



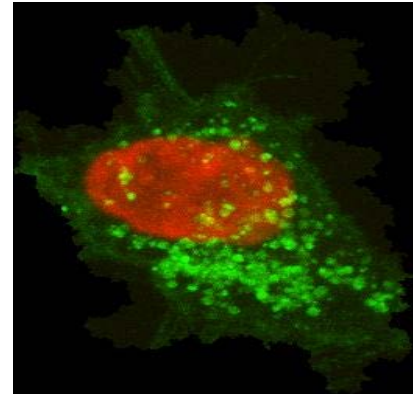
Generative modeling of cellular components with deep learning techniques

Xiongtao Ruan

12/14/2017

Cellular Pattern Recognition

- Describe cell patterns using numerical features
- Do classification, etc. to assign terms
- First described in Boland, Markey & Murphy (1998) and Boland & Murphy (2001)
- Later popularized in packages such as CellProfiler, WND-CHARM, Ilastik, CellCognition, etc.



Drawback

- Image features are typically not transferable across images from different sources (widefield vs. confocal vs. superresolution, differences in magnification or camera pixel size, pixel bit depth, etc.)

Another drawback

- Term assignment/classification approaches are incomplete and do not make full use of information in images
- “Is this an apple or an orange?” is a *discriminative* question; can be answered with 1 or 2 features
- “What does an apple look like?” requires a *generative model*

Generative models?

- Human cognition

 examples

Learn 

 "mental model"

Write 



Generated examples

- Image-based models

 Training images



MODEL Statistical generative model

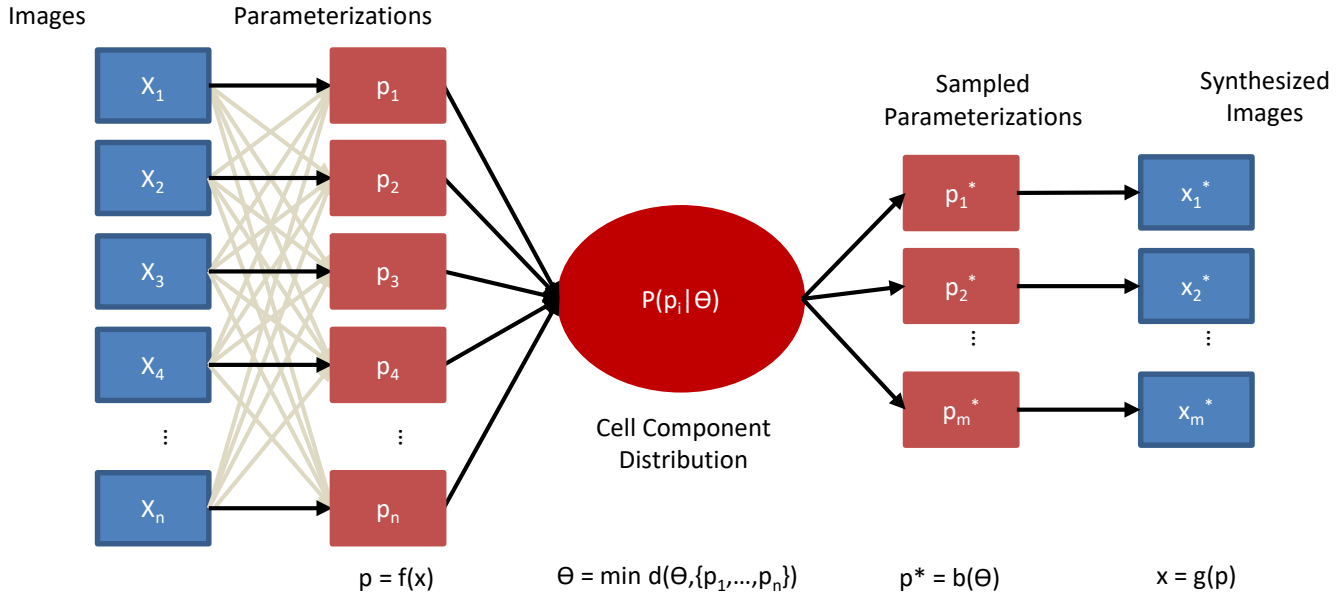


 Generated image

Parametric models

- Computer vision problems such as this have traditionally been tackled by hand-constructing models and learning their parameters from images

