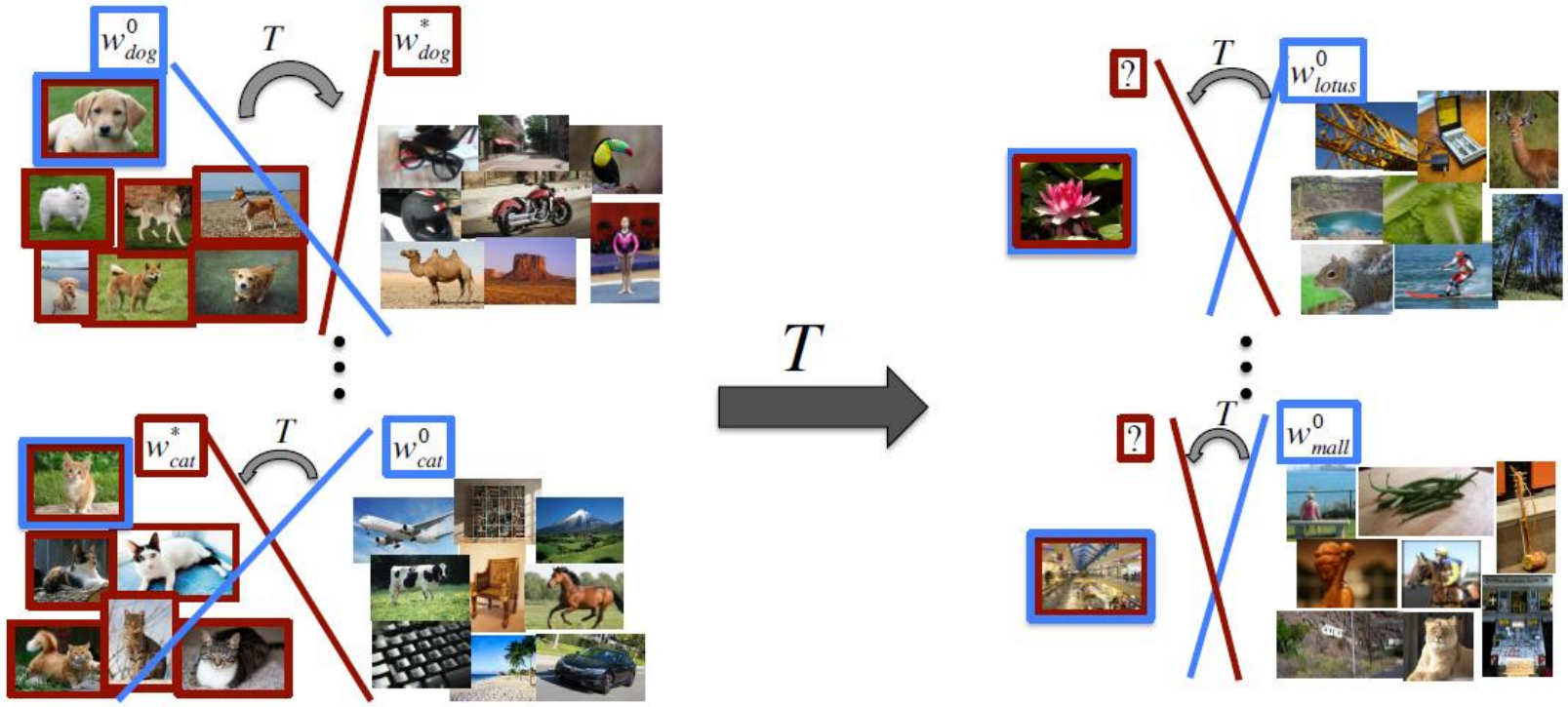
Drawback – all depends on 1 most similar class



Sources: Aytar, Yusuf, and Andrew Zisserman. "Tabula rasa: Model transfer for object category detection." *ICCV, 2011*.
Yang, Jun, Rong Yan, and Alexander G. Hauptmann. "Adapting SVM classifiers to data with shifted distributions." *ICDM, 2007.*

Model Regression Network

Model SVM boundary transformation - regression



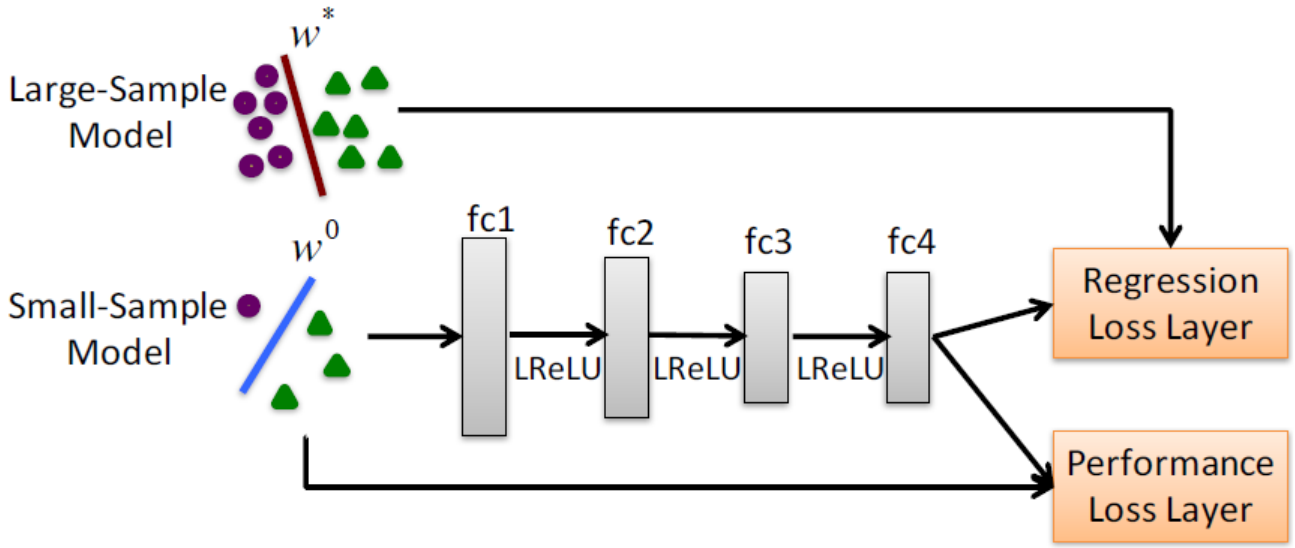
Blue – small sample model

Red – large sample model

Source: Wang, Yu-Xiong, and Martial Hebert. "Learning to learn: Model regression networks for easy small sample learning." *ECCV*, 2016.

Model Regression Network

Feed-forward neural network - to model regression



Drawback – better for medium-size training (5+ examples)

Source: Wang, Yu-Xiong, and Martial Hebert. "Learning to learn: Model regression networks for easy small sample learning." *ECCV*, 2016.

Our approach - general

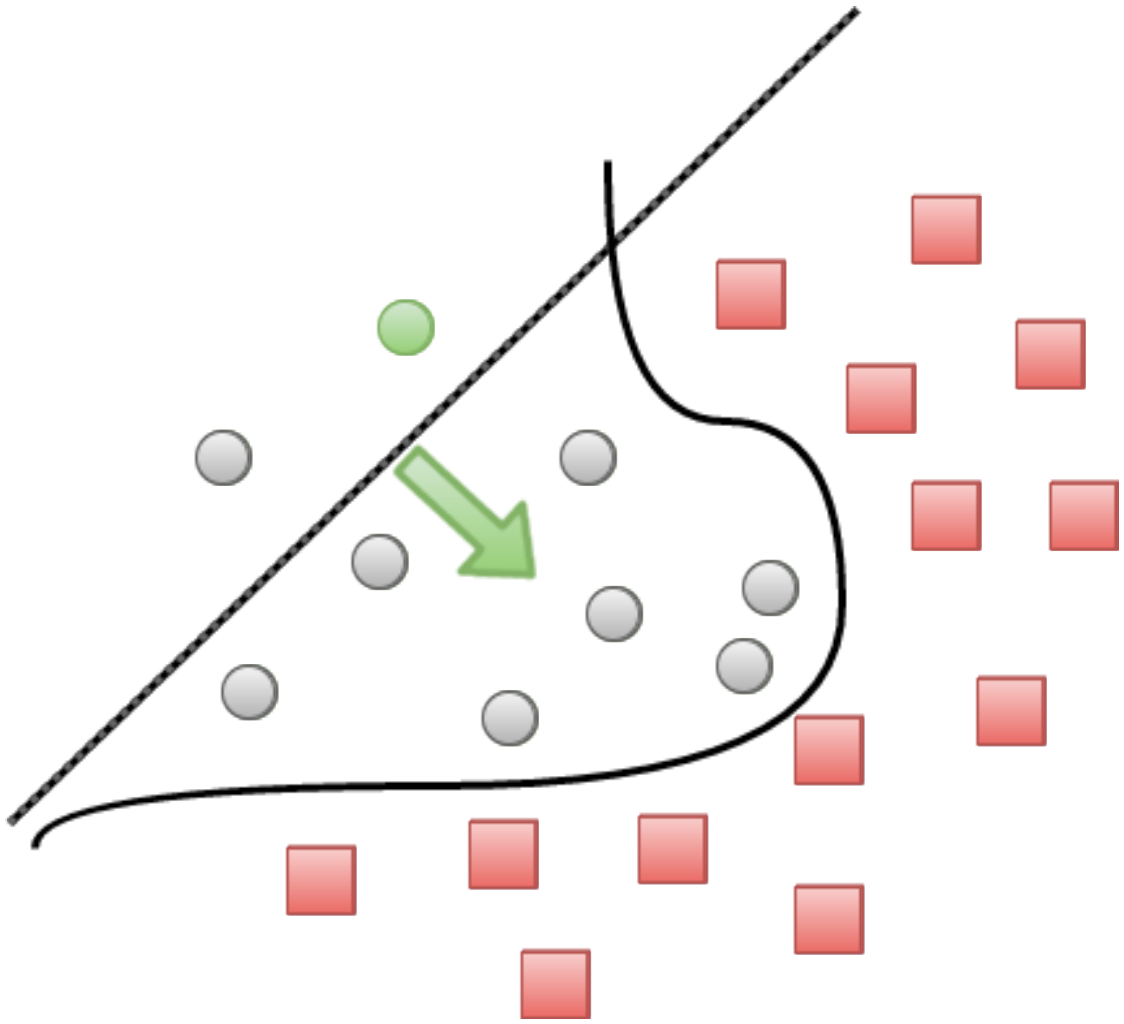
Our approach

Model how the decision boundary changes when you add positive training examples

Predict how your decision boundary should look like if you would have more examples.

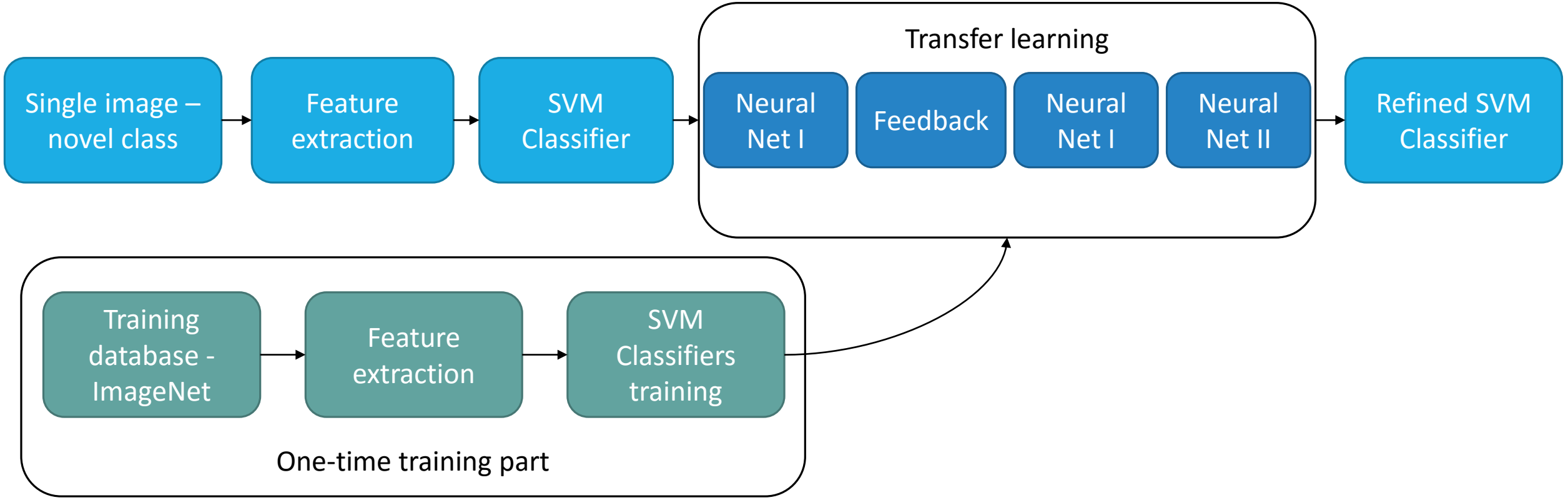
Assumption # 1 – Have only single training example

Assumption # 2 – Include user feedback



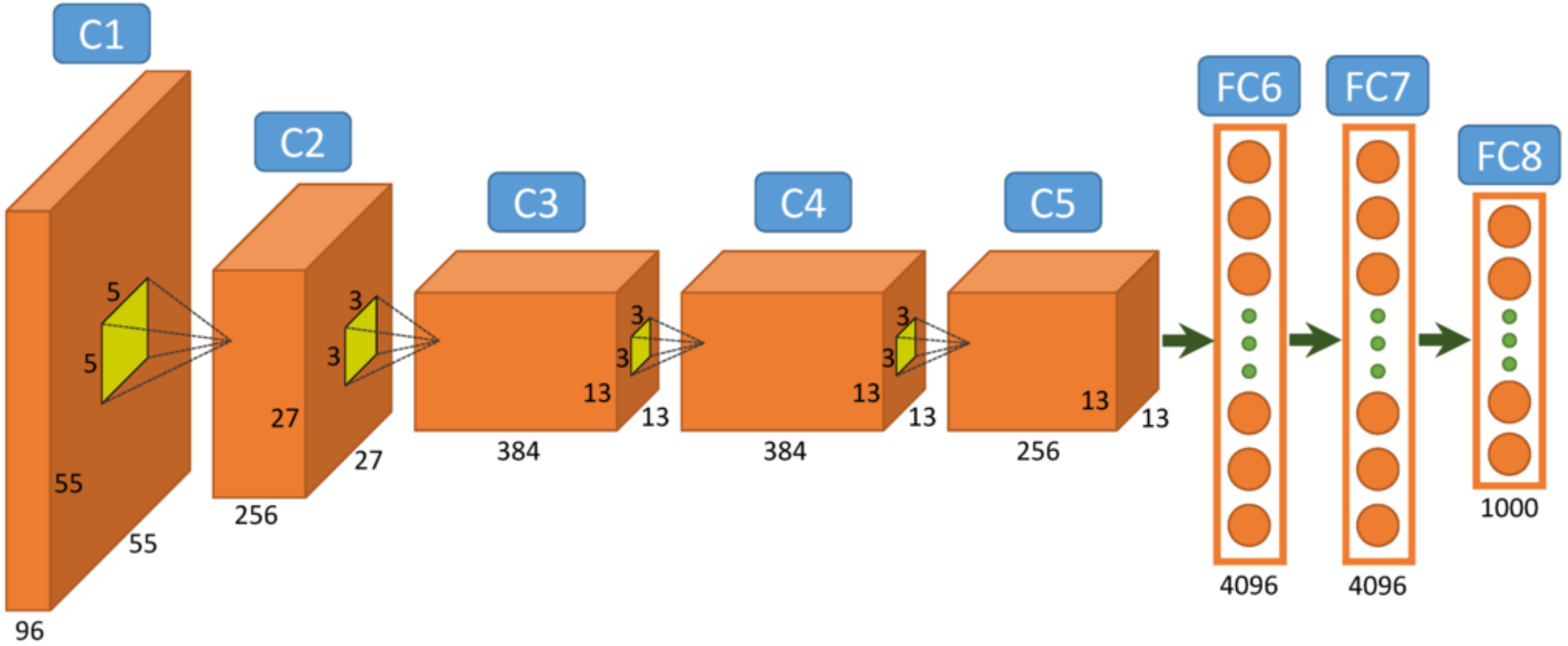
General idea

General overlook of the method



Feature extraction - AlexNet

Image is fed to layer C1 and our output is 4096 dimensional vectors of layer FC6



Source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *NIPS*, 2012.
Image source: saagie.com

Training part

We need a lot of examples to learn – use ImageNet:

~1.2 million images

1000 object categories

Labeled data

Feature extraction step gives us ~1.2 million feature vectors.

