Time-series Generative Adversarial Networks

UCLA; University of Cambridge

NeurIPS 2019

Introduction

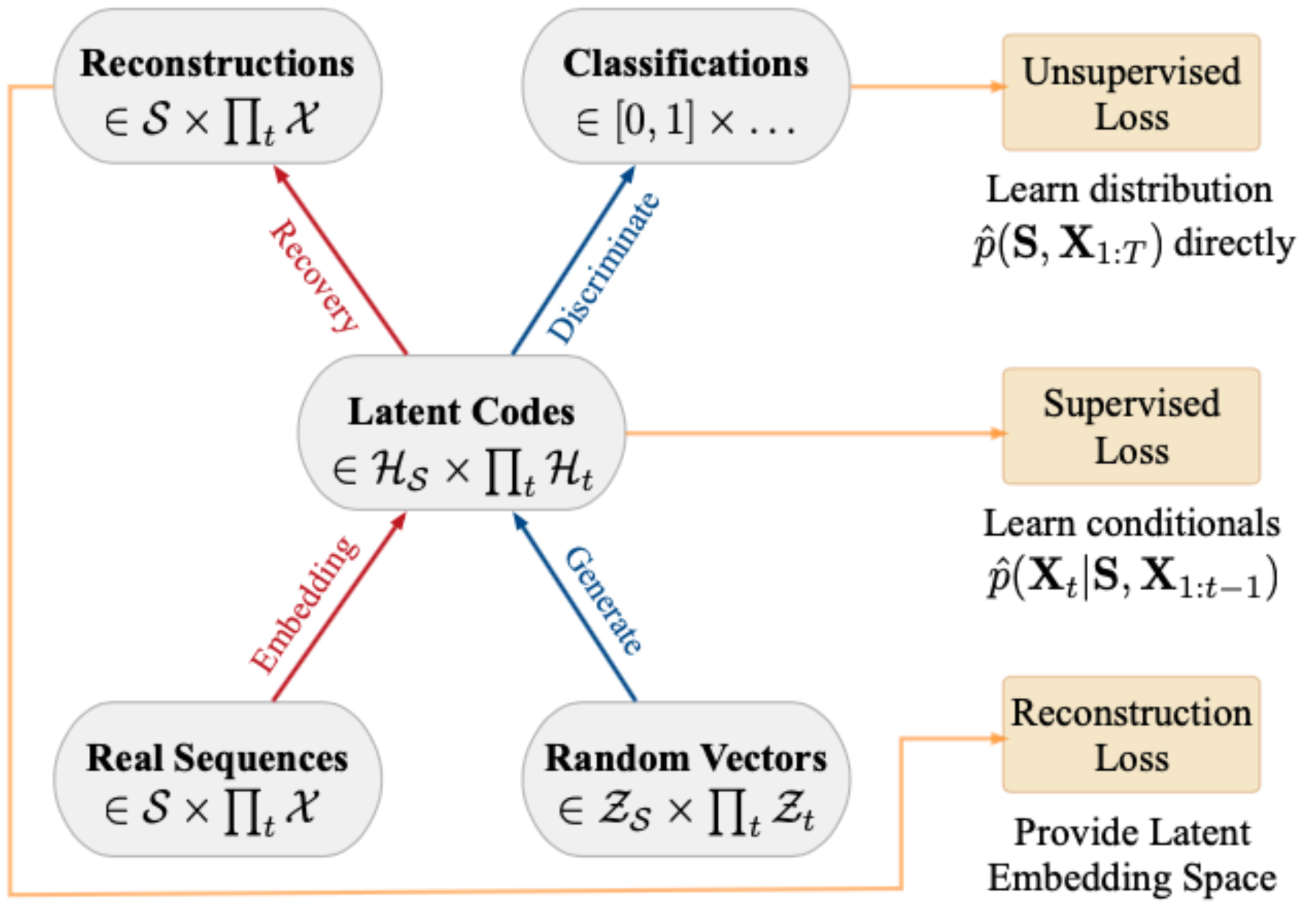
Autoregression model: $\prod_t p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$

- Useful in forecasting;
- Deterministic, not generative.

