Deep Neural Networks and Discrete Choice Models (Part 3)

Shenhao Wang

191105

Part 0. Recap

Outline

1

Decision-making under uncertainty: prospect theory

(25 min)

2

Modeling time uncertainty in transportation

(5 min)

3

Working paper: theory-based deep residual network

(20 min)

4

Multitask & transfer learning

(10 min)

5

Working paper: MTLDNN to combine RP & SP

(20 min)

Part 1. Decision-making under uncertainty: prospect theory

Examples: decision-making under uncertainty

New technology adoption (e.g. autonomous vehicles)

Gamble

Insurance

Smoking: health risk

Asset investment

Natural disasters

Governance: belt and smoking regulations

Consumption: quality uncertainty

Urban transportation: time uncertainty

Decision-making under time uncertainty

Option A (Ride Hailing)

Travel cost: $5

Travel time: **15** minutes

Option B (Ride Sharing)

Travel cost: $3

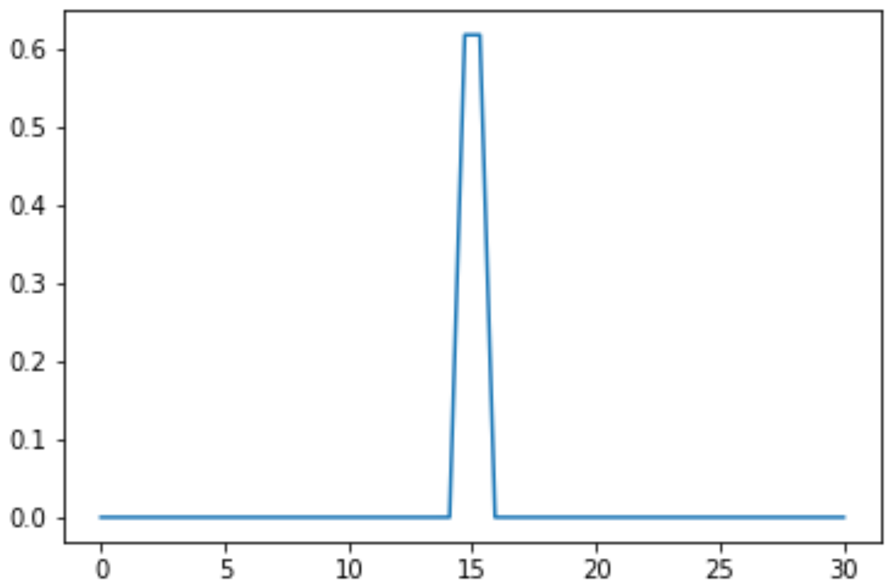
Travel time: between **15-25** minutes

Decision-making under time uncertainty

Option A (Ride Hailing)

Travel cost: $5

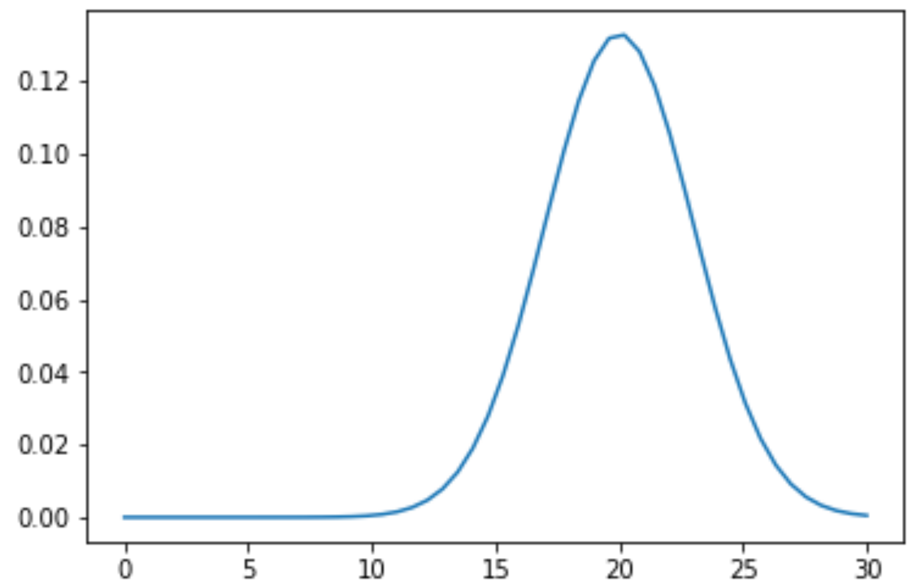
Travel time: **15** minutes



Option B (Ride Sharing)

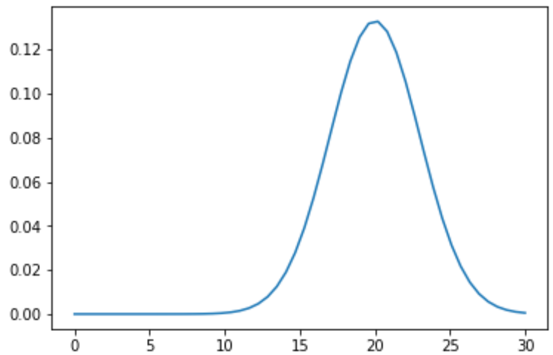
Travel cost: $3

Travel time: between **15-25** minutes

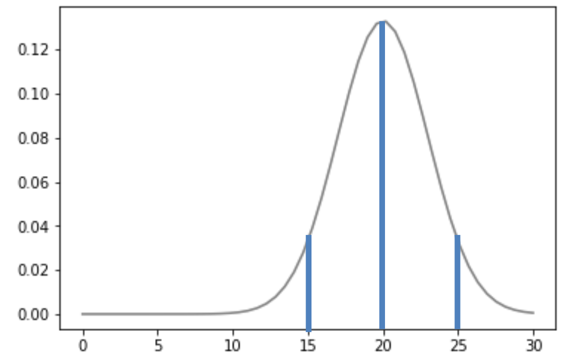


How to characterize uncertainty? Math concepts

1. Full continuous distribution



2. Discrete distribution (in survey)



3. Mean and variance

Mean: 20 minutes

STD: 5 minutes

4. Min and max (or 5% and 95 percentiles)

Min: 10 minutes

Max: 30 minutes

5% percentile: 12 minutes

95% percentile : 28 minutes

Confusing concepts: risk, uncertainty, and “black swan” events

1. Risk vs. uncertainty

    Academic difference: full probability distribution vs. no full probability distribution

    Colloquial difference : only loss vs. both loss and gains

2. Totally unpredictable: “black swan” events.

    Self-eliminating: why bother?

    Self-contradictory: extremely small probability vs. unpredictable.

    etc.

I will always use mathematical concepts, but avoid these confusing concepts.

Models

Given wealth W, a person needs to choose between two **monetary** options:

$$(\$x_1, p_1;\ \$x_2, p_2)\ vs.\ \$0$$

e.g. $(+\$1{,}000, 50\%; -\$1{,}000{,}50\%)\ vs.\ \$0$

Note: options can be any probability distribution.

How to compute utilities of the two options to predict choices?

1) Expected utility : $p_1 * v(W + x_1) + p_2 * v(W + x_2)$ vs. $\mathrm{v}(W)$;

2) **Prospect theory**: $\pi(p_1)v(x_1) + \pi(p_2)v(x_2)$ vs. $v(0)$

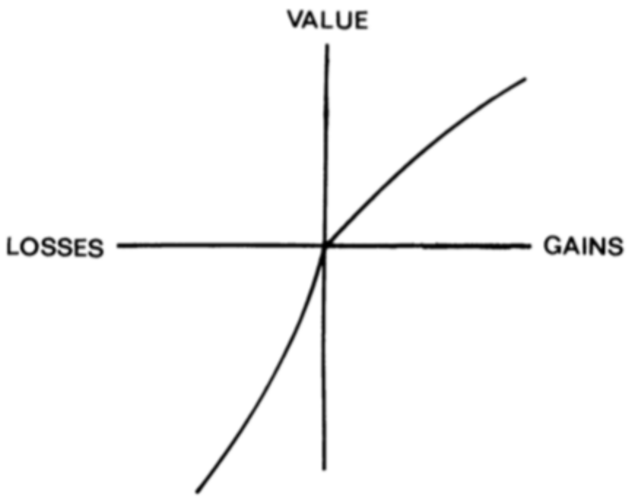
   W & reference point

   $\pi(p)$

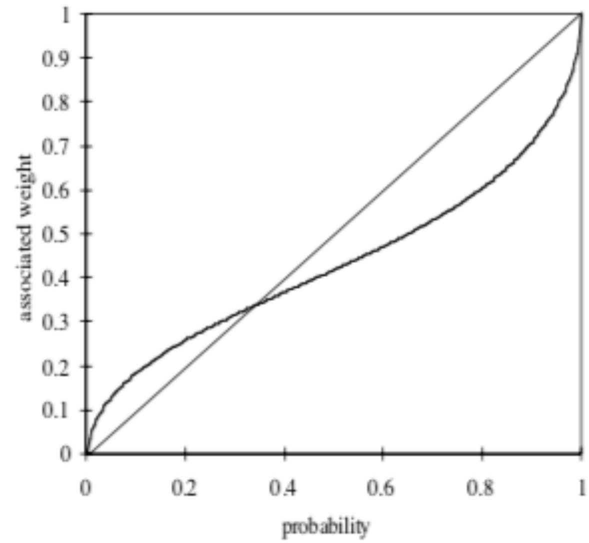
3) Black box: $f\colon (\$x_1, p_1;\ \$x_2, p_2; W) \rightarrow (0{,}1)$

Prospect theory: five characteristics

$$\pi(p_1)v(x_1) + \pi(p_2)v(x_2)$$



Value Function $v(x_1)$



Probability weighting function $\pi(p)$

1) Concavity over gains
2) Convexity over losses
3) Framing over gains and losses (reference dependent)
4) Loss aversion
5) Probability weighting

Prospect Theory

$$\pi(p_1)v(x_1) + \pi(p_2)v(x_2)$$

#1) Concavity over gains (risk averse)

(50%, $1000; 50%, $0)

<

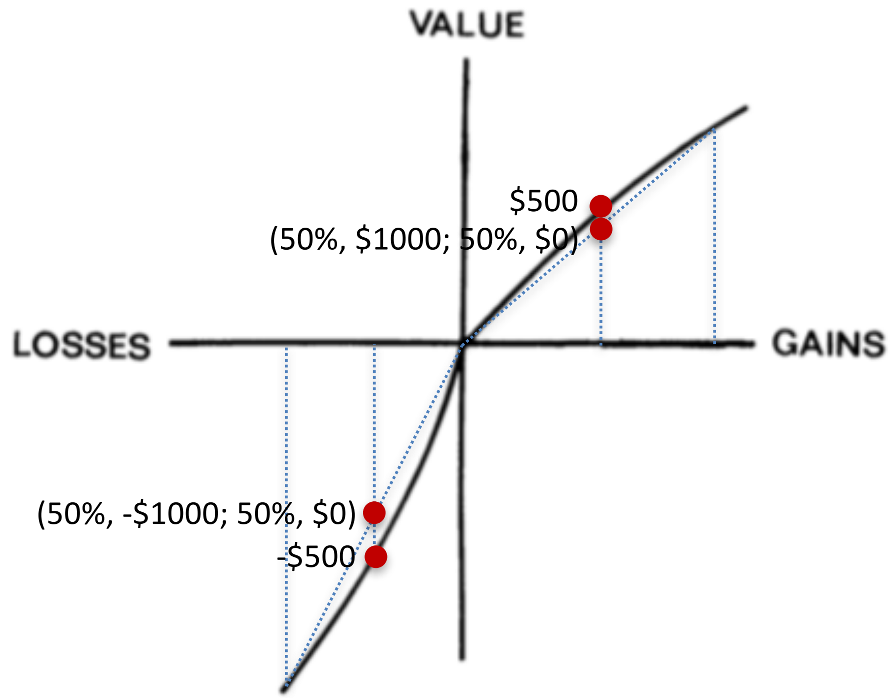
$500

#2) Convexity over losses (risk seeking)