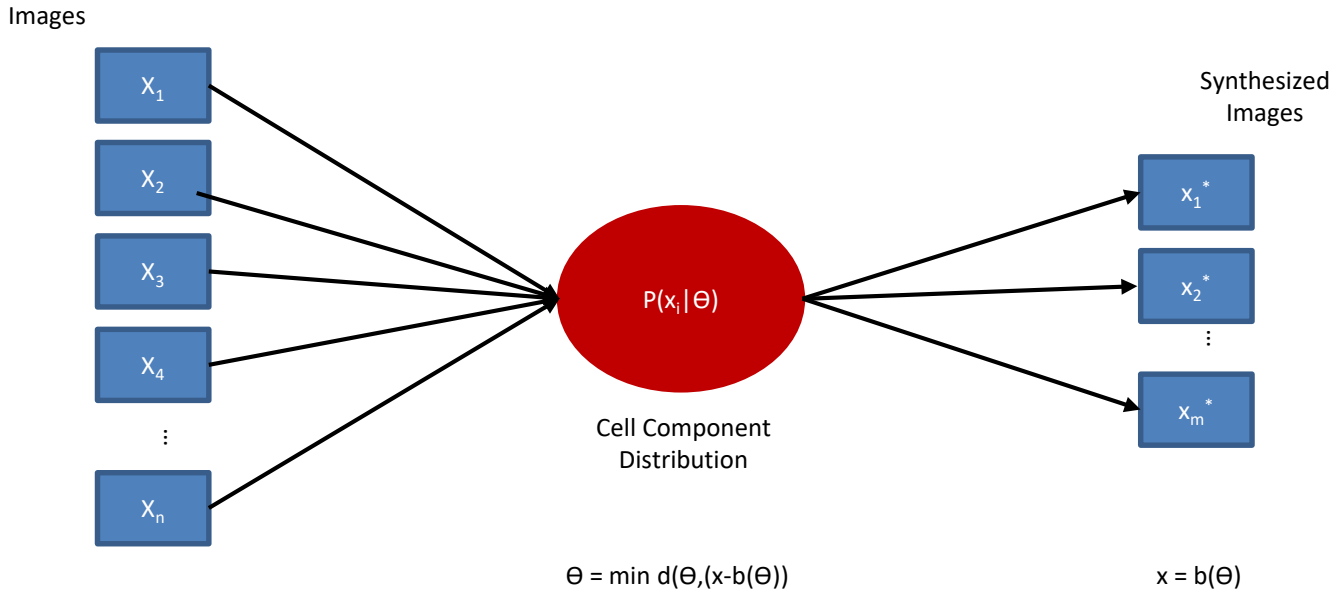
Parametric modeling (e.g., CellOrganizer)



“Deep” learning

- If large numbers of training examples are available, “deep learning” methods can learn directly from images without need for custom design

Deep learning models (e.g., autoencoders)



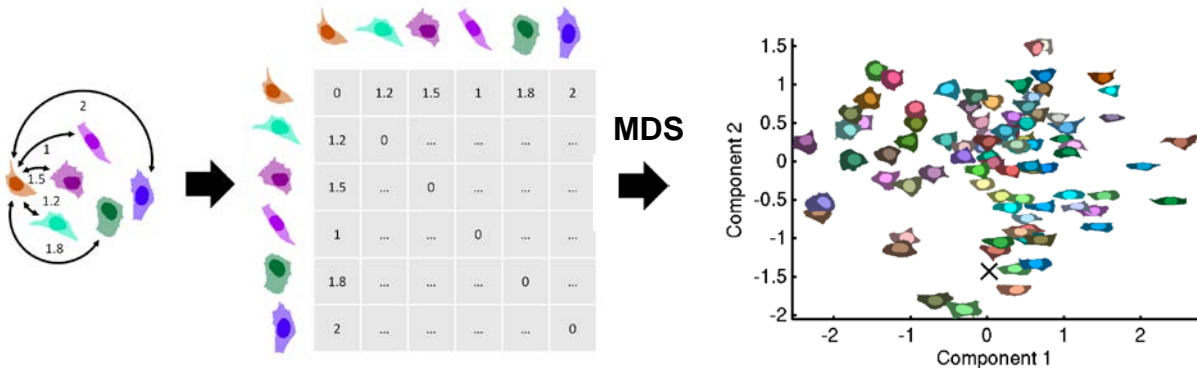
MODELING CELL SHAPE WITHIN AND BETWEEN CELL TYPES

Methods for shape analysis

- Many methods have been described
- Descriptive
 - Features
- Generative
 - PCA on outline coordinates
 - Diffeomorphic distance embedding
 - Neural networks

Diffeomorphic model

- Use large deformation diffeomorphic metric mapping (LDDMM) algorithm to define distance that measures dissimilarity between shapes.
- The distances are put into the distance matrix and the shape space is inferred through embedding using Multidimensional scaling (MDS).



PCA based model

- Extract landmarks
- Perform PCA analysis on landmarks
- Use first k principle components as the model



