Position-dependent partial convolutions for supervised spatial interpolation

Hirotaka Hachiya HHACHIYA@WAKAYAMA-U.AC.JP
Graduate School of Systems Engineering, Wakayama University/Center for AIP, RIKEN

Kotaro Nagayoshi NAGAYOSHI.KOTARO@G.WAKAYAMA-U.JP
Graduate School of Systems Engineering, Wakayama University/Center for AIP, RIKEN

Asako Iwaki IWAKI@BOSAI.GO.JP
National Research Institute for Earth Science and Disaster Resilience

Takahiro Maeda TMAEDA@BOSAI.GO.JP
National Research Institute for Earth Science and Disaster Resilience

Naonori Ueda NAONORI.UEDA@RIKEN.JP
Center for AIP, RIKEN

Hiroyuki Fujiwara FUJIWARA@BOSAI.GO.JP
National Research Institute for Earth Science and Disaster Resilience

Editors: Emtiyaz Khan and Mehmet Gönen

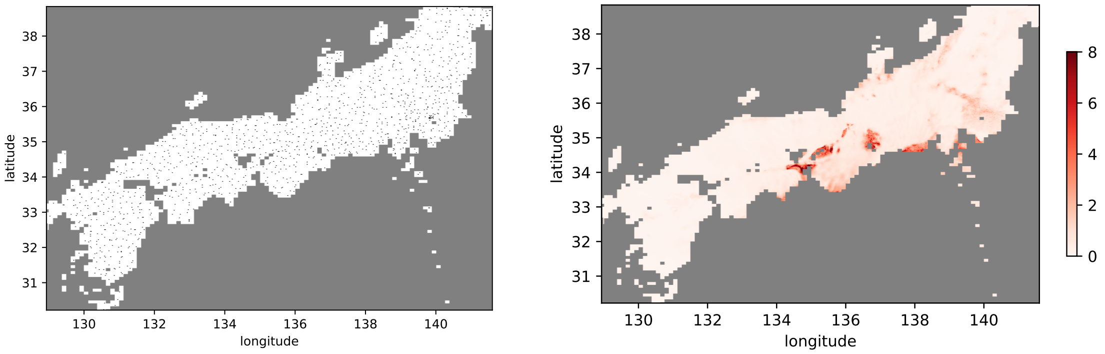
Abstract

Acquiring continuous spatial data, e.g., spatial ground motion is essential to assess the damaged area and appropriately assign rescue and medical teams. To this purpose, spatial interpolation methods have been developed to estimate the value of unobserved points linearly from neighbor observed values, i.e., inverse distance weighting and Kriging. Recently, realistic spatial continuous environmental data with various scenarios can be generated by 3D finite difference methods with a high-resolution structure model. It enables us to collect supervised data even for unobserved points. Along this line, we propose a framework of supervised spatial interpolation and apply highly advanced deep inpainting methods where we treat spatially distributed observed points as a masked image and non-linearly expand them through convolutional encoder-decoder networks. However, the property of translation invariance would avoid locally fine-grained interpolation since the relation between the target and surrounding observation points varies over regions due to its topography and subsurface structure. To overcome this problem, we propose introducing position-dependent convolution where kernel weights are adjusted depending on their position on an image based on the trainable position-feature map. We show the effectiveness of our proposed method, called, PoDIM (Position-dependent Deep Inpainting Method), through experiments using simulated ground-motion data.

Keywords: spatial interpolation; image inpainting; partial convolution; deep network

1. Introduction

In recent years, many signs of a megathrust earthquake have occurred in Japan. Such earthquakes will likely cause severe damage over a wide area of Japan shortly. In order to reduce the damage, gathering continuous spatial data of ground motion just after the



(a) Observation sites

(b) Spatial continuous simulation

Figure 1: (a) Distribution of seismic observation sites in the region from Kyushu to Tohoku in Japan. Black dots represent the location of strong motion stations K-NET and KiK-net operated by NIED (National Research Institute for Earth Science and Disaster Resilience (NIED) (a), National Research Institute for Earth Science and Disaster Resilience (NIED) (b))—there are totally 1740 sites throughout Japan. White and gray areas represent unobserved locations on land and sea, respectively. (b) an example of spatial continuous simulated velocity response spectra [m/s] for a megathrust earthquake occurred in Nankai trough—the value is clipped at 8.0 for visualization purposes.

earthquake would play a significant role—it will enable us to assess the damaged area and make adequate plans for rescue operations and emergency medical services. However, environmental data like seismic motion are typically collected from point sources due to the cost and difficulty of installing sensors.

More specifically, Fig. 1(a)subfigure depicts the location of strong motion stations K-NET and KiK-net operated by NIED (National Research Institute for Earth Science and Disaster Resilience (NIED) (a), National Research Institute for Earth Science and Disaster Resilience (NIED) (b)) in the region from Kyusyu to Tohoku in Japan. The figure indicates that although observation points (in black) are distributed broadly over Japan, there are large blank areas (in white) between the points, especially in coastal areas (in white and gray boundaries) and central areas corresponding to mountains.

To acquire the values even in unobserved points, spatial interpolation methods have been studied and put to practical use in geostatistics and environmental fields, such as mining, meteorology, and soil science (Li and Australia (2008), Yang and Xing (2021)). These methods estimate the value of an unobserved target point by a weighted sum of the ones of neighbor observed points. The weights are set based on the distance between the target and observed points, e.g., in IDW (inverse distance weighting) (Lu and Wong (2008), Li et al. (2018)), and on the variogram models, e.g., in Kriging (Matheron (1963), Li and Australia (2008)).

In recent years, with high-resolution and large-scale structure models, 3D finite-difference methods enable us to simulate realistic environmental surface data. For instance, ground-motion data for the Nankai trough megathrust earthquakes in southwest, Japan can be generated with various rapture scenarios given parameters, e.g., source and magnitude, a pattern of asperities, and rapture initiation points (Maeda et al., Moschetti et al.). Fig. 1(b)subfigure depicts an example of spatial continuous simulated velocity response spectra [m/s] for a megathrust earthquake occurred in Nankai trough.

Given these highly accurate spatial continuous data, we can formulate the problem of spatial interpolation in a supervised learning manner where supervised values are given even in unobserved points in the training phase. As a supervised spatial interpolation, one can apply deep inpainting methods. Image inpainting is a computer vision task of reconstructing missing regions in an image and is applied to the enhancement of damaged images, e.g., with scratches and noises and object removal from images (Elharrouss et al. (2020)). Recently, deep convolutional encoder-decoder network based approaches have been actively developed (Liu et al. (2018), Yan et al. (2018), Zeng et al. (2019)). The semantic context of an image is encoded into a latent feature space by the encoder, and then semantic-coherent patches are generated by the decoder. These deep inpainting methods enable adaptive and nonlinear spatial interpolation model based on the pattern of surrounding observed values, allowing us to spatially coherent interpolation, e.g., smooth surface.