(50%, -$1000; 50%, $0)

>

-$500

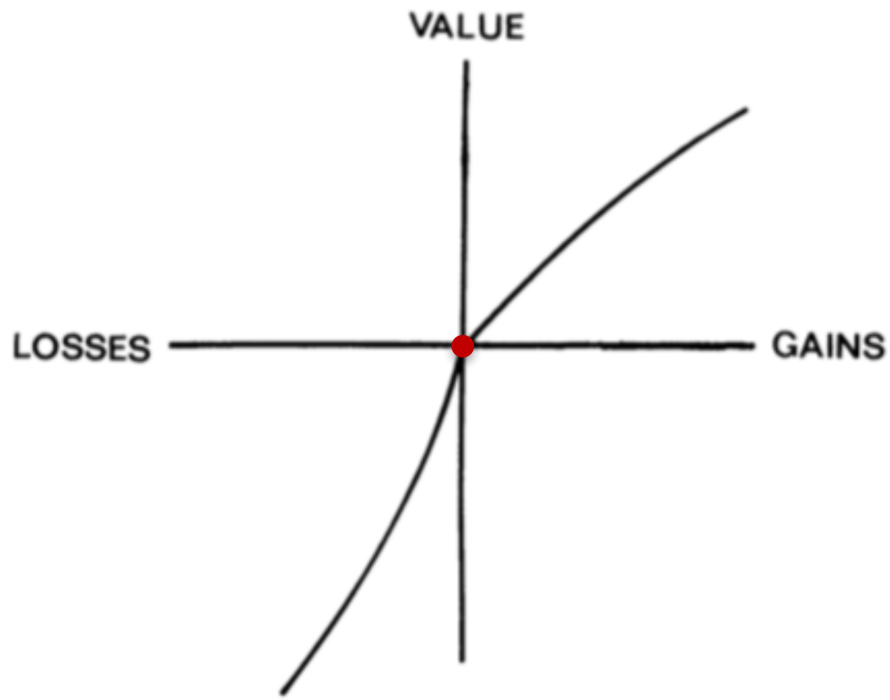


Prospect Theory

$$\pi(p_1)v(x_1) + \pi(p_2)v(x_2)$$

3) **Framing** over gains and losses

Asian disease experiment



PT: Framing over gains and losses (Asian disease experiment)

”Imagine that the U.S. is preparing for the outbreak of an unusual Asian disease, which is expected to kill 600 people. Two alternative programs to combat the disease have been proposed.”

Experiment 1 (A vs. B)

A **[72%]**	B **[28%]**
200 people will be saved	there is a 1/3 probability that 600 people will be saved, and a 2/3 probability that no people will be saved

Experiment 2 (C vs. D)

C **[22%]**	D **[78%]**
400 people will die	there is a 1/3 probability that nobody will die, and a 2/3 probability that 600 people will die

A = C

B = D

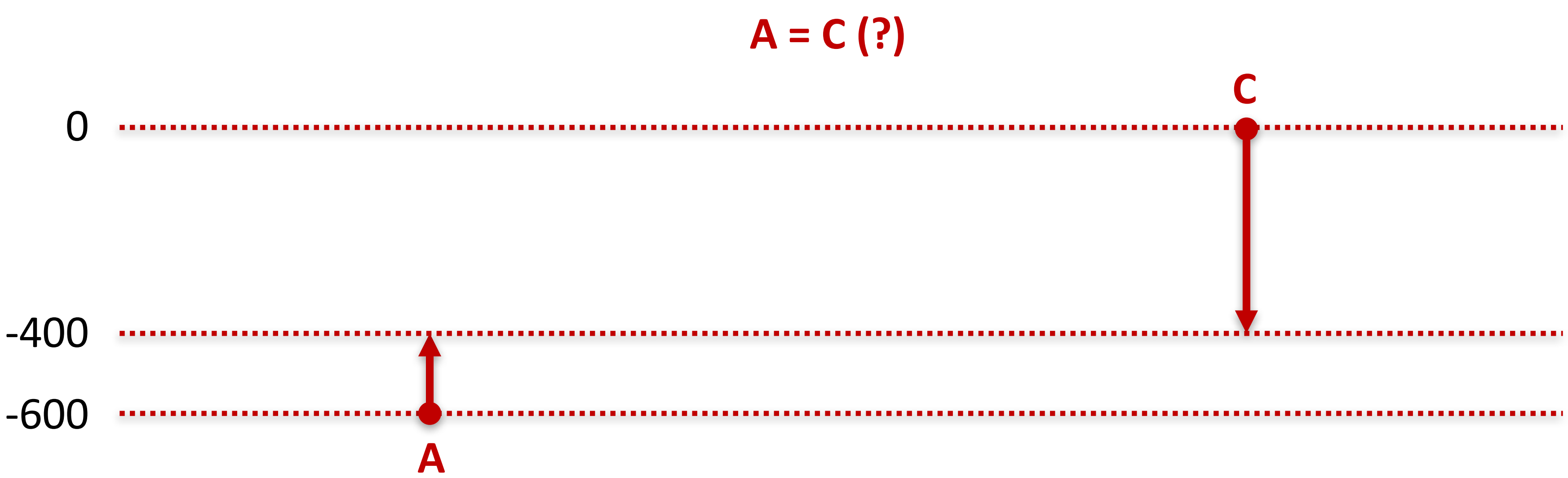
PT: Framing over gains and losses (Asian disease experiment)

Experiment 1 (A vs. B)

A [72%]	B [28%]
200 people will be saved	there is a 1/3 probability that 600 people will be saved, and a 2/3 probability that no people will be saved

Experiment 2 (C vs. D)

C [22%]	D [78%]
400 people will die	there is a 1/3 probability that nobody will die, and a 2/3 probability that 600 people will die



Prospect Theory

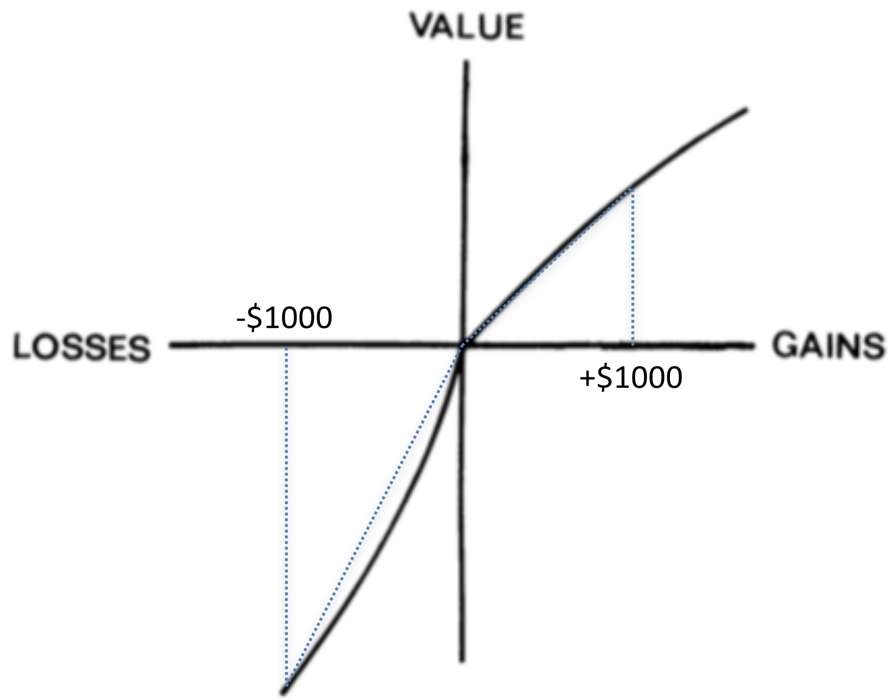
$$\pi(p_1)v(x_1) + \pi(p_2)v(x_2)$$

4) Loss aversion (the kink)

0

>

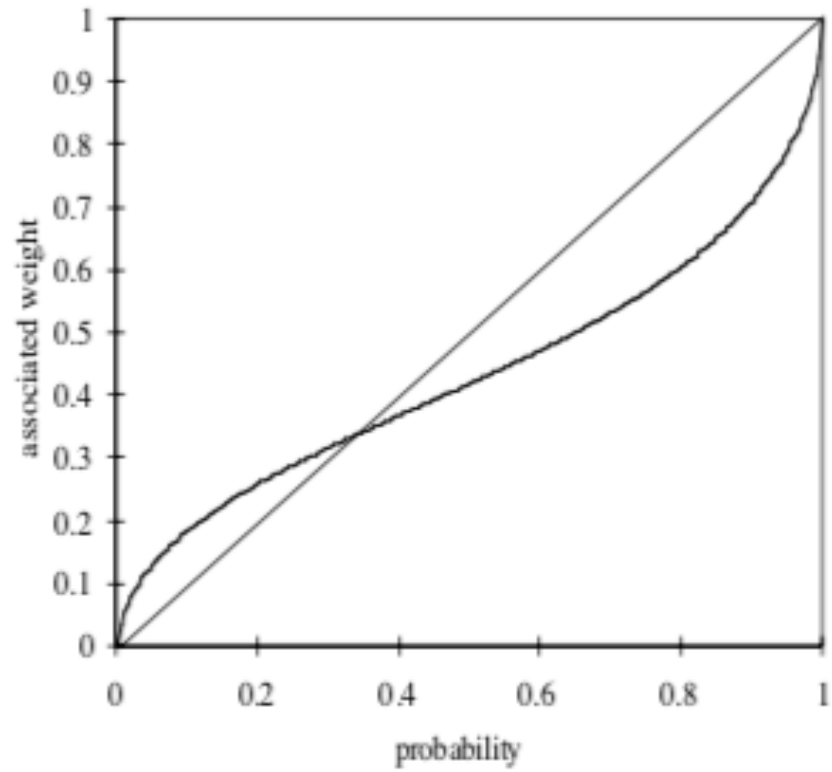
(50%, +$1000; 50%, -$1000)



Prospect Theory

$$\pi(p)v(x) + \pi(q)v(y)$$

5) Probability weighting



Russian Roulette Game

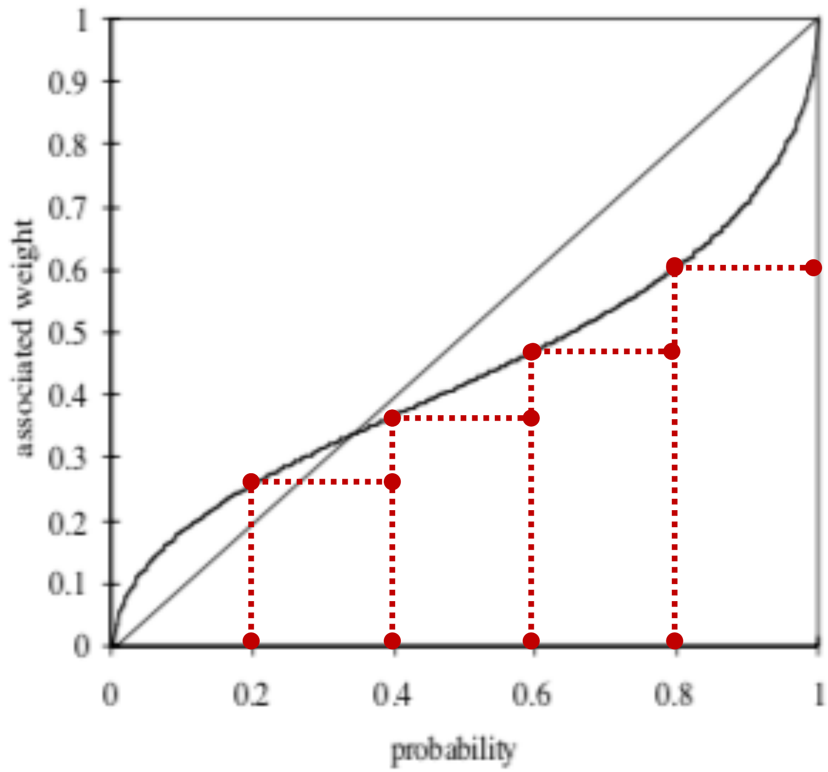
$$\pi(p)v(x) + \pi(q)v(y)$$

5) Probability weighting intuition

WTP from 5 to 4 bullets: WTP_54
WTP from 4 to 3 bullets: WTP_43
WTP from 3 to 2 bullets: WTP_32
WTP from 2 to 1 bullets: WTP_21
WTP from 1 to 0 bullets: WTP_10

