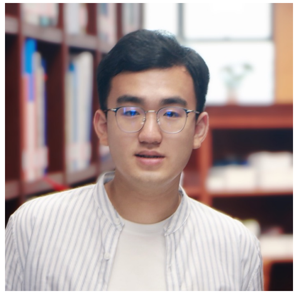
Rank-N-Contrast: Learning Continuous Representations for Regression


Kaiwen Zha[1,*]


Peng Cao[1,*]


Jeany Son[2]


Yuzhe Yang[1]


Dina Katabi[1]

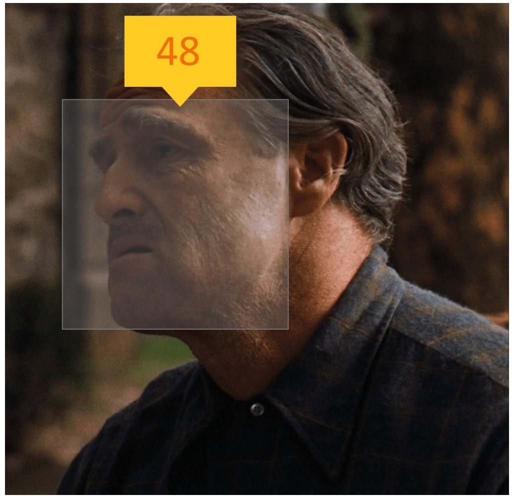
[1] MIT CSAIL, [2] GIST * Equal Contribution

NeurIPS 2023 (Spotlight)

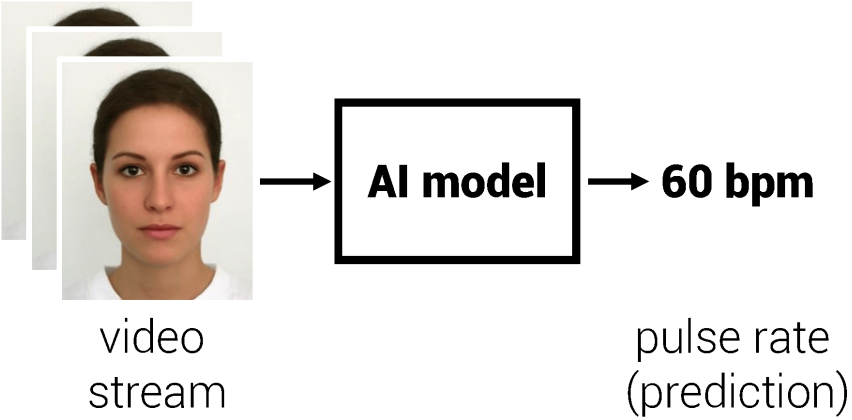
Regression problems are common in the real world

A regression problem: Predicting the target $y \in \mathbb{R}^{d_t}$ based on the input data $x \in \mathcal{X}$.

