3D Multiresolution Velocity Model Fusion With Probability Graphical Models

Zheng Zhou, Peter Gerstoft, K. B. Olsen

February 9, 2024

Corresponding author: Zheng Zhou, zhz039@ucsd.edu

Abstract

The variability in spatial resolution of seismic velocity models obtained via tomographic methodologies is attributed to many factors, including inversion strategies, ray path coverage, and data integrity. Integration of such models, with distinct resolutions, is crucial during the refinement of community models, thereby enhancing the precision of ground motion simulations. Toward this goal, we introduce the Probability Graphical Model (PGM), combining velocity models with heterogeneous resolutions and non-uniform data point distributions. The PGM integrates data relations across varying-resolution subdomains, enhancing detail within low-resolution domains by utilizing information and prior knowledge from high-resolution subdomains through a maximum posterior (MAP) problem. Assessment of efficacy, utilizing both 2D and 3D velocity models—consisting of synthetic checkerboard models and a fault zone model from Ridgecrest, CA—demonstrates noteworthy improvements in accuracy, compared to state-of-the-art fusion techniques. Specifically, we find reductions of 30% and 44% in computed travel-time residuals for 2D and 3D models, respectively, as compared to conventional smoothing techniques. Unlike conventional methods, the PGM's adaptive weight selection facilitates preserving and learning details from complex, non-uniform high-resolution models and applies the enhancements to the low-resolution background domain.

Introduction

The integration of tomography velocity models with different resolutions is important for refining community models, especially in applications such as ground motion estimation or dynamic rupture modeling, where varying scales are imperative (e.g., Ajala and Persaud, 2022; Yeh and Olsen, 2023).

The fusion of high-resolution (HR) and low-resolution (LR) models poses challenges due to the potential emergence of sharp boundaries and misaligned patterns. Apart from being physically unrealistic, such patterns can result in artifacts in ground motion simulations.

To address the velocity model fusion problem, several notable techniques have been developed. The Gaussian kernel filter (Ghosh, 2018), a widely used method in image and signal processing, applies a Gaussian kernel to data for smoothing, with the degree of smoothing determined by the kernel's bandwidth or standard deviation. This technique is effective in enhancing the clarity of the data and requires precise parameter tuning to avoid data distortion. The cosine taper window interpolation, as discussed in Ajala and Persaud (2021) and Ajala et al. (2022), is another method focusing on noise reduction. It employs a cosine taper to reduce signal amplitude at the sequence ends, thereby merging the two velocity models effectively while preserving their overall characteristics. Finally, Dictionary Learning Smoothing, explored in studies by Yang et al. (2012), Bianco and Gerstoft (2018), and Zhang and Ben-Zion (2023), offers an advanced approach for smoothing signals or images. This technique involves creating a sparse representation of data through dictionary learning, enabling effective noise removal while maintaining the data's underlying structure, albeit with high computational demands and extensive training data requirements.

Inspired by advancements in image super-resolution (Cheung et al., 2018) and image editing (Dhamo et al., 2020; Zhang et al., 2018), we present a method to fuse seismic tomography models employing Probability Graphical Models (PGMs). Our proposed fusion technique not only accentuates the local HR structure but also safeguards global smoothness in the resulting model, a step forward in addressing the complexities in 3D tomography model fusion.

The problem of combining multiscale models appears across various geophysical fields, revealing the scale-dependent nature of anisotropy and introducing substantial implications for our understanding of Earth's structure (Van Houtte et al., 2006). Notably, the problem of synthesizing models becomes more complicated when considering the powerful spatial and directional dependency of tomographic resolution, which might induce direction-oriented smoothing (Dhamo et al., 2020). While conventional Gaussian kernel smoothing (Ghosh, 2018) has shown good efficacy in simple cases, their capacity to capture the intricate nuances of Earth's structure might be limited for complex models.