Intuition:
WTP_54 > WTP_32
WTP_10 > WTP_32

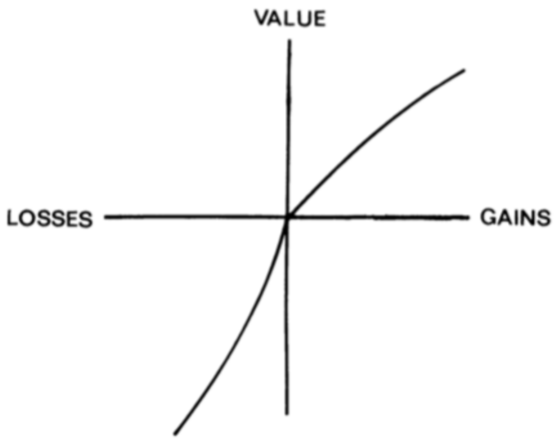
Example: gamble



Russian roulette game: trade your life with probabilities

Function forms of prospect theory

Tversky and Kahneman (1992)

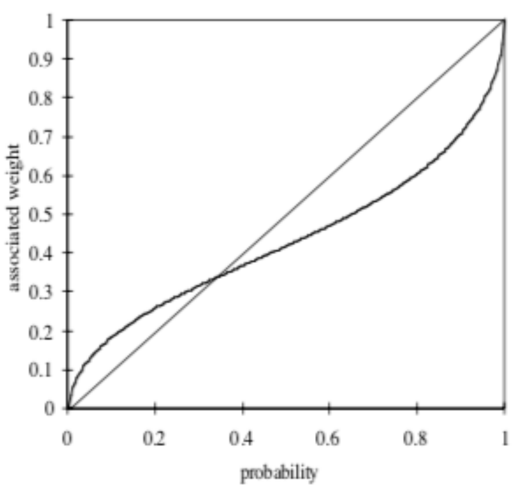


Value function:

$$v(x) = \begin{cases} (x - r)^{.88} & \text{if } x \geq r; \\ -2.25\,(-(x - r))^{.88} & \text{if } x < r, \end{cases}$$

Probability weighting function:

$$w(p) = \frac{p^{.65}}{\left(p^{.65} + (1 - p)^{.65}\right)^{1/.65}}$$



Other value functions

Other probability weighting functions

One open question: what is a right reference point?

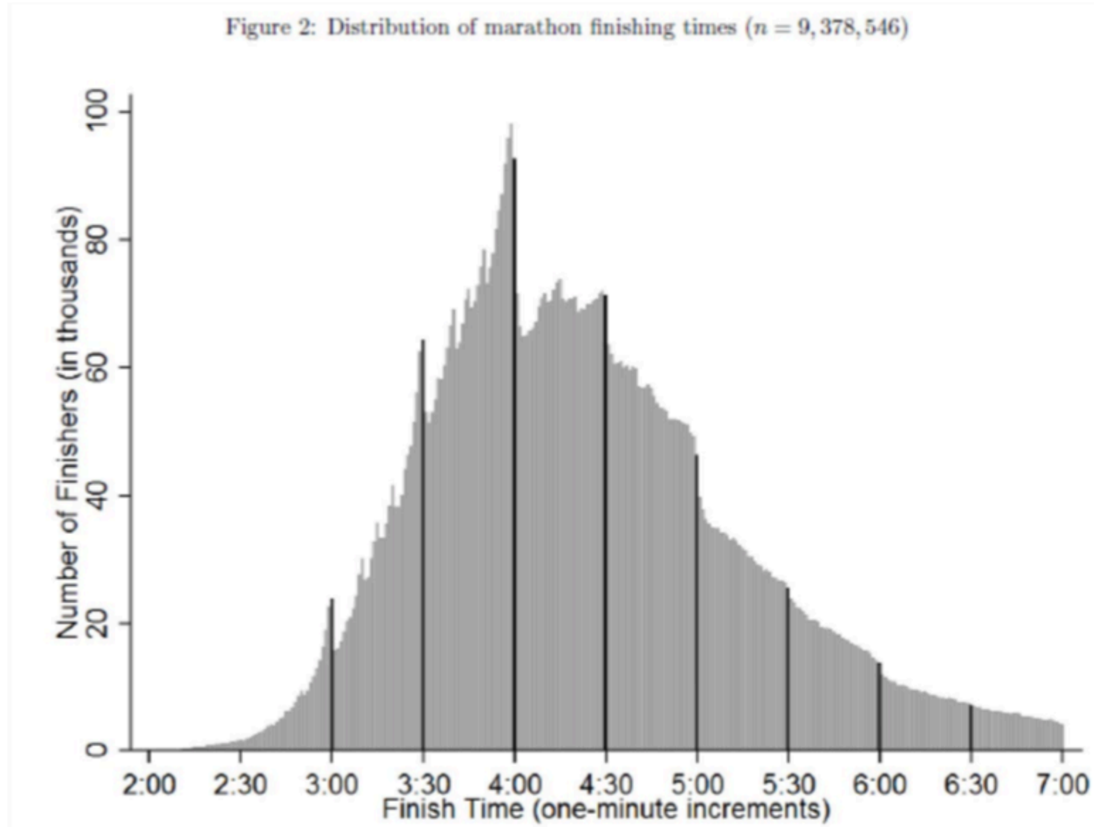
Possible reference points

- Status quo (PT 79)
- Past values/prices
- Aspirations/goals
- Social comparison
- Expectations

Critiques

- Overfitting/refutability/complexity (e.g. coin flip example)
- Model training: identification challenges

Four generations of PT

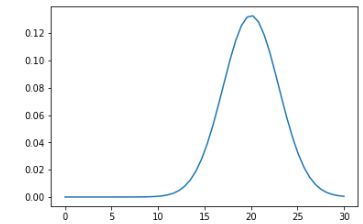


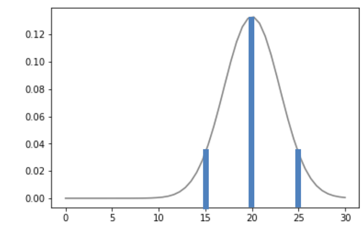
(Allen et al. 2014)

Part 2. Modeling travel time uncertainty in urban transportation

Different levels of uncertainty

1. Continuous probability distribution
2. Discrete probability distribution
3. Mean and variance
4. Min and max (or 5% and 95% percentiles)





Comments: current practice of using PT and DCM for time uncertainty

PT targets the first two cases (#1 and #2 uncertainty information).

PT is not commonly used in travel behavioral research.

DCMs with #3 and #4 uncertainty information are most common.

Travel mode choice with time uncertainty

Example 1. mean-variance model

$$U = \beta_T\, T + \beta_{SD}\, SD(T) + \beta_C C$$

Value of time (VOT): β_T/β_C;

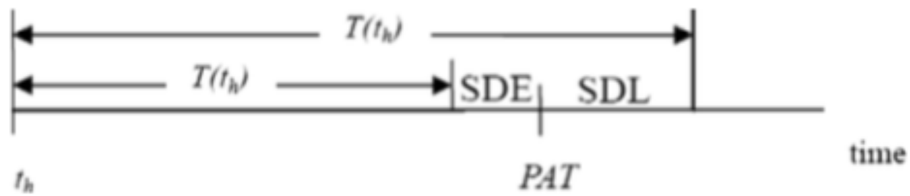
Value of reliability (**VOR**): β_{SD}/β_C;

Reliability ratio: VOR/VOT

Travel mode choice with time uncertainty

Example 2. scheduling model

$$U = \beta_T T + \beta_{SDE} SDE + \beta_{SDL} SDL + \beta_C C$$



Choice A

SDE = (7 + 4 + 1 + 0 + 0)/5 = 2.4

SDL = (0 + 0 + 0 + 5 + 9)/5 = 2.8

Related to PT

- Reference dependence: PAT
- Loss aversion
- Linear approximation to PT

PLEASE CIRCLE EITHER CHOICE **A** OR CHOICE **B**

Average Travel Time **9 minutes** You have an equal chance of arriving at any of the following times:	Average Travel Time **9 minutes** You have an equal chance of arriving at any of the following times:
7 minutes early **4 minutes early** **1 minute early** **5 minutes late** **9 minutes late**	**3 minutes early** **3 minutes early** **2 minute early** **2 minutes early** **On time**
Your cost: **\$0.25**	Your cost: **\$1.50**
Choice A	**Choice B**

Do people use PT in modeling travel time uncertainty?

Rarely seen: Li and Hensher (2017)

Reasons

1) We may not need the full PT in urban transportation. e.g. VOR
2) It is hard to estimate the full PT