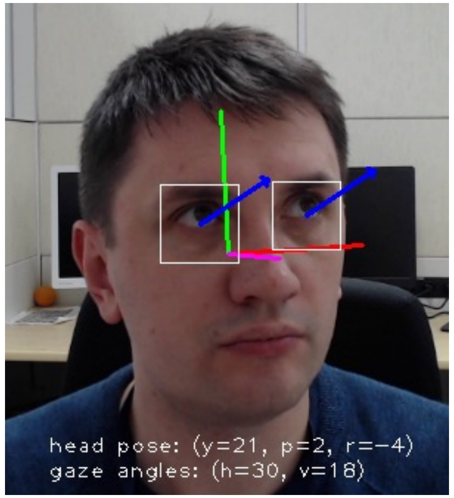
Age Estimation



Prediction of a Health Metric



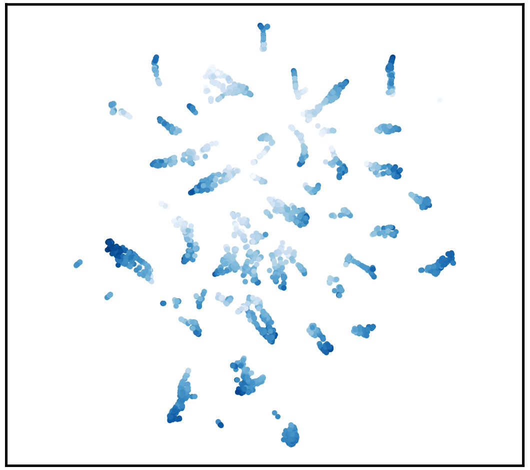
Gaze Direction Detection



Prior Methods → Fragmented Representation

Example: Predict the temperature from outdoor webcam images

L_1

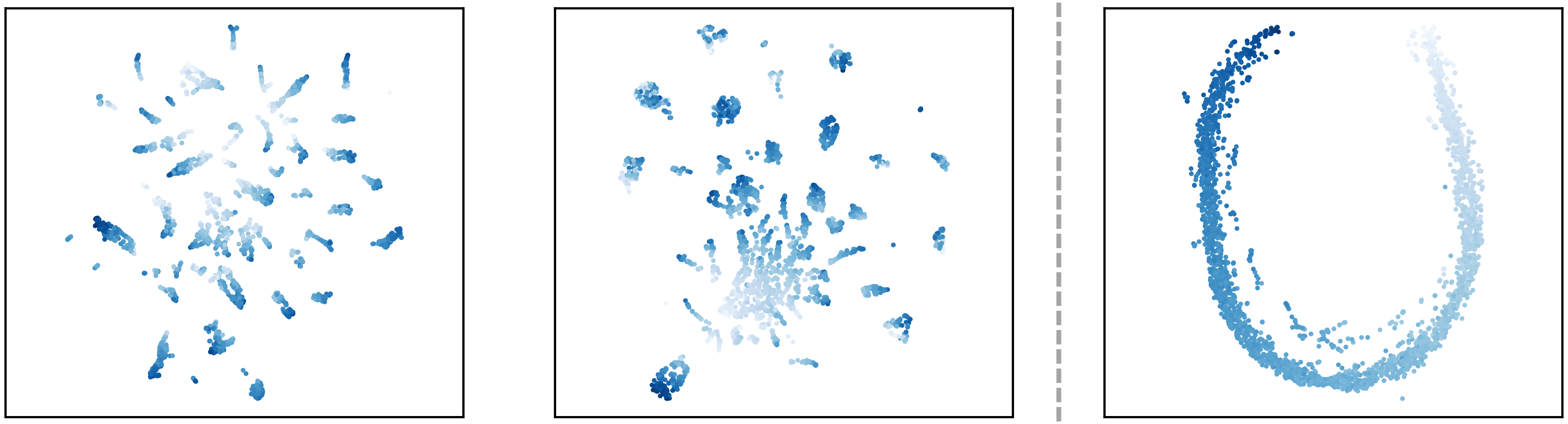


Fragmented representation fails to capture that the temperature is continuous.

Prior Methods → Fragmented Representation

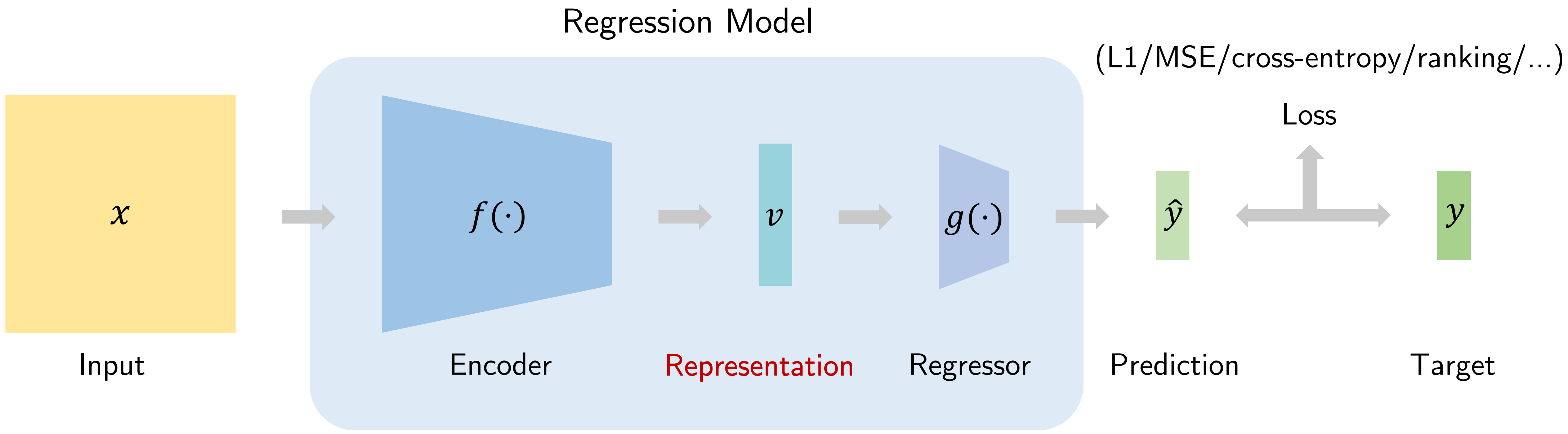
Example: Predict the temperature from outdoor webcam images