However, the property of translation invariance induced by convolutional operations in the encoder-decoder network, would limit further performance improvement since the relationship between the target and its surrounding observed points varies from region to region depending on the local environment, e.g., topography and subsurface structure.

To overcome this translation-invariant problem in deep inpainting methods for spatial interpolation, we propose a position-dependent convolution kernel in which kernel weights are adaptively adjusted depending on the position of an image. More specifically, we obtain a position-feature map adaptively, each pixel corresponding to the scale or bias adjustment value of kernel weights at each position, through the training process of the encoder-decoder network. Then, we apply it at a specific encoder layer to extract position-dependent context and generate position-coherent patches by the decoder.

Through experiments with simulated ground-motion data, we show the effectiveness of our proposed method, called, PoDIM (POsition-dependent Deep Inpainting Method), on spatial interpolation in comparison with classical linear models and deep inpainting methods.

2. Related works

In this section, we review existing spatial interpolation methods and deep inpainting methods.

2.1. Spatial interpolation methods

Spatial interpolation methods estimate the value of an unobserved target point from its neighbor observed values (Li and Australia (2008)). Let us denote observed spatial data of e-th event by the m-pair of point $\boldsymbol{x}^{\mathrm{obs}}$ and its observed value y^{obs}, i.e., $d_e^{\mathrm{obs}} = \left\{(\boldsymbol{x}_i^{\mathrm{obs}}, y_{ei}^{\mathrm{obs}})\right\}_{i=1}^{m}$ where observed points $\{\boldsymbol{x}_i^{\mathrm{obs}}\}_{i=1}^{m}$ are fixed over events. Then, we estimate the value y_j of a point $\boldsymbol{x}_j$ given the observation d_e^{obs}. In the spatial interpolation methods, a linear model

$f_{\text{linear}}(\cdot)$, i.e, a weighted sum the observed values $\{y_{ei}^{\text{obs}}\}_{i=1}^{m}$ (Lu and Wong (2008)) is used as

$$\widehat{y_{ej}} = f_{\text{linear}}\left(\boldsymbol{x}_j \mid d_e^{\text{obs}}\right) \equiv \sum_{i=1}^{m} w_{ji} y_{ei}^{\text{obs}}. \tag{1}$$

where w_{ji} is the weight for the observed value y_{ei}^{obs} and it is set based on the relationship between two points, i.e., $\boldsymbol{x}_j$ and $\boldsymbol{x}_i^{\text{obs}}$.

2.1.1. Inverse distance weighting

IDW (inverse distance weighting) (Lu and Wong (2008), Li et al. (2018)) sets the weight based on the distance between a point $\boldsymbol{x}_j$ and its neighbor observed points $\{\boldsymbol{x}_i^{\text{obs}}\}_{i \in S(\boldsymbol{x}_j)}$ as

$$w_{ji} = \frac{r(\boldsymbol{x}_j, \boldsymbol{x}_i)^{-\alpha}}{\sum_{i' \in S(\boldsymbol{x}_j)} r(\boldsymbol{x}_j, \boldsymbol{x}_{i'})^{-\alpha}}, \quad r(\boldsymbol{x}, \boldsymbol{x}') \equiv \|\boldsymbol{x} - \boldsymbol{x}'\|_2, \tag{2}$$

where $S(\boldsymbol{x}_j)$ is a set of indices of neighbor observation points of the point $\boldsymbol{x}_j$ and α is a power parameter. The power parameter is usually set at 1 or 2 and can be adaptively set based on the relative distance w.r.t. the expected distance (Lu and Wong (2008)). IDW assumes neighbor observation points are uniformly distributed w.r.t. the target point $\boldsymbol{x}_j$. However, the neighbor observed points could be distributed with a cluster in a specific direction in reality, and the clustered observed values strongly influence the estimated value. The influence of the imbalanced observed points could be alleviated by adjusting weights based on the similarities between observation points (Li et al. (2018)).

2.1.2. Kriging

Kriging optimizes weights w so as to minimize the variance of estimated value $\widehat{y_{ej}}$ which is computed using the variance $\gamma(r)$ of the deviation of values y between two points in a distance r as

$$\gamma(r) = \frac{1}{2}\text{var}\left[y(\boldsymbol{x} + r) - y(\boldsymbol{x})\right], \tag{3}$$

where $y(\boldsymbol{x})$ is the value at point $\boldsymbol{x}$. The variance, called variogram, is modeled by linear, exponential, and Gaussian functions, and parameters of which are estimated using neighbor observations $\{(\boldsymbol{x}_i^{\text{obs}}, y_i^{\text{obs}})\}_{i \in S(\boldsymbol{x}_j)}$ (Matheron (1963), Li and Australia (2008)). Since the larger the distance r usually causes the higher variogram $\gamma(r)$, the weights are tuned so that nearer observation points are likely to have higher weights, similarly to IDW. In addition, to estimate the variogram accurately, Kriging assumes that the target point $\boldsymbol{x}_j$ is surrounded by neighbor observation points. Thus, in practice, neighbor observations are selected using Delaunay triangulation—vertices of the triangle containing the target point $\boldsymbol{x}_j$, i.e., 1^{st}-order triangle, and vertices of the triangle and its adjacent triangles, 2^{nd}-order triangle, are used as neighbor observed points $S(\boldsymbol{x}_j)$ (Hessami et al. (2001)).

2.1.3. Problem of linear model

These spatial interpolation methods estimate the value of unobserved points linearly with their neighbor observed values. This linear model estimation would be acceptable in regions with dense observation sites; however, it is likely to degrade the interpolation accuracy in sparse sites due to its high non-linearity against the observed values.

2.2. Image inpainting

Image inpainting is a computer vision task of reconstructing missing regions in an image (Elharrouss et al. (2020)). Let us denote training data by $N^{\rm tr}$-pair of image $Y \in \mathbb{R}^{d_{\rm h} \times d_{\rm w}}$ and its mask $M \in \mathbb{R}^{d_{\rm h} \times d_{\rm w}}$, i.e., $\mathcal{D}^{\rm tr} = \{(Y_e, M_e)\}_{e=1}^{N^{\rm tr}}$ where $d_{\rm h}$ and $d_{\rm w}$ are the height and width of an image, and mask M is defined as

$$M[h, w] = \begin{cases} 0 & \text{if pixel at } (h, w) \text{ is missing} \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Then, we consider an inpainting function $f(\cdot)$ to convert masked images into unmasked ones as

$$\widehat{Y_e} = f\big(Y_e \odot M_e\big). \tag{5}$$

where $\odot$ is element-wise multiplication. Recently, deep convolutional encoder-decoder network based approaches have been actively developed (Liu et al. (2018), Yan et al. (2018), Zeng et al. (2019)).