Train large number of SVM classifiers (~700k) and use that to train two neural networks that will model this transition

Source: Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision* 115.3 (2015)

Training part

We need 3 different kinds of SVM classifiers for training:

Single sample model

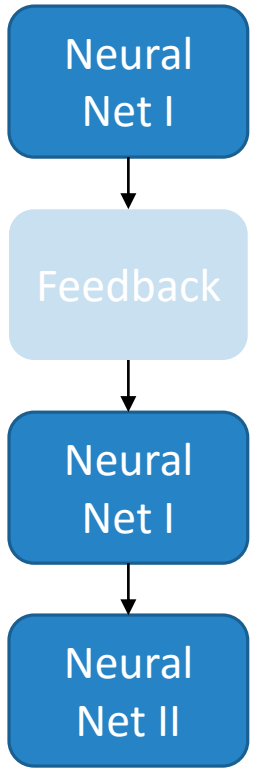
- 1 positive example
- Represent future inputs

Small sample model

- 5-10 positive examples
- Represents middle step that allows to refine the method

Full data model

- ~1000 positive examples
- Represents ideal classifier given sufficient data

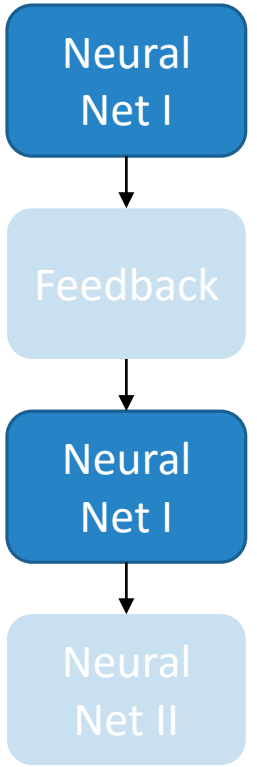


Our approach – details

Neural Network I

Training parameters:

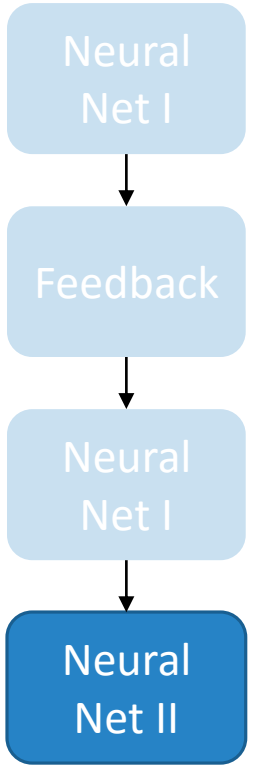
- For each of 1000 categories:
  - Subsample 600 images
  - 600 pairs:
    - SVM from one sample (input) - SVM from all samples (target)
  - Split it:
    - 590 for training, 10 for testing
- So:
  - 590k training examples
  - 10k testing examples



Neural Network II

Training parameters:

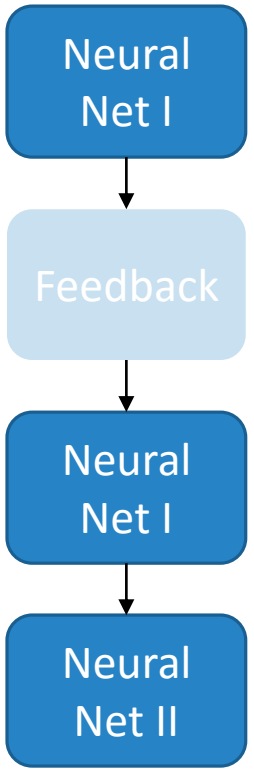
- For each of 1000 categories:
  - Create classifiers:
    - Using {1, 2, 3, …, 9, 10, 15, 20, …, 100} samples (subsampling)
    - Using 5 different SVM regularization parameters {0.01, 0.1, 1, 10, 100}
    - Repeating 5 times for each combination
  - 700 pairs:
    - subsampled SVM (input) - SVM from all samples (target)
  - Split it:
    - 685 for training, 15 for testing
- So:
  - 685k training examples
  - 15k testing examples



Source: Wang, Yu-Xiong, and Martial Hebert. "Learning to learn: Model regression networks for easy small sample learning." *ECCV*, 2016.

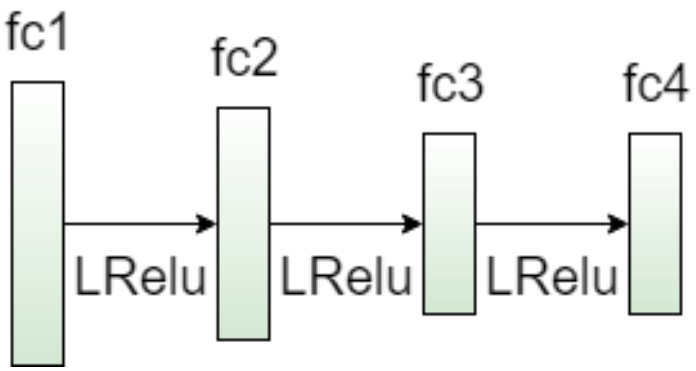
Dimensionality Reduction

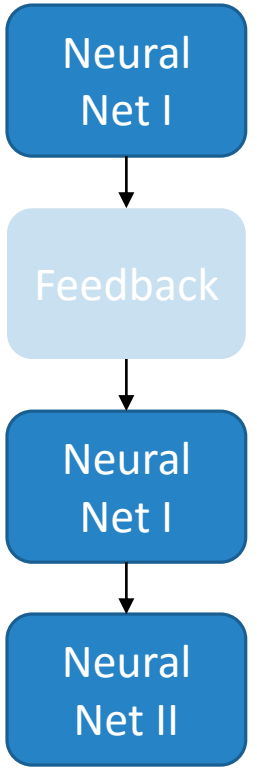
- Problems training high-dimensional network (~ 93 million weights)
- Decided to lower the dimensionality to improve chances of successful training
- Linear Discriminant Analysis
  - 4096 → 500 dimensional feature vectors
  - New network with ~1.3 million weights



Networks architecture

- Both networks have the same architecture
- Layers dimensions:
  - 750, 625, 501, 501
  - Fully connected layers
- leaky ReLU units with parameter 0.1





Relevance Feedback

Using your SVM classifier – run it on your unlabeled testing set

Retrieve top K positive images found

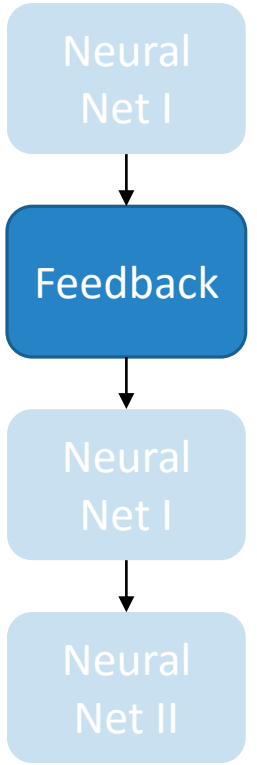
Possible to retrieve also top K negative images found

Automated feedback:

- Assume algorithm was right without asking user
- Add those found images to training part of next SVM classifier

User feedback:

- Show user retrieved images along with assumed label. Ask if it's right
- Add those found images to training part of next SVM classifier



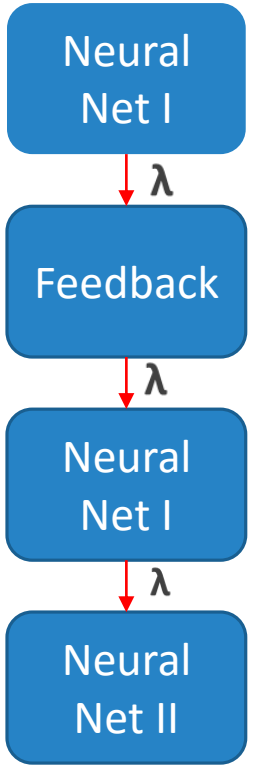
Smoothened DMT

- During testing we discovered that:
  - Relevance Feedback increases **precision** significantly (very low **recall**)
  - DMT increases slowly **recall** (**precision** slowly decreases)

- Introduce new learning parameter **λ**:

$$Hyperplane_{new} = \lambda * Hyperplane_{new} + (1 - \lambda) * Hyperplane_{old}$$

- Use λ-smoothing between every step of DMT

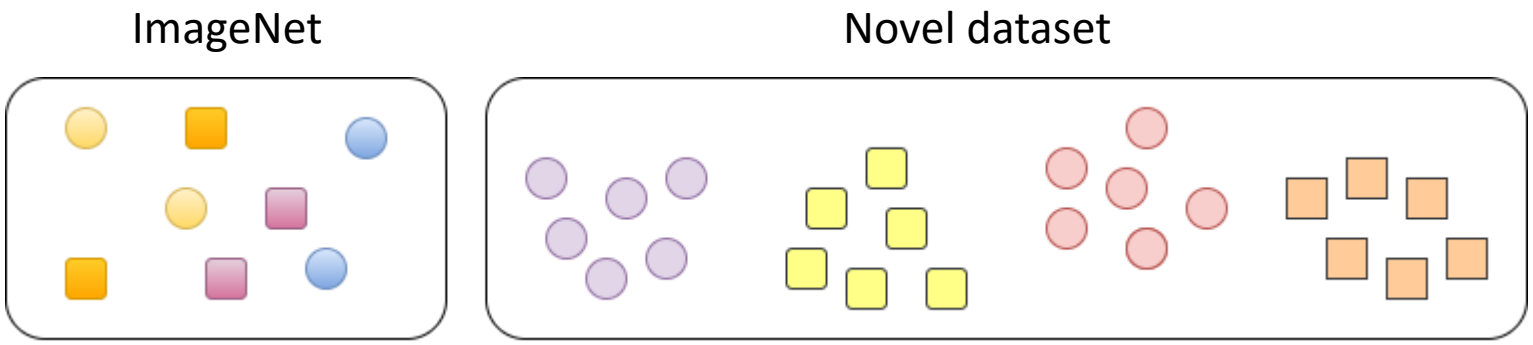


How to create single sample SVM

Two approaches:

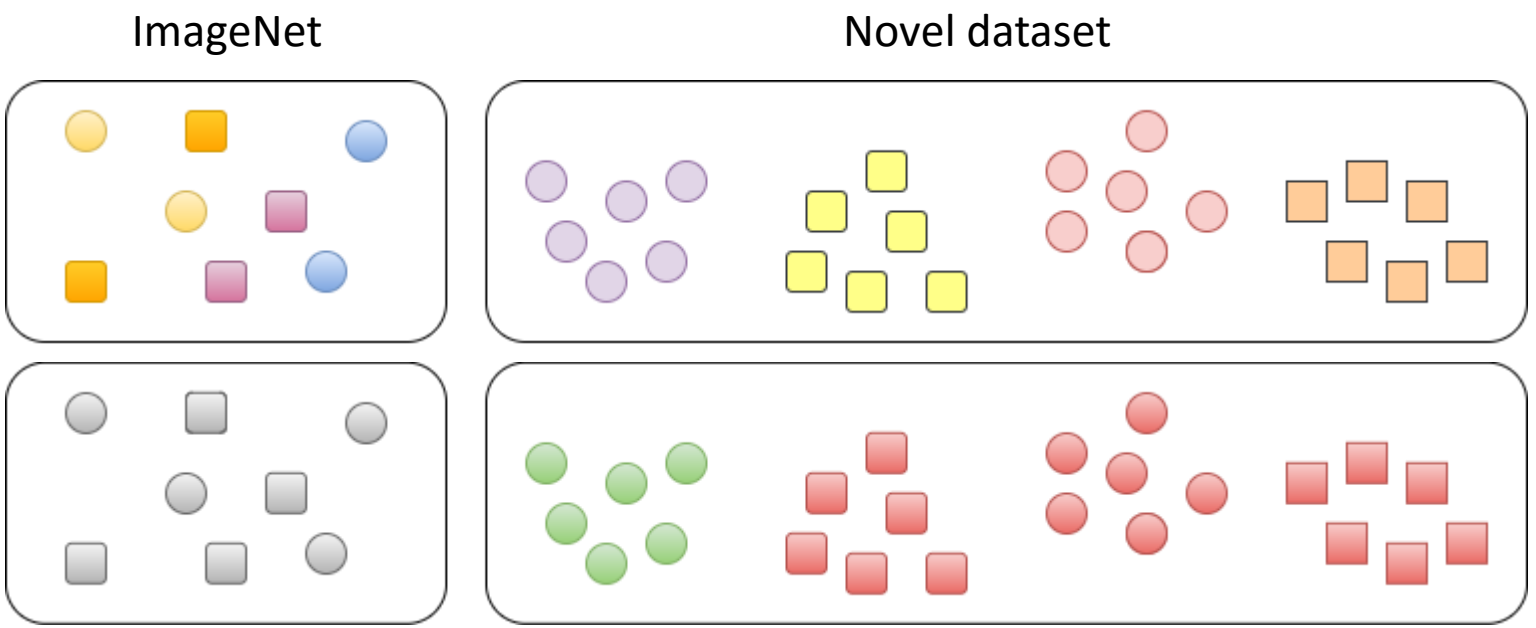
- Zero Knowledge method
  - Dataset completely unlabeled
  - Negative examples from ImageNet
- Next Category method
  - Dataset partially labeled (e.g. when increasing your dataset, adding new category)
  - Negative examples from this dataset