L_1 | SupCon | RnC (Ours)



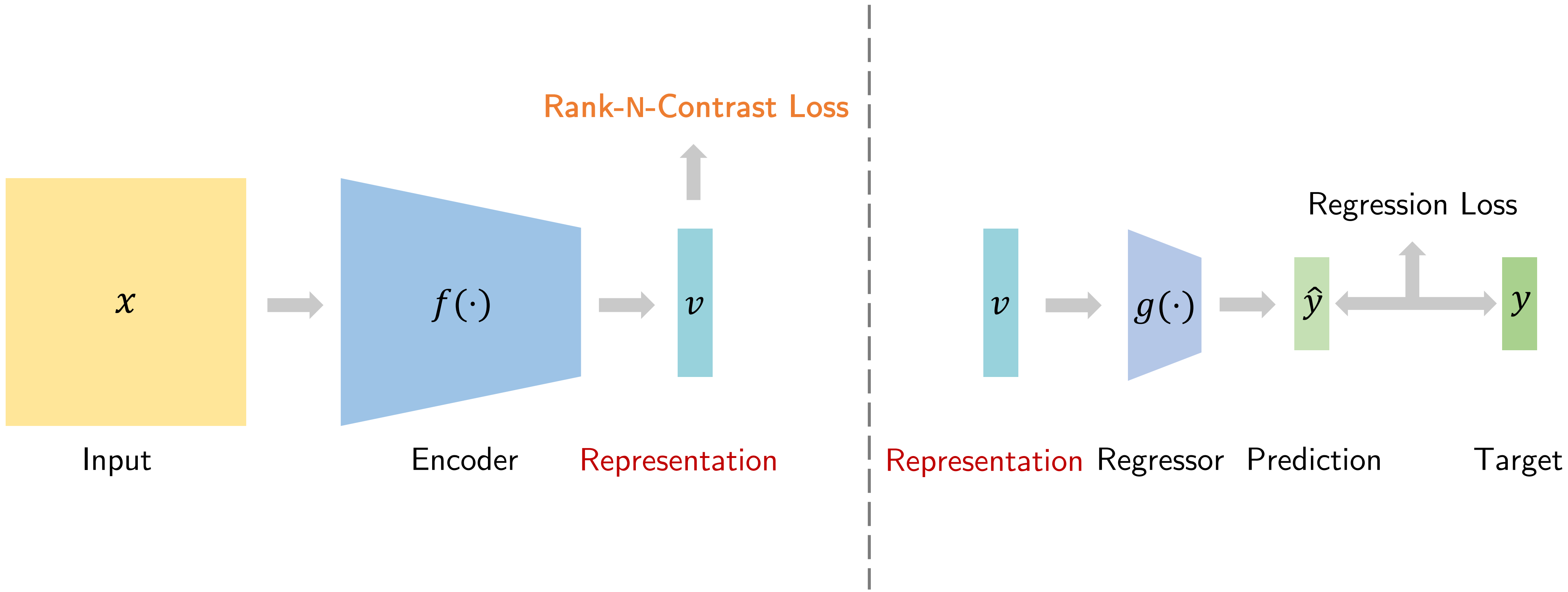
In RnC, the representation is continuous.