2.2.1. Partial convolution

Partial-conv is one of the state-of-the-art deep inpainting methods (Liu et al. (2018)) where the convolution operation is performed against a masked image as

$$\begin{aligned} \widehat{y^{l+1}} &= \begin{cases} \mathrm{pconv}(Y^l, M^l)\frac{\mathrm{sum}(\mathbb{1})}{\mathrm{sum}(c(M^l))} + b & \text{if } \mathrm{sum}(c(M^l)) > 0 \\ 0 & \text{otherwise,} \end{cases} \\ \mathrm{pconv}(Y, M) &\equiv \mathrm{sum}\Big(W \odot c\big(Y_M\big)\Big), \\ Y_M &\equiv Y \odot M \end{aligned} \tag{6}$$

where l, $W \in \mathbb{R}^{s \times s}$ and b are a layer index, kernel weight matrix and bias respectively, and $\mathrm{sum}(X)$ and $c(X) \in \mathbb{R}^{s \times s}$ are the sum of all elements in a matrix X and the cropped map X for the current convolution window respectively. In addition, $\mathbb{1} \in \mathbb{R}^{s \times s}$ and $\frac{\mathrm{sum}(\mathbb{1})}{\mathrm{sum}(c(M^l))}$ are the matrix of all elements being 1 and the ratio of missing area in the current convolution window, to adjust the scale of output $\widehat{y}$. We note that this operation of partial-conv without the bias-term at one layer is exactly same as the linear spatial interpolation model $f_{\rm linear}(\cdot)$ in Eq. 1.

The mask is also updated at each layer as

$$m^{l+1} = \begin{cases} 1 & \text{if } \mathrm{sum}(c(M^l)) > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where since the updated mask M is set to 1 even if the mask $c(M)$ contains only one observed pixel, the ratio of the missing area is shrunk layer by layer. Then, encoder-decoder network architecture, i.e., U-Net, is used as an inpainting function $f(\cdot)$ as

$$\widehat{Y_e} = f(Y_e \odot M_e) = (\mathrm{dec}_\phi \circ \mathrm{enc}_\theta)(Y_e \odot M_e), \tag{8}$$

where $\circ$ denotes function composition, and θ and ϕ are parameters of convolution in encoder and decoder networks respectively. Parameters in networks are tuned to minimize the loss function.

Through the encoder-decoder architecture with partial convolution and loss functions, partial-conv enables us to estimate the value of missing regions non-linearly with neighbor observed values with a smooth surface in terms of both values and semantic context.

2.2.2. Problem of translation invariance

The property of translation invariance in convolutional operations, would be adequate for general image inpainting tasks; that is, the position of target objects with missing regions, i.e., person, building, and scene, is arbitrary, and a common interpolation strategy would be helpful for all locations. However, in the spatial interpolation of environmental data, the relationship between the target and its surrounding observed point values varies from region to region depending on the environment, e.g., topography and subsurface structure, even if the surrounding observation patterns are the same. Therefore, the property of translation invariance could disturb fine-grained spatial interpolation.

3. Proposed method

To realize nonlinear and position-dependant fine-grained spatial interpolation in environmental data, we propose to apply and extend existing deep inpainting methods by introducing adaptive position-dependant convolution.

3.1. Creation of mask-map and image from environmental data

To apply deep inpainting methods to spatial interpolation problems, we encode the location of observation sites and spatial environmental surface data into images. Let us denote the coordinate transformation function from image pixel (h, w) to the world coordinate $\boldsymbol{x}_{hw}$, e.g., latitude and longitude, by $\boldsymbol{x}_{hw} = p(h, w)$. Then, we create mask map M expressing observation sites and an image expressing environmental surface data as

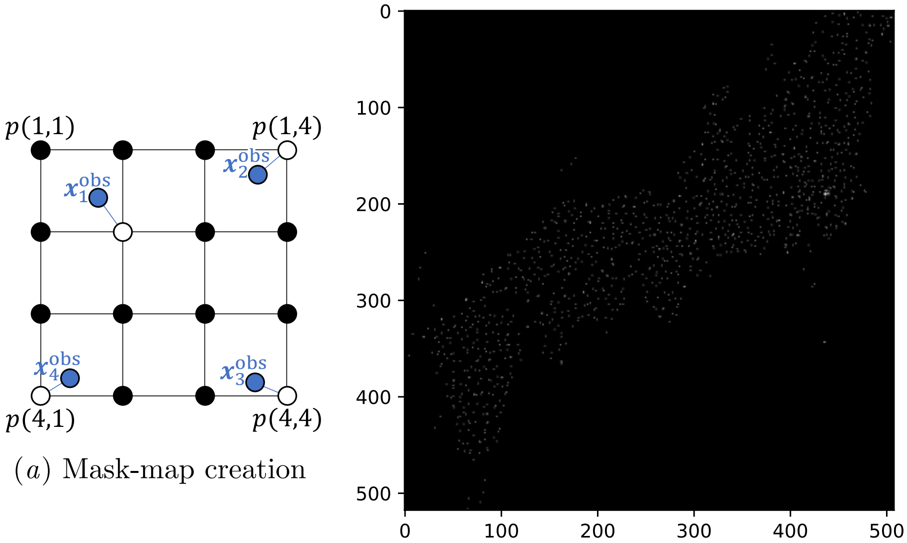
$$M[h, w] = \begin{cases} 1 & \text{if } (h, w) = \underset{(h', w') \in \{1 \dots d_{\mathrm{h}}\} \times \{1 \dots d_{\mathrm{w}}\}}{\operatorname{argmin}} \|\boldsymbol{x}^{\mathrm{obs}} - p(h', w')\|_2 \quad \forall \boldsymbol{x}^{\mathrm{obs}} \in X^{\mathrm{obs}} \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$$Y[h, w] = y\big(p(h, w)\big), \tag{10}$$

where $X^{\mathrm{obs}} = \{\boldsymbol{x}_i^{\mathrm{obs}}\}_{i=1}^{m}$ and the resolution of mask M and image Y, i.e., $d_{\mathrm{h}} \times d_{\mathrm{w}}$ is set as the same resolution of environmental surface data. Fig. 2(a)subfigure depicts the illustration of the process of mask-map creation and Fig. 2(b)subfigure depicts an example of mask created for seismic observation sites in Fig. 1(a)subfigure—the resolution of mask is set as the same resolution of the environmental surface data, $(d_{\mathrm{h}}, d_{\mathrm{w}}) = (518, 508)$. We note that the mean absolute error between actual seismic observation sites and mask observation points in latitude and longitude are as small as 0.00426 and 0.00626 degrees corresponding to about 0.47 and 0.57 km, respectively.