Further steps: PT and DNN

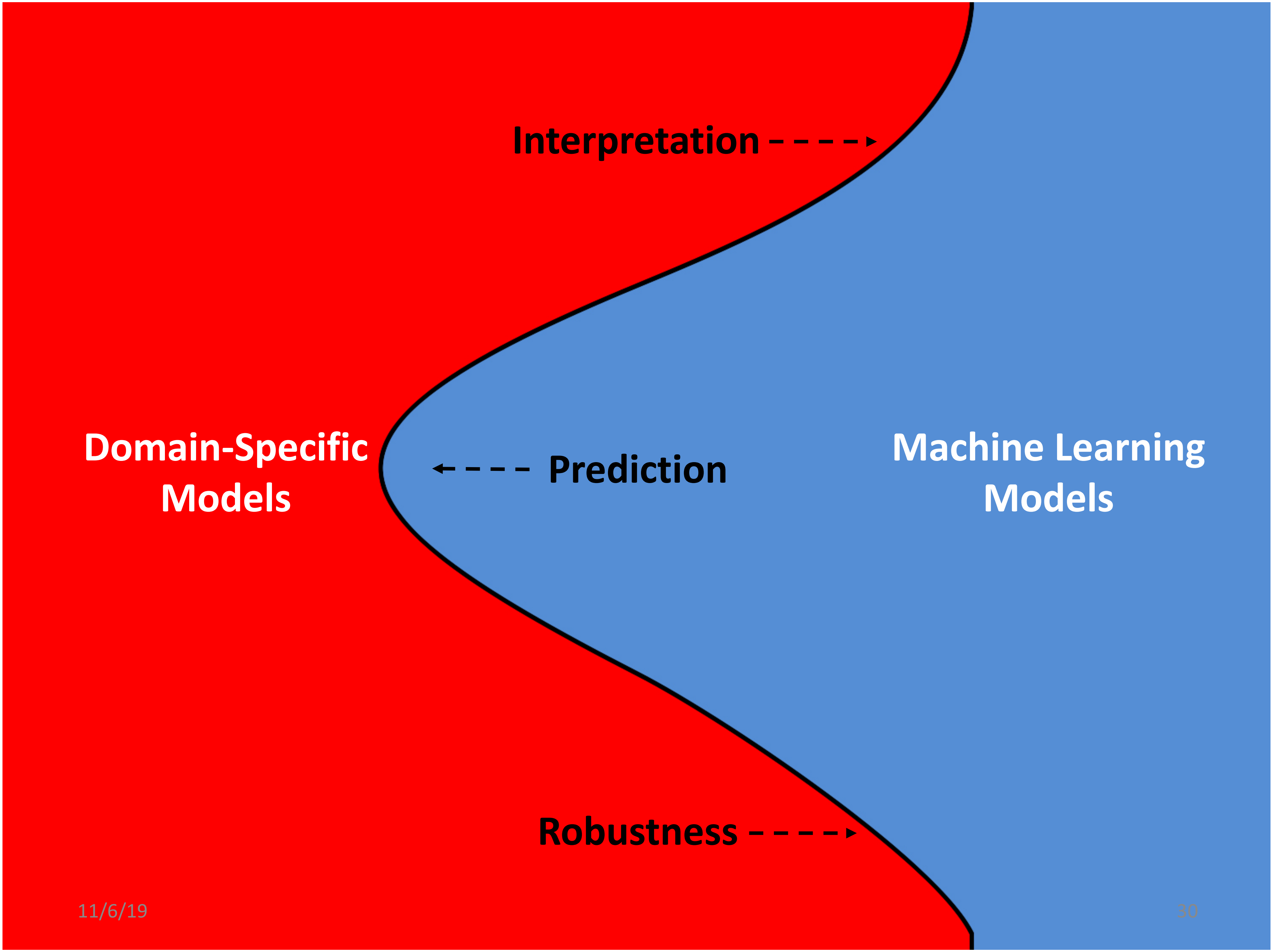
- A competitive view: can we use ML classifiers (DNN) to achieve higher prediction accuracy? Research is missing…
- **A complementary view: can we jointly use PT (or DCM) and DNN to achieve a better result?**

Part 3. Theory-based deep residual network for individual decision-making

Domain-Specific Models	Machine Learning Models
Spatial-temporal prediction	CNN/RNN/LSTM
Demand analysis (DCM, PT)	Supervised learning (DNN)
Network analysis	Graphical neural networks
Feedback & system control	Reinforcement learning

Domain-Specific Models

Machine Learning Models


Interpretation
Domain-Specific Models
Prediction
Machine Learning Models
Robustness

How to provide mutual benefits between domain-specific and generic-purpose models for individual decision-making?

$$V(x) = V_T(x) + \delta V_{DNN}(x)$$

1. δ controls the ratio between utility theory and DNN utilities. (Use λ regularization constant to implement it; λ is roughly the inverse of δ)

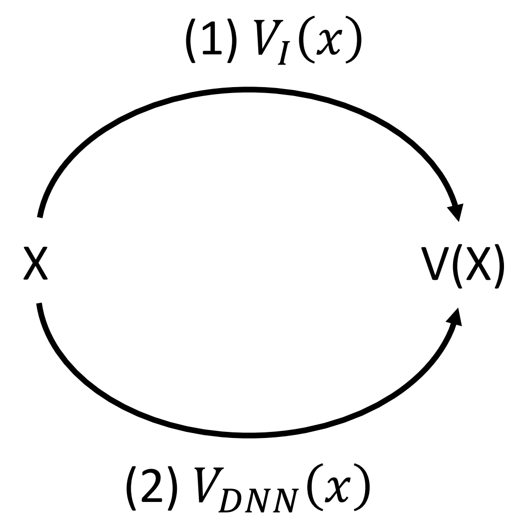
2. Two-stage training: (1) $V_T(x)$ and (2) $V_{DNN}(x)$
   - Information theory
   - Simultaneous training is unreasonable
   - "Politically correct"

3. Generic for any utility maximization framework and DNN architectures

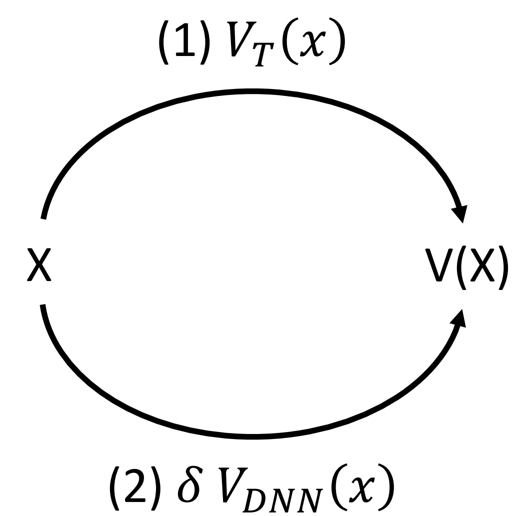
Theory-Based Residual Neural Network (TB-ResNet)

$$V(x) = V_T(x) + \delta V_{DNN}(x)$$

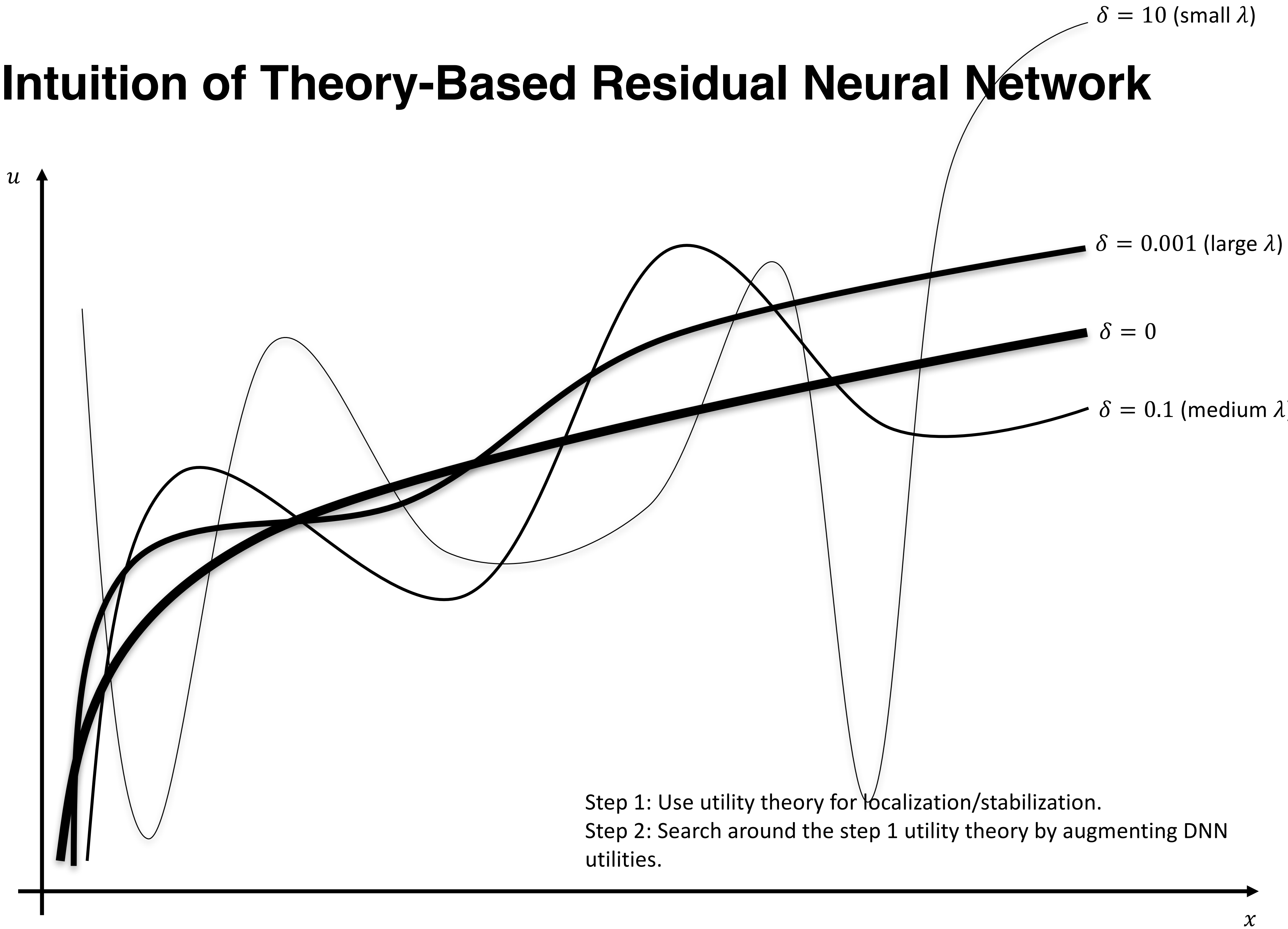
ResNet



TB-ResNet



Intuition of Theory-Based Residual Neural Network



Step 1: Use utility theory for localization/stabilization.
Step 2: Search around the step 1 utility theory by augmenting DNN utilities.

Three Instances of Theory-Based Residual Neural Network

Three Instances of TB-ResNets $\boldsymbol{V(x) = V_T(x) + \delta V_{DNN}(x)}$	1. CM-ResNet (choice modeling) ○ e.g. choose between K alternatives 2. PT-ResNet (prospect theory) ○ e.g. choose between two risky payoffs (x, p) 3. HD-ResNet (hyperbolic discounting) ○ e.g. temporal decisions (x, t)

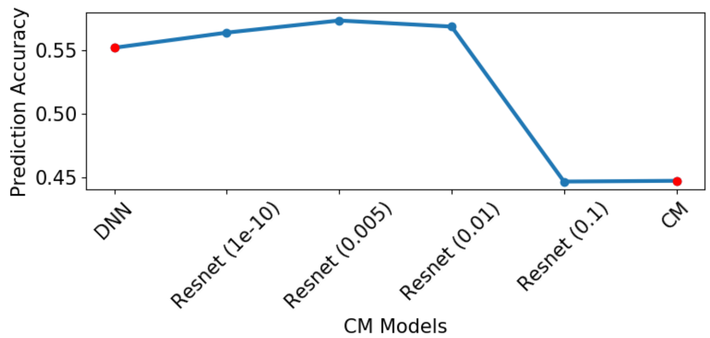
Comparing TB-ResNet to DNNs and Theories Based Model on Three Metrics

$$V(x) = V_T(x) + \delta V_{DNN}(x)$$

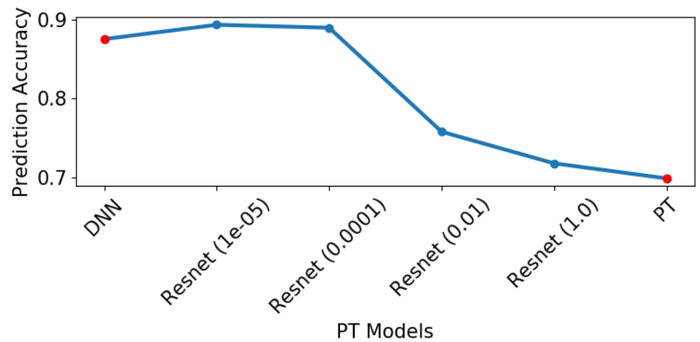
1. Prediction Accuracy
2. Interpretation (local information)
3. Robustness