GAN:

- Model directly $p(x_{1:T})$ without leveraging AR prior;
- May not capture stepwise dependencies.

The model



(a) Block Diagram

Embedding & Reconstructions:

S – static feature;
X – temporal features;
Hs – latent vector space of S;
Ht – latent vector space of X;

Embedding function:

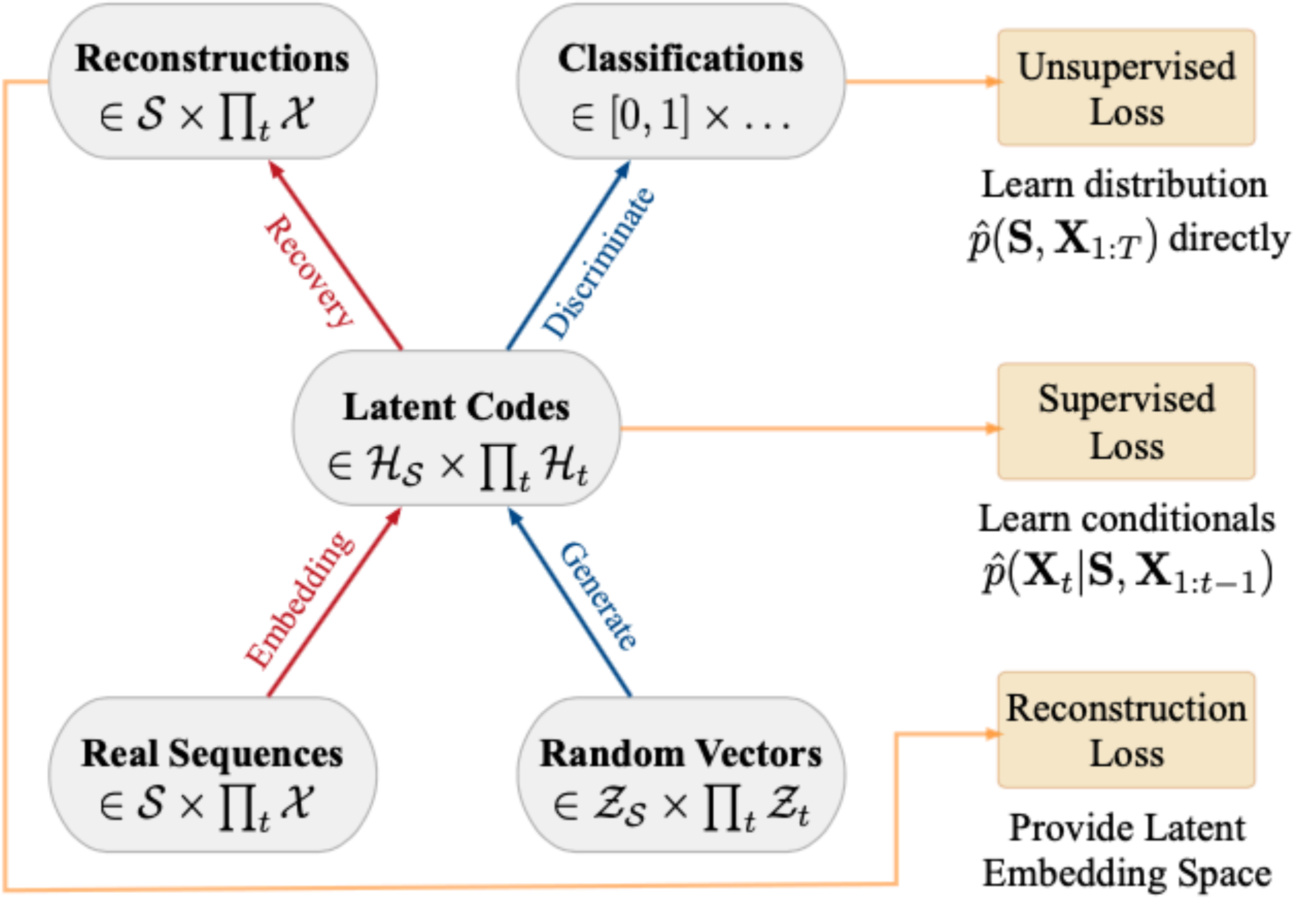
$$\mathbf{h}_\mathcal{S} = e_\mathcal{S}(\mathbf{s}), \qquad \mathbf{h}_t = e_\mathcal{X}(\mathbf{h}_\mathcal{S}, \mathbf{h}_{t-1}, \mathbf{x}_t)$$