Traditional Regression Learning



Prior regression learning methods constrains the final predictions, not the representation.

Our Approach: Rank-N-Contrast (RNC)

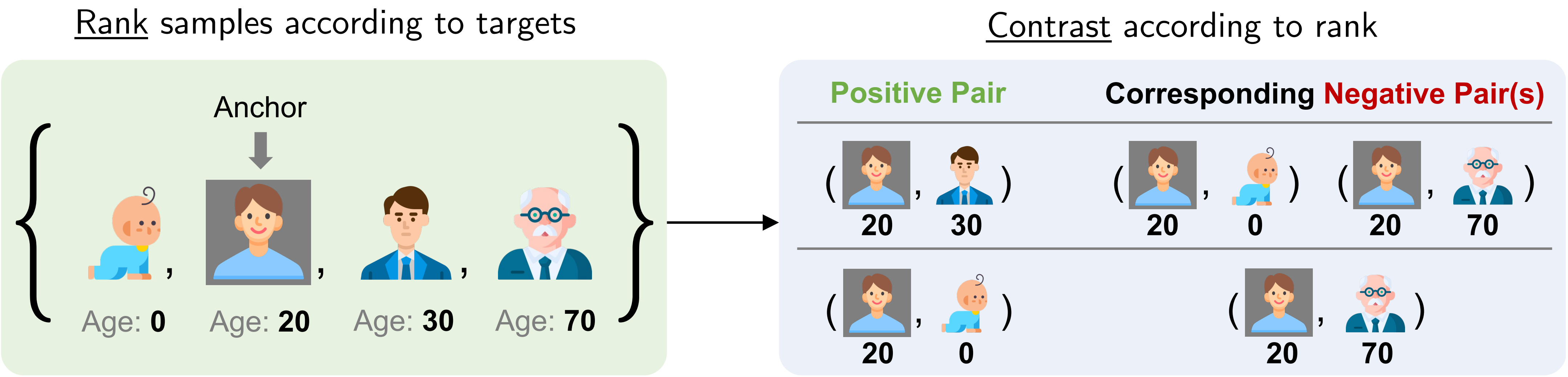


We first train an encoder $f(\cdot): X \to \mathbb{R}^{d_e}$ with the Rank-N-Contrast Loss, and then fix the encoder to train a regressor $g(\cdot): \mathbb{R}^{d_e} \to \mathbb{R}^{d_t}$ with a regression loss.

