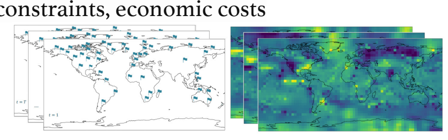
ESC-GAN: Extending Spatial Coverage of Physical Sensors

Xiyuan Zhang, Ranak Roy Chowdhury, Jingbo Shang, Rajesh Gupta, Dezhi Hong
xiyuanzh@ucsd.edu,{rrchowdh,jshang,gupta,dehong}@eng.ucsd.edu

Motivation

- Geo-sensors help monitor our ecosystem
  - Global warming, flood forecasting, agriculture
- However, geo-sensors are sparsely deployed!
  - Physical constraints, economic costs



(a) Original Data Coverage (b) Extended Data

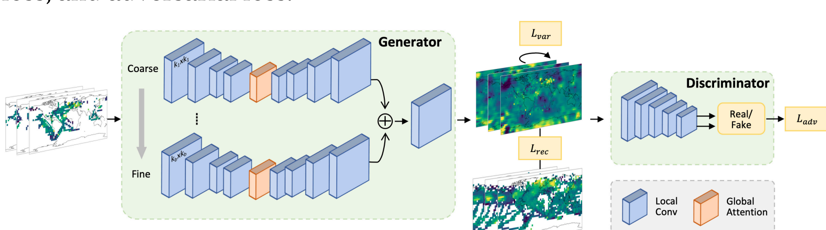
Goal: seek a cost-effective approach to Extend the Spatial Coverage (ESC) of sensory data without deploying additional sensors

Challenges & Contributions

- Complete miss of temporal information
  - Fundamentally different from traditional imputation task
  - ESC-GAN framework
- Existence of local and global context
  - 3D partial convolution to learn local correlations, global attention module to capture global attention information
- Multi-scale structure
  - Fine-grained data for accurate local patterns, coarse-grained data for "macro" view
  - Multi-branch generator to exploit information of different granularity
- Irregular and stochastic forms, high variations
  - Adversarial training

Model Overview

Our model takes sparse map as input and produces map with all the missing data reconstructed. We feed the recovered maps together with ground-truth maps to the discriminator for a real or fake classification. We combine three loss functions, i.e., reconstruction loss, variation loss, and adversarial loss.



reconstructed loss $$L_{rec} = \frac{1}{N_{masked}} \sum_t \sum_y \sum_x (1 - M_{t,y,x})(Z_{t,y,x} - X_{t,y,x})^2$$

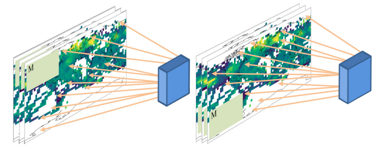
variation loss $$L_{var} = \frac{1}{N}\Big(\sum_{(y,x)\in R,(y+1,x)\in R} ||\tilde{X}_{t,y+1,x} - \tilde{X}_{t,y,x}||_1 + \sum_{(y,x)\in R,(y,x+1)\in R} ||\tilde{X}_{t,y,x+1} - \tilde{X}_{t,y,x}||_1 \Big)$$

adversarial loss $$L_D = \mathbb{E}_{\mathbf{x}\sim P_X(\mathbf{x})}[RELU(1 - D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_Z(\mathbf{z})}[RELU(1 + D(\mathbf{z})]$$
$$L_{adv} = L_G = -\mathbb{E}_{\mathbf{z}\sim P_Z(\mathbf{z})}[D(\mathbf{z})]$$

overall optimization $$L = L_{rec} + \lambda_{var} L_{var} + \lambda_{adv} L_{adv}$$

Local 3D Partial Convolution

At each iteration, we apply a training mask removing a random subset of locations. The masking during training helps the model learn how to recover data over time in the masked-out cell. Values in some locations are invalid, so we partially convolve only on locations with data.