Due to the inherent characteristics of graphical methods, our proposed approach embodies the property of invariance under rotations or angular transformations. Furthermore, it easily accommodates the extension of larger neighboring patch sizes, facilitating the adaptation to varying volumes of training data and allowing for the accommodation of varying data quantities during training. Finally, our approach enables more inclusive, adaptive, and precision-enhanced modeling of Earth's subsurface structures, showing promise for the PGM in the fusion of 3D tomographic models spanning varied resolutions.

PGMs, capable of processing complex structures due to their ability to discern inherent relations among images (Ortega et al., 2018; Shuman et al., 2013), represent a promising tool for seismic analysis, including the study of reflection and seismic attributes surrounding low-velocity zones. Expanding beyond standard graphs, PGMs have been extended to higher-dimensional spaces, such as multilayer graphs (Das and Ortega, 2020) and hypergraphs (Zhang et al., 2022), and have been used in several seismic applications (e.g., Mu and Yuen, 2016; Zhang et al., 2023; Zhao et al., 2022).

Within all the PGMs, Markov Random Field (MRF) is a prevalent and highly effective approach for tackling supervised structure learning tasks that encompass the intricate mapping of complex geometric structures, as articulated by Murphy (2001). MRFs have been instrumental in the area of image restoration and edition, which was initially conceived by Geman and Graffigne (1986). This approach is rooted in Bayesian inference principles, applied to a spatially stochastic model. In stark contrast to convolution-based methods, the MRF procedure has been empirically validated to yield optimal and mathematically tractable results in the context of image processing, as substantiated by Blake et al. (2011).

To combine realistic tomography velocity models with unevenly distributed patterns, we propose a PGM that captures the relations between subdomains with different resolutions. Our focus is on models that distinctly segregate high-resolution (HR) and low-resolution (LR) areas. By learning information from the HR subdomain, we aim to enhance the details within the LR regions. This enhancement is achieved through a maximum likelihood formulation, incorporating prior knowledge from the HR areas.

Tests are performed on both a checkerboard and a fault zone model derived from the 2019 Ridgecrest, CA, region to demonstrate its efficacy. Generally, a lower travel time misfit indicates a more accurate velocity model of the Earth's subsurface, which in turn leads to more precise ground motion simulations (Edwards and Fäh, 2013). Our model is evaluated by the misfit between observed and calculated travel times and demonstrates that our PGM is generally superior to widely used conventional methods (see Experiments section).

The contributions of this article are two-fold: we introduce (1) a PGM for combining 3D tomography models with various resolutions, and (2) an anisotropic mechanism as a guide for the graph learning process.

Fundamental Model Setup

For two partially observed velocity fields $\boldsymbol{A}_{\mathrm{LR}}$ and $\boldsymbol{A}_{\mathrm{HR}}$, the task is to estimate the true velocity field $\boldsymbol{A}$. Here, we let $\boldsymbol{A}_{HR}$ represent a high-resolution velocity field on pixels inside the low-resolution $\boldsymbol{A}_{\mathrm{LR}}$ velocity field. In this paper, we focus on optimally merging borders between $\boldsymbol{A}_{\mathrm{LR}}$ and $\boldsymbol{A}_{\mathrm{HR}}$, as illustrated in Fig 1 using an excerpt from the Statewide California Earthquake Center (SCEC) Community Velocity Model (CVM) version S-4.26 as well as a HR fault zone model from the Ridgecrest, CA, area. We illustrate our method using 6 labels, a choice that will later be shown to be optimal in the trade-off between model complexity and computational cost (see

Section Experiments).

In our graphical model, a discrete class label map helps tie the spatial velocity field together. The label maps represent different geological structures, defined by their association with certain velocity intervals. The label map is initialized from the continuous velocity map $\boldsymbol{A}$ (Fig. 2, left), where we define a 6-cluster discrete label map $\boldsymbol{X}$ (Fig. 2, right) containing 6 velocity intervals (labeled 1–6), which is obtained from the continuous velocity maps $\boldsymbol{A}$.