(a) $b_1 = -3\sqrt{A_1}$



(b) $b_2 = 0$



(c) $b_1 = +3\sqrt{A_1}$



(d) $b_2 = -3\sqrt{A_2}$



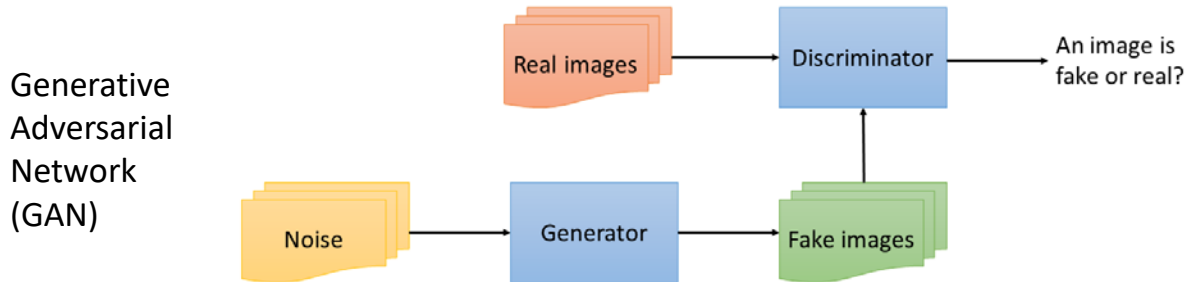
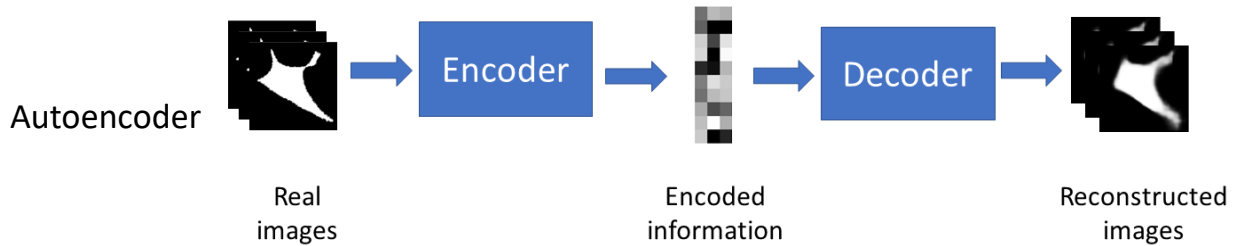
(e) $b_2 = 0$

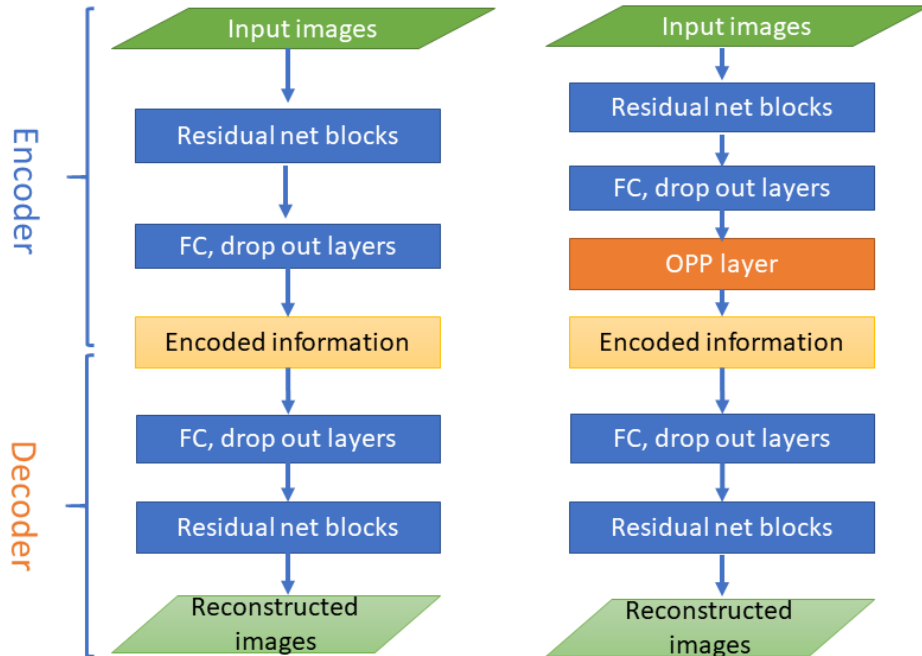


(f) $b_2 = +3\sqrt{A_2}$

Unsupervised deep learning

- Deep networks that try to learn representations of data without label information





FC Autoencoder

OPP Autoencoder

Comparison of methods

- All methods represent (encode) a given cell shape with a set of features/parameters
- We sought to compare the accuracy of representation (reconstruction error) across methods

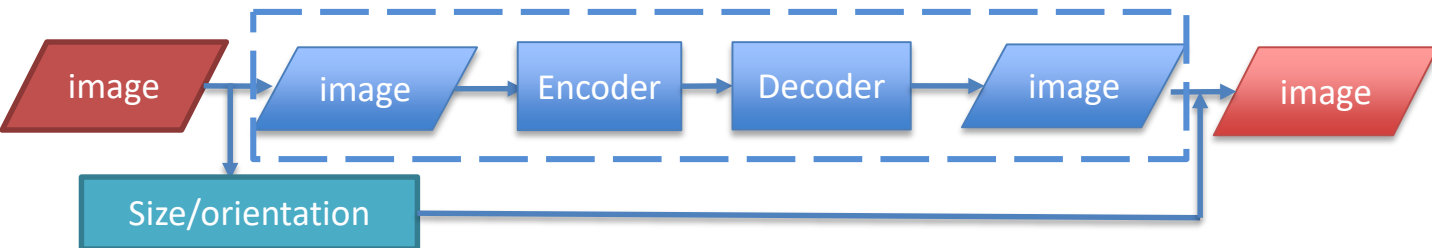
Comparison of methods

- Used training and testing sets of 10,000 images each from HPA
- Calculated average difference between original testing image and image after encoding/decoding