1. Prediction Accuracy

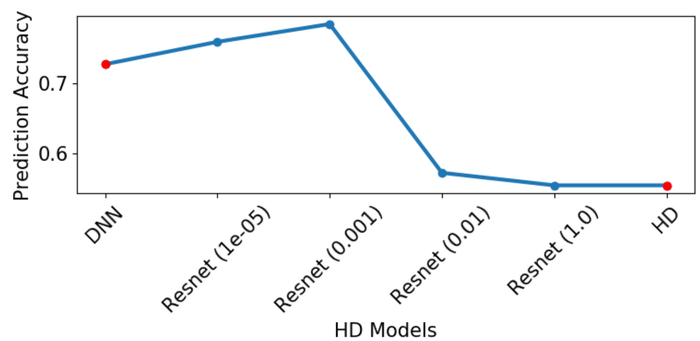
CM



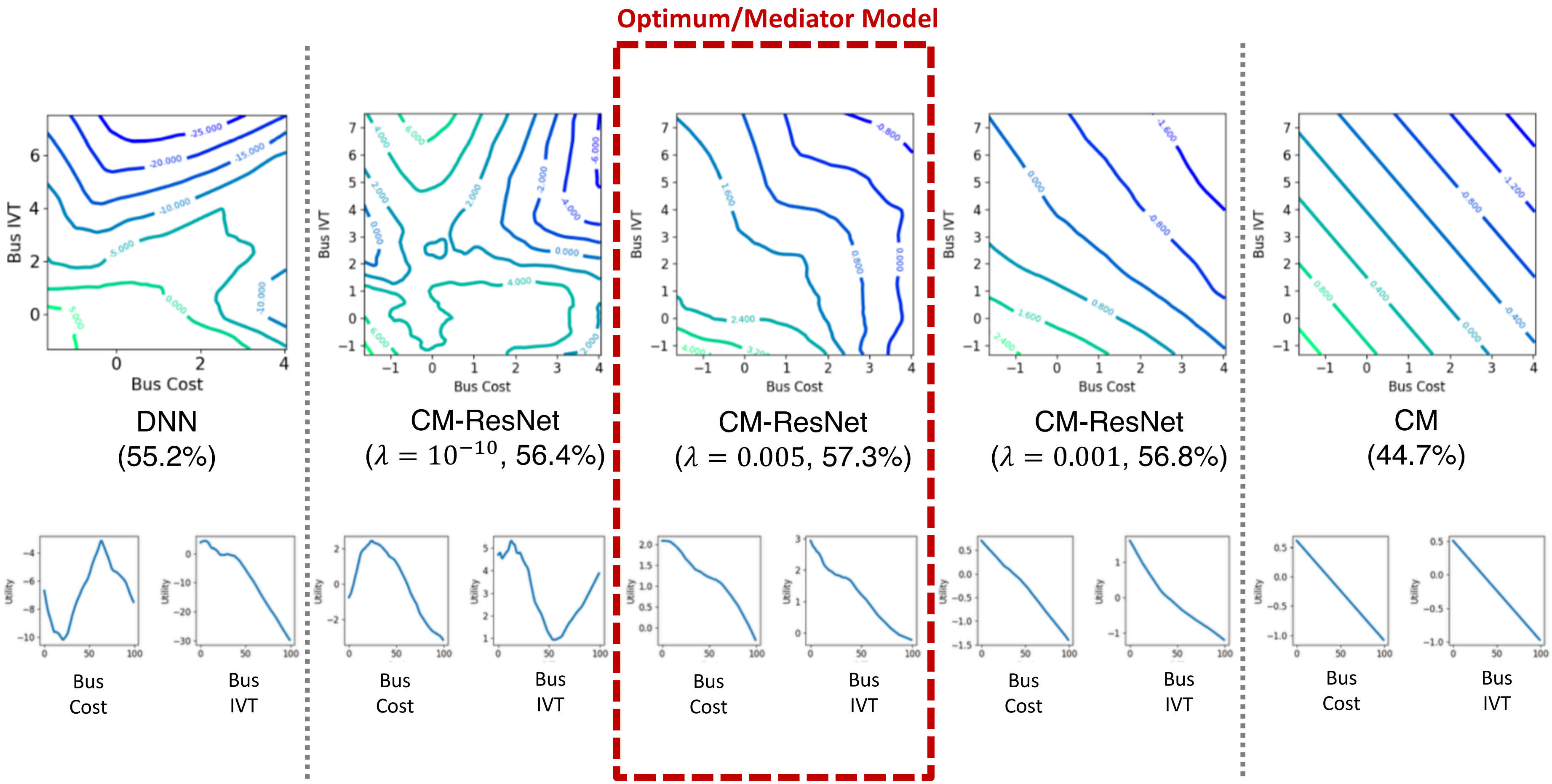
PT



HD

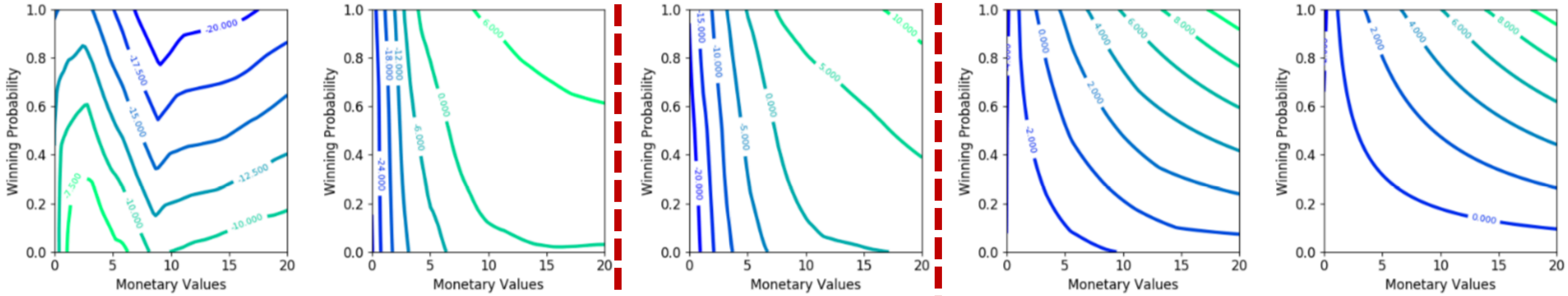


2. Interpretability of Utility Function in the CM Scenario

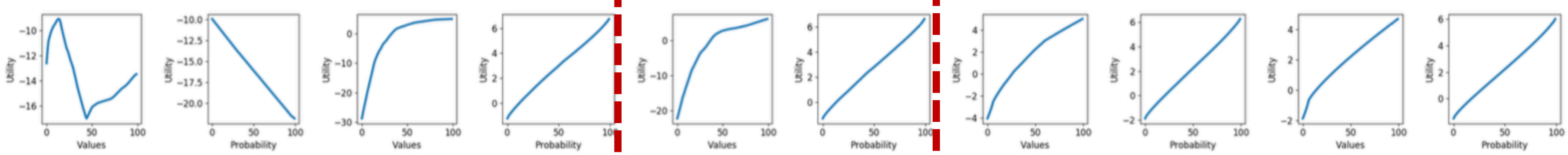


2. Interpretability of Utility Function in the PT Scenario

Optimum/Mediator Model



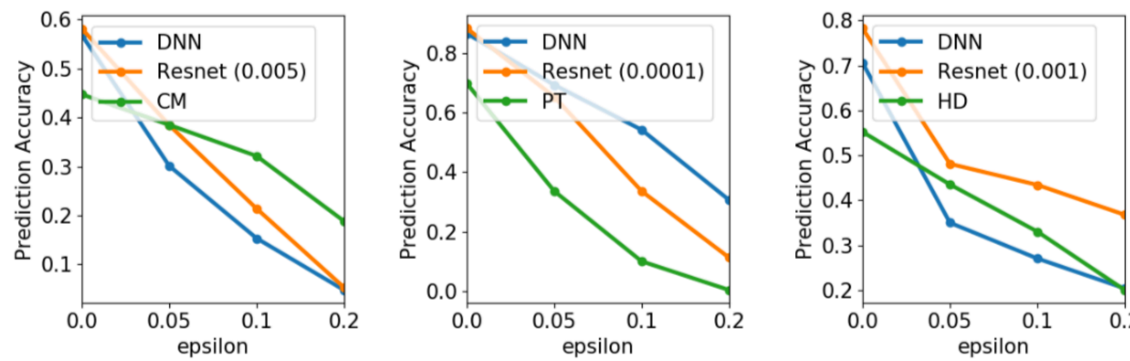
(a) DNN (87.5%) (b) PT Resnet (λ = $1e-5$; 89.3%) (c) PT Resnet (λ = 0.0001; 89.0%) (d) PT Resnet (λ = 0.01; 75.8%) (e) PT (69.9%)



(f) x0 (g) x1 (h) x0 (i) x1 (j) x0 (k) x1 (l) x0 (m) x1 (n) x0 (o) x1

3. Robustness

First Gradient Sign Method (FGSM) and
Target Gradient Sign Method (TGSM)



(a) CM FGSM (b) PT FGSM (c) HD FGSM

Comparing TB-ResNet to DCMs and DNNs

	Compare to CM, PT, and HD	Compare to DNNs
Prediction Accuracy	Significant Improvement (by addressing function misspecification)	Marginal Improvement (by localization and regularization)
Interpretability	Significant Improvement (by augmenting and enriching utility functions)	Significant Improvement (by stabilizing local information)
Robustness	NA	Significant Improvement (by stabilizing local information)

Conclusion

A neat and generic framework

Flexible combination of DCMs and DNNs

Analogy to ResNet

Provide mutual benefits to DCMs and DNNs

- Higher prediction accuracy
- Better interpretability (substitution patterns)
- Robust to various adversarial attacks (pointwise in-sample, out-of-sample, attacks beyond pointwise, etc.)

Future potentials: TB-ResNet for all of them?

Domain-Specific Models		Machine Learning Models
Spatial-temporal prediction	←- - - - - - - - - -→	CNN/RNN/LSTM
Demand analysis	←- - - - - - - - - -→	Supervised learning
Network analysis	←- - - - - - - - - -→	Graphical neural networks
Feedback & system control	←- - - - - - - - - -→	Reinforcement learning

Part 4. Multitask Learning & Transfer Learning

Baseline, Transfer Learning, and Multitask Learning

Reality always involve multiple similar tasks.