In 3D models, the pixels are described by (i, j, k) coordinates and contain both a label $X_{i,j,k}$ and a velocity $A_{i,j,k}$. The velocity $A_{i,j,k}$ with the label $X_{i,j,k} = n$ (n represents the labels, an example is shown in Fig. 2, right) follows a Gaussian distribution $\mathscr{N}(\mu_n, \sigma_n^2)$ with mean μ_n and variance σ_n^2. Velocities at different pixels but with the same labels follow the same distribution. Thus, in a graph, the velocities $\boldsymbol{A}$ are on top of the labels $\boldsymbol{X}$ (Fig. 3). d denotes the set of all possible labels of $X_{i,j,k}$ (here, $n = \{1, \dots, 6\}$), and D represents the set of all possible combinations of labels $\boldsymbol{X}$ for the entire map. The whole map is tied together via the class labels $\boldsymbol{X}$ that depend on the neighboring class labels indicated by the graphical grid structure in Fig. 3. For each point (i, j, k) the neighboring class $\mathscr{N}_{i,j,k}$ is defined by its four immediate points. Note that we use the points at certain regions $\mathscr{V}$, i.e., at the border between the low- and high-resolution maps.

Markov Random Field Models (MRFs)

Bayesian Estimation

Given the prior probabilities $P(\boldsymbol{X})$ of label $\boldsymbol{X}$ and the likelihood densities $P(\boldsymbol{A} \mid \boldsymbol{X})$ of the observed velocity $\boldsymbol{A}$, the posterior probability can be formulated through Bayes' theorem as:

$$P(\boldsymbol{X} \mid \boldsymbol{A}) = \frac{P(\boldsymbol{A} \mid \boldsymbol{X})P(\boldsymbol{X})}{P(\boldsymbol{A})} \propto P(\boldsymbol{A} \mid \boldsymbol{X})P(\boldsymbol{X}). \tag{1}$$

Here, the probability density function (PDF) $P(\boldsymbol{A})$ of $\boldsymbol{A}$ is a fixed probability distribution (for given $\boldsymbol{A}$) and does not affect the maximum a posteriori (MAP) estimation solution. The Bayesian labeling problem requires finding the MAP configuration. The MAP of labeling for observation $\boldsymbol{A}$ is given by:

$$\boldsymbol{X}^* = \arg\max_{\boldsymbol{X}\in D} P(\boldsymbol{X} \mid \boldsymbol{A}), \tag{2}$$

where D denotes a set of possible candidates of the discrete labels $\boldsymbol{X}$, and $\boldsymbol{A}$ represents the observation of the continuous velocities (Dudik et al., 2004). To derive the MAP solution, both the prior probability and the likelihood function are needed. The likelihood function $P(\boldsymbol{A} \mid \boldsymbol{X})$ captures the conditional relation between the observation (refers to the continuous velocity in our research) and the hidden states (the variable, which corresponds to the discrete labels, cannot be directly observed here).

MRF Prior and Posterior Energy

A model can be considered a valid MRF if and only if the probability distribution $P(\boldsymbol{X})$ of its configurations adheres to an exponential distribution with appropriate normalization, described in the subsequent form

$$P(\boldsymbol{X}) = \frac{1}{Z_1} e^{-U_{\text{prior}}(\boldsymbol{X})}, \tag{3}$$

where Z_1 is a normalizing constant, and $U_{\text{prior}}(\boldsymbol{X})$ is the prior energy (Section 4.2 in Koller and Friedman, 2009). The prior energy $U_{\text{prior}}(\boldsymbol{X})$ can be expressed as the sum of neighboring potentials.

$$U_{\text{prior}}(\boldsymbol{X}) = \sum_{n\in\mathscr{N}} \theta_n(\boldsymbol{X}) = \sum_{\{(i,j,k)\}\in\mathscr{N}^0_{i,j,k}} \theta_0\left(X_{i,j,k}\right) + \sum_{\{(i',j',k')\}\in\mathscr{N}^1_{i,j,k}} \theta_1\left(X_{i,j,k}, X_{i',j',k'}\right) + \cdots, \tag{4}$$

where $\mathscr{N}$ denotes the complete set of potential neighboring systems. The 0th- and 1st-order neighboring systems are represented as $\mathscr{N}^0_{i,j,k}$ and $\mathscr{N}^1_{i,j,k}$, respectively, with the corresponding potentials given by θ_0 and θ_1. The 0th-order neighboring system is defined by considering every possible index (i, j, k). The 1st-order neighboring system is defined by considering every index (i', j', k') with a grid (Manhattan) distance 1, see Fig. 4. For the sake of brevity, only the 0th-order and 1st-order neighboring potentials are retained, while the higher-order potentials are truncated in Eq. (4).