Rank-N-Contrast Loss

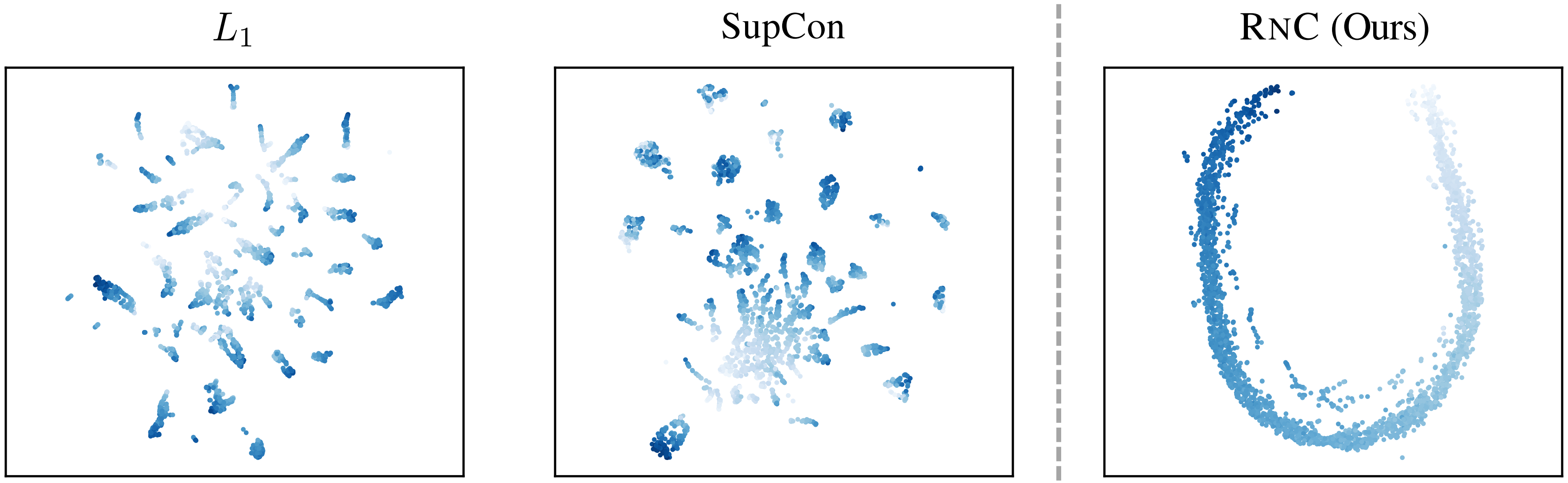
$$\mathcal{L}_{\text{RNC}} = \frac{1}{2N}\sum_{i=1}^{2N} l_{\text{RNC}}^{(i)} = \frac{1}{2N}\sum_{i=1}^{2N}\frac{1}{2N-1}\sum_{j=1,\ j\neq i}^{2N} -\log \frac{\exp(\text{sim}(\boldsymbol{v}_i,\boldsymbol{v}_j)/\tau)}{\sum_{\boldsymbol{v}_k\in\mathcal{S}_{i,j}}\exp(\text{sim}(\boldsymbol{v}_i,\boldsymbol{v}_k)/\tau)}.$$

$S_{i,j}$: the set of samples that are of higher ranks than $\boldsymbol{v}_j$ in terms of target distance with respect to $\boldsymbol{v}_i$.

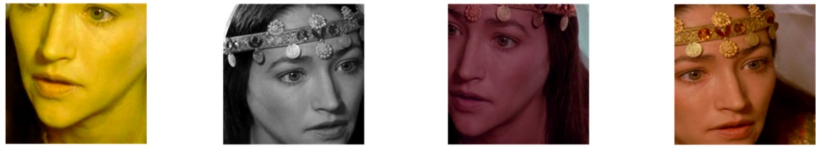
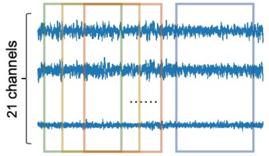
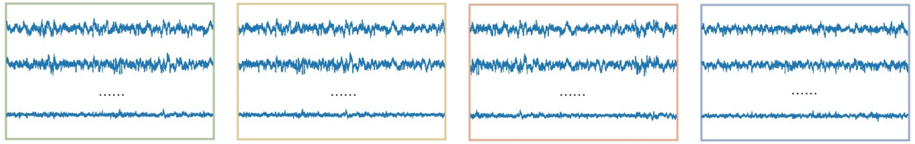
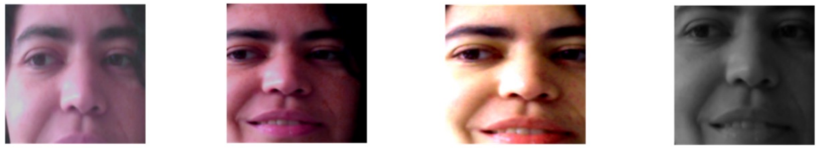
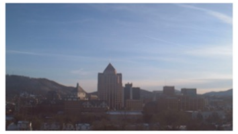
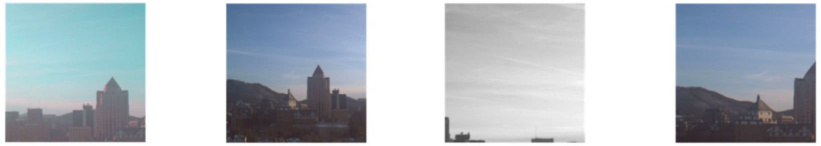


Learned representations by RNC

- We theoretically prove that the Rank-N-Contrast Loss learns continuous representations that are ordered according to targets, leading to a better generalization bound.
- Empirically, RNC delivers representations with the intrinsic order in the data.