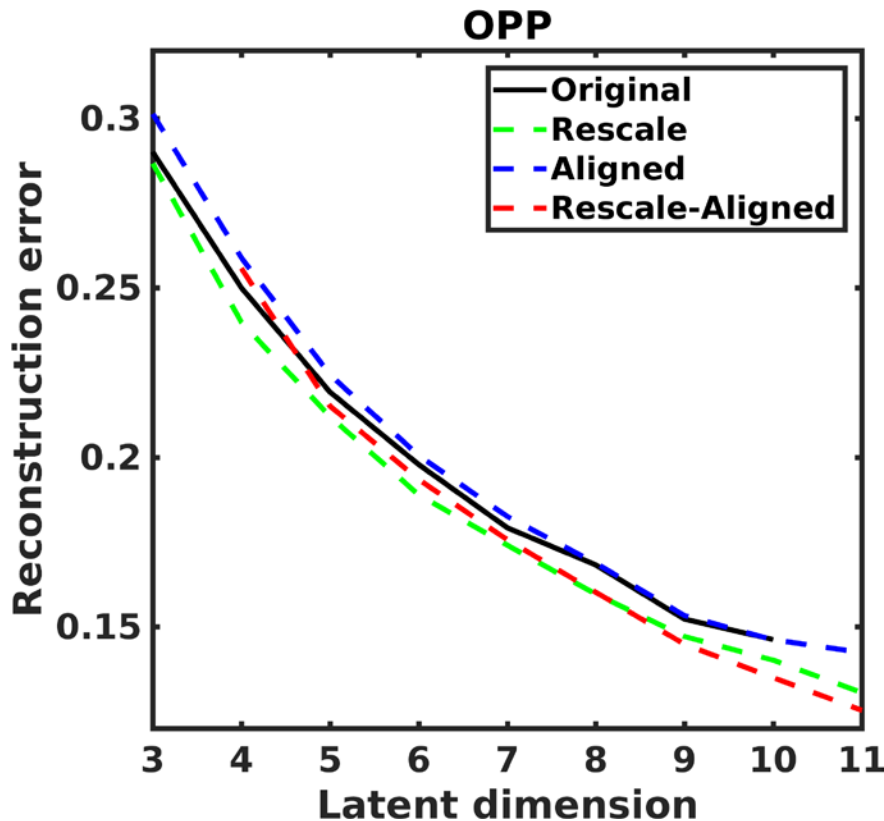
	PCA	Diffeomorphic	AE FC	AE OPP
Error (%)	28.0	46.1	18.5	17.9
Time (hours)	< 0.2	$\sim 1 \times 10^4$	~ 2	~ 2

Size, orientation and shape

- Autoencoder with OPP performed the best
- But all of these methods “mix” size, orientation and shape for each cell
- Can we get better performance by removing variation in size or orientation before training?



- Compare errors for different numbers of latent dimensions
- Slight improvement when removing size



Improve training for small-scale data

- We can train the model with removal of size/orientation with fewer images

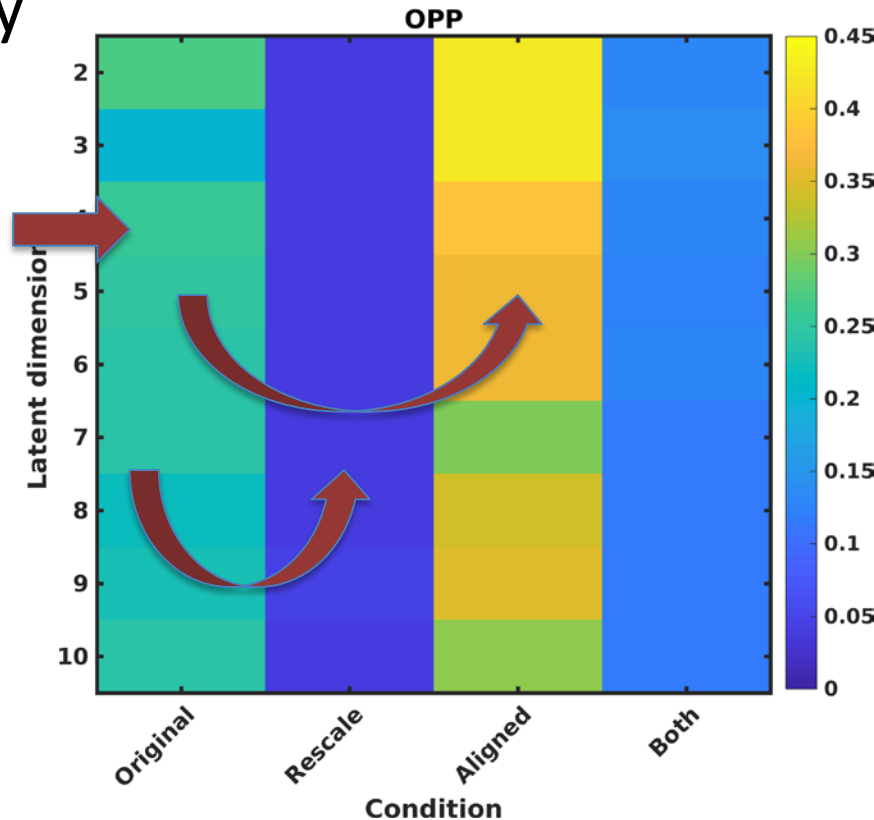
Train num	Latent dim	2	3	4	5	6	7	8	9	10
10k	Original	0.3629	0.3038	0.2448	0.2180	0.1974	0.1854	0.1688	0.1533	0.1528
5k	Original	0.3866	0.3094	0.2525	0.2227	0.2020	0.1833	0.1737	0.1611	0.1532
	Rescale		0.2971	0.2471	0.2148	0.1966	0.1785	0.1639	0.1520	0.1441
	Both			0.2650	0.2206	0.1977	0.1809	0.1641	0.1491	0.1387
2k	Original	0.4276	0.3348	0.2685	0.2353	0.2403	0.1969	0.1968	0.1769	0.2032
	Rescale		0.3456	0.2649	0.2253	0.2231	0.1925	0.1835	0.1678	0.1632
	Both			0.2850	0.2312	0.2028	0.1851	0.1699	0.1584	0.1540
1k	Original	0.4659	0.3901	0.3142	0.2567	0.2389	0.2328	0.2277	0.2147	0.2577
	Rescale		0.3804	0.2906	0.2452	0.2499	0.2185	0.2277	0.2171	0.1789
	Both			0.3060	0.2562	0.2187	0.2259	0.2014	0.1730	0.1596