Given the assumption that the velocities $\boldsymbol{A}$ associated with specific labels $\boldsymbol{X}$ adhere to Gaussian distributions, it is possible to represent the likelihood function in the exponential format

$$P(\boldsymbol{A} \mid \boldsymbol{X}) = \frac{1}{Z_2} e^{-U_{\text{like}}(\boldsymbol{A}|\boldsymbol{X})}, \tag{5}$$

where $U_{\text{like}}(\boldsymbol{A} \mid \boldsymbol{X})$ is the likelihood energy. Invoking the Bayes rule as presented in Eq. (1), it can be easily inferred that the posterior probability follows an exponential distribution

$$P(\boldsymbol{X} \mid \boldsymbol{A}) = \frac{1}{Z_3} e^{-U_{\text{post}}(\boldsymbol{X}|\boldsymbol{A})}, \tag{6}$$

where Z_2 and Z_3 are normalization constants, and U_{post} is the posterior energy. Taking the negative logarithm in Eqs. (5)–(6) gives the posterior energy

$$U_{\text{post}}(\boldsymbol{X} \mid \boldsymbol{A}) = U_{\text{prior}}(\boldsymbol{X}) + U_{\text{like}}(\boldsymbol{A} \mid \boldsymbol{X}) + C, \tag{7}$$

where C is a constant associated with the normalization constants Z_1, Z_2, and Z_3 and U_{prior} is the prior energy. Consequently, for a group of given $\boldsymbol{A}, X$ is defined as an MRF depending on d with a space of all the possible states $\mathscr{N}$. The MAP solution is determined equivalently by

$$\boldsymbol{X}^* = \arg\min_{\boldsymbol{X} \in D} U_{\text{prior}}(\boldsymbol{X} \mid \boldsymbol{A}), \tag{8}$$

which minimizes a negative log-likelihood problem in Eq. (8).

In summary, the methodology for MRF modeling is delineated in the following sequential steps:

1. Specification of a neighborhood system, represented as $\mathscr{N}$.

2. Definition of prior potentials, denoted as θ_0 and θ_1.

3. Derivation of the likelihood energy, given by $U_{\text{like}}(\mathbf{A}|\mathbf{X})$.

4. Computation of the posterior energy, $E(\mathbf{X})$, which can be expressed as a summation of neighboring potential functions.

From Eqs. (5)–(8), the posterior probability $P(\boldsymbol{X}|\boldsymbol{A})$ can be decomposed into the prior energy $U_{\text{prior}}(\boldsymbol{X})$, that can be quantified via multiple potentials and the likelihood function energy $U_{\text{like}}(\boldsymbol{A}|\boldsymbol{X})$. This observation substantiates the rationale for employing MRF priors, as it enables the assessment of conditional probabilities $P(\boldsymbol{X}|\boldsymbol{A})$ without the need for knowledge of their specific mathematical expressions.

Probability Graphical Model (PGM)

Our PGM follows a first-order MRF setting (Fig. 4) where each random variable has four neighbors on which it is conditionally dependent. The full conditional probability of the discrete random variable $X_{i,j,k} \in \{1,\ldots,n\}$ is the exponential of the sum of potentials (four 1st-order neighboring potentials θ_1 between cluster labels and one 0th-order center data potential θ_0 between cluster label and velocity) in conventional MRF settings. In image processing problems, optimizing the entire map can be broken down into a group of suboptimization problems that optimize each pixel iteratively (Pulli et al., 2012). Inserting (4) into (7), we have