Experiments

Dataset	Original	Augmented
`AgeDB`		
`TUAB`		
`MPIIFaceGaze`		
`SkyFinder`		

Age Estimation

Brain-age Prediction

Gaze Direction Detection

Temperature Prediction

Improved Regression Performance

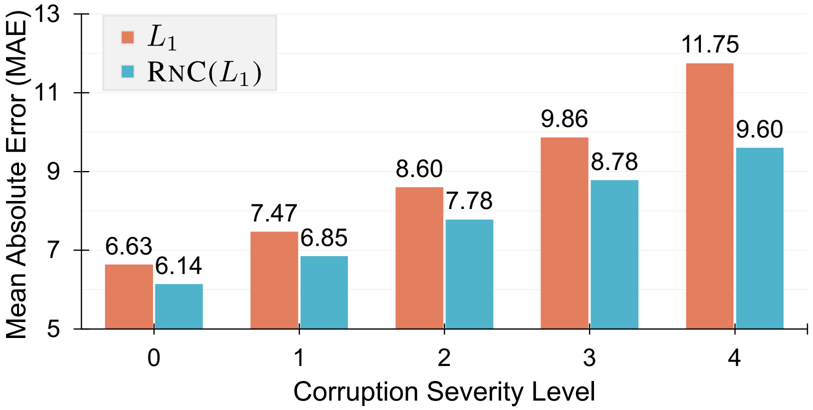
RNC is compatible to end-to-end regression methods, and consistently improves the performance.

	AgeDB		TUAB		MPIIFaceGaze		SkyFinder	
Metrics	MAE↓	R^2↑	MAE↓	R^2↑	Angular↓	R^2↑	MAE↓	R^2↑
L_1	6.63	0.828	7.46	0.655	5.97	0.744	2.95	0.860
RNC(L_1)	**6.14** (+0.49)	**0.850** (+0.022)	**6.97** (+0.49)	**0.697** (+0.042)	**5.27** (+0.70)	**0.815** (+0.071)	2.86 (+0.09)	**0.869** (+0.009)
MSE	6.57	0.828	8.06	0.585	6.02	0.747	3.08	0.851
RNC(MSE)	**6.19** (+0.38)	**0.849** (+0.021)	**7.05** (+1.01)	**0.692** (+0.107)	**5.35** (+0.67)	**0.802** (+0.055)	**2.86** (+0.22)	**0.869** (+0.018)
HUBER	6.54	0.828	7.59	0.637	6.34	0.709	2.92	0.860
RNC(HUBER)	**6.15** (+0.39)	**0.850** (+0.022)	**6.99** (+0.60)	**0.696** (+0.059)	**5.15** (+1.19)	**0.830** (+0.121)	**2.86** (+0.06)	**0.869** (+0.009)
DEX [36]	7.29	0.787	8.01	0.537	5.72	0.776	3.58	0.778
RNC(DEX)	**6.43** (+0.86)	**0.836** (+0.049)	**7.23** (+0.78)	**0.646** (+0.109)	**5.14** (+0.58)	**0.805** (+0.029)	**2.88** (+0.70)	**0.865** (+0.087)
DLDL-v2 [14]	6.60	0.827	7.91	0.560	5.47	0.799	2.99	0.856
RNC(DLDL-v2)	**6.32** (+0.28)	**0.844** (+0.017)	**6.91** (+1.00)	**0.697** (+0.137)	**5.16** (+0.31)	**0.802** (+0.003)	**2.85** (+0.14)	**0.869** (+0.013)
OR [33]	6.40	0.830	7.36	0.646	5.86	0.770	2.92	0.861
RNC(OR)	**6.34** (+0.06)	**0.843** (+0.013)	**7.01** (+0.35)	**0.688** (+0.042)	**5.13** (+0.73)	**0.825** (+0.055)	**2.86** (+0.06)	**0.867** (+0.006)
CORN [40]	6.72	0.811	8.11	0.597	5.88	0.762	3.24	0.819
RNC(CORN)	**6.44** (+0.28)	**0.838** (+0.027)	**7.22** (+0.89)	**0.663** (+0.066)	**5.18** (+0.70)	**0.820** (+0.058)	**2.89** (+0.35)	**0.862** (+0.043)