Zero Knowledge method



Initial configuration

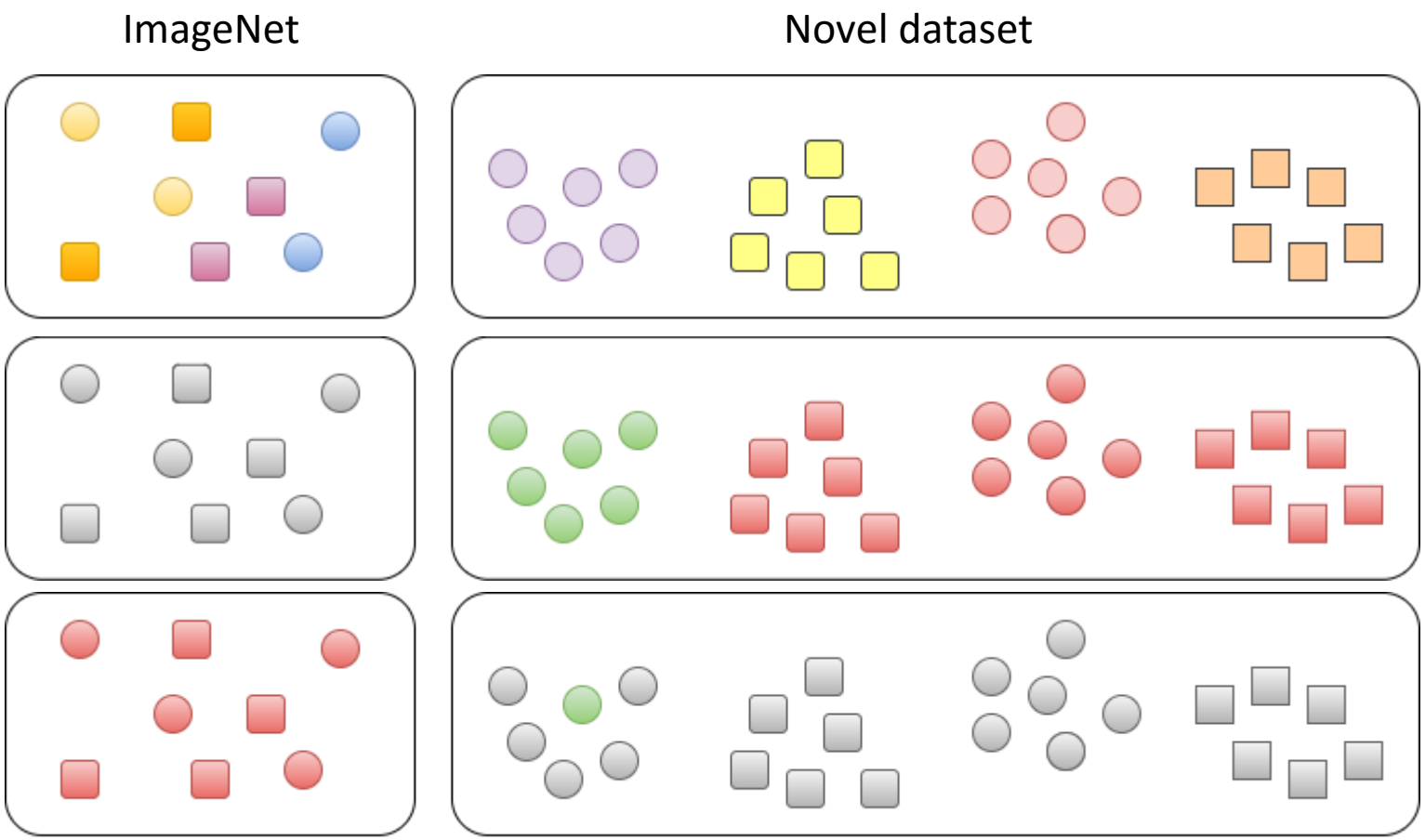
Zero Knowledge method



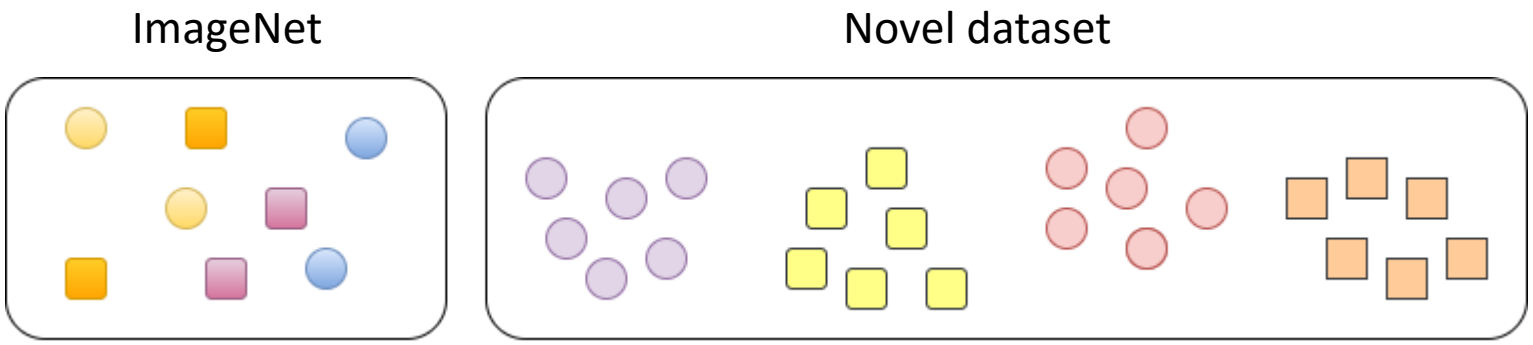
Initial configuration

Full data SVM training

Zero Knowledge method

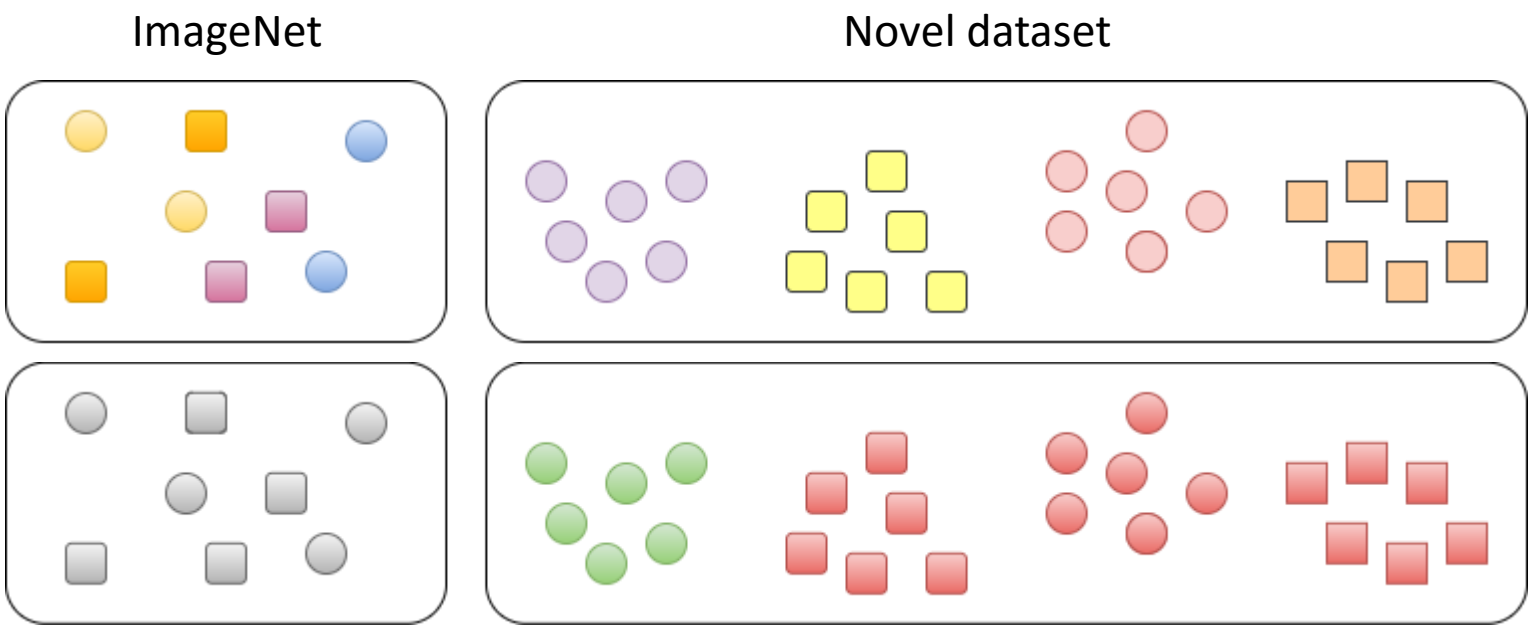


Adding Next Category method

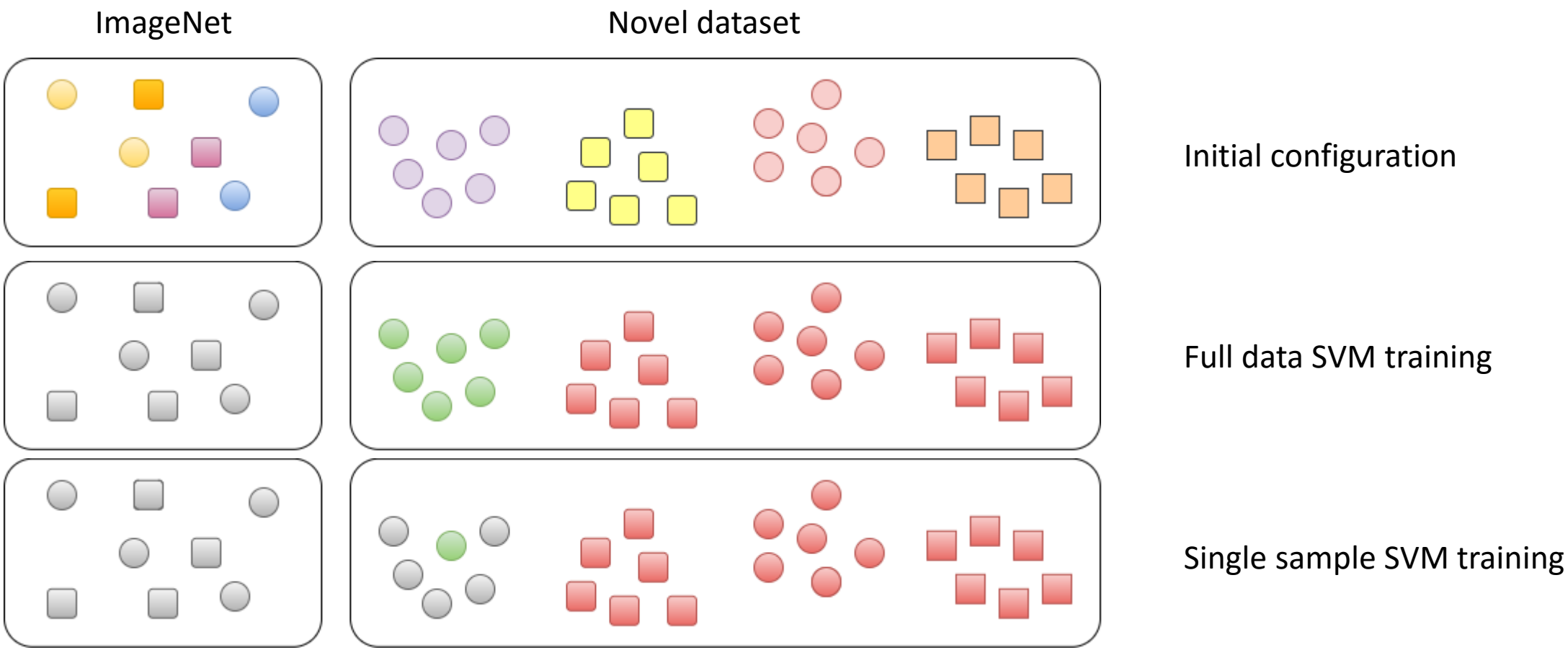


Initial configuration

Adding Next Category method



Adding Next Category method



Results

Results – visual domain

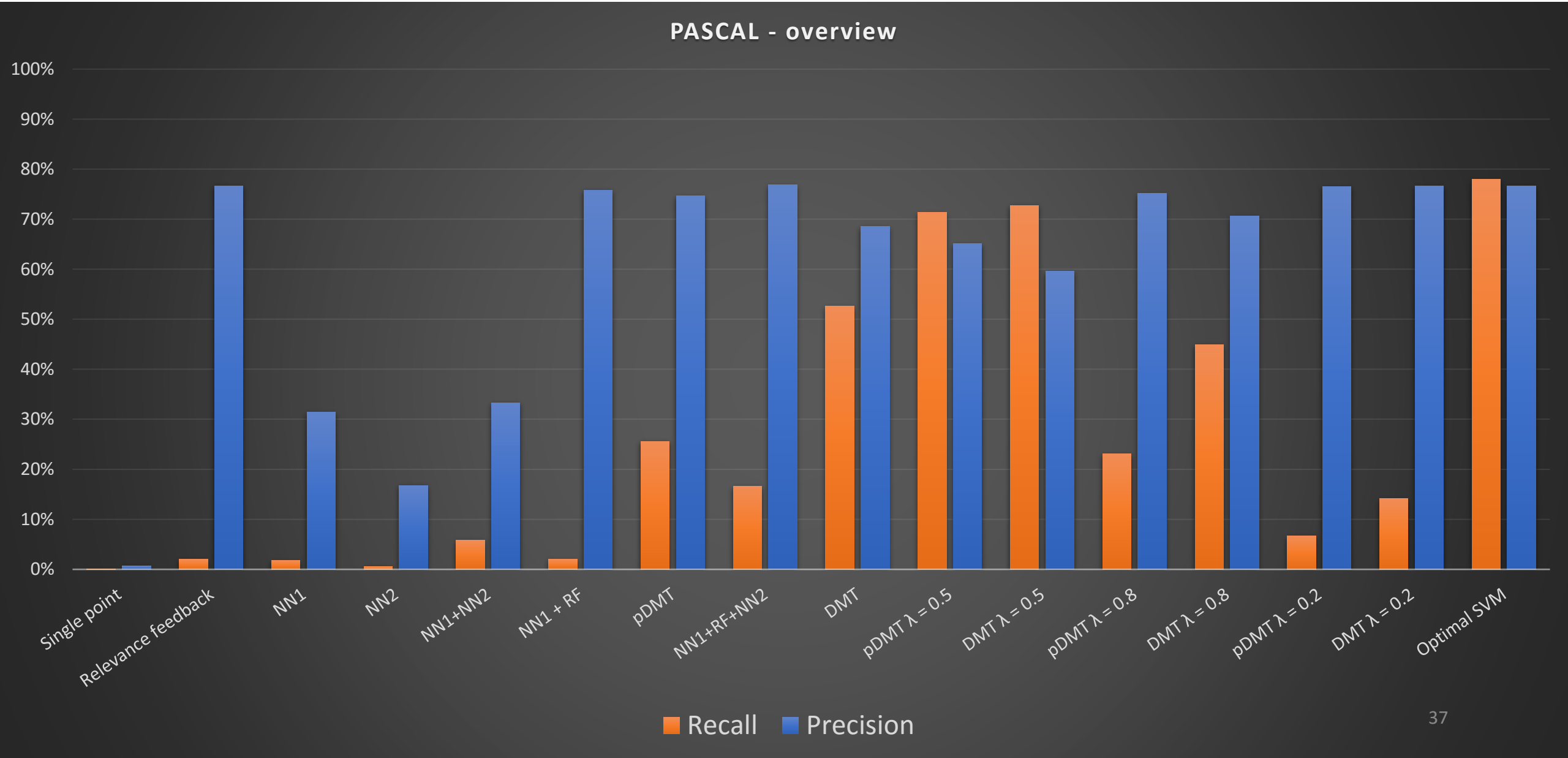
We tested our approach on 3 benchmark datasets:

- PASCAL VOC 2007
  - 20 categories
  - 5011 images (train + val)
- CALTECH 256
  - 256 categories
  - 30607 images
- CALTECH-UCSD Birds 200
  - 200 categories
  - 6,033 images

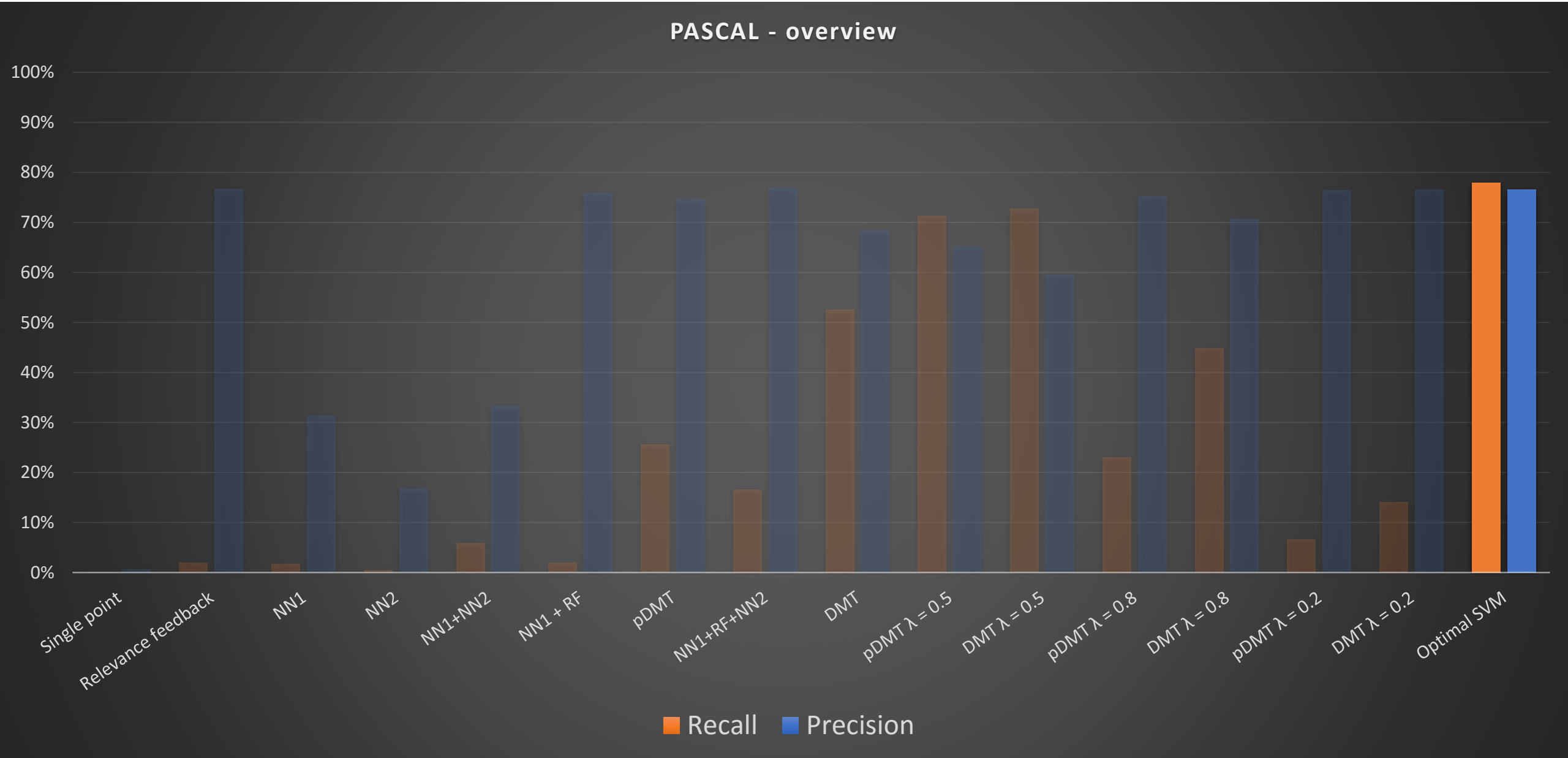
Split every category: 50% of images for training, 50% for testing