Comparison within and among cell lines

- Next asked whether cultured lines differ in shape
- Constructed dataset from HPA for 10 cell lines (A-431, A-549, CACO-2, HEK 293, HeLa, Hep-G2, MCF-7, Pc-3, U-251 MG, and U-2 OS)
- Asked how distinguishable they were (average of all pairwise comparison) using different autoencoder representations

Distinguishability

- Cell lines are distinguishable using original images
- Better if aligned
- Not distinguishable if size is normalized

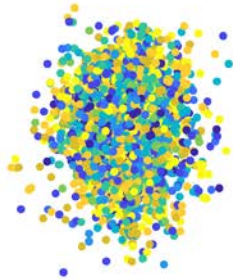


Shape Spaces for 10 cell lines

Original



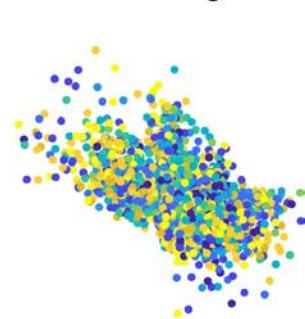
Rescale



Aligned



Rescale & Aligned



JOINT MODELING OF CELL AND NUCLEAR SHAPES

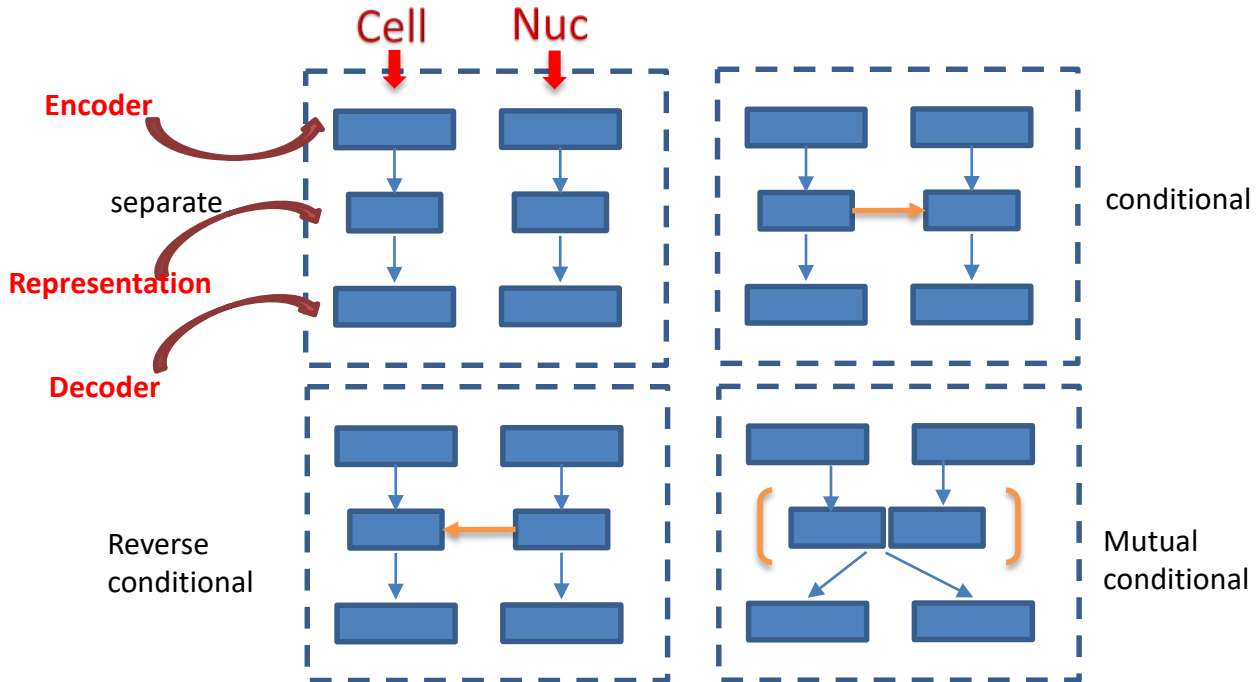
Question

- Johnson et al. 2015, showed cell and nuclear shapes are dependent in terms of prediction for each other with diffeomorphic model. The question is: could deep learning techniques capture the dependency relationship?
- From another point of view, the question is to see whether joint modeling can improve the prediction between each other.

Proposed network structures

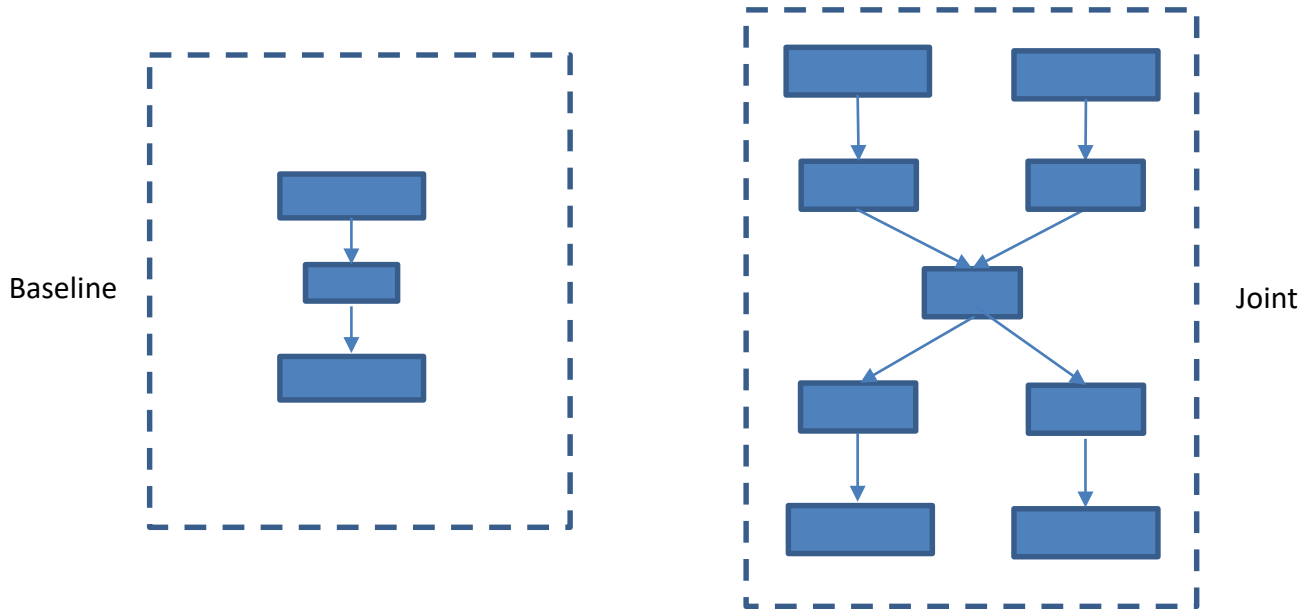
- **Separate models:** model cell and nuclear shape separately.
- **Baseline:** represent cell and nuclear shapes as indexed image, and use a single autoencoder
- **Conditional:** nuclear dependent on cell
- **Reverse conditional:** cell dependent on nuclear
- **Mutual conditional:** both, concatenate the encoded information as input for decoders.
- **Joint:** two encoders joint to generate same encoded information

Illustration of network structures



Left: cell, right: nuclear

Illustration of network structures



Left: cell, right: nuclear

Evaluation criterion

- Use Jaccard index between original and reconstructed images for cell and nuclear shape, respectively.
- The overall error is the average of cell and nuclear errors.