Recovery function:

$$\tilde{\mathbf{s}} = r_\mathcal{S}(\mathbf{h}_s), \qquad \tilde{\mathbf{x}}_t = r_\mathcal{X}(\mathbf{h}_t)$$

The e and r can be any function;
Here
e is recurrent net and
r is attention-based decoder.

The model



(a) Block Diagram

Generator & Discriminator

Hs – latent vector space of S;
Ht – latent vector space of X;
Zs – Gaussion dist.;
Zt – Wiener process.

Generating function:

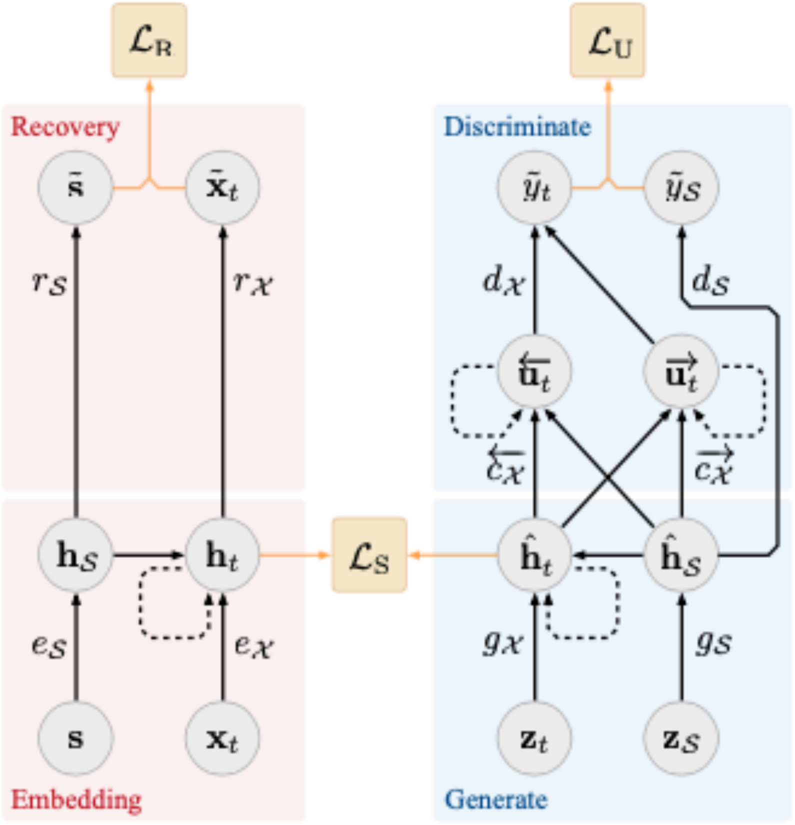
$$\hat{\mathbf{h}}_\mathcal{S} = g_\mathcal{S}(\mathbf{z}_\mathcal{S}), \qquad \hat{\mathbf{h}}_t = g_\mathcal{X}(\hat{\mathbf{h}}_\mathcal{S}, \hat{\mathbf{h}}_{t-1}, \mathbf{z}_t)$$

Discriminator function:

$$\tilde{y}_\mathcal{S} = d_\mathcal{S}(\tilde{\mathbf{h}}_\mathcal{S}) \qquad \tilde{y}_t = d_\mathcal{X}(\overleftarrow{\mathbf{u}}_t, \overrightarrow{\mathbf{u}}_t)$$

The g and d can be any function;
Here
g is recurrent net and
d is bi-directional recurrent net