Single sample SVM

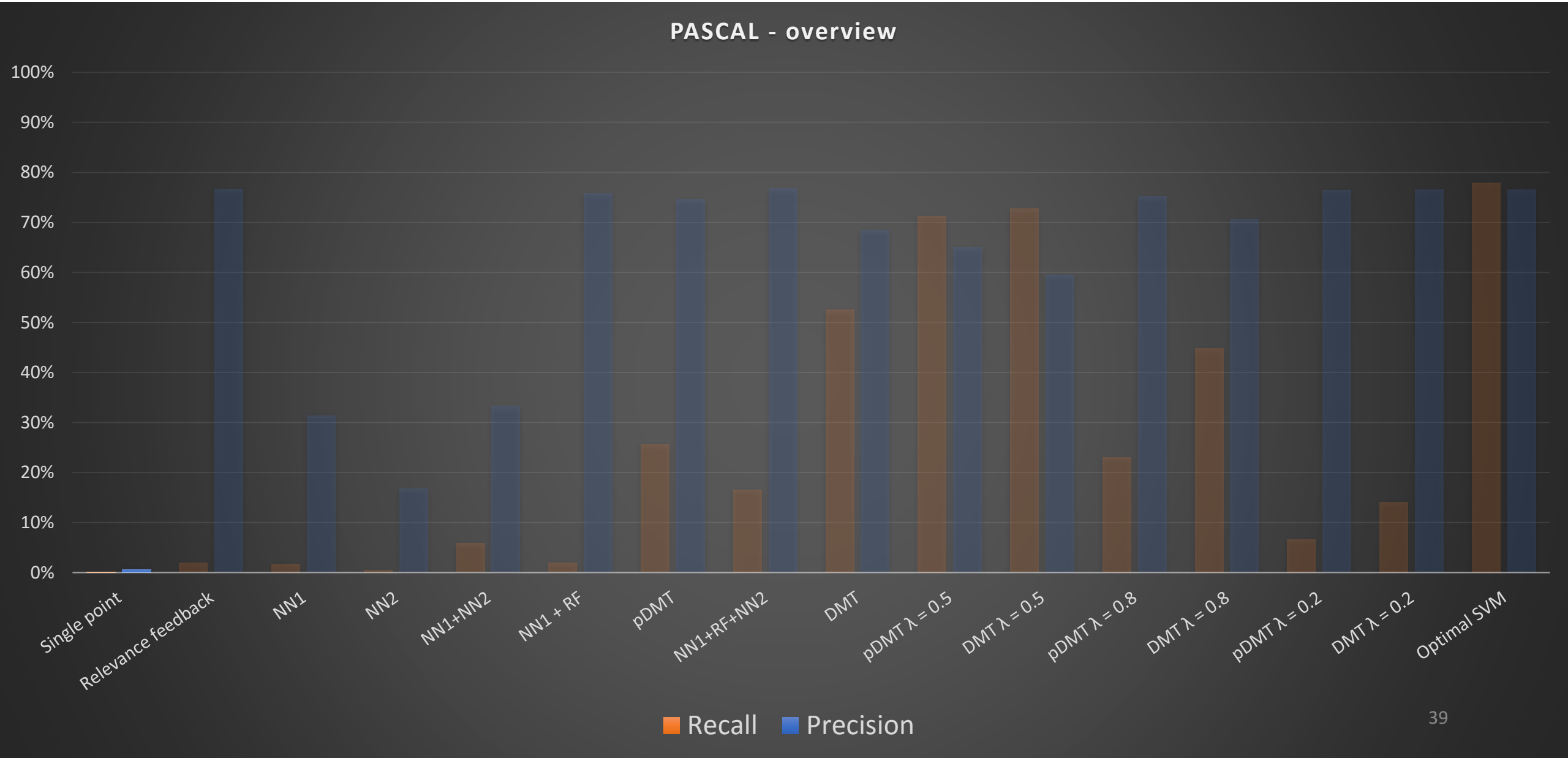
Refined SVM



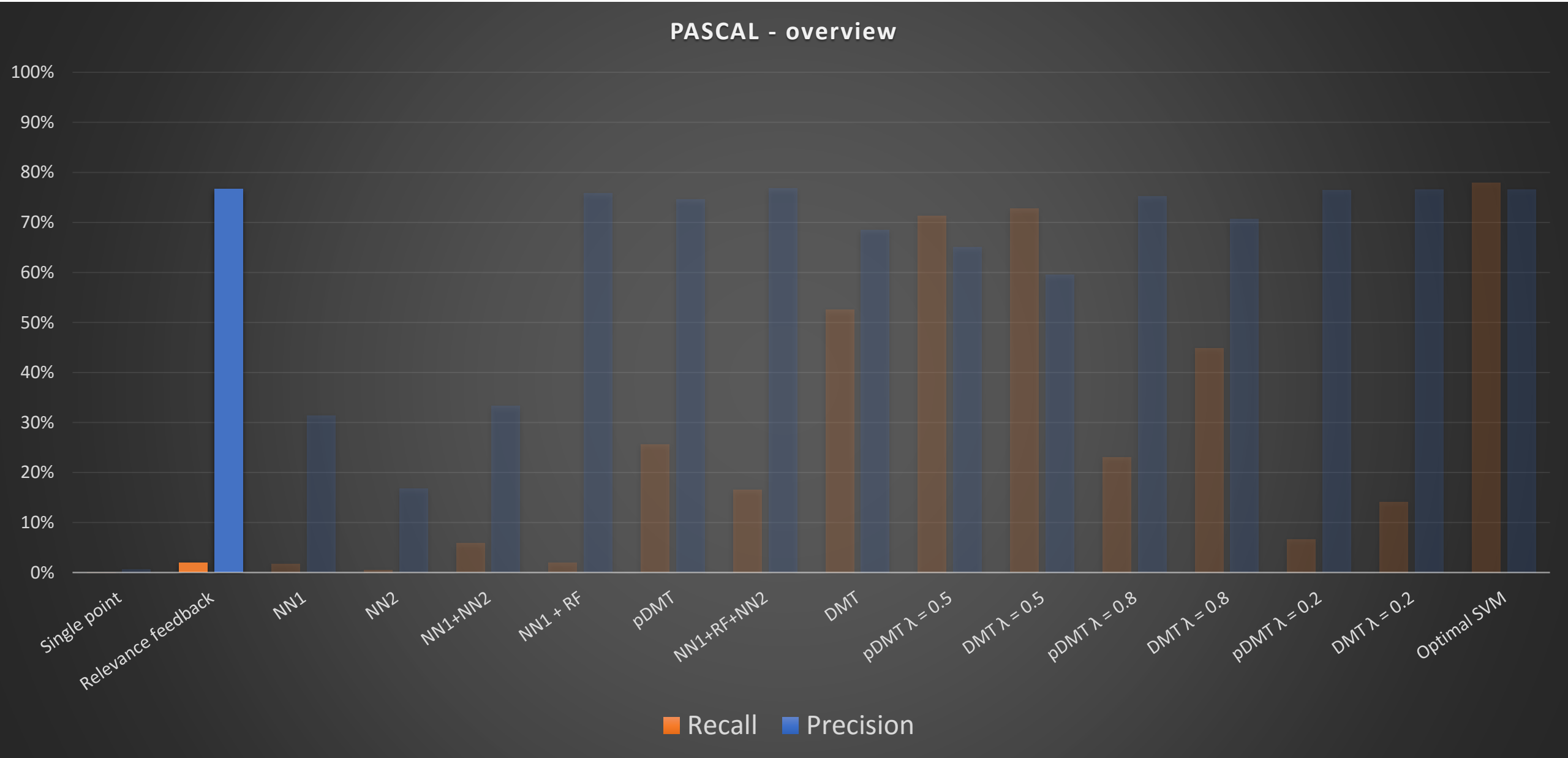
Full data SVM



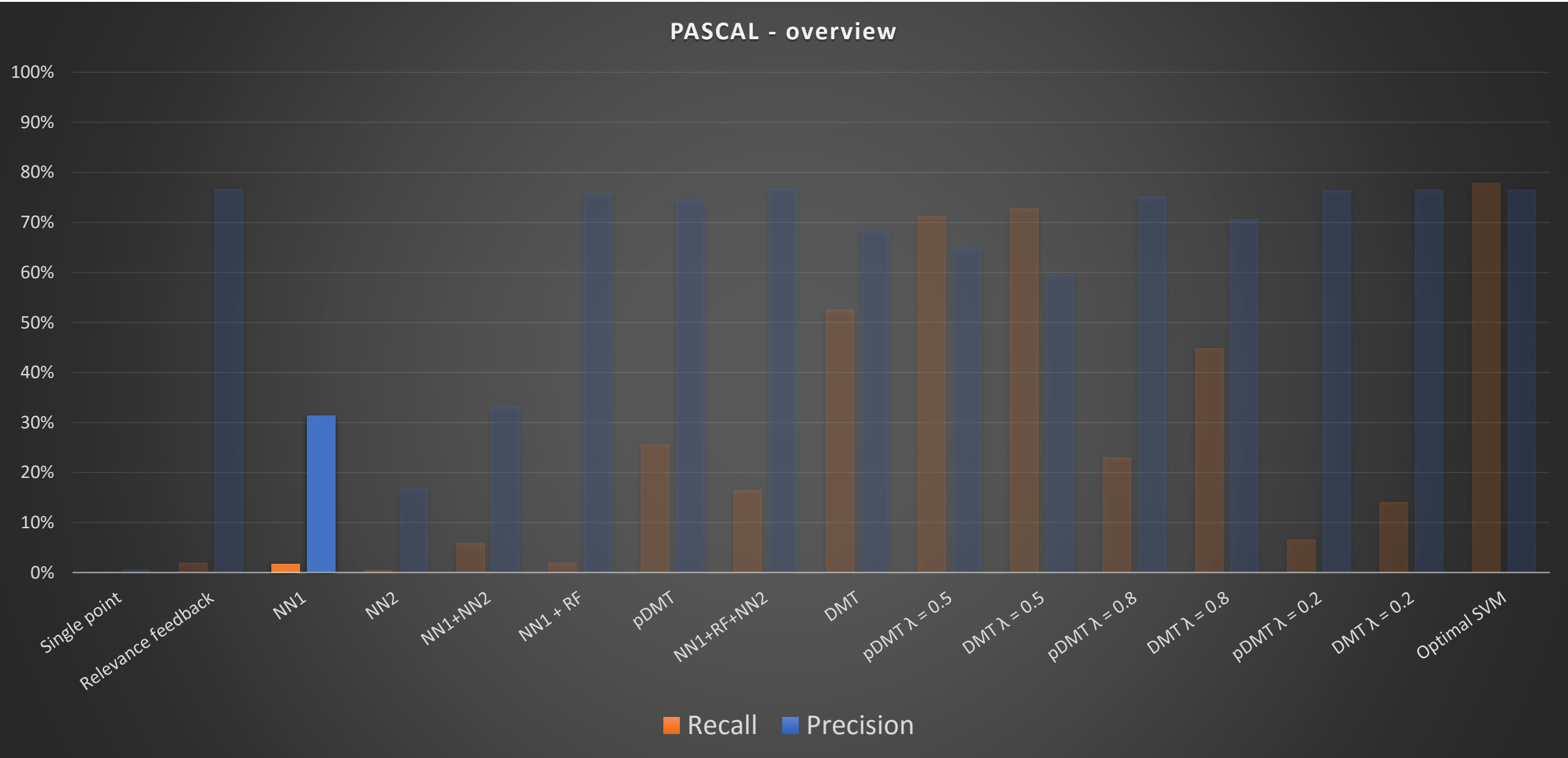
Single sample SVM → Refined SVM



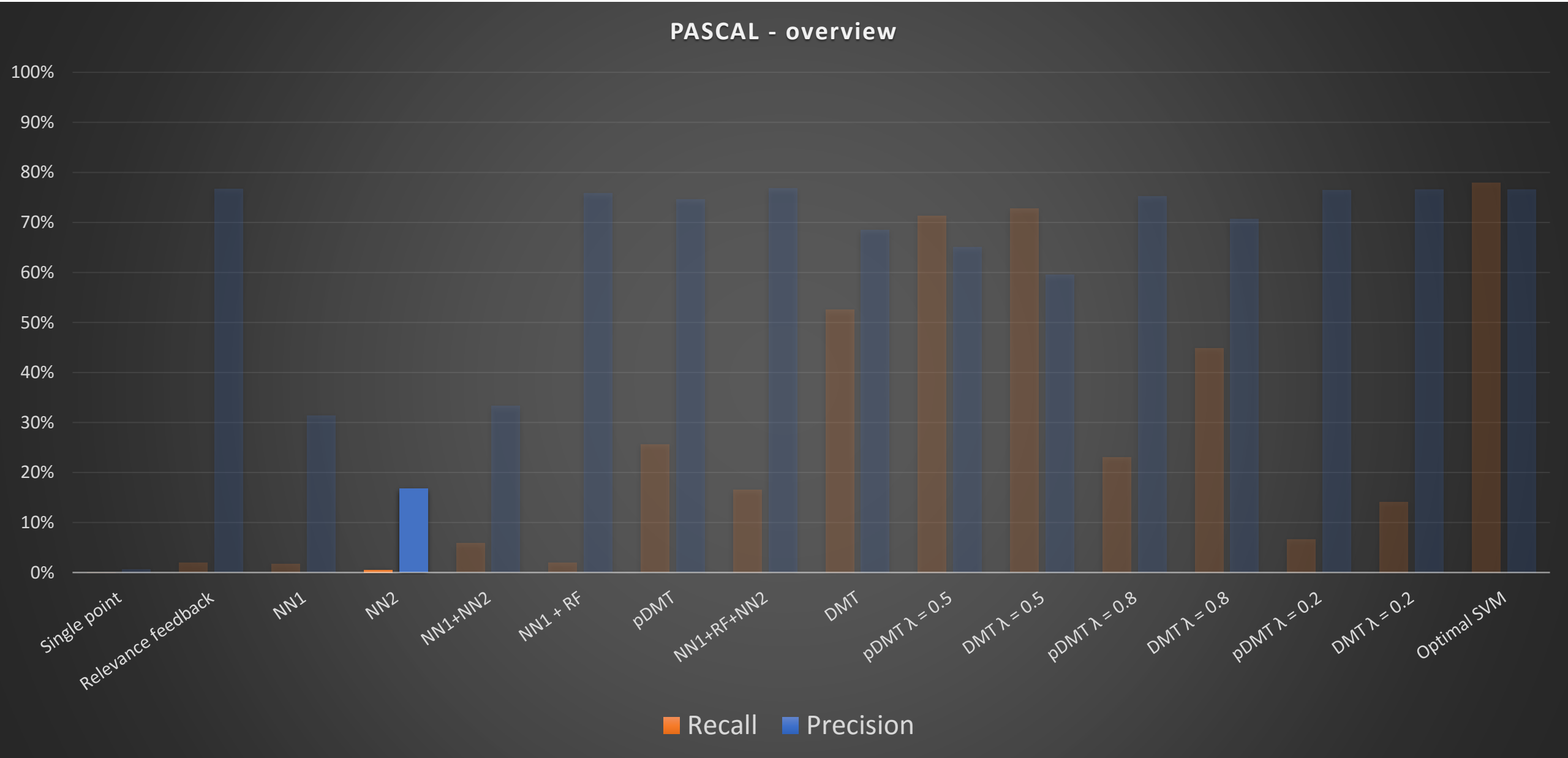
Single sample SVM → Feedback → Refined SVM

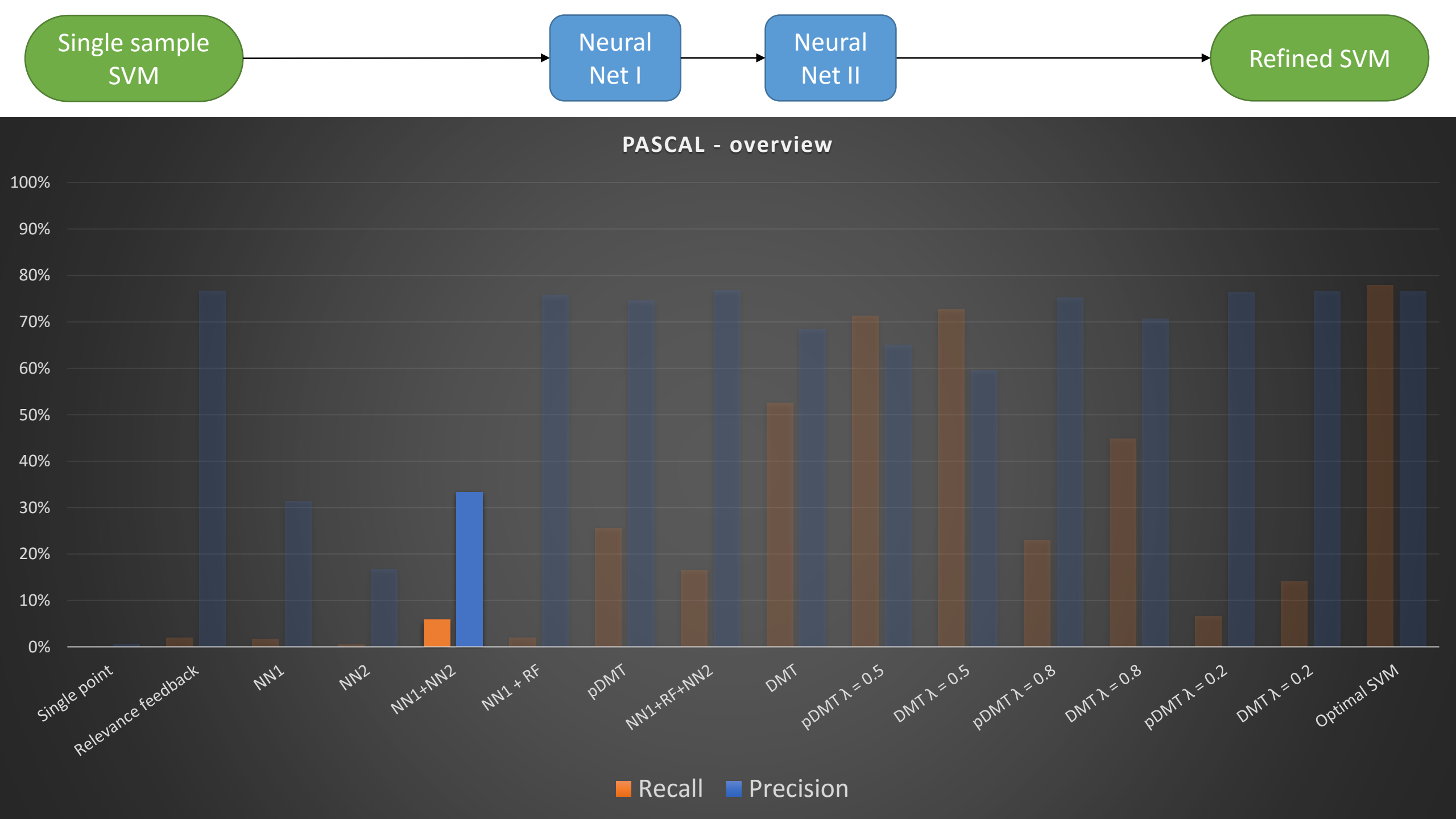


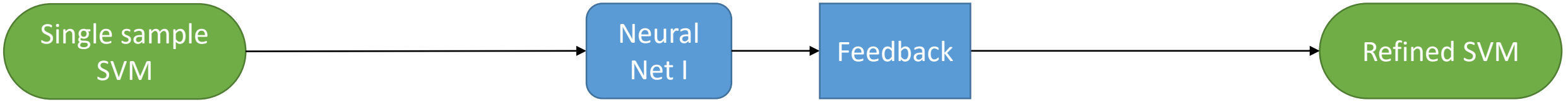
Single sample SVM → Neural Net I → Refined SVM