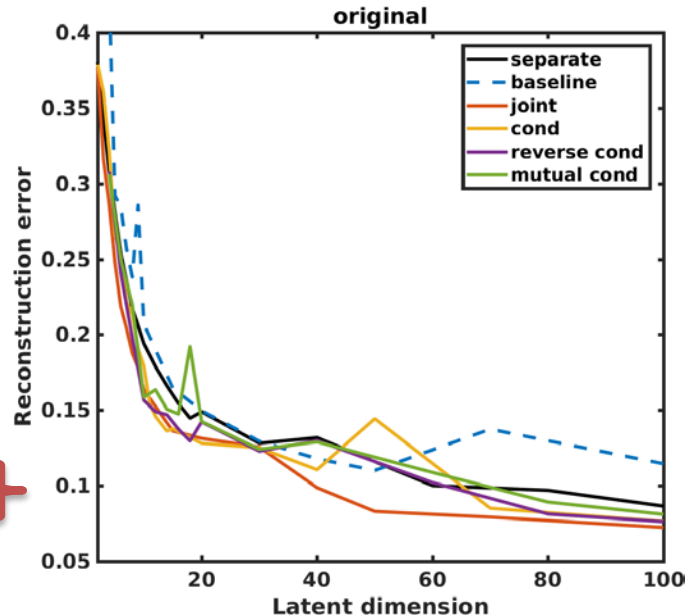
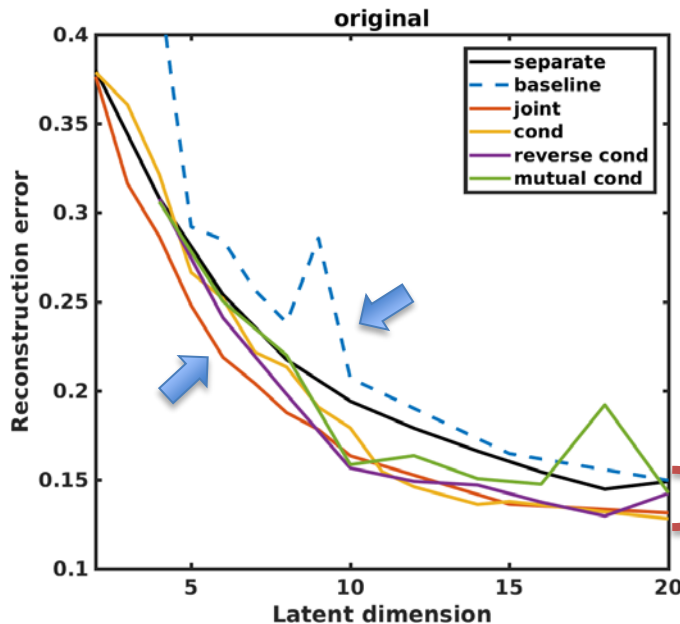
Compare with PCA model

- Total latent dimension: 7
- For separate models, cell dimension: 4, nuclear dimension: 3

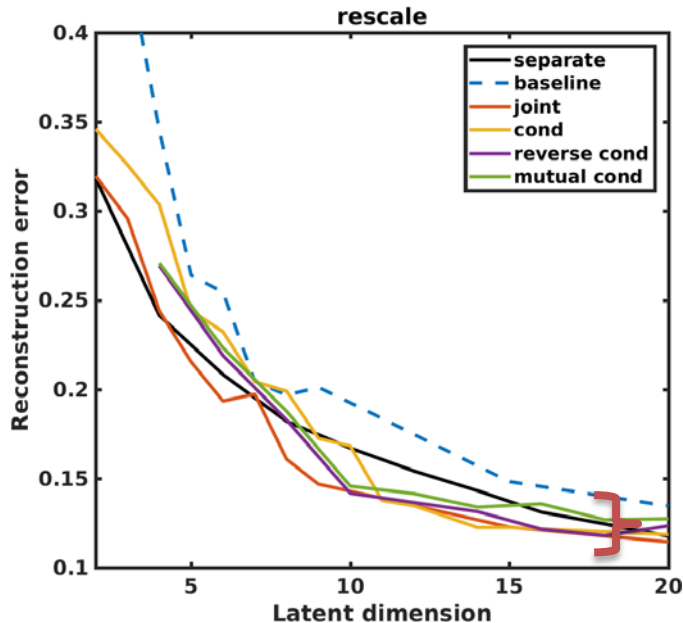
	PCA		Deep autoencoder	