Approaches



(a) TimeGAN

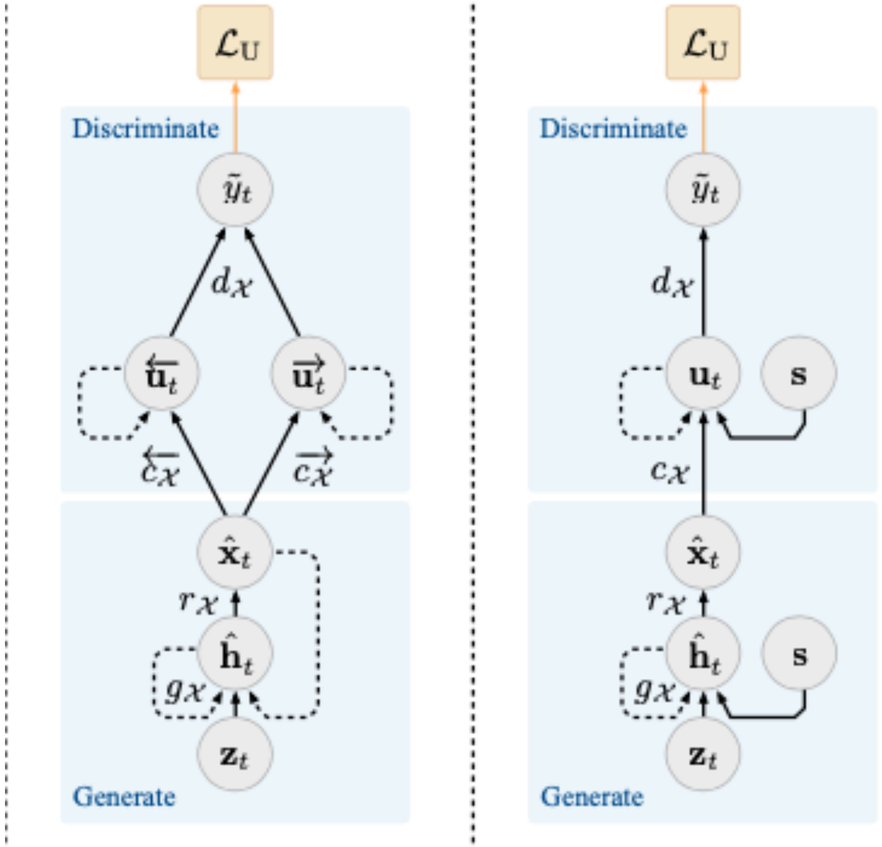
$$\mathcal{L}_R = \mathbb{E}_{\mathbf{s},\mathbf{x}_{1:T}\sim p}\big[\|\mathbf{s} - \tilde{\mathbf{s}}\|_2 + \textstyle\sum_t \|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2\big]$$

$$\mathcal{L}_U = \mathbb{E}_{\mathbf{s},\mathbf{x}_{1:T}\sim p}\big[\log y_S + \textstyle\sum_t \log y_t\big]$$
$$+ \mathbb{E}_{\mathbf{s},\mathbf{x}_{1:T}\sim \hat{p}}\big[\log(1 - \hat{y}_S) + \textstyle\sum_t \log(1 - \hat{y}_t)\big]$$

$$\mathcal{L}_S = \mathbb{E}_{\mathbf{s},\mathbf{x}_{1:T}\sim p}\big[\textstyle\sum_t \|\mathbf{h}_t - g_X(\mathbf{h}_S, \mathbf{h}_{t-1}, \mathbf{z}_t)\|_2\big]$$

Other Approaches

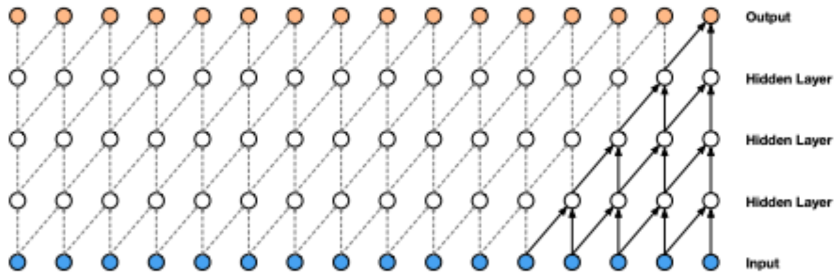
TimeGAN
RCGAN
C-RNN-GAN
T-Forcing
P-Forcing
WaveNet
WaveGAN