3.2. Position-dependent convolution

To overcome the translation invariance problem of convolution operation for spatial interpolation tasks, we propose a position-dependent kernel by introducing an adaptive position-feature map. Let us denote the position-feature map by $P \in \mathbb{R}^{d_{\mathrm{h}} \times d_{\mathrm{w}}}$ each element of which



(a) Mask-map creation

(b) Example of mask-map

Figure 2: (a) Illustration of mask creation process for $(d_{\text{h}}, d_{\text{w}}) = (4, 4)$ —if grid point $p(h, w)$ is the nearest point from one of observed points X^{obs}, then $M[h, w]$ is set to 1 (in white), otherwise 0. (b) An example of mask-map M created for observation sites in Fig. 1(a)subfigure—the resolution of mask is set as the same resolution of the environmental surface data, $(d_{\text{h}}, d_{\text{w}}) = (518, 508)$ in Fig. 2(b)subfigure. White dots and black surfaces indicate $M[h, w] = 1$ and $M[h, w] = 0$ respectively.

is a trainable scalar parameter. Then, as shown in Fig. 3, we transform kernel weight W dependently on the position of an image it applies using the position-feature map P as

$$W' = g\big(W, c(P)\big), \quad \text{ppconv}(Y, M, P) = \text{sum}\Big(W' \odot c(Y_M)\Big), \tag{11}$$

where $g(\cdot)$ and ppconv$(\cdot)$ are a kernel transformation and position-dependent convolution functions respectively. We consider various types of transformation.

3.2.1. Scale-adjustment function

We adjust the scale of each element of kernel-weight W dependent on the position as

$$g_{\text{scale}}\big(W, c(P)\big) = W \odot c(P + \mathbb{1}). \tag{12}$$

where $\mathbb{1}$ is the matrix of all elements being 1 with the same size of the position-feature map P. The position-dependent convolution ppconv$(\cdot)$ with the scale-adjustment can be

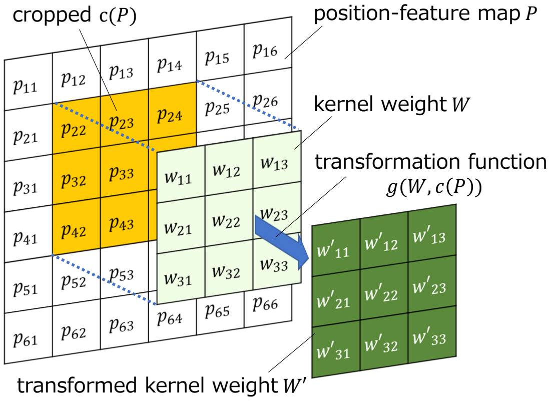


Figure 3: Illustration of position-dependent convolution in which kernel-weight W is transformed based on position-feature map P using transformation function $g(W, c(P))$.

performed with modification of partial-conv pconv$(\cdot)$ in Eq. 6 as

$$\begin{aligned}
\mathrm{ppconv}_{\mathrm{scale}}(Y, M, P) &= \mathrm{sum}\Big(\big(W \odot c(P + \mathbb{1})\big) \odot c(Y_M)\Big) \\
&= \mathrm{sum}\Big(W \odot c\big(Y_M\big)\Big) + \mathrm{sum}\Big(W \odot c\big(Y_{PM}\big)\Big), \\
&= \mathrm{pconv}(Y, M) + \mathrm{sum}\Big(W \odot c\big(Y_{PM}\big)\Big), \\
Y_{PM} &\equiv Y \odot P \odot M. \qquad (13)
\end{aligned}$$

where $\mathrm{sum}\Big(W \odot c\big(Y_{PM}\big)\Big)$ is a variant of partial-conv against masked and scale-adjusted image Y_{PM}.

3.2.2. Bias-adjustment function

We adjust the bias of each element of kernel-weight W dependent on the position as

$$g_{\mathrm{bias}}\big(W, c(P)\big) = W + c(P). \qquad (14)$$

The position-dependent convolution operation with the bias adjustment can also be performed with a small modification of pconv$(\cdot)$ as

$$\begin{aligned}
\mathrm{ppconv}_{\mathrm{bias}}(Y, M, P) &= \mathrm{sum}\Big(\big(W + c(P)\big) \odot c(Y_M)\Big) \\
&= \mathrm{sum}\Big(W \odot c\big(Y_M\big)\Big) + \mathrm{sum}\Big(c\big(P \odot Y_M\big)\Big) \\
&= \mathrm{pconv}(Y, M) + \mathrm{sum}\Big(c\big(Y_{PM}\big)\Big). \qquad (15)
\end{aligned}$$