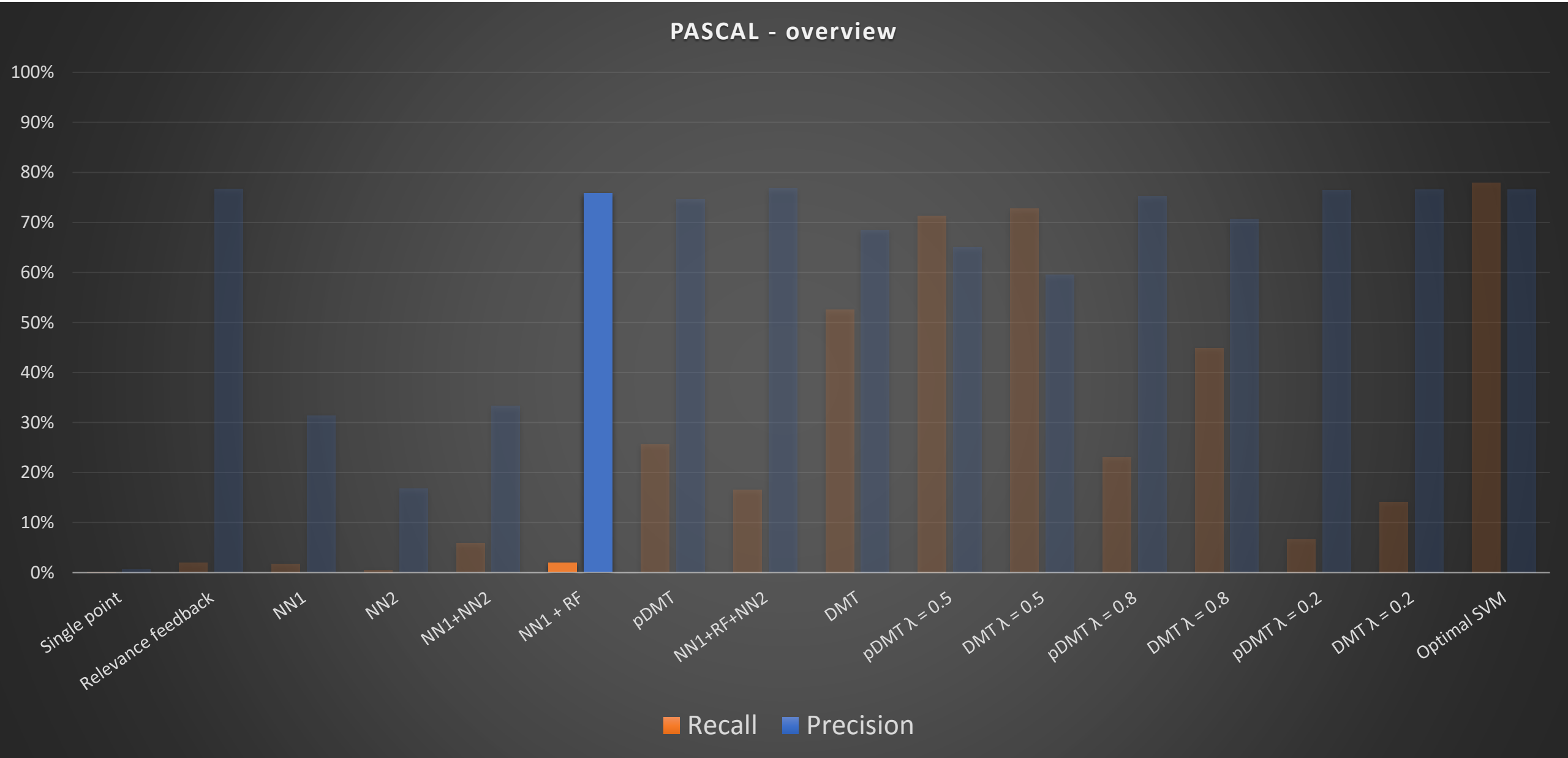


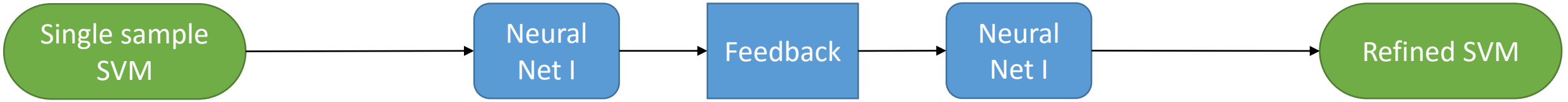
Single sample SVM → Neural Net II → Refined SVM

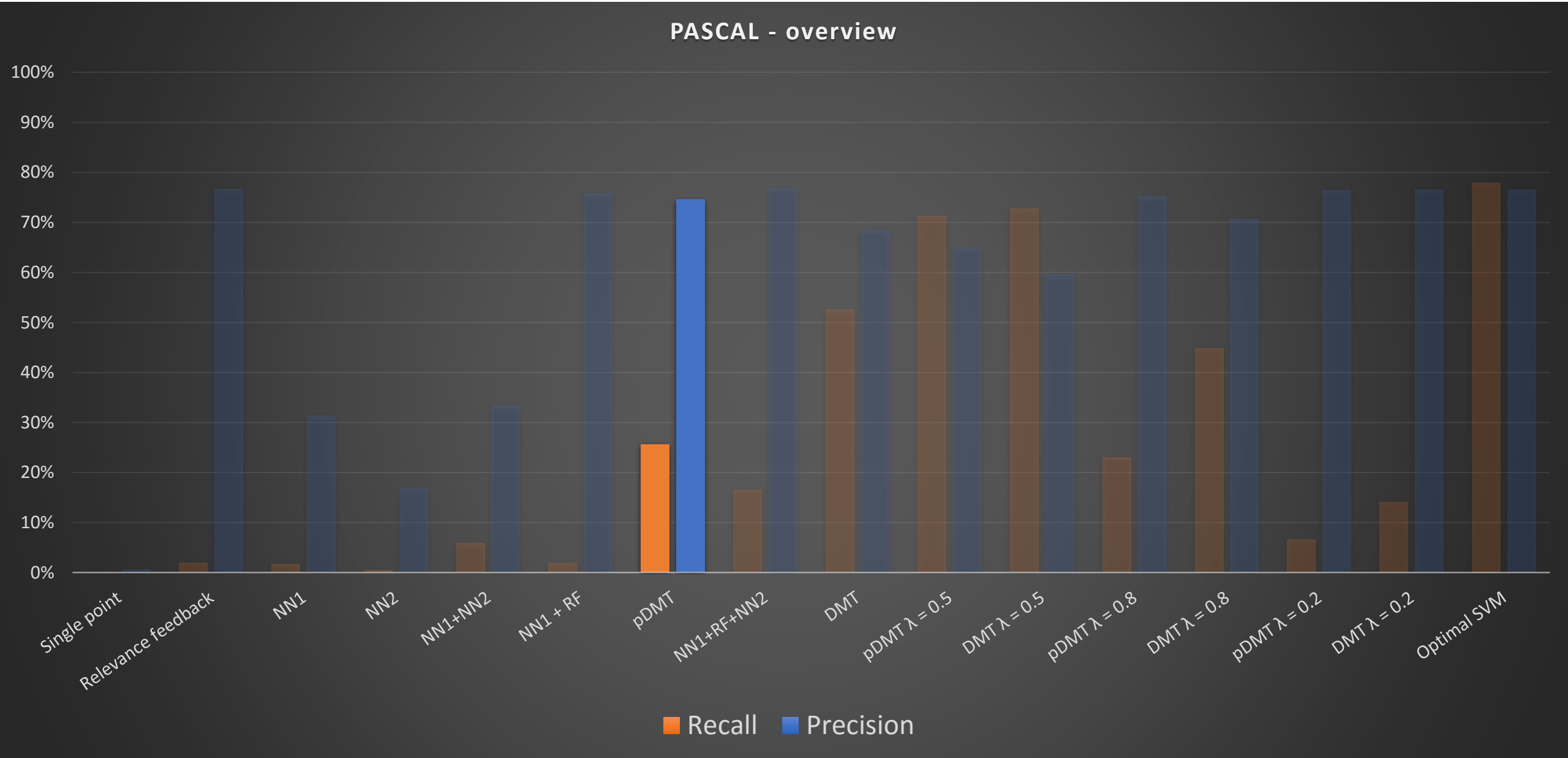


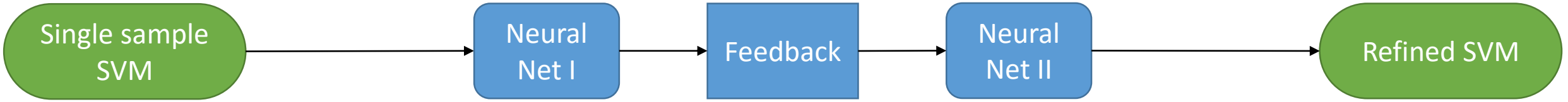

Single sample SVM
Neural Net I
Neural Net II
Refined SVM
PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