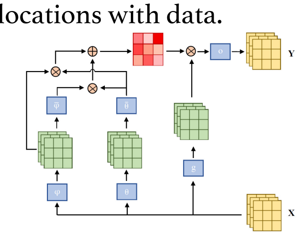


(a) iteration i (b) iteration j

$$Y = \begin{cases} \mathbf{W}^T(\mathbf{X} \odot \mathbf{M})\frac{|\mathbf{1}|_1}{|\mathbf{M}|_1} + b, & \text{if } |\mathbf{M}|_1 > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Global Attention Module

Values in some locations are invalid, so we partially convolve only on locations with data.



$$f(\mathbf{X}, \mathbf{X}') = \frac{e^{(\theta(\mathbf{X})-\mu_\theta)^T(\phi(\mathbf{X}')-\mu_\phi)+\mu_\theta^T\phi(\mathbf{X}')}}{\sum_{\mathbf{X}'} e^{(\theta(\mathbf{X})-\mu_\theta)^T(\phi(\mathbf{X}')-\mu_\phi)+\mu_\theta^T\phi(\mathbf{X}')}}$$

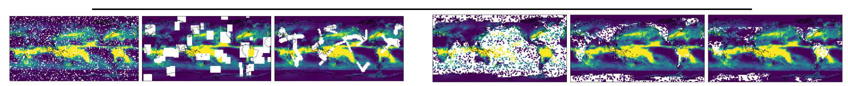
$$O_{t,y,x} = \sum_{\forall t',y',x'} f(X_{t,y,x}, X_{t',y',x'})g(X_{t',y',x'})$$

Multi-Scale Structure

- parallel branches with convolution filters of different receptive fields to extract multi-resolution features
- the number of branches could be decided by the input granularity

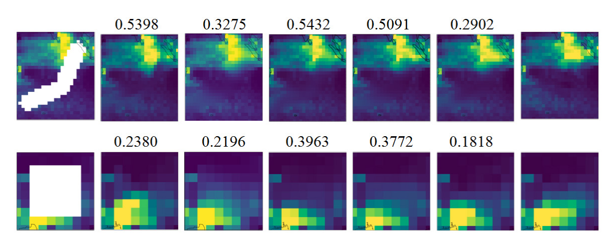
Datasets

Dataset	Lat	Lon	Time	Granularity	#Grid Cells
HadCRUT	36	72	2004	5°× 5 °	5,194,368
CMAP	72	144	503	2.5°× 2.5 °	5,215,104
KDD CUP 2018	6	8	8736	0.0167°× 0.0175°	96096



(a) Scatter (b) Regular Cluster (c) Irregular Cluster (a) Ocean (b) High-altitude (c) High-vegetation