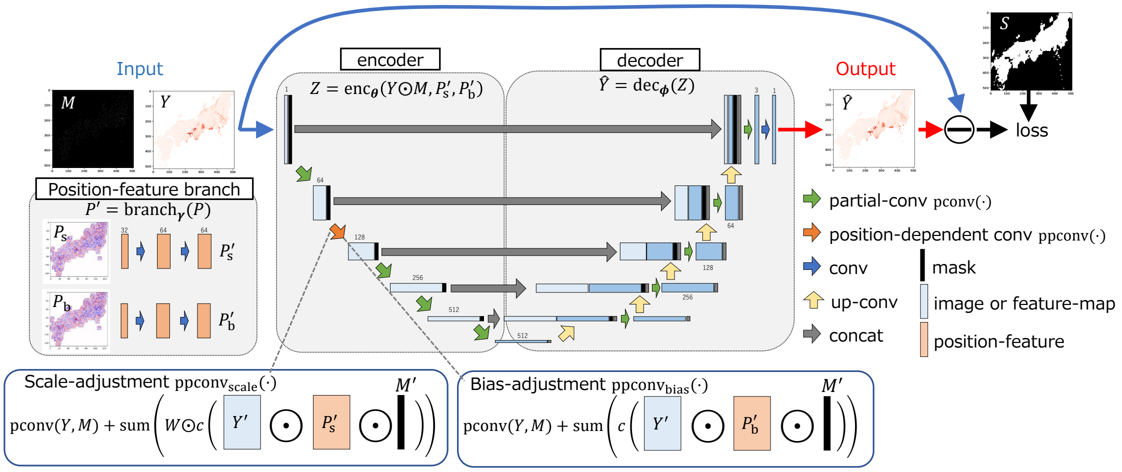

Input
M
Y
encoder
$Z = \mathrm{enc}_{\boldsymbol{\theta}}(Y \odot M, P'_\mathrm{s}, P'_\mathrm{b})$
decoder
$\hat{Y} = \mathrm{dec}_{\boldsymbol{\phi}}(Z)$
Output
$\hat{Y}$
S
loss
Position-feature branch
$P' = \mathrm{branch}_{\boldsymbol{\gamma}}(P)$
P_s
P_b
32
64
64
P'_s
P'_b
1
64
128
256
512
512
256
128
64
3
1
partial-conv $\mathrm{pconv}(\cdot)$
position-dependent conv $\mathrm{ppconv}(\cdot)$
conv
mask
up-conv
image or feature-map
concat
position-feature
Scale-adjustment $\mathrm{ppconv}_\mathrm{scale}(\cdot)$
M'
$\mathrm{pconv}(Y, M) + \mathrm{sum}\Big(W \odot c\big(Y' \odot P'_\mathrm{s} \odot\big)\Big)$
Bias-adjustment $\mathrm{ppconv}_\mathrm{bias}(\cdot)$
M'
$\mathrm{pconv}(Y, M) + \mathrm{sum}\Big(c\big(Y' \odot P'_\mathrm{b} \odot\big)\Big)$

Figure 4: Illustration of the framework of our proposed method, PoDIM for spatial interpolation, consisting of U-Net and position-feature branch. In encoder enc$(\cdot)$ and decoder dec$(\cdot)$ networks, partial convolutions are repeatedly applied to extract feature map from observed pixels. In position-feature branch, initial 1-channel position-feature maps P_s and P_b are converted to multi-channel feature maps P'_s and P'_b through convolution networks. Then, at a target layer of the encoder-network, e.g., $m = 2$, position-dependent convolution using transformation function; scale-adjustment $g_\mathrm{scale}(\cdot)$, bias-adjustment $g_\mathrm{bias}(\cdot)$ or affine-transformation $g_\mathrm{affine}(\cdot)$ is applied.

3.2.3. Affine transformation function

We adjust both the scale and bias of each element of kernel-weight W dependent on the position as

$$g_\mathrm{affine}\big(W, c(P_\mathrm{s}), c(P_\mathrm{b})\big) = W \odot c(P_\mathrm{s} + \mathbb{1}) + c(P_\mathrm{b}), \tag{16}$$

where P_s and P_b are position-feature maps for scale and bias adjustment. The convolution operation with the affine transformation can be performed by the combination of scale and bias adjustment functions as

$$\begin{aligned}\mathrm{ppconv}_\mathrm{affine}(Y, M, P_\mathrm{s}, P_\mathrm{b}) &= \mathrm{sum}\Big(\big(W \odot c(P_\mathrm{s} + \mathbb{1}) + c(P_\mathrm{b})\big) \odot c\big(Y_M\big)\Big) \\ &= \mathrm{ppconv}_\mathrm{scale}(Y, M, P_\mathrm{s}) + \mathrm{sum}\Big(c\big(Y_{PM}\big)\Big).\end{aligned} \tag{17}$$

3.3. Position-feature branch

To apply position-dependent convolution to arbitrary target layer m, we prepare position-feature map $P^m \in \mathbb{R}^{d^m_\mathrm{h} \times d^m_\mathrm{w}}$ with the same resolution as the one of the feature-map of the m-th target-layer and then convert it to $P^{m'} \in \mathbb{R}^{c^m \times d^m_\mathrm{h} \times d^m_\mathrm{w}}$ in position-feature branch with convolution operations as

$$P^{m'} = \mathrm{branch}_\gamma(P^m), \tag{18}$$

where γ is the parameter of convolution in the position-feature branch and $(c^m, d_{\mathrm{h}}^m, d_{\mathrm{w}}^m)$ are the channel, height, and width of the feature-map in the m-th layer.

3.4. Entire architecture and training

The architecture of our proposed method, called PoDIM (Position-dependent Deep Inpainting Method), is illustrated in Fig. 4, consisting of U-Net with partial and position-dependent convolution layers and position-feature branch. Thus, the inpainting function $f(\cdot)$ in our architecture is defined as

$$\widehat{Y}_e = f(Y_e \odot M_e) = (\mathrm{dec}_\phi \circ \mathrm{enc}_\theta)(Y_e \odot M_e, P_{\mathrm{s}}^{m'}, P_{\mathrm{b}}^{m'}). \tag{19}$$

Parameters θ, ϕ, P_{s}, P_{b} and γ are tuned so as to minimize the loss function as

$$\min_{\theta,\phi,\gamma} \left[\frac{1}{N^{\mathrm{tr}}} \sum_{e=1}^{N^{\mathrm{tr}}} \left\{10\mathcal{L}_e^{\mathrm{obs}} + \mathcal{L}_e^{\mathrm{unobs}}\right\}\right], \tag{20}$$

where $\mathcal{L}^{\mathrm{obs}}$ and $\mathcal{L}^{\mathrm{unobs}}$ are losses for observed and unobserved points, respectively, defined as

$$\begin{aligned} \mathcal{L}_e^{\mathrm{obs}} &\equiv \left\| S \odot M_e \odot \left(Y_e - \widehat{Y}_e\right) \right\|_1, \\ \mathcal{L}_e^{\mathrm{unobs}} &\equiv \left\| S \odot \left(1 - M_e\right) \odot \left(Y_e - \widehat{Y}_e\right) \right\|_1. \end{aligned} \tag{21}$$

where $S \in \mathbb{R}^{d_{\mathrm{h}} \times d_{\mathrm{w}}}$ is the evaluation target-mask to exclude outer region, e.g., sea and lakes, from the evaluation, defined as

$$S[h, w] = \begin{cases} 0 & \text{if pixel } (h, w) \text{ is in the outside of evaluation-region} \\ 1 & \text{otherwise.} \end{cases} \tag{22}$$

To train both encoder-decoder networks and position-feature branch effectively, we use multi-phase training. As the phase-1, we freeze position-feature map P_{s} and P_{b}, and parameters γ in position-feature branch after setting initial values as $P_{\mathrm{s}} = P_{\mathrm{b}} = 0$ so as to disable kernel transformation function $p(\cdot)$, i.e., $p_{\mathrm{scale}}(\cdot) = p_{\mathrm{bias}}(\cdot) = p_{\mathrm{affine}}(\cdot) = \mathrm{pconv}(\cdot)$. That is, in phase-1, the entire network is equivalent to partial-conv. In phase-2, we freeze parameters θ and ϕ in the encoder-decoder network and train only position-feature map and position-feature branches. Lastly, in phase-3, we refine the encoder-decoder networks with a fixed position-feature map and position-feature branch.