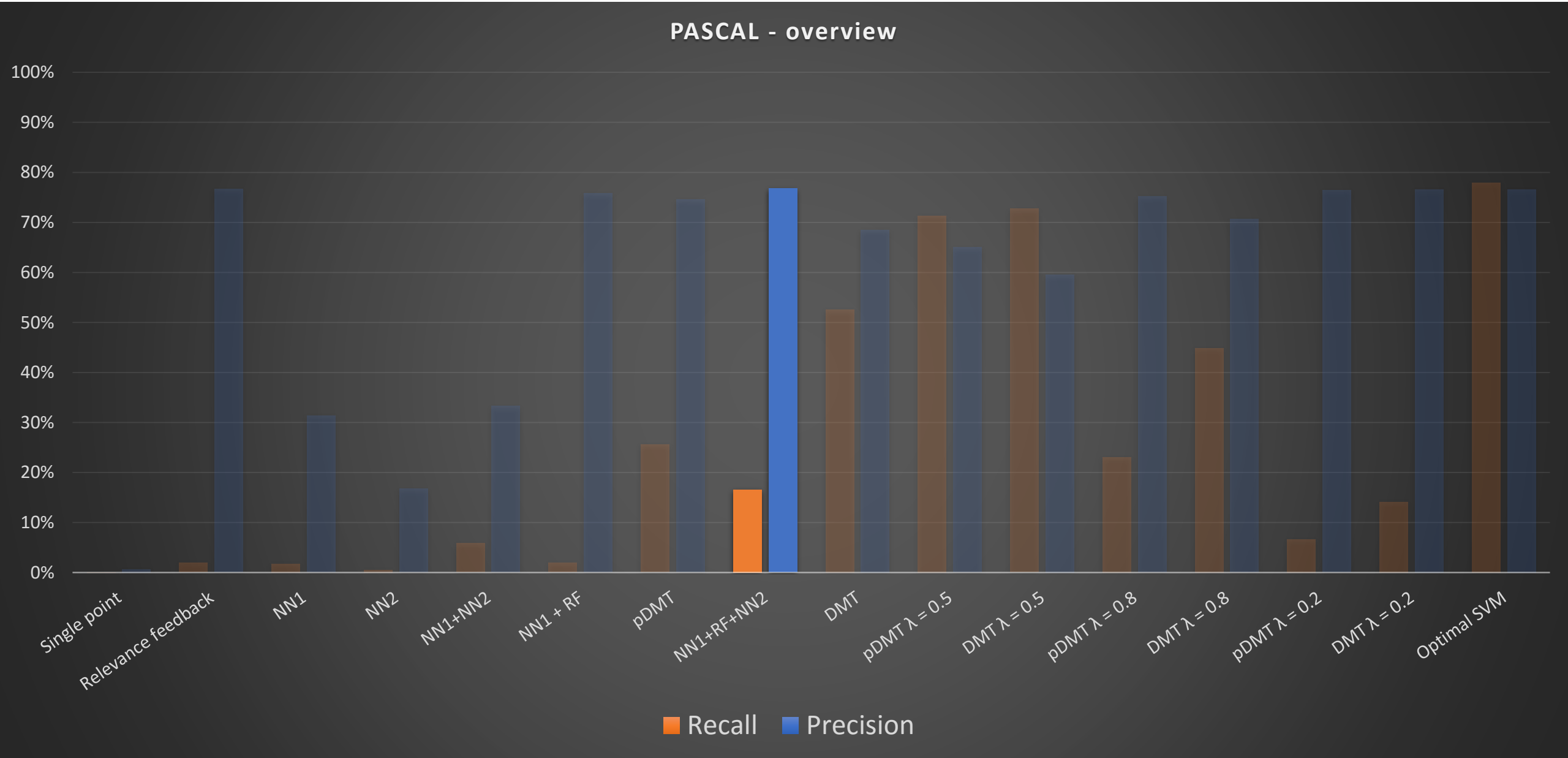

Single sample SVM
Neural Net I
Feedback
Refined SVM

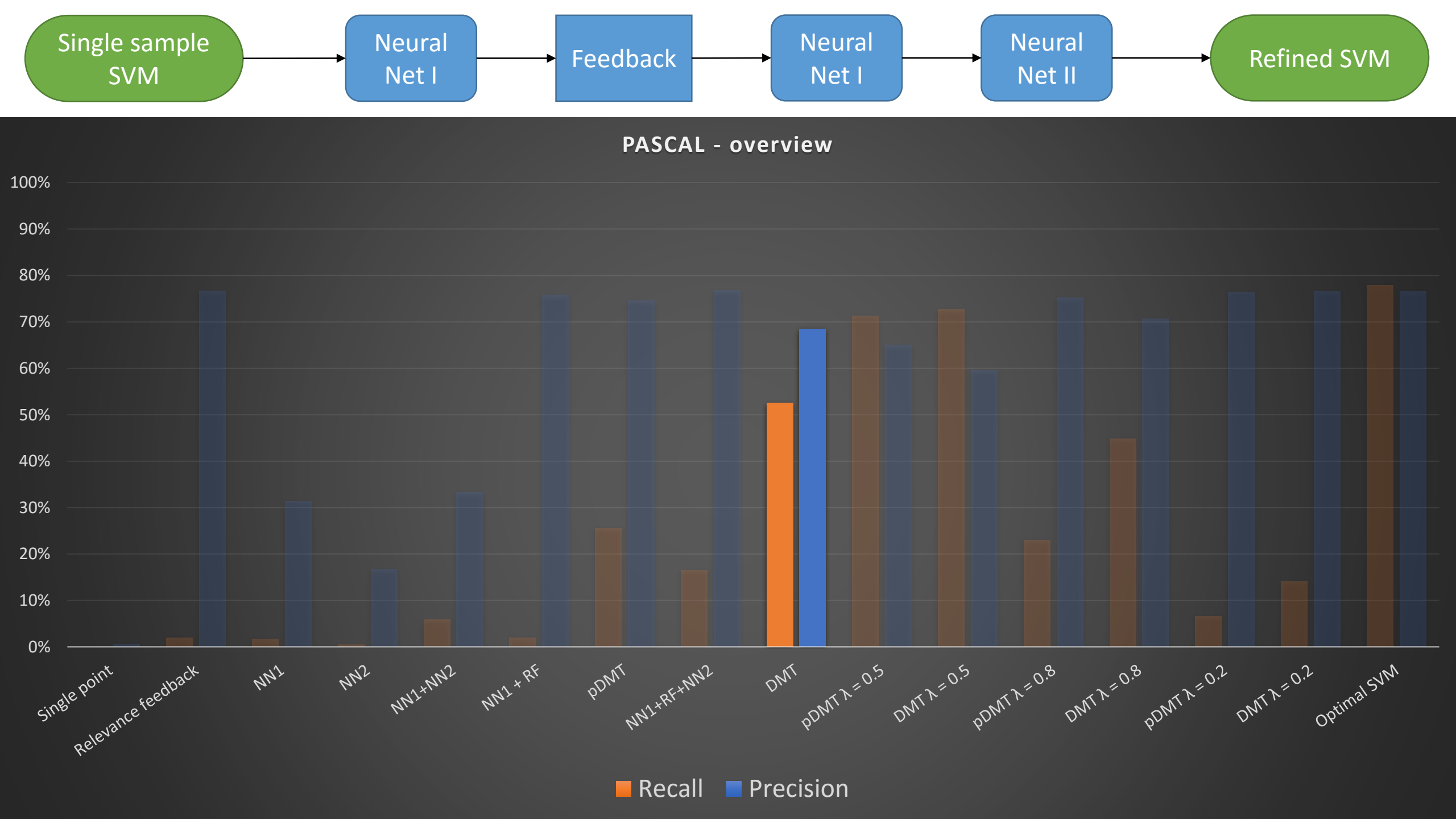

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

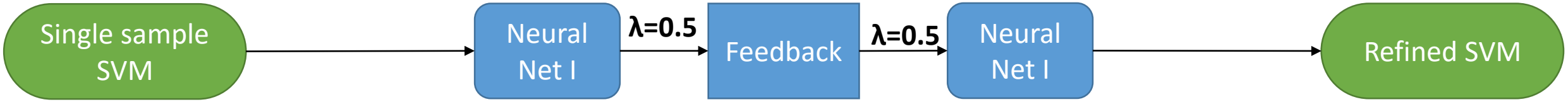

Single sample SVM
Neural Net I
Feedback
Neural Net I
Refined SVM

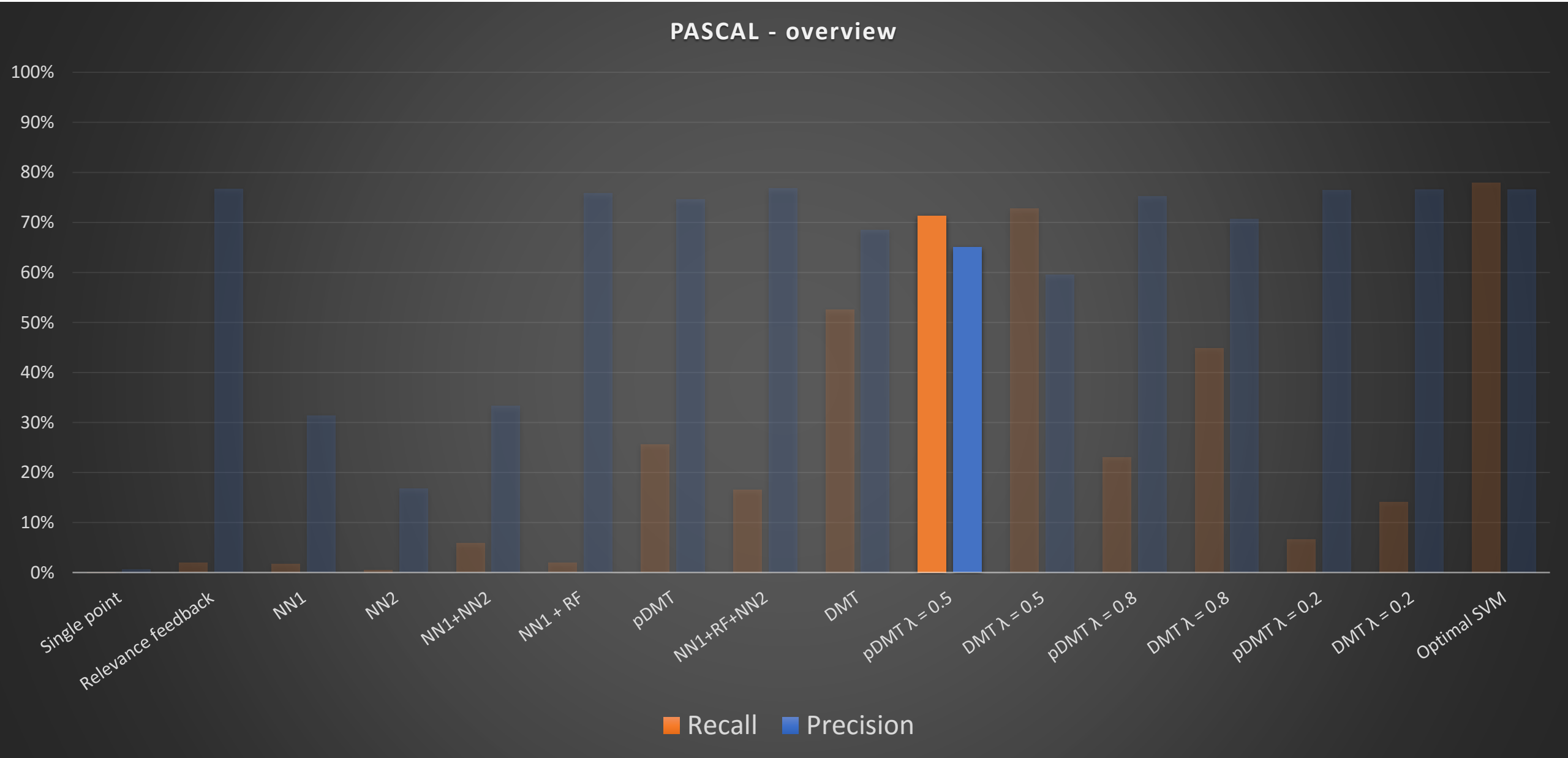

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

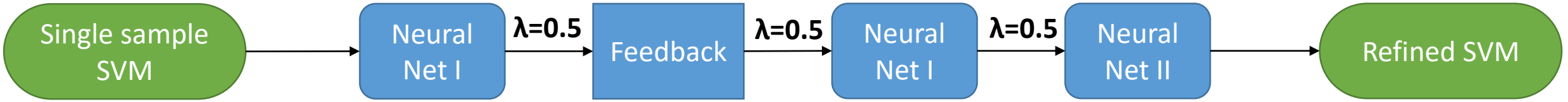

Single sample SVM
Neural Net I
Feedback
Neural Net II
Refined SVM

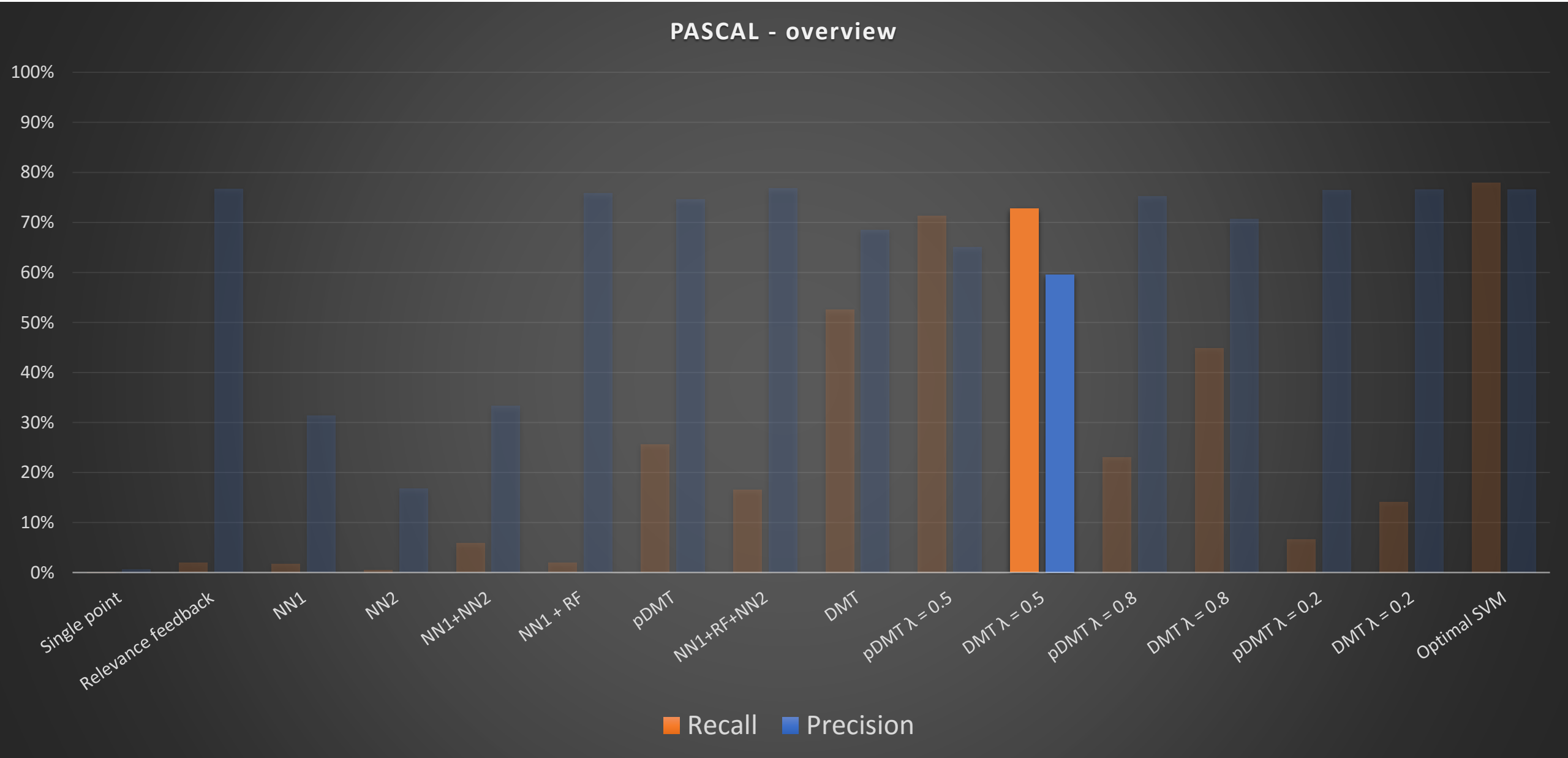

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

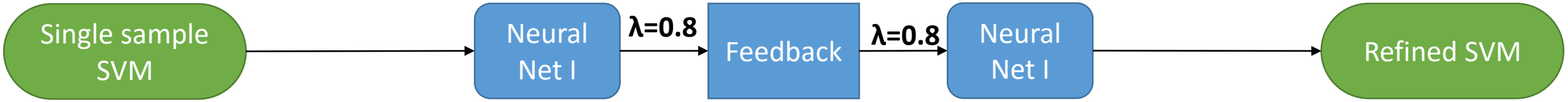

Single sample SVM
Neural Net I
Feedback
Neural Net I
Neural Net II
Refined SVM
PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

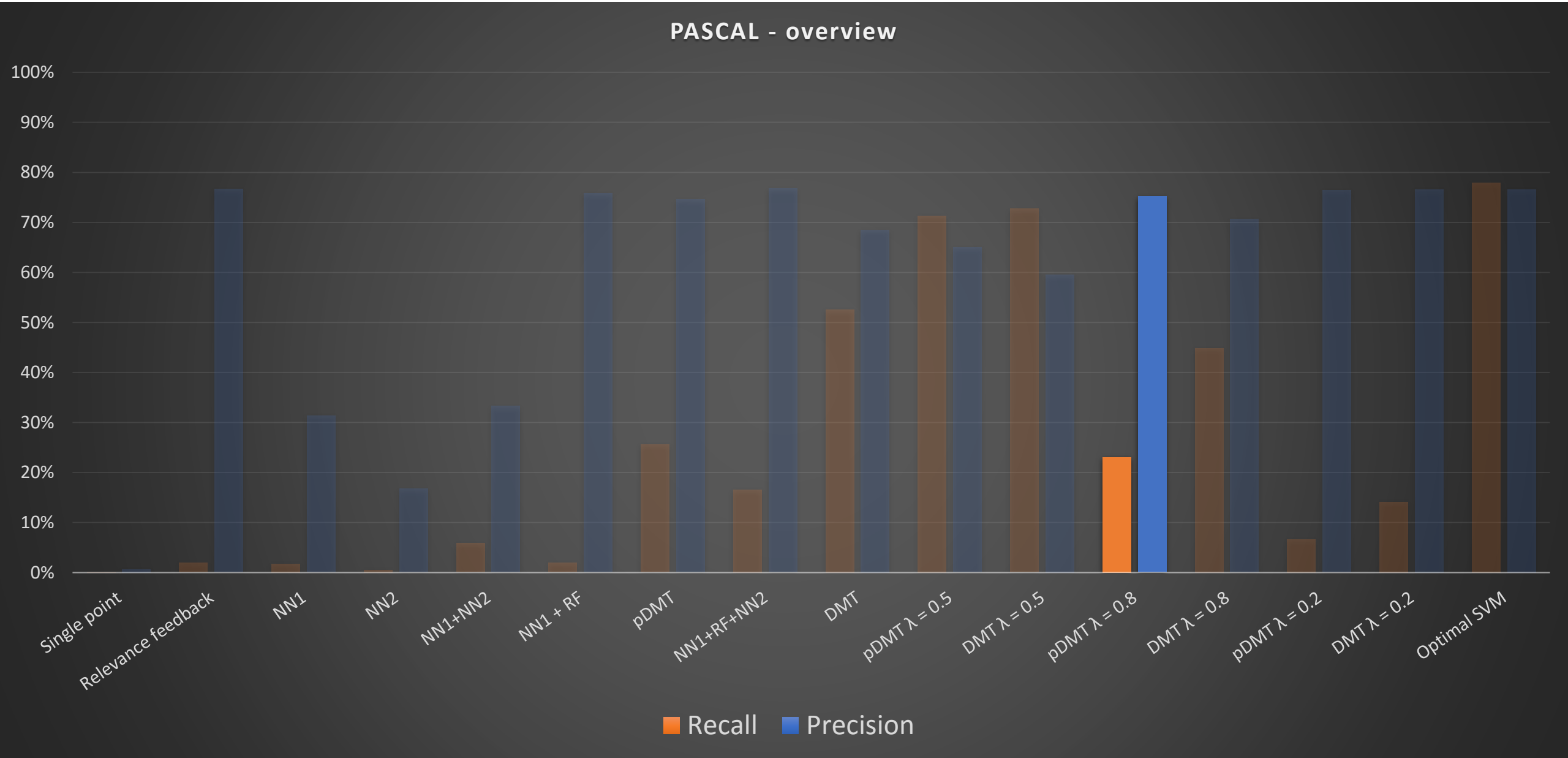

Single sample SVM
Neural Net I
λ=0.5
Feedback
λ=0.5
Neural Net I
Refined SVM

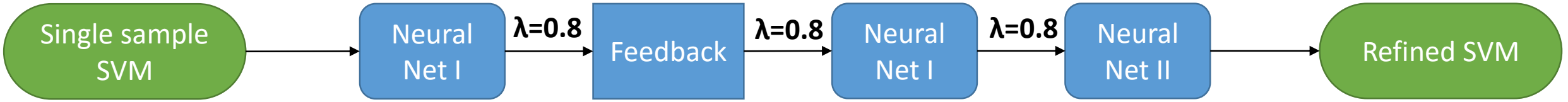

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

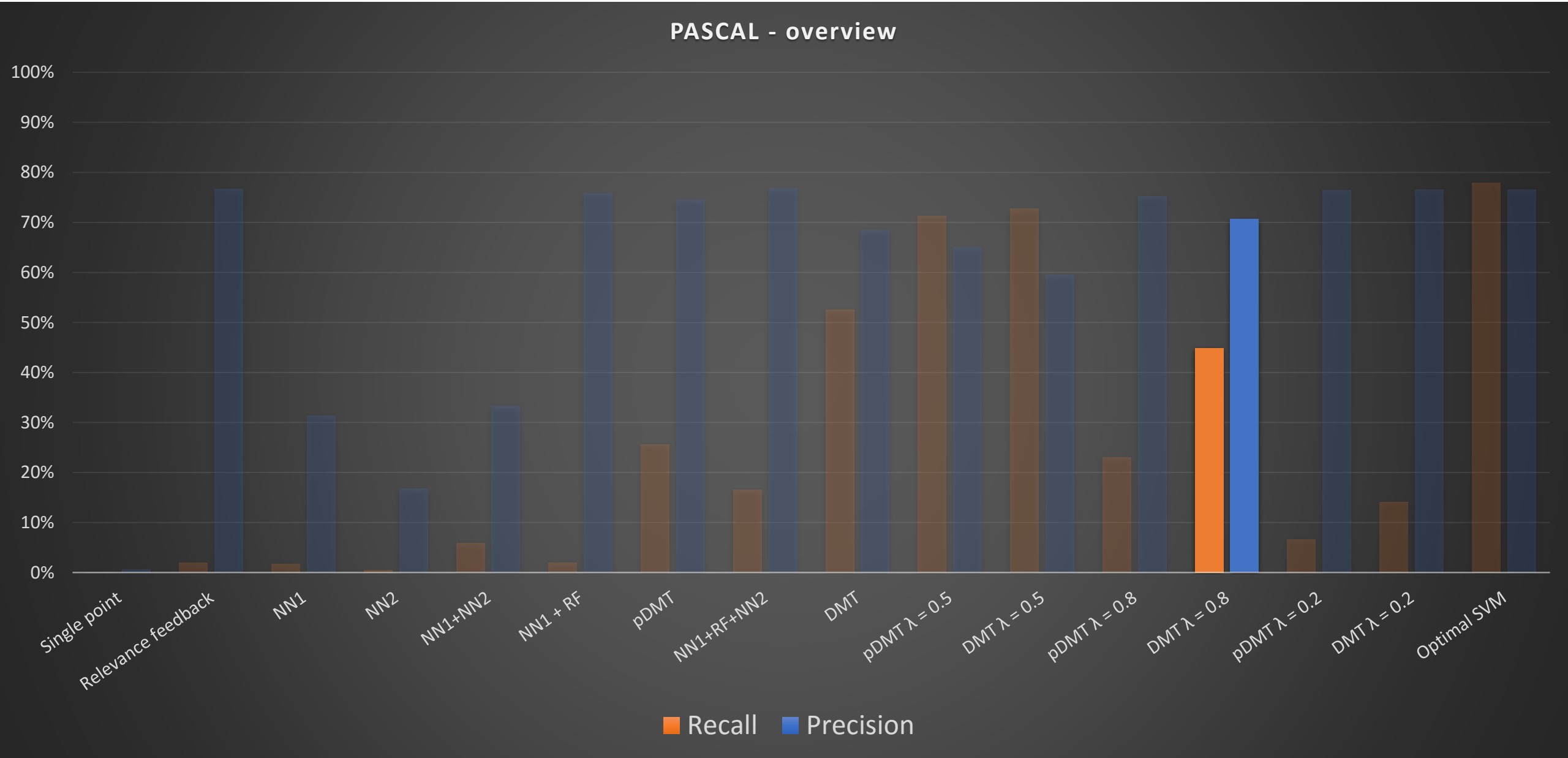

Single sample SVM
Neural Net I
λ=0.5
Feedback
λ=0.5
Neural Net I
λ=0.5
Neural Net II
Refined SVM