(b) C-RNN-GAN (c) RCGAN

RNNs:
T-Forcing: teacher- forcing
P-Forcing: professor-forcing

CNNs:
WaveNet:
WaveGAN: GAN counter part of WaveNET



Results

(1) Visualization:

- tSNE, PCA

(2) Discriminative Score :

- Original sequence -> *real;* generated sequence -> *not real*.
- An off-the-shelf (RNN) classifier is trained to distinguish between the two classes.
- Classification error on the held-out test set.

(3) Predictive Score:

- Using the synthetic dataset, we train a post-hoc sequence-prediction model (by optimizing a 2-layer LSTM) to predict next-step temporal vectors over each input sequence.
- Then, we evaluate the trained model on the original dataset.
- Performance is measured in terms of the mean absolute error (MAE);

Expt #1

we experiment on sequences from autoregressive multivariate Gaussian models

$$\mathbf{x}_t = \phi \mathbf{x}_{t-1} + \mathbf{n}, \text{ where } \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{1} + (1-\sigma)\mathbf{I})$$

Table 1: Results on Autoregressive Multivariate Gaussian Data (Bold indicates best performance).

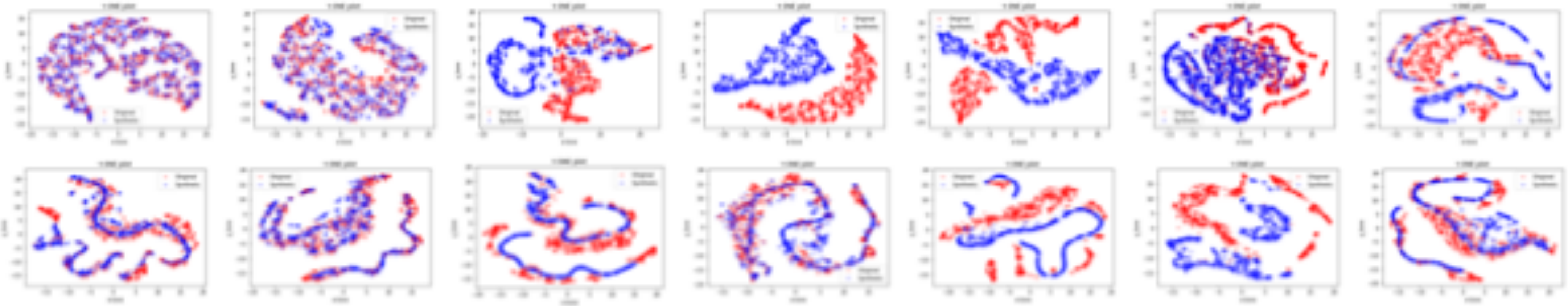
	Temporal Correlations (fixing $\sigma = 0.8$)			Feature Correlations (fixing $\phi = 0.8$)		
Settings	$\phi = 0.2$	$\phi = 0.5$	$\phi = 0.8$	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 0.8$
Discriminative Score (Lower the better)						
TimeGAN	**.175±.006**	**.174±.012**	**.105±.005**	**.181±.006**	**.152±.011**	**.105±.005**
RCGAN	.177±.012	.190±.011	.133±.019	.186±.012	.190±.012	.133±.019
C-RNN-GAN	.391±.006	.227±.017	.220±.016	.198±.011	.202±.010	.220±.016
T-Forcing	.500±.000	.500±.000	.499±.001	.499±.001	.499±.001	.499±.001
P-Forcing	.498±.002	.472±.008	.396±.018	.460±.003	.408±.016	.396±.018
WaveNet	.337±.005	.235±.009	.229±.013	.217±.010	.226±.011	.229±.013
WaveGAN	.336±.011	.213±.013	.230±.023	.192±.012	.205±.015	.230±.023
Predictive Score (Lower the better)						
TimeGAN	**.640±.003**	**.412±.002**	**.251±.002**	**.282±.005**	**.261±0.002**	**.251±.002**
RCGAN	.652±.003	.435±.002	.263±.003	.292±.003	.279±.002	.263±.003
C-RNN-GAN	.696±.002	.490±.005	.299±.002	.293±.005	.280±.006	.299±.002
T-Forcing	.737±.022	.732±.012	.503±.037	.515±.034	.543±.023	.503±.037
P-Forcing	.665±.004	.571±.005	.289±.003	.406±.005	.317±.001	.289±.003
WaveNet	.718±.002	.508±.003	.321±.005	.331±.004	.297±.003	.321±.005
WaveGAN	.712±.003	.489±.001	.290±.002	.325±.003	.353±.001	.290±.002