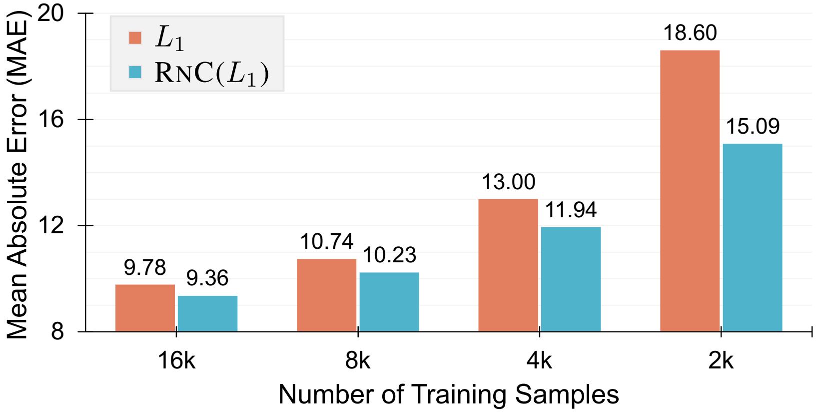
RNC surpasses state-of-the-art methods

Method	AgeDB	TUAB	MPIIFaceGaze	SkyFinder
Representation learning methods (Linear Probing):				
SIMCLR [4]	9.59	11.01	9.43	4.70
DINO [3]	10.26	11.62	11.92	5.63
SUPCON [25]	8.13	8.47	9.27	3.97
Representation learning methods (Fine-tuning):				
SIMCLR [4]	6.57	7.57	5.50	2.93
DINO [3]	6.61	7.58	5.80	2.98
SUPCON [25]	6.55	7.41	5.54	2.95
Regression learning methods:				
L_1	6.63	7.46	5.97	2.95
LDS+FDS [44]	6.45	–	–	–
L2CS-NET [1]	–	–	5.45	–
LDE [7]	–	–	–	2.92
RANKSIM [17]	6.51	7.33	5.70	2.94
ORDINAL ENTROPY [50]	6.47	7.28	–	2.94
RNC(L_1)	**6.14**	**6.97**	**5.27**	**2.86**
GAINS	**+0.31**	**+0.31**	**+0.18**	**+0.06**

Intriguing properties of RNC



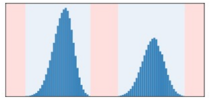
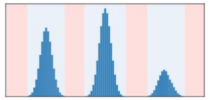
RNC is more robust to data corruptions.



RNC is more resilient to reduced training data.

	AgeDB → IMDB-WIKI (subsampled, 2k)				IMDB-WIKI (subsampled, 32k) → AgeDB			
	Linear Probing		*Fine-tuning*		*Linear Probing*		*Fine-tuning*	
Metrics	MAE↓	R^2↑	MAE↓	R^2↑	MAE↓	R^2↑	MAE↓	R^2↑
L_1	12.25	0.496	11.57	0.528	7.36	0.801	6.36	0.848
RNC(L_1)	**11.12** (+1.13)	**0.556** (+0.060)	**11.09** (+0.48)	**0.546** (+0.018)	**7.06** (+0.30)	**0.812** (+0.011)	**6.13** (+0.23)	**0.850** (+0.002)

RNC delivers better transfer learning results.

Label Distribution	Method	All	Seen	Unseen
	L_1	12.53	10.82	18.40
	RNC(L_1)	**11.69** (+0.84)	**10.46** (+0.36)	**15.92** (+2.48)
	L_1	11.94	10.43	14.98
	RNC(L_1)	**10.88** (+1.06)	**9.78** (+0.64)	**13.08** (+1.90)

RNC has better zero-shot generalization to unseen targets.

Thanks!

Paper: https://arxiv.org/abs/2210.01189

Code: https://github.com/kaiwenzha/Rank-N-Contrast