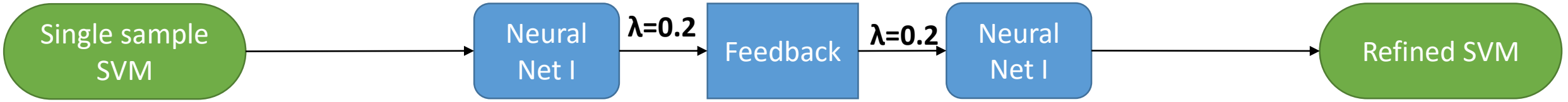

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

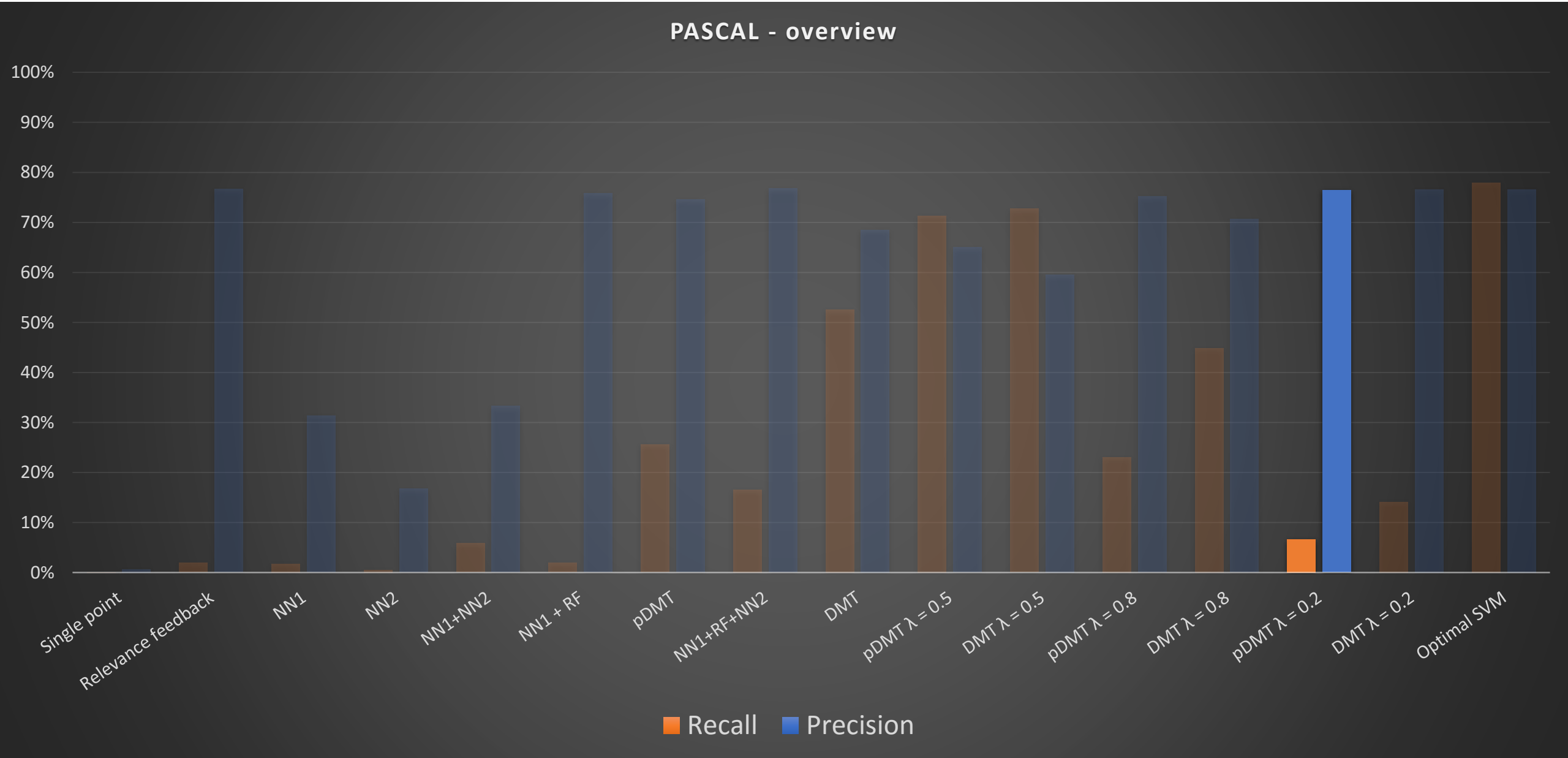

Single sample SVM
Neural Net I
λ=0.8
Feedback
λ=0.8
Neural Net I
Refined SVM

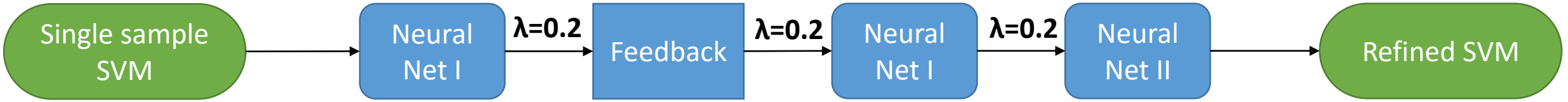

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

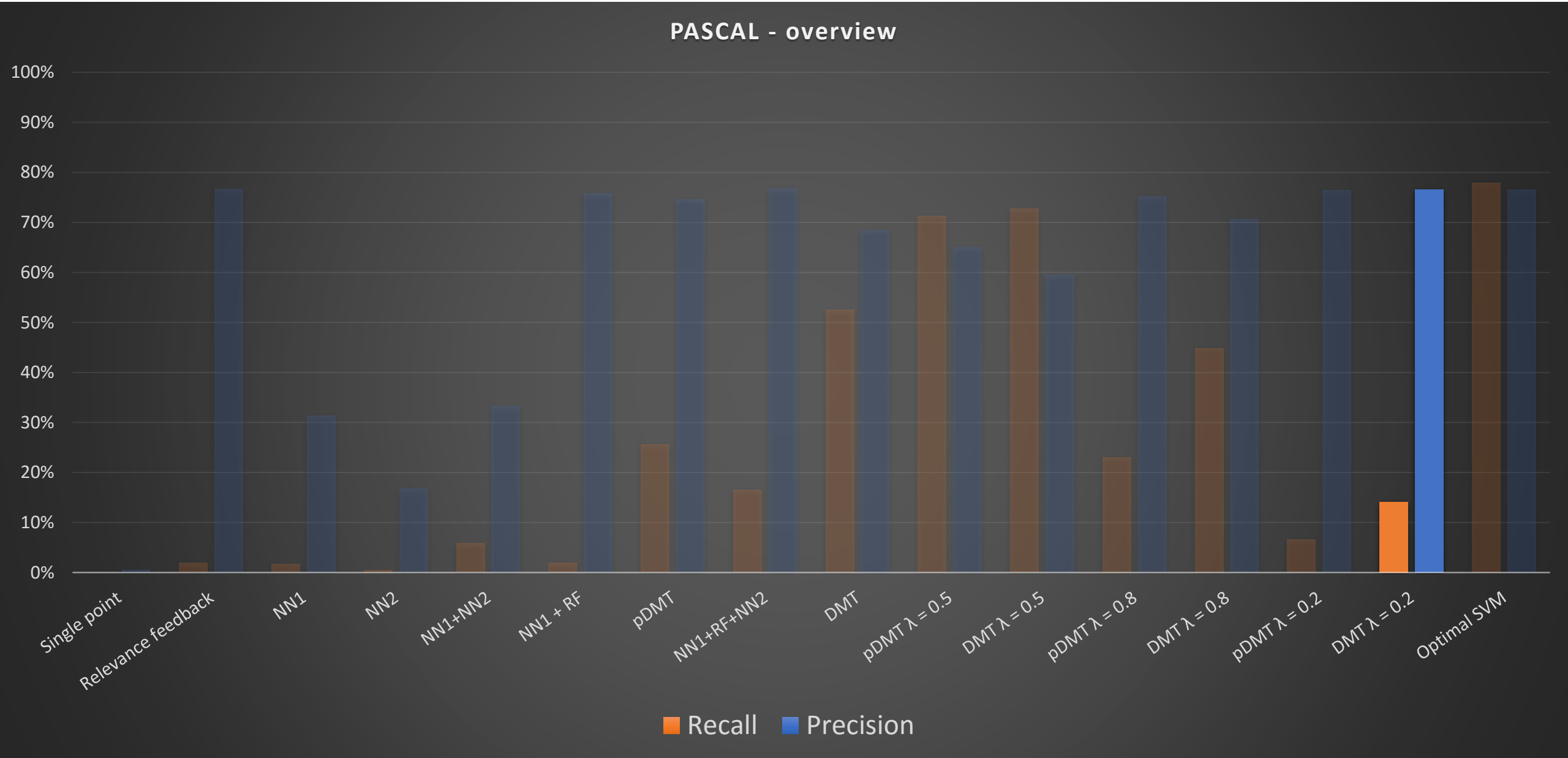

Single sample SVM
Neural Net I
λ=0.8
Feedback
λ=0.8
Neural Net I
λ=0.8
Neural Net II
Refined SVM

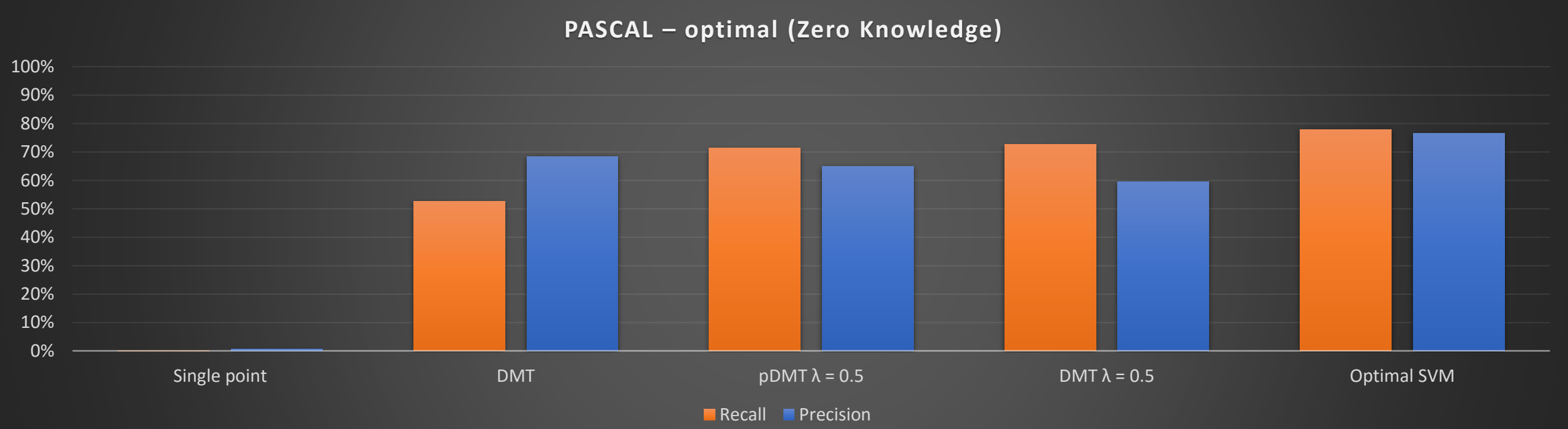

PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

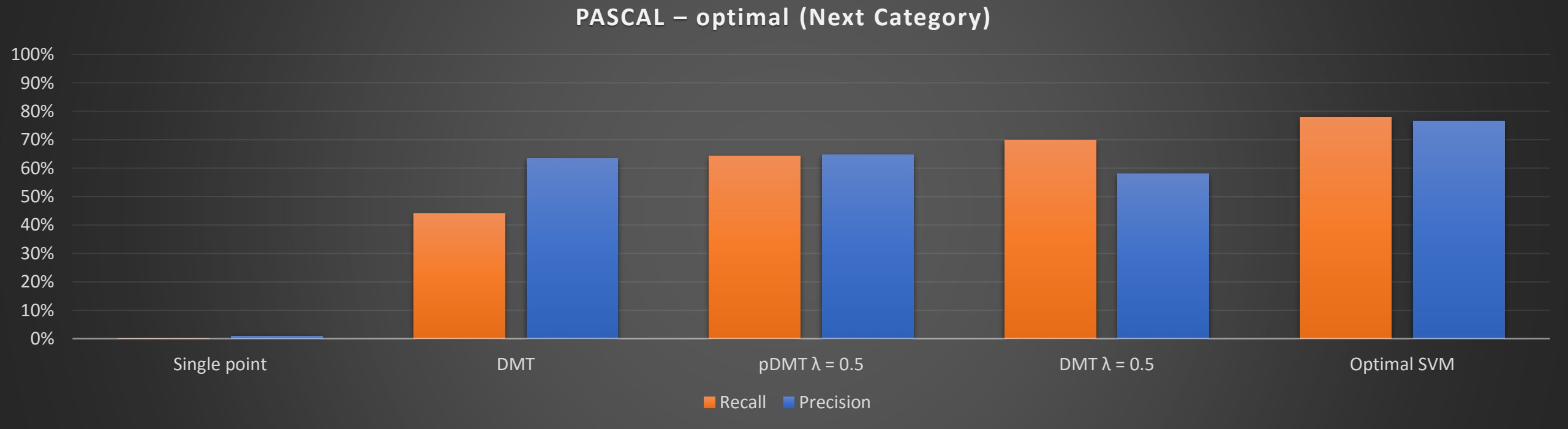

Single sample SVM
Neural Net I
λ=0.2
Feedback
λ=0.2
Neural Net I
Refined SVM


PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

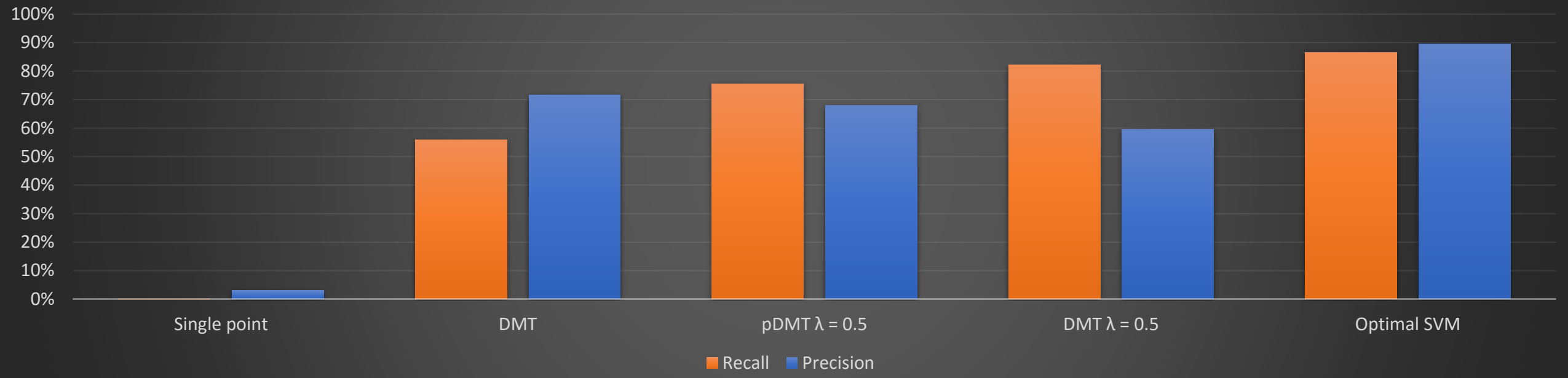

Single sample SVM
Neural Net I
λ=0.2
Feedback
λ=0.2
Neural Net I
λ=0.2
Neural Net II
Refined SVM


PASCAL - overview
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
Relevance feedback
NN1
NN2
NN1+NN2
NN1 + RF
pDMT
NN1+RF+NN2
DMT
pDMT λ = 0.5
DMT λ = 0.5
pDMT λ = 0.8
DMT λ = 0.8
pDMT λ = 0.2
DMT λ = 0.2
Optimal SVM
Recall
Precision

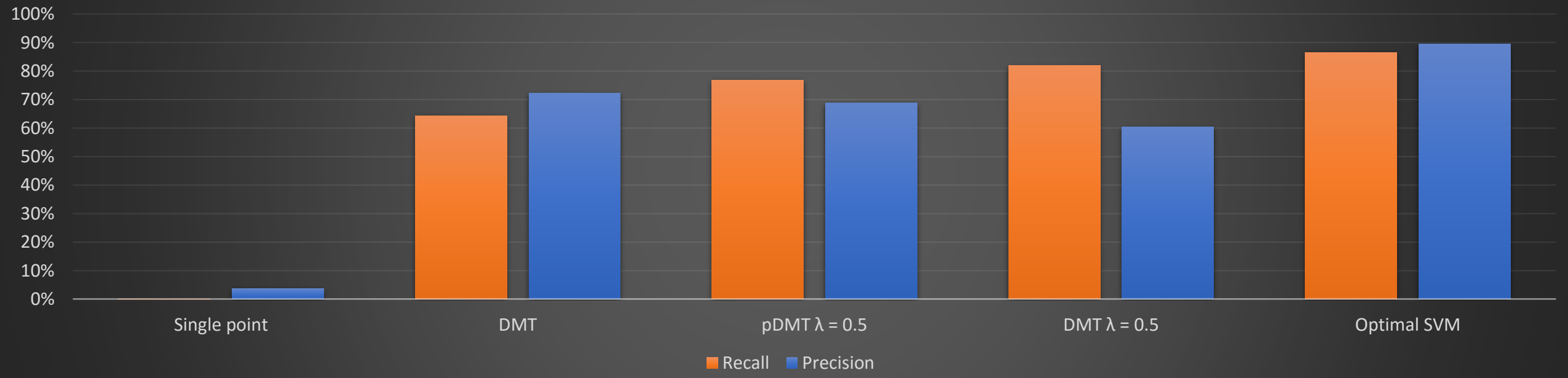

PASCAL – optimal (Zero Knowledge)
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
DMT
pDMT λ = 0.5
DMT λ = 0.5
Optimal SVM
Recall
Precision