	Separate	Joint	Separate	Joint
Cell	0.2616	0.2274	0.2175	0.2509
Nuclear	0.2827	0.2599	0.2483	0.1573
Overall	0.2722	0.2437	0.2329	0.2041

Comparison of joint errors

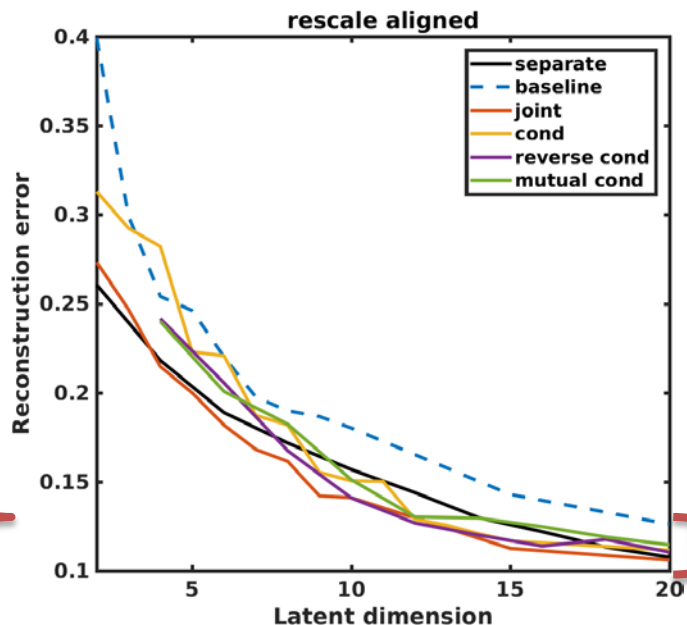
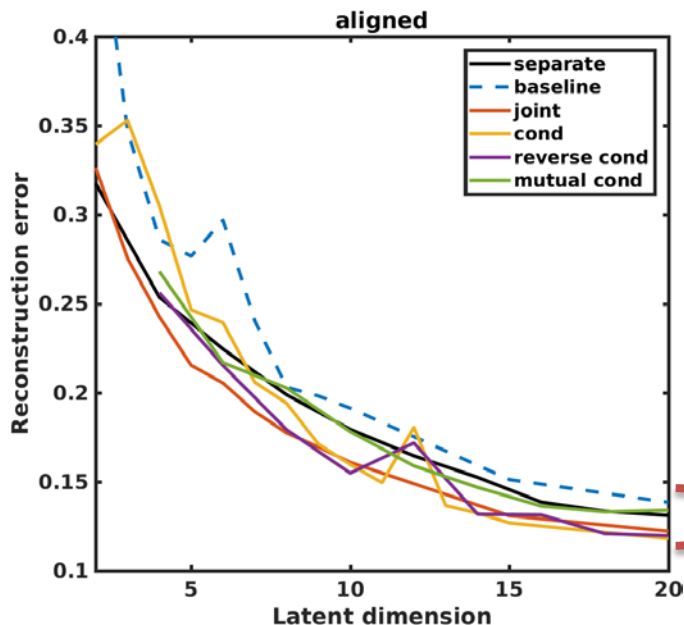
- Left: small range, right: broader range