Examples (travel mode choice)

Target task: travel mode choice in MA. Source task: travel mode choice in CA. (Geographical difference)

Target task: travel mode choice in 2010. Source task: travel mode choice in 2000. (Temporal difference)

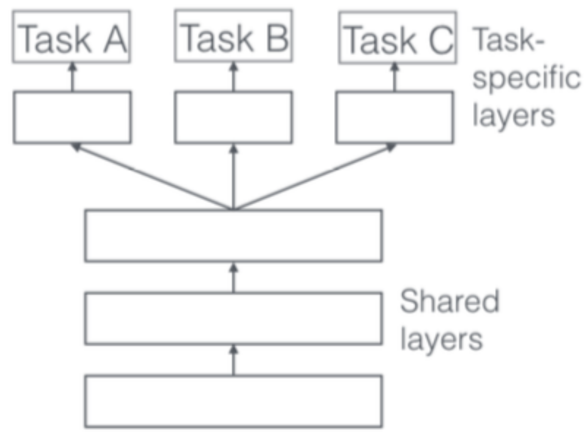
Target task: auto ownership in MA. Source task: travel mode choice in MA. (Output difference)

Target task: travel mode choice with an experiment. Source task: travel mode choice with NHTS dataset. (Dataset difference)

Target task: field experiment for travel mode choice. Source task: some lab experiment for travel mode choice. (Procedure difference)

	Training	**Testing**
Baseline machine learning	Task 1	Task 1
Transfer learning (TL)	Task 1	Task 2
Multitask learning (MTL)	Task 1 & Task 2	Task 1 & Task 2

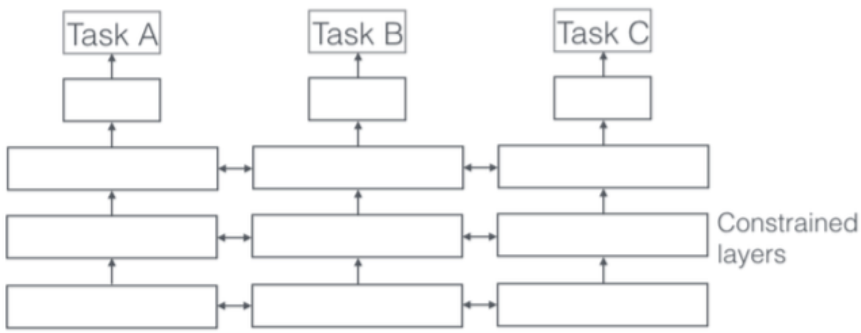
Multitask learning baseline (Caruana, 1997)



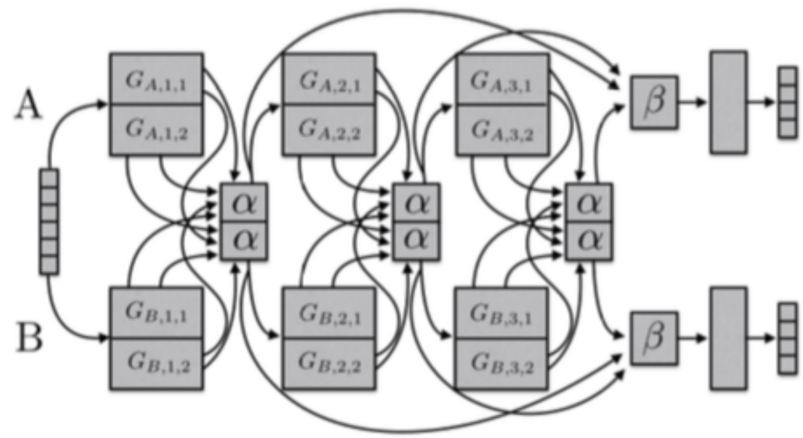
Key intuition: control similarities and differences

Multitask learning examples

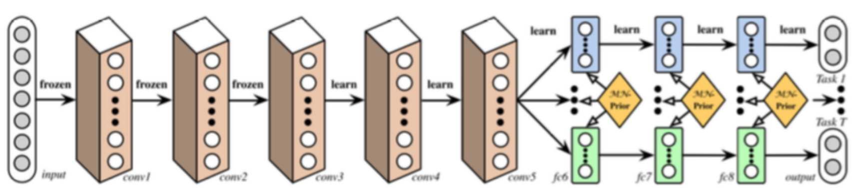
Duong et al., 2015



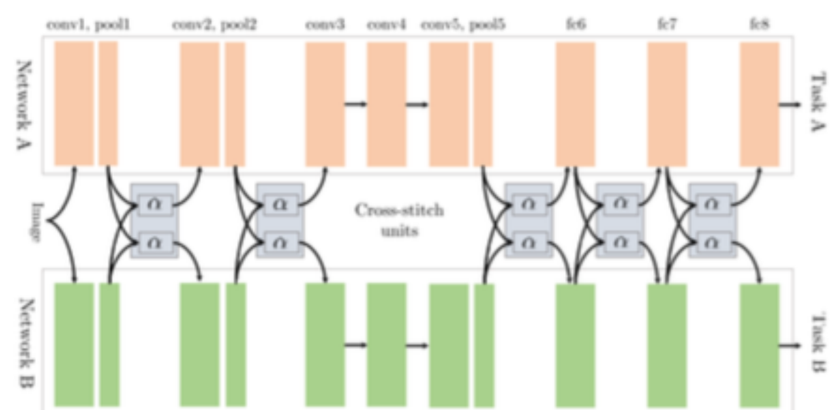
Ruder et al., 2017



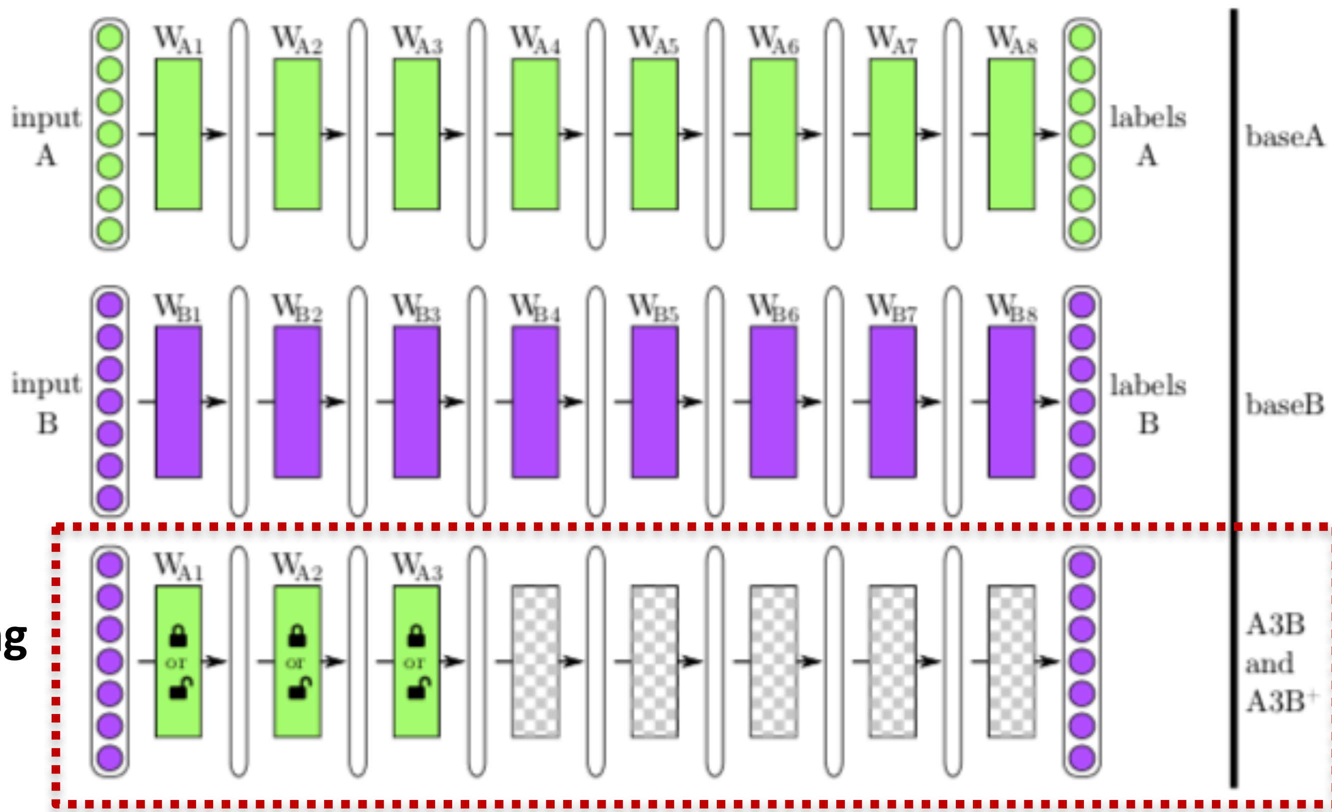
Long and Wang, 2015



Misra et al., 2016



A baseline transfer learning example (Yosinski et al., 2014)

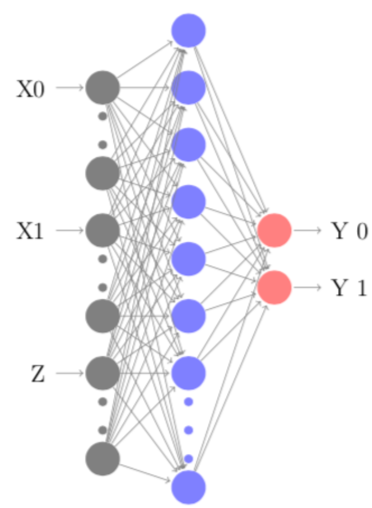


Transfer Learning DNN

1. Freeze lower layers
2. Initialize lower layers

Transfer learning example: intuition

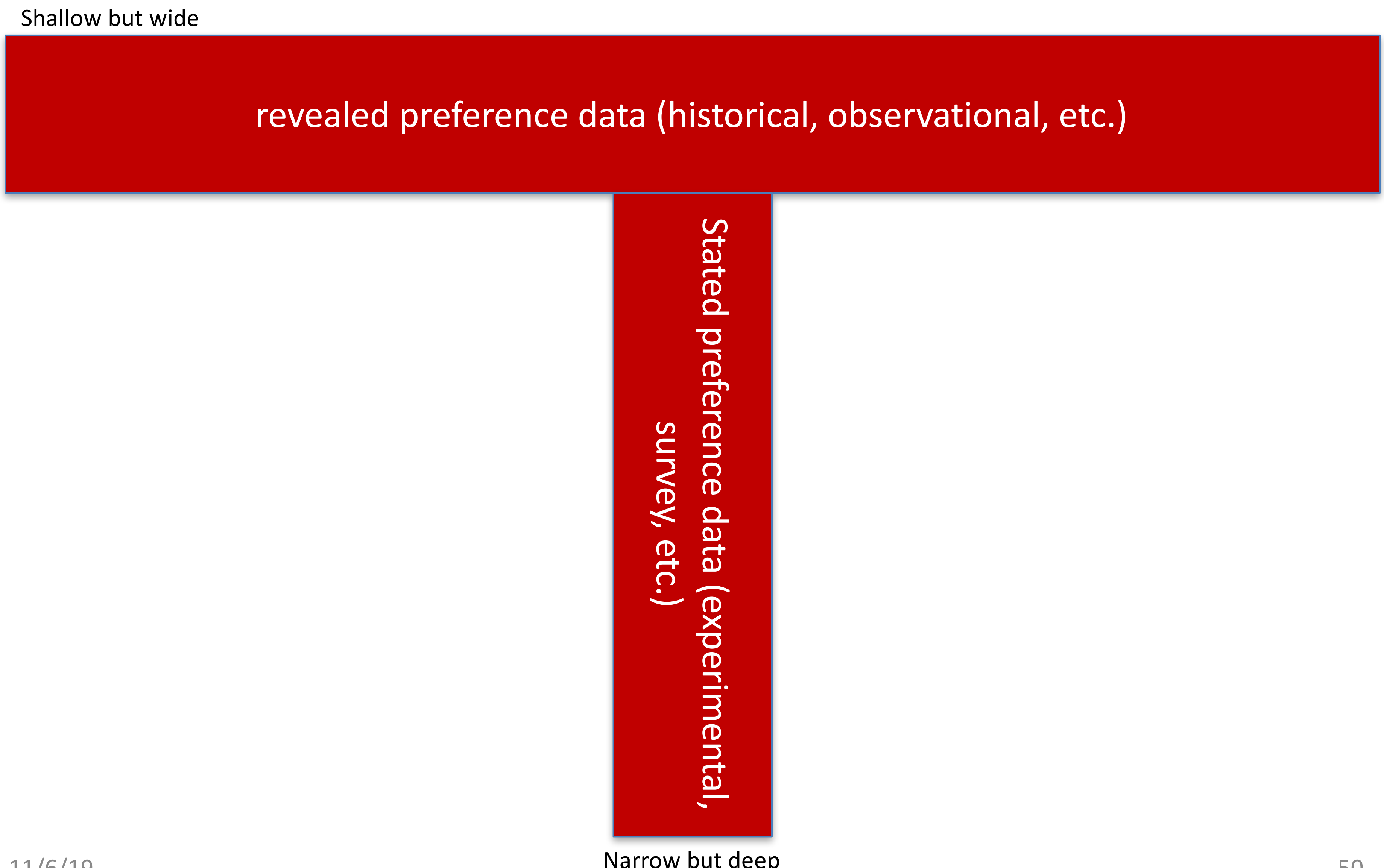
1. Classical Frequentist choice models as freezing.
2. Classical Bayesian models as initialization.



$$U(k) = \boldsymbol{\beta}^{\boldsymbol{T}} \phi(x_k, z) + \epsilon_k$$

Part 5. Multitask Learning Deep Neural Networks to Combine Revealed and Stated Preference Data

"T-shaped" datasets for the demand analysis of new product/service (e.g. AV)



Background

RP+SP as a Classical Question

- Pros and cons of RP and SP (Ben-Akiva et al., 1994; Hausman et al., 1998)
- Joint RP+SP (Ben-Akiva et al., 1994; Hensher and Bradley, 1993; Polydoropoulou and Ben-Akiva, 1994)
- Nested logit model as one classical method (Hensher and Bradley, 1993; Louviere et al. 1999)

MTLDNNs as a New Method

- A multitask learning perspective
- "Multiple tasks arise naturally…" (Caruana, 1997)
- Wide applications: NLP (Collobert and Weston, 2008; Hashimoto et al. 2016); healthcare drug discovery (Ramsundar et al. 2015); etc.