PASCAL – optimal (Next Category)
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
DMT
pDMT λ = 0.5
DMT λ = 0.5
Optimal SVM
Recall
Precision

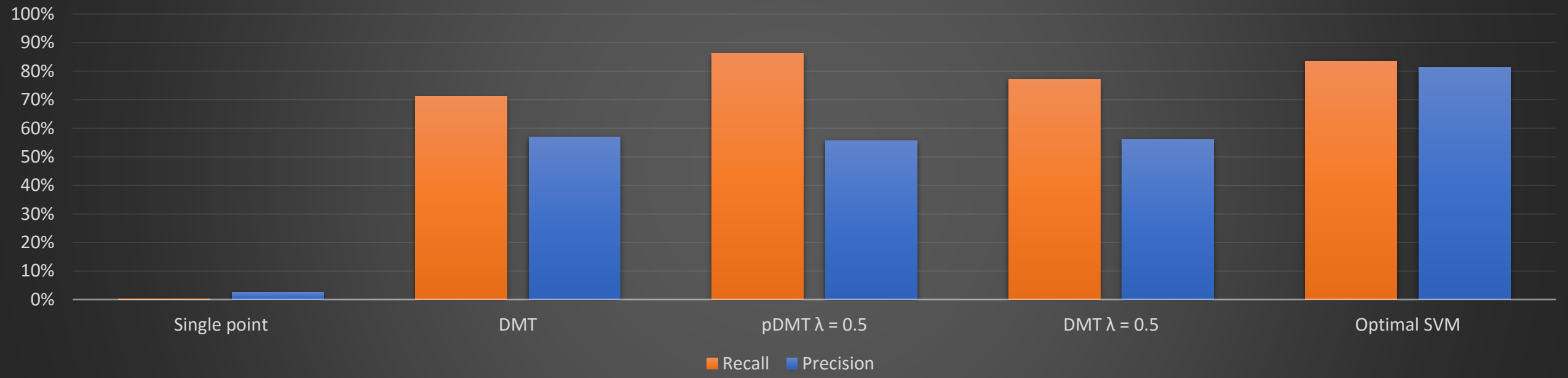
CALTECH 256 – optimal (Zero Knowledge)



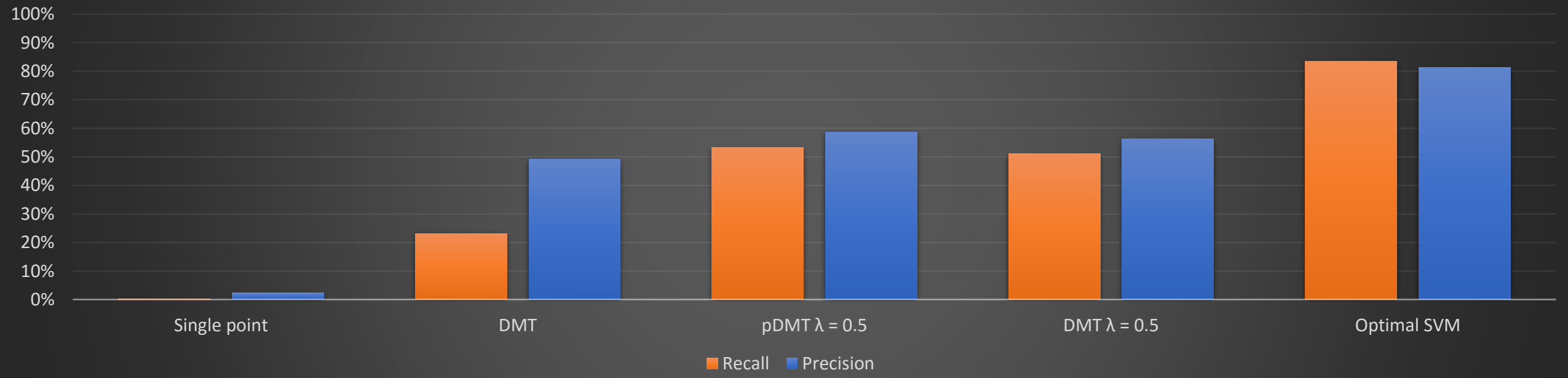
CALTECH 256 – optimal (Next Category)

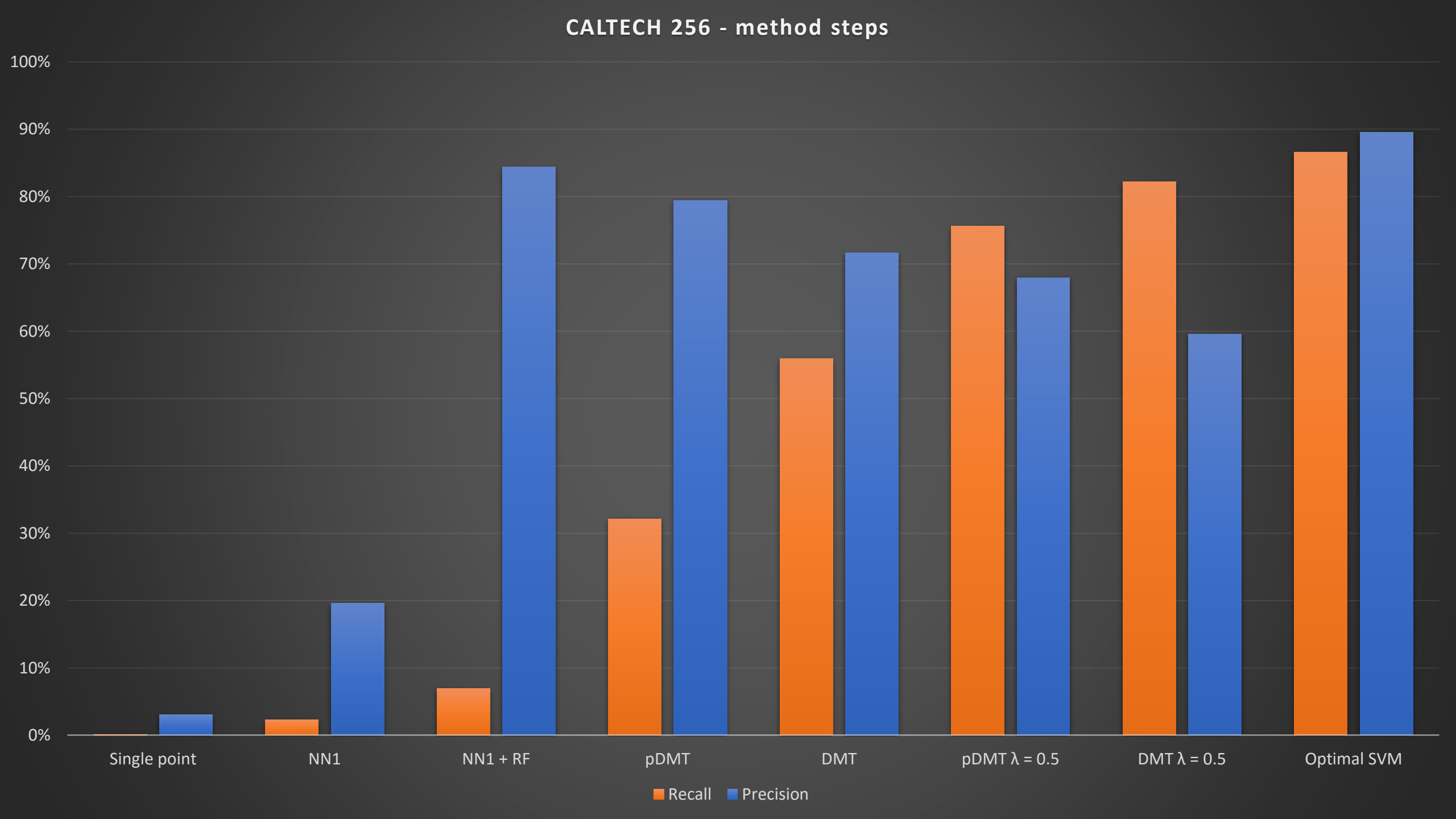


BIRDS – optimal (Zero Knowledge)



BIRDS – optimal (Next Category)




CALTECH 256 - method steps
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Single point
NN1
NN1 + RF
pDMT
DMT
pDMT λ = 0.5
DMT λ = 0.5
Optimal SVM
Recall
Precision

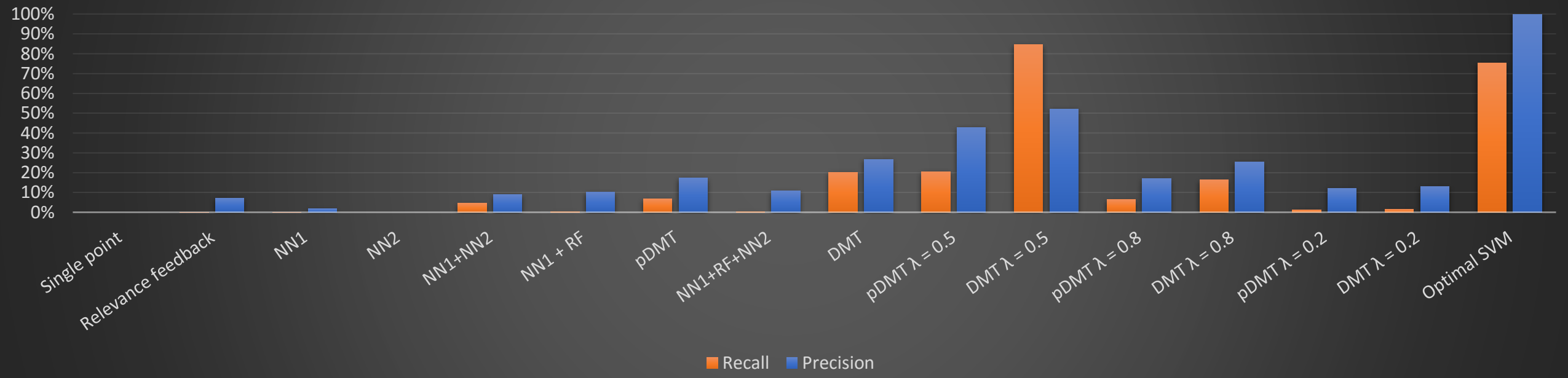
Results – other domains

We tested our approach on 2 datasets from non-visual domains:

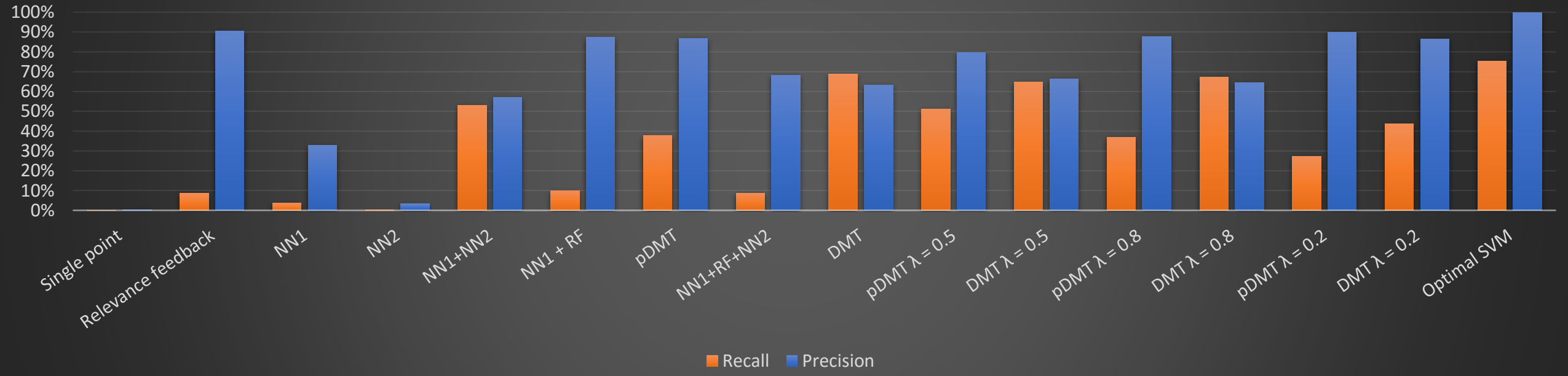
- CNAE-9
  - **1080 documents** of free text business descriptions of Brazilian companies categorized into a subset of **9 categories**
  - Each document was represented as a vector, where the weight of each word is its frequency in the document
  - **857** dimensional feature vector
  - Highly sparse
- UCI HAR
  - Human Activity Recognition Using Smartphones Data Set
  - 30 people recording daily activities wearing smartphone on their waist
  - Each person performed **six activities** (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING)
  - **561-feature vector** with time and frequency domain variables
  - **10299 instances**

Each dataset has reduced dimensionality using PCA to 500

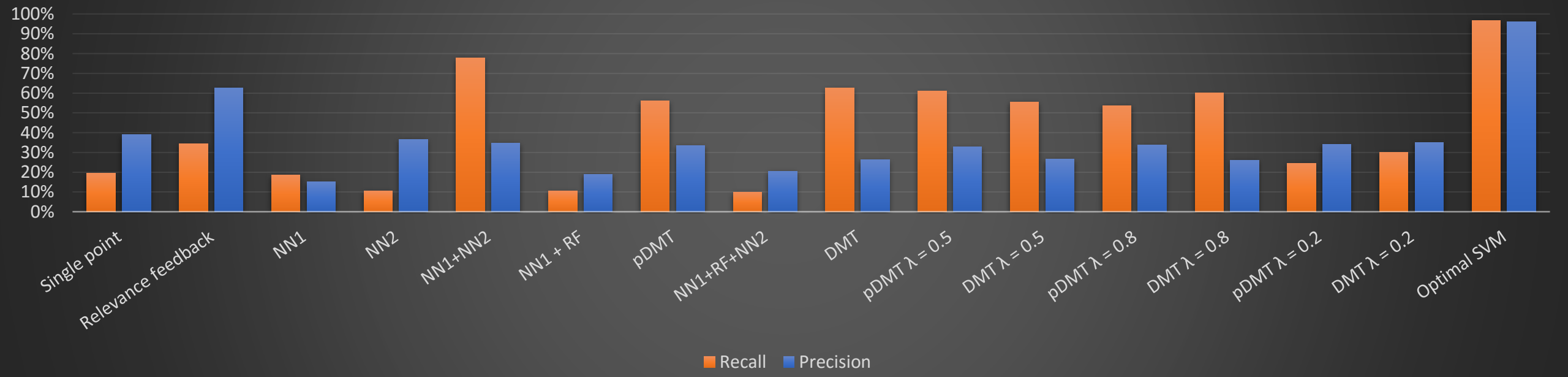
CNAE-9 (Zero Knowledge)



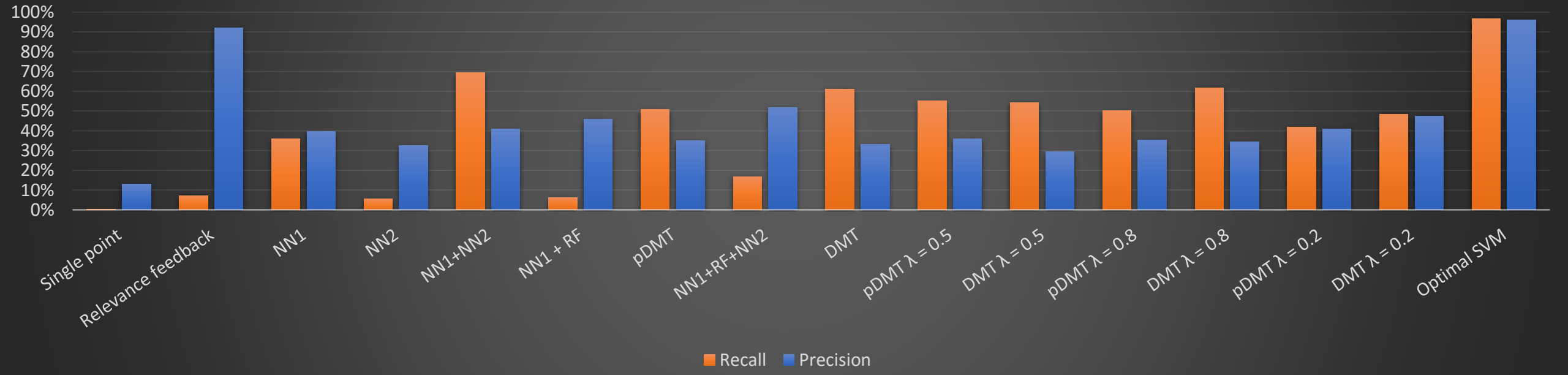
CNAE-9 (Next Category)



UCI-HAR (Zero Knowledge)



UCI-HAR (Next Category)



What now

Future work

1

User feedback

Accommodate to different datasets (hierarchy problem)

2

Ensemble of DMTs

Boost accuracy of regular classifiers

3

End-user experience

Build your own classifiers on-the-fly (AR, smartphone)

Thank you!