4. Experimental evaluation

We evaluate the performance of spatial interpolation of our proposed method, PoDIM using simulated ground-motion data, in comparison with classical spatial interpolation methods: IDW (in Sec. 2.1.1) and Kriging (in Sec. 2.1.2), and deep image inpainting method: partial-conv (in Sec. 2.2.1). We use PSNR (Peak Signal-to-Noise Ratio) for the performance evaluation of spatial interpolation, defined as

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{MAX}^2}{\mathrm{MAE}},$$
$$\mathrm{MAE} \equiv \frac{1}{N^{\mathrm{te}}\mathrm{sum}(S)}\mathrm{sum}\Big(S \odot \big(Y_e - \widehat{Y}_e\big) \odot \big(Y_e - \widehat{Y}_e\big)\Big). \tag{23}$$

where N^{te} and MAX are the number of test data and the maximum possible error, respectively. We perform L-fold cross-validation where data $\mathcal{D}$ are split into L-group $\{\mathcal{D}^l\}_{l=1}^L$ and a model is trained using $\{\mathcal{D}^l\}_{l\neq l'}$ and its performance is evaluated using $\mathcal{D}^{l'}$ for each $l' = 1, 2, \ldots, L$. Thus, in total, all data are used for the evaluation with L different models, i.e., $\mathcal{D}^{\mathrm{te}} = \{\mathcal{D}^l\}_{l=1}^L$.

We compare the performance of spatial interpolation generated by the following methods:

- IDW, $\alpha = 1$ & 1^{st}-triangle (in Sec. 2.1.1)—the power parameter is set as $\alpha = 1$ and three vertices of the triangle including a target point $\boldsymbol{x}^{\mathrm{tar}}$ are used as observation points $\{\boldsymbol{x}_i^{\mathrm{obs}}\}_{i=1}^3$. Triangles are generated by Delaunay triangulation as shown in Fig. 4 in the supplementary document.
- IDW, $\alpha = 1$ & 2^{nd}-triangle (in Sec. 2.1.1)—the vertices of adjacent triangles and the triangle including a target point $\boldsymbol{x}^{\mathrm{tar}}$ are used as observation points $\{\boldsymbol{x}_i^{\mathrm{obs}}\}_{i=1}^{3\sim 6}$.
- IDW, $\alpha = 2$ & 1^{st}-triangle (in Sec. 2.1.1)—the power parameter is set as $\alpha = 2$
- IDW, $\alpha = 2$ & 2^{nd}-triangle (in Sec. 2.1.1)
- Kriging, linear & 1^{st}-triangle (in Sec. 2.1.2)—linear variogram model is used
- Kriging, linear & 2^{nd}-triangle (in Sec. 2.1.2)
- Kriging, exp & 1^{st}-triangle (in Sec. 2.1.2)—exponential variogram model is used
- Kriging, exp & 2^{nd}-triangle (in Sec. 2.1.2)
- Partial-conv (in Sec. 2.2.1)—10-layer encoder-decoder-network with skip-connection, which corresponds to the model trained at the phase-1 in our proposed method, PoDIM (see Sec. 3) with disabled position-feature branch, as shown in Fig. 4
- PoDIM, $m = 2$ & scale (in Sec. 3)—our proposed method where scale adjustment function $g_{\mathrm{scale}}(\cdot)$ is applied at $m = 2$ layer as shown in Fig. 4, in which given the model trained at the phase-1 for partial-conv, it is updated through the phase-2 and -3.
- PoDIM, $m = 2$ & bias (in Sec. 3)—our proposed method where bias adjustment function $g_{\mathrm{bias}}(\cdot)$ is applied at $m = 2$ layer
- PoDIM, $m = 2$ & affine (in Sec. 3)—our proposed method where affine transformation function $g_{\mathrm{affine}}(\cdot)$ is applied at $m = 2$ layer

As for training partial-conv and PoDIM, early stopping based on the PSNR of randomly selected validation data from $\mathcal{D}^{\mathrm{tr}}$ with the patience of 10-epoch is used, and the model with the maximum PSNR in the training epochs is saved.

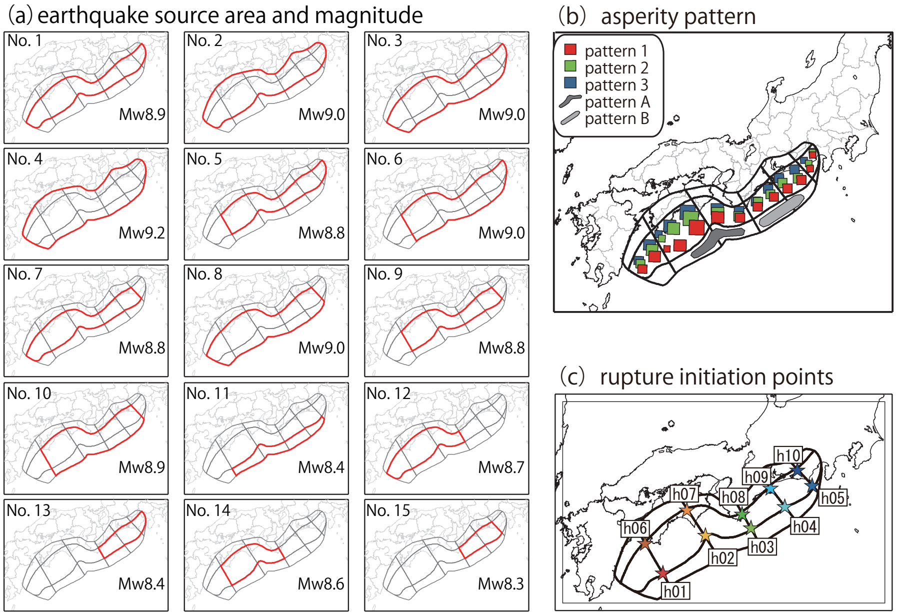


Figure 5: Variability of the possible earthquake rupture parameters for the anticipated megathrust earthquakes in the Nankai trough (Maeda et al.). (a) 15 patterns of the earthquake source area, denoted by red lines, and the earthquake magnitude labeled as Mw. (b) Spatial patterns of the location of the asperities. Asperities are set according to the combination of three patterns of the square-shaped patches in the deeper zone on the fault with two patterns of banana-shaped patches in the shallower zone near the trench. (c) 10 patterns of the location of the rupture initiation points, denoted by stars.