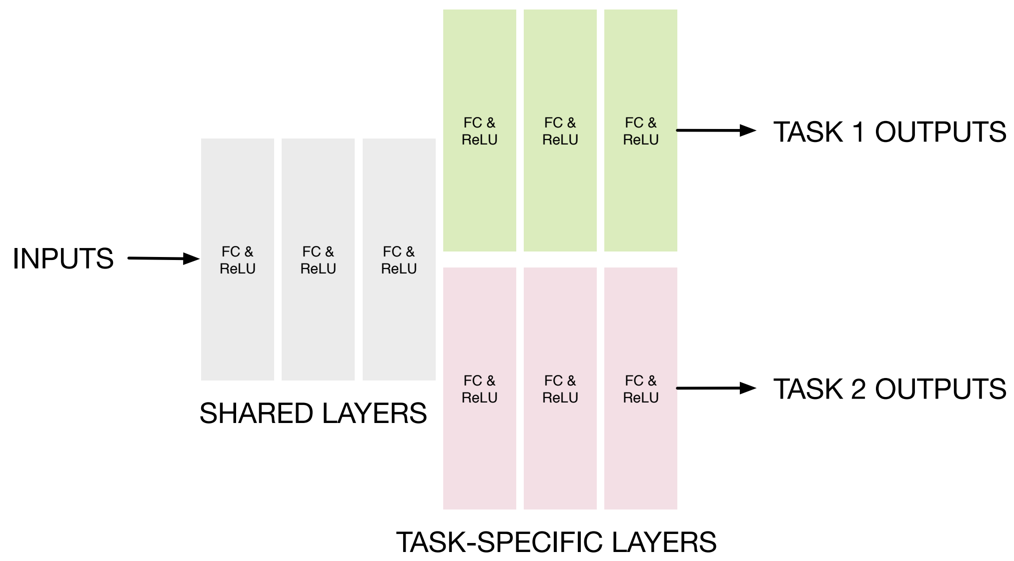
A MTLDNN Example (Caruana, 1997)

Block: layers in DNNs

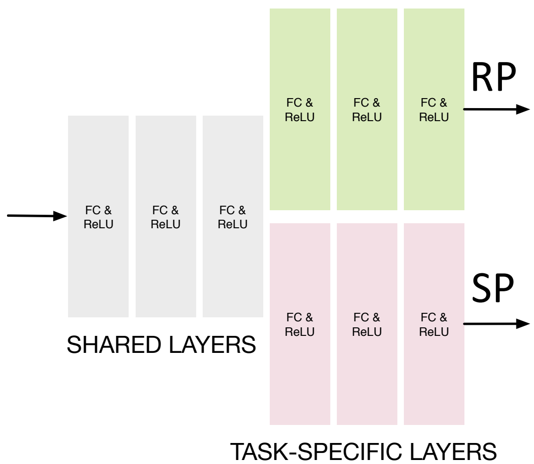
Grey: **shared** layers

Green/Red layers: **task-specific** layers

Flexible MTLDNN architecture: different depth and width



Formulation of MTLDNNs



Feature Transformation

$$V_{k_r,i} = (g_r^{M_2,k_r} \circ g_r^{M_2-1} \circ ... \circ g_r^1) \circ (g_0^{M_1} \circ g_0^{M_1-1} \circ ... \circ g_0^1)(x_{r,i})$$

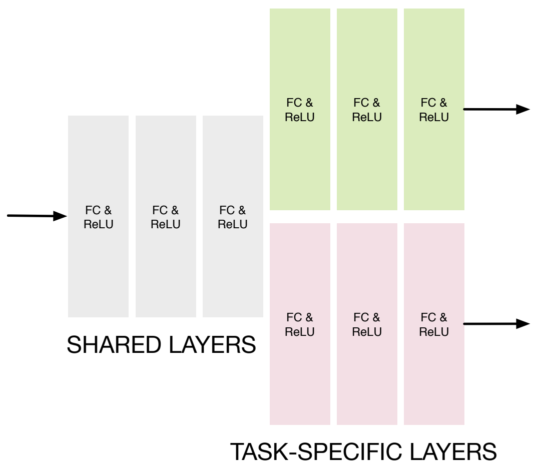
$$V_{k_s,t} = (g_s^{M_2,k_s} \circ g_s^{M_2-1} \circ ... \circ g_s^1) \circ (g_0^{M_1} \circ g_0^{M_1-1} \circ ... \circ g_0^1)(x_{s,t})$$

Softmax Activation

$$P(y_{k_r,i}; w_r, w_0) = \frac{e^{V_{k_r,i}}}{\sum_{j_r=1}^{K_r} e^{V_{j_r,i}}}$$

$$P(y_{k_s,t}; w_s, w_0, T) = \frac{e^{V_{k_s,t}/T}}{\sum_{j_s=1}^{K_s} e^{V_{j_s,t}/T}}$$

Formulation of MTLDNNs



Empirical Risk Minimization

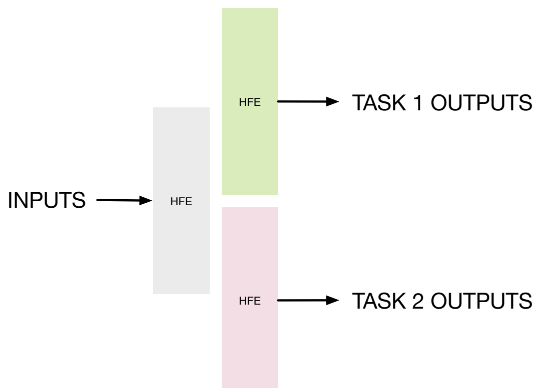
$$\min_{w_r, w_s, w_0, T} R(X, Y; w_r, w_s, w_0, T; c_H) =$$

1 Cross-Entropy Loss

$$\min_{w_r, w_s, w_0, T} \Bigg\{ \boxed{- \frac{1}{N_r} \sum_{i=1}^{N_r} \sum_{k_r=1}^{K_r} y_{k_r} \log P(y_{k_r,i}; w_r, w_0; c_H) - \frac{\lambda_0}{N_s} \sum_{t=1}^{N_s} \sum_{k_s=1}^{K_s} y_{k_s} \log P(y_{k_s,t}; w_r, w_0, T; c_H)} + \boxed{\lambda_1 ||w_0||_2^2 + \lambda_2 ||w_s||_2^2 + \lambda_3 ||w_s - w_r||_2^2} \Bigg\}$$

2 Regularizations: Scale Controls

Formulation of NLs



Feature Transformation

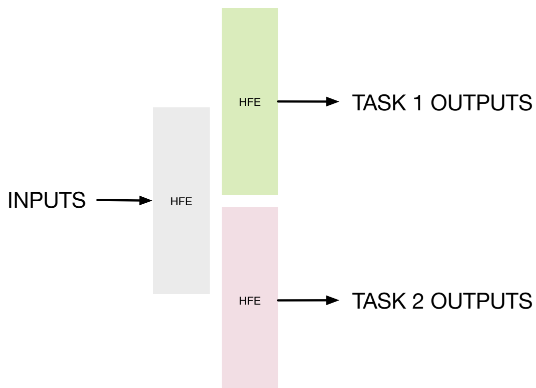
$$V_{k_r,i} = \beta_{k_r}^T \phi(x_{r,i})$$
$$V_{k_s,t} = \beta_{k_s}^T \phi(x_{s,t})$$

Softmax Activation

$$P(y_{k_r,i};\beta_r) = \frac{e^{\beta_{k_r}^T \phi(x_{r,i})}}{\sum_{j_r=1}^{K_r} e^{\beta_{j_r}^T \phi(x_{r,i})}}$$

$$P(y_{k_s,t};\beta_s) = \frac{e^{\beta_{k_s}^T \phi(x_{s,t})/\theta}}{\sum_{j_s=1}^{K_s} e^{\beta_{j_s}^T \phi(x_{s,t})/\theta}}$$

Formulation of NLs



Empirical Risk Minimization

$$\min_{\beta_r,\beta_s} R(X,Y;\beta_r,\beta_s) =$$

1. Cross-Entropy Loss

$$\min_{\beta_r,\beta_s}\boxed{\Big\{ -\frac{1}{N}\Big[\sum_{i=1}^{N_r}\sum_{k_r=1}^{K_r} y_{k_r,i}\log P(y_{k_r,i};\beta_r) + \sum_{t=1}^{N_s}\sum_{k_s=1}^{K_s} y_{k_s,t}\log P(y_{k_s,t};\beta_s)\Big]\Big\}}$$

MTLDNNs are More Generic than NLs.