Results

(1) Sine; (2) Stocks; (3) Energy; (4) Events

Sine

Stocks



(a) TimeGAN (b) RCGAN (c) CRNNGAN (d) T-Forcing (e) P-Forcing (f) WaveNet (g) WaveGAN

Results

(1) Sine; (2) Stocks; (3) Energy; (4) Events

Table 2: Results on Multiple Time-Series Datasets (Bold indicates best performance).

Metric	Method	Sines	Stocks	Energy	Events
Discriminative Score (Lower the Better)	TimeGAN	**.011±.008**	**.102±.021**	**.236±.012**	**.161±.018**
	RCGAN	.022±.008	.196±.027	.336±.017	.380±.021
	C-RNN-GAN	.229±.040	.399±.028	.499±.001	.462±.011
	T-Forcing	.495±.001	.226±.035	.483±.004	.387±.012
	P-Forcing	.430±.027	.257±.026	.412±.006	.489±.001
	WaveNet	.158±.011	.232±.028	.397±.010	.385±.025
	WaveGAN	.277±.013	.217±.022	.363±.012	.357±.017
Predictive Score (Lower the Better)	TimeGAN	**.093±.019**	**.038±.001**	**.273±.004**	**.303±.006**
	RCGAN	.097±.001	.040±.001	.292±.005	.345±.010
	C-RNN-GAN	.127±.004	**.038±.000**	.483±.005	.360±.010
	T-Forcing	.150±.022	**.038±.001**	.315±.005	.310±.003
	P-Forcing	.116±.004	.043±.001	.303±.006	.320±.008
	WaveNet	.117±.008	.042±.001	.311±.005	.333±.004
	WaveGAN	.134±.013	.041±.001	.307±.007	.324±.006
	Original	.094±.001	.036±.001	.250±.003	.293±.000

Results

Table 3: Source-of-Gain Analysis on Multiple Datasets (via Discriminative and Predictive scores).

Metric	Method	Sines	Stocks	Energy	Events
Discriminative Score (Lower the Better)	TimeGAN	**.011±.008**	**.102±.021**	**.236±.012**	**.161±.018**
	w/o Supervised Loss	.193±.013	.145±.023	.298±.010	.195±.013
	w/o Embedding Net.	.197±.025	.260±.021	.286±.006	.244±.011
	w/o Joint Training	.048±.011	.131±.019	.268±.012	.181±.011
Predictive Score (Lower the Better)	TimeGAN	**.093±.019**	**.038±.001**	**.273±.004**	**.303±.006**
	w/o Supervised Loss	.116±.010	.054±.001	.277±.005	.380±.023
	w/o Embedding Net.	.124±.002	.048±.001	.286±.002	.410±.013
	w/o Joint Training	.107±.008	.045±.001	.276±.004	.348±.021

Conclusion

- We propose TimeGAN:
  - a novel framework for time-series generation that combines the versatility of the unsupervised **GAN approach** with the control over conditional temporal dynamics afforded by supervised **autoregressive models**.
- State-of-the-art benchmarks
- Future: privacy framework.

Thanks!