4.1. Ground-motion data

For the experiments, we used simulated ground-motion data obtained from 3-D earthquake ground-motion simulations for as many as 360 rupture scenarios for the anticipated megathrust earthquakes in the Nankai trough (Maeda et al.). Ground-motion simulation is a numerical computation (finite-difference) of seismic wave propagation from the earthquake source to the ground surface within a 3-D medium, as shown in the schematic drawing in Fig. 5 in the supplementary document. The earthquake source fault, or the rupture area within Earth's crust, is modeled as an ellipse-like shape extending beneath the islands and consists of several asperities, or the patches on the fault that are modeled to generate stronger seismic waves compared to the other area. In this study, the rupture process of each earthquake source model is referred to as rupture scenario. As depicted in Fig. 5, 360 rupture scenarios have been set to describe the variability of the possible earthquake rupture parameters, including (a) the earthquake source area and magnitude, (b) the spatial pattern of the asperity locations, and (c) the location of the rupture initiation point, etc.; therefore, the simulated ground motion varies from one scenario to another, reflecting the difference in the rupture parameters.

The simulated ground-motion waveforms are transformed into 5% damped velocity response spectra (Sv) at the period of 5 seconds, a type of ground-motion intensity often used for earthquake engineering purposes. The simulated Sv data are obtained at a total of 77609 grids on the ground surface with approximately 2 km intervals. Fig. 1(b)subfigure depicts an example of simulated ground-motion data when parameters (a), (b) and (c) in Fig. 5 are set to No.1, the combination of pattern-1 and pattern-A, and h01, respectively. Note that we clip the value of response spectra in the range $[0, 8]$ and normalize it to $[0, 1]$ for experimental evaluation purpose.

We consider the following three experimental settings;

- All—all 360 scenarios using $L = 10$ cross-validation
- East—150 scenarios with east rapture initiation points, h04, h05, h09 and h10 (see Fig. 5 (c)) using $L = 5$ cross-validation
- West—140 scenarios of west rapture initiation points, h01, h02, h06 and h07 (see Fig. 5 (c)) using $L = 5$ cross-validation

4.2. Evaluation results

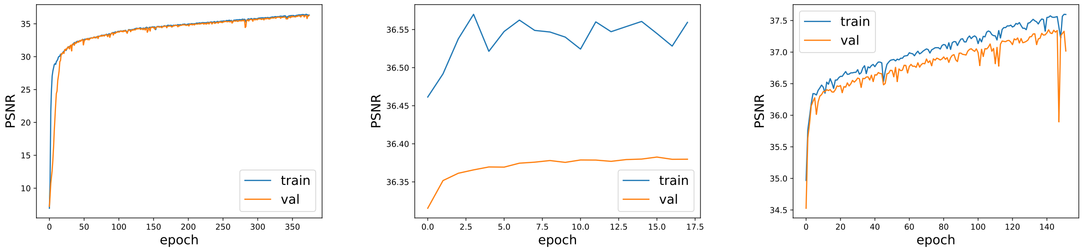
Table 1 depicts the performance comparison using PSNR for three settings: All, East, and West. The table shows that among classical methods, more complex models, e.g., IDW with $\alpha = 2$ and Kriging with exponential variogram, using a greater number of observations, e.g., 2^{nd}-triangle tend to have higher performance, indicating that spatial interpolation in ground-motion would require models to have high expression capability. Related to this point, the table shows that deep inpainting methods tend to have much higher performance than all variants of classical interpolation methods since inpainting methods have highly nonlinear representation capability w.r.t. observed values $Y \odot M$ and high-dimensional models are optimized through supervised learning. Our proposed method, PoDIM, further improves the performance over partial-conv in all data sets; All, East and West, more than about 1-point, indicating position-dependent convolution applied at the layer $m = 2$ is effective for locally fine-grained interpolation. Fig. 6 depicts examples of PSNR values computed using training and validation data in the phase-1, -2, and -3 of training of our proposed method, PoDIM. The figures show that although the training of phase-1, corresponding to partial-conv converges, by training the position-feature map P and position-feature branch $\text{branch}_\gamma(\cdot)$ in the phase-2, PSNR further improves in phase-3, indicating position-dependent convolution enables fine-grained interpolation.

Fig. 7 depicts examples of masked image $Y \odot M$, true image Y, and interpolated image $\widehat{Y}$ and Fig. 8 depicts true vs. predicted values. Figures show that given extremely sparse observed values (b), our proposed method, PoDIM (f), can produce fine-grained interpolated images which look much similar to the true one (a) and have higher PSNR scores than existing methods.

Overall, these experimental results show that our proposed method, PoDIM, could be an effective spatial interpolation method, as our proposed method outperforms existing methods.

Table 1: Performance comparison using PSNR for ground-motion data. The method with the best score is indicated in bold.

Group	Method	All	East	West
Classic	IDW, $\alpha = 1$ & 1^{st}-triangle	29.10	29.76	27.82
	IDW, $\alpha = 1$ & 2^{nd}-triangle	28.84	29.57	27.57
	IDW, $\alpha = 2$ & 1^{st}-triangle	29.16	29.79	27.86
	IDW, $\alpha = 2$ & 2^{nd}-triangle	29.18	29.83	27.91
	Kriging, linear & 1^{st}-triangle	28.64	29.30	27.41
	Kriging, linear & 2^{nd}-triangle	28.28	29.02	27.05
	Kriging, exp & 1^{st}-triangle	28.65	29.32	27.42
	Kriging, exp & 2^{nd}-triangle	28.90	29.55	27.65
Inpainting	Partial-conv	35.32	33.27	31.61
	PoDIM, $m = 2$ & scale	**36.79**	**33.95**	**32.44**
	PoDIM, $m = 2$ & bias	36.44	33.87	32.70
	PoDIM, $m = 2$ & affine	36.30	33.83	32.35