Main Results

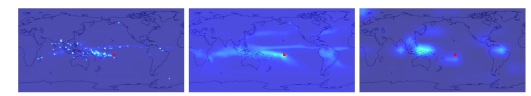


(a) Missing Area (b) IDW (c) ST-MVL (d) PConv (e) 3DGated (f) ESC-GAN (g) GT

Method	HadCRUT						CMAP					
	Scatter		Reg Cluster		Irr Cluster		Scatter		Reg Cluster		Irr Cluster	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Zero	1.0396	0.6638	0.8551	0.5983	0.7446	0.5879	1.0290	0.6830	1.2861	0.7620	1.2869	0.7390
Mean	0.9697	0.6375	0.7985	0.5744	0.6971	0.5677	1.0272	0.6804	1.3294	0.7548	1.2969	0.7364
sKNN	0.3756	0.3991	0.4645	0.4649	0.3791	0.4210	0.1120	0.1785	0.7159	0.4863	0.4888	0.3960
IDW	0.3524	0.3868	0.4440	0.4535	0.3596	0.4087	0.1042	0.1719	0.7036	0.4792	0.4658	0.3839
Kriging	0.9517	0.6308	0.7995	0.5767	0.6906	0.5703	0.8838	0.5863	0.9709	0.6279	1.1257	0.6545
MF	0.6181	0.5216	0.7669	0.5782	0.6111	0.5390	0.1721	0.2395	0.8583	0.5753	0.5942	0.4974
BTTF	0.5867	0.5225	0.6798	0.5553	0.5764	0.5332	0.2474	0.3137	0.9423	0.6237	0.5723	0.5154
ST-MVL	0.3648	0.3964	0.4710	0.4655	0.3581	0.4084	0.1162	0.1832	0.7202	0.4919	0.5039	0.4177
IGKNN	0.7214	0.5492	0.7212	0.5501	0.6405	0.5491	0.8474	0.6132	1.1710	0.7285	1.1254	0.7170
NAOMI	1.0391	0.6637	0.8550	0.5983	0.7442	0.5877	1.0288	0.6833	1.2859	0.7620	1.2863	0.7396
PConv	0.3908	0.4211	0.4759	0.4784	0.4122	0.4494	0.1008	0.1704	0.6469	0.4492	0.2969	0.3024
3DGated	0.3610	0.3907	0.4265	0.4454	0.3581	0.4066	0.1400	0.2071	0.5532	0.4381	0.2952	0.3033
ESC-GAN	**0.3354**	**0.3800**	**0.4097**	**0.4295**	**0.3418**	**0.3971**	**0.0802**	**0.1531**	**0.5441**	**0.4308**	**0.2739**	**0.3017**

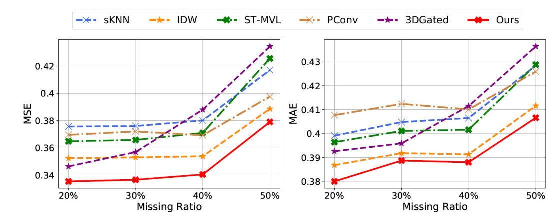
Visualization

We visualize the softmax attention score between a randomly chosen query region and all the other regions. We mark the query regions with red rectangles and compare with ground truth attention (figure (a), (b)).



(a) Dot Product GT (b) Cosine GT (c) ESC-GAN

Robustness study



Method	Ocean		High-altitude		High-vegetation	
	MSE	MAE	MSE	MAE	MSE	MAE
sKNN	0.3162	0.3160	0.0865	0.1428	0.1857	0.2178
IDW	0.2738	0.2929	0.0786	0.1335	0.1687	0.2060
ST-MVL	0.2855	0.2972	0.0784	0.1347	0.1710	0.2061
PConv	0.2174	0.2517	0.0743	0.1303	0.1588	0.2004
3DGated	0.2989	0.3093	0.1231	0.1878	0.2254	0.2644
ESC-GAN	**0.1929**	**0.2512**	**0.0663**	**0.1234**	**0.1399**	**0.1911**

Generalization to Random Missing

traditional spatio-temporal imputation task for random missing values

%Missing	20%	30%	40%	50%	60%	70%	80%	90%
Last	1.073	0.894	0.901	0.990	1.040	1.236	1.689	2.870
Mean	0.916	0.907	0.914	0.923	0.973	0.935	0.937	1.002
KNN	0.892	0.803	0.776	0.798	0.856	0.852	0.873	1.243
MF	0.850	0.785	0.787	0.772	0.834	0.805	0.860	1.196
MTSI	0.844	0.780	0.753	0.743	0.803	0.780	0.837	1.018
BRITS	0.455	0.421	0.372	0.409	0.440	0.482	0.648	0.725
DCRNN	0.579	0.565	0.449	0.506	0.589	0.622	0.720	0.861
CDSA	0.373	0.393	0.287	0.291	0.387	0.495	0.521	0.631
ESC-GAN	**0.207**	**0.229**	**0.232**	**0.231**	**0.274**	**0.299**	**0.326**	**0.434**

Code

https://github.com/xiyuanzh/ESC-GAN