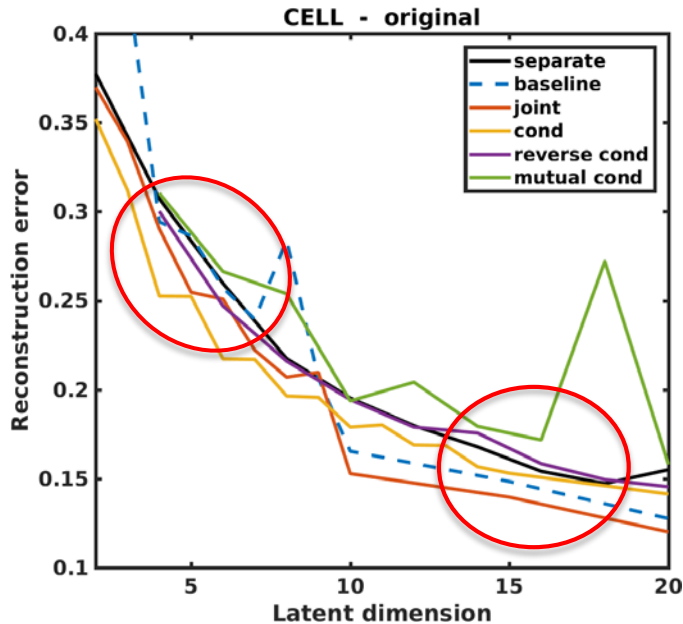
Size and orientation contribute to the dependency relationships



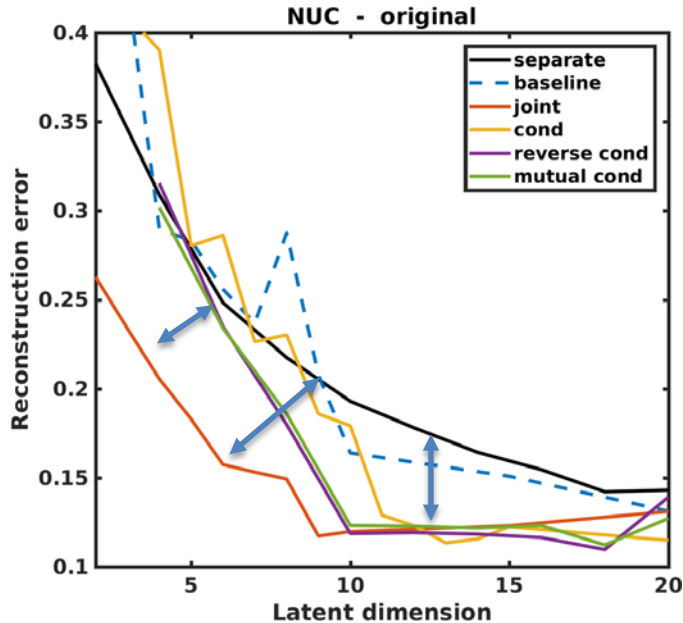
Size and orientation contribute to the dependency relationships



Reconstruction error for cell shapes is reduced with joint modeling



Reconstruction error for nuclear shapes is reduced with joint modeling



Conclusion

- For cell shape only, autoencoders outperform PCA and diffeomorphic model a lot.
- Scale and orientation are two major attributes for shapes and removing them improves the reconstruction accuracy.
- Joint modeling of cell and nuclear shapes shows better performances than separate models.
- Scale and orientation are also involved in the dependency relationship.

Future work

- Do more experiments, i.e. compare with diffeomorphic model, and also use other datasets to compare different methods.
- Continue to improve the performance of autoencoders for joint modeling, i.e. changing network structures, tuning hyperparameters.