(a) phase-1 (b) phase-2 (c) phas-3

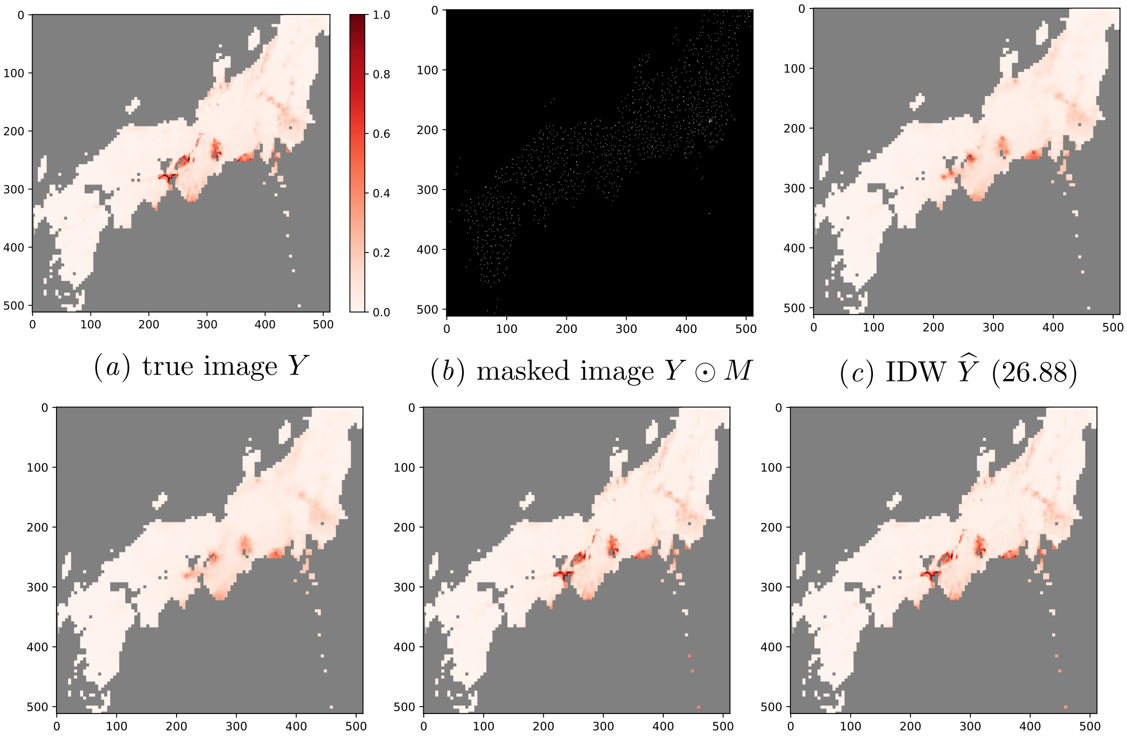
Figure 6: Examples of training curves; PSNR values computed using training and validation data over epochs in phase-1, -2, and -3 training in our proposed method, PoDIM with scale-adjustment for $l = 1$-fold cross validation in all seismic motion data.

5. Conclusion

In this work, we proposed a new deep learning framework for spatial interpolation for environmental data by introducing position-dependent convolution into U-Net-based image inpainting architecture where the scale and bias of kernel weights are adjusted depending on its position on an image. We showed the effectiveness of our proposed methods through experiments with seismic ground-motion data. Further analysis of the proposed method with other environmental data would be future work.

References

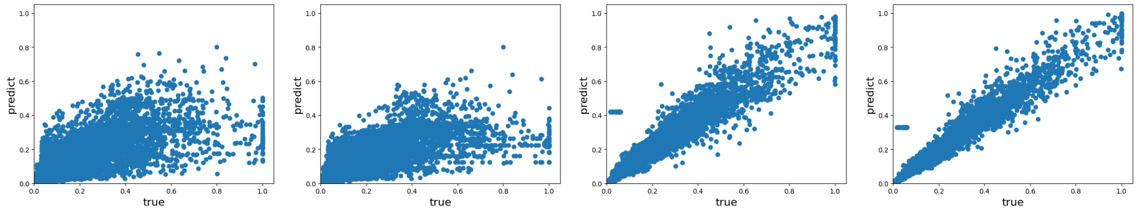
O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari. Image inpainting: A review. *Neural Processing Letters*, 51(5):2007–2028, 2020.



(a) true image Y (b) masked image $Y \odot M$ (c) IDW $\widehat{Y}$ (26.88)

(d) Kriging $\widehat{Y}$ (26.53) (e) Partial-conv $\widehat{Y}$ (34.08) (f) PoDIM scale $\widehat{Y}$ (36.72)

Figure 7: Examples of true (a), masked (b), and interpolated images (c)-(f) for the earthquake with rupture initiation point of h01 (see Fig. 1(b)subfigure). The value in parenthesis () indicates the PSNR score. Note that true Y and interpolated image $\widehat{Y}$ are masked (in gray) by evaluation target-mask S for the visualization purpose.



(a) IDW (b) Kriging (c) partial-conv (d) PoDIM scale

Figure 8: True vs. predicted values for Fig. 7.

M. Hessami, F. Anctil, and A. A. Viau. Delaunay implementation to improve kriging computing efficiency. *Computers & Geosciences*, 27:237–240, 2001.

J. Li and Geoscience Australia. *A Review of Spatial Interpolation Methods for Environmental Scientists*. Geoscience Australia, 2008.

Z. Li, K. Wang, H. Ma, and Y. Wu. An adjusted inverse distance weighted spatial interpolation method. In *Proceedings of the 3rd International Conference on Communications, Information Management and Network Security (CIMNS2018)*, pages 128–132, 2018.

G. Liu, F. A. Reda, K. J. Shih, T. C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

G. Y. Lu and D. W. Wong. An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences*, 34(9):1044–1055, 2008.

T. Maeda, A. Iwaki, N. Morikawa, S. Aoi, and H. Fujiwara. Seismic-hazard analysis of long-period ground motion of megathrust earthquakes in the nankai trough based on 3d finite-difference simulation. *SEISMOLOGICAL RESEARCH LETTERS*, (6):1265–1273, 11 .

G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

M. P. Moschetti, S. Hartzell, L. Ramirez-Guzman, A. D. Frankel, S. J. Angster, and W. J. Stephenson. 3d ground-motion simulations of m w 7 earthquakes on the salt lake city segment of the wasatch fault zone: Variability of long-period (t $\geq$ 1s) ground motions and sensitivity to kinematic rupture parameters. *Bulletin of the Seismological Society of America*, (64):1704–1723.

National Research Institute for Earth Science and Disaster Resilience (NIED) . Nied k-net, kik-net, a.

National Research Institute for Earth Science and Disaster Resilience (NIED) . Site list. https://www.kyoshin.bosai.go.jp/kyoshin/db/index_en.html?all, b.

Z. Yan, X.Li, M. Li, W. Zuo, and S. Shan. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–17, 2018.

R. Yang and B. Xing. A comparison of the performance of different interpolation methods in replicating rainfall magnitudes under different climatic conditions in chongqing province (china). *Atmosphere*, 12(10), 2021.

Y. Zeng, J. Fu, H. Chao, and B. Guo. Shift-net: Image inpainting via deep feature rearrangement. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1486–1494, 2019.