MTLDNNs

1. Automatic feature learning
2. "Soft" constraints to describe the similarities between RP and SP
   - Architectural design (e.g. # of shared vs. task-specific layers)
   - Regularizations (e.g. λ_3)

NLs

1. Handcrafted feature learning
2. "Hard" constraints to describe the similarities between RP and SP
   - Shared vs. task-specific parameters (e.g. β_r vs. β_s)

Experiment Setup: Data and Training

Dataset: online survey collected in Singapore

- RP: four travel modes (walking, public transit, ridesharing, and driving)
- SP: add AV

Sample size: RP (1,592) + SP (8,418)

Training vs. testing sets (4:1)

Hyperparameter searching and comparison for MTLDNNs

- Depth & width of MTLDNN architectures
- Regularization constants

Experiment Setup: Comparing Four Groups (Eight Models)

1. Top 1 MTLDNN (MTLDNN)
2. Top 10 MTLDNN Ensemble (MTLDNN-E)

3. Feedforward DNN separately trained for RP and SP (DNN-SPT)
4. Feedforward DNN jointly trained for RP and SP (DNN-JOINT)

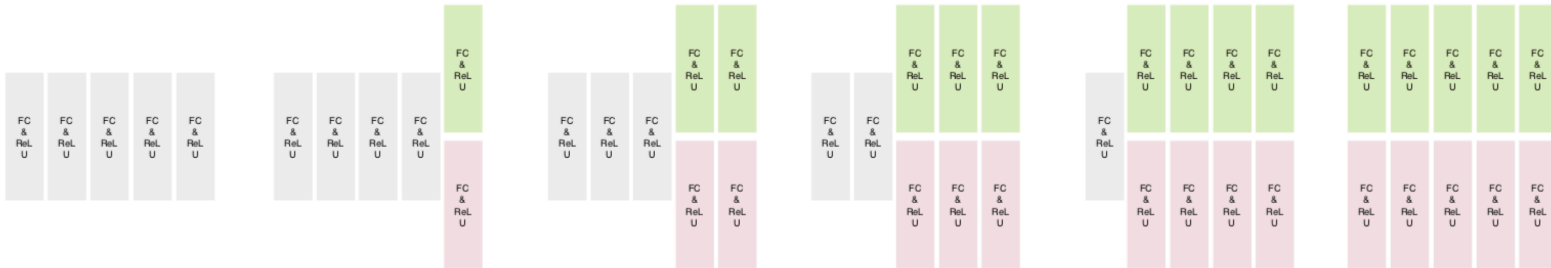
5. Nested logit model with full parameter constraints (NL-C)
6. Nested logit model without parameter constraints (NL-NC)

7. Multinomial logit model separately trained for RP and SP (MNL-SPT)
8. Multinomial logit model jointly trained for RP and SP (MNL-JOINT)

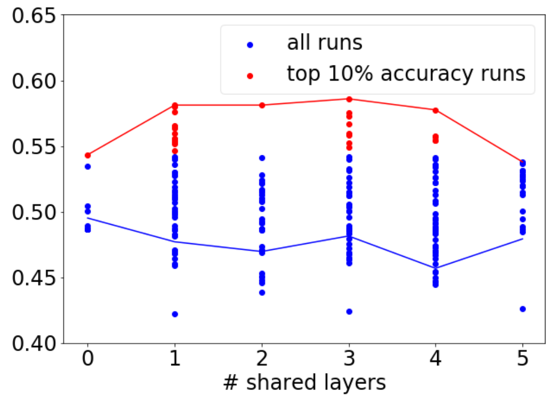
1) Prediction: MTLDNNs perform better than NLs by about 5% prediction accuracy

	MTLDNN (Top1)	MTLDNN-E (Top10)	DNN-SPT	DNN-JOINT	NL-C	NL-NC	MNL-SPT	MNL-JOINT
			Panel 1: Prediction Accuracy					
Joint RP+SP (Testing)	60.0%	58.7%	53.4%	53.8%	55.4%	55.0%	55.0%	51.9%
RP (Testing)	69.9%	66.6%	65.8%	65.8%	65.4%	64.7%	64.5%	44.0%
SP (Testing)	58.2%	57.2%	51.1%	51.5%	53.5%	53.2%	53.2%	53.5%

2) Causes: the soft constraints specific to multitask learning are effective in improving prediction accuracy



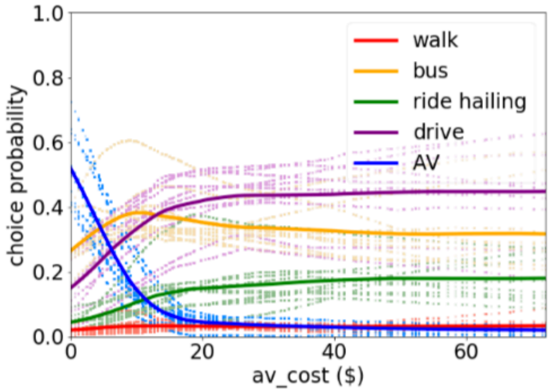
(a) Six Different Architectures: (5-0);(4-1);(3-2);(2-3);(1-4);(0-5)



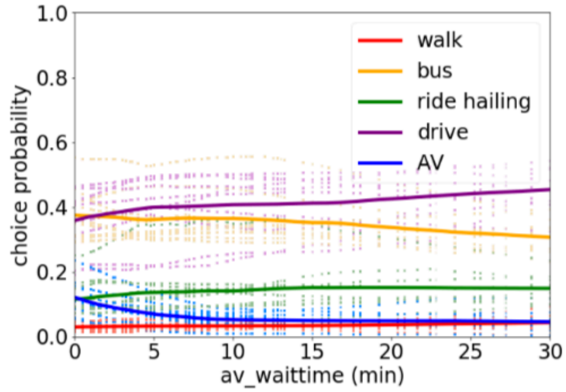
Performance of Six Architectures

We should not naively use feedforward DNN architectures. Model design specific to multitask learning is important!

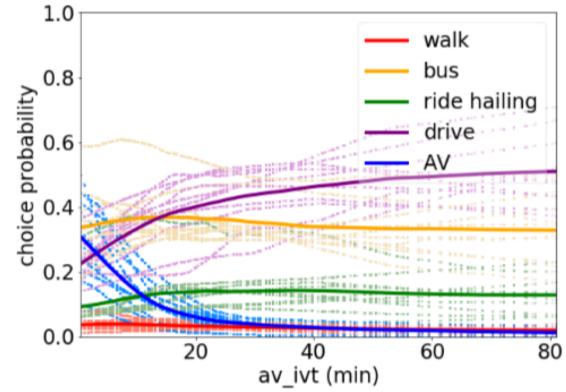
3) Interpretation: extracting the substitution patterns of AVs with other alternatives from MTLDNNs



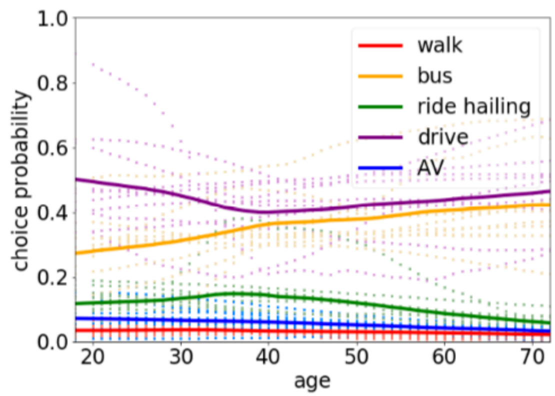
(a) AV Cost



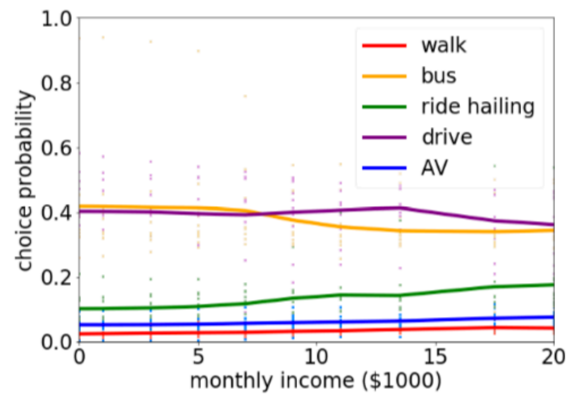
(b) AV Wait Time



(c) AV In-Vehicle Time



(d) Age



(e) Income

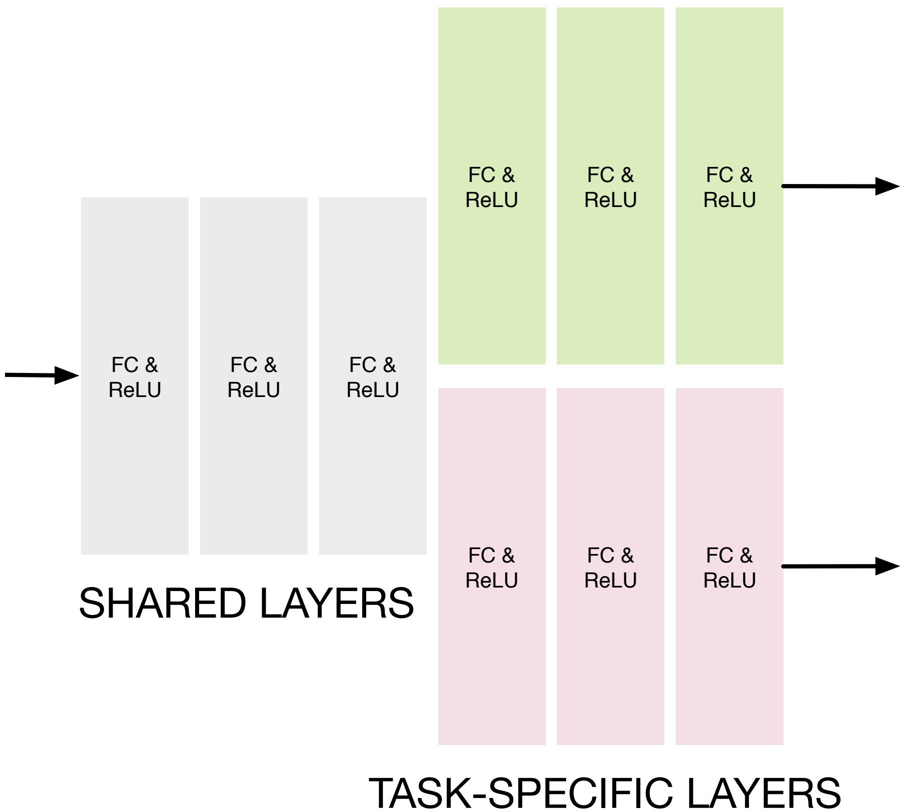
3) Interpretation: rank the importance of input variables by computing elasticities for AV adoption

Variable	Elasticity
AV Cost	-0.981
AV In-Vehicle Time	-0.905
Age	-0.561
AV Wait Time	-0.375
Income	0.102

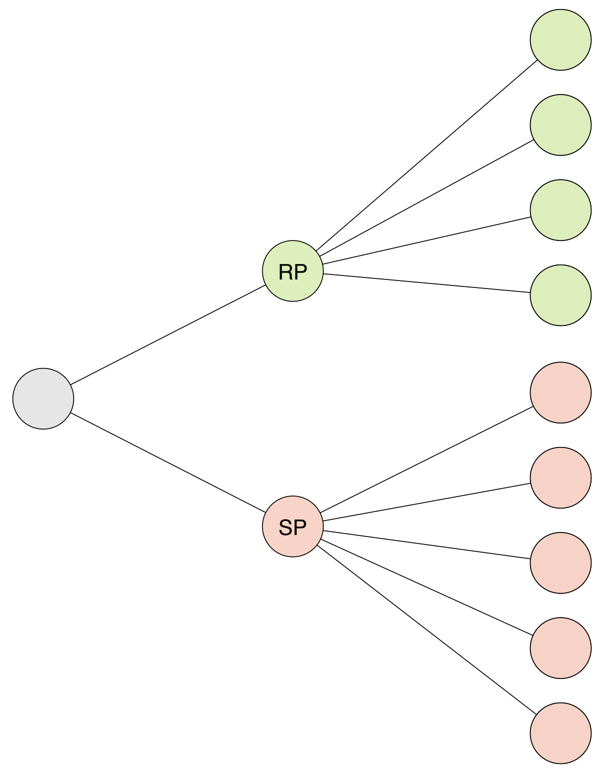
Elasticity Table

An intriguing question: MTLDNNs and NLs

MTLDNN Visualization



NL Visualization



Future Studies

Other applications

- Data fusion (e.g. across cities, states, etc.)
- Joint decisions (e.g. activity pattern, mode choice, etc.)
- More than two tasks.
- etc.

Other MTLDNN architectures

Using the transfer learning framework

Summary

1. **Introduce: MTLDNNs and RP&SP**

2. **MTLDNNs are more general than NLs**

3. **Results**

   - **Empirically MTLDNNs outperform NLs in prediction**

   - **The better performance can be attributed to the soft constraints (e.g. architectures & regularizations)**

   - **MTLDNN provides valuable information for AV adoption.**

4. **Future directions: other MTLDNNs and applications**

End & Thank You