$$\begin{aligned} -\log p(X_{i,j,k} \mid A_{i,j,k}) &= U_{\text{post}}(X_{i,j,k} \mid A_{i,j,k}) \\ &\propto \theta_0(X_{i,j,k}, A_{i,j,k}) + \sum_{(i',j',k') \in \mathcal{N}_{i,j,k}} \theta_1(X_{i,j,k}, X_{i',j',k'}) + C, \end{aligned} \tag{9}$$

$$\theta_0\left(X_{i,j,k}, A_{i,j,k}\right) = \frac{(A_{i,j,k} - \mu_n)^2}{\sigma_n^2}, \tag{10}$$

where θ_0 is the 0th-order neighboring potential (Li, 2012) (also known as the data cost function) that relates $X_{i,j,k}$ to the observed velocity data $A_{i,j,k}$. μ_n and σ_n^2 are the mean and variance, respectively, of all pixels with the same cluster label $n = X_{i,j,k}$. It promotes that continuous velocity values $\boldsymbol{A}$ sharing pixels with the same discrete label $\boldsymbol{X}$ follow the same Gaussian distribution.

$$\theta_1 \left(X_{i,j,k}, x_{i',j',k'} \right) = 1 - \delta(X_{i,j,k}, X_{i',j',k'}) \tag{11}$$

is the 1st-order neighboring potential (Li, 2012) (also known as the smoothness cost function), where δ is the Dirac delta function, that relates $X_{i,j,k}$ to the 1st-order neighboring variable $X_{i',j',k'}$ (see Fig. 4). This function encourages the neighboring pixels to share the same discrete label $X_{i,j,k}$, promoting the model's local smoothness.

The performance of standard or potential function-based MRF schemes can be limited when dealing with complex geological structures (Zhou et al., 2023). Assigning different neighboring pixels with various importance weights based on anisotropy patterns can effectively remove the non-uniform direction-dependent features of the model gradients, leading to improved inversion results, especially relevant for real, complex geological structures.

The objective function for the MAP problem of $X_{i,j,k}$ becomes

$$\begin{aligned} X^*_{i,j,k} = &\arg\max_{\boldsymbol{X}_{i,j,k}} p(X_{i,j,k} \mid A_{i,j}) = \arg\min_{\boldsymbol{X}_{i,j,k}} \omega^0_{i,j,k} \theta_0(X_{i,j,k}, A_{i,j,k}) \\ &+ \sum_{(i',j',k') \in \mathscr{N}_{i,j,k}} \omega^1_{i',j',k'} \theta_1(X_{i,j,k}, X_{i',j',k'}) + C, \end{aligned} \tag{12}$$

where $\omega_{i,j,k}$ and $\omega_{i',j',k'}$ are the weights which balance the anisotropic characteristics. These weights are typically set to uniform default values, given by the number of pixels in the local neighborhood, here $\omega^0_{i,j,k} = 1/1$ and $\omega^1_{i',j',k'} = 1/6$. This implies an equal contribution from each neighboring pixel.

MCMC and Gibbs Sampling

Markov Chain Monte Carlo (MCMC) is a statistical method used to sample probability distributions (Melas and Wilson, 2002; Sambridge and Mosegaard, 2002). Gibbs sampling is a specific MCMC algorithm that can be used to iteratively sample a multivariate probability distribution from the conditional distributions of each variable given the current values of the other variables (Carlo, 2004). Combining MCMC with Gibbs sampling enables estimating complex probability distributions without explicit knowledge of the distribution.

We employ the MCMC method with Gibbs sampling to solve Eq. (8). Gibbs sampling generates a new sample of $X_{i,j,k}$ directly from its distribution conditioned on the labels of its neighbors $X_{i',j',k'}$ and $A_{i,j,k}$. In the MRF structure, the update is achieved by calculating the probability for each of the possible labels (here, $n \in \{1, \cdots, 6\}$) at (i, j, k) using Eq. (12) and randomly selecting from this distribution (refer to Fig. 5).

The velocity map $\boldsymbol{A}$ is initialized with the superimposed HR and LR velocity maps, see Fig. 1 (a) and (b), and the label map $\boldsymbol{X}$ is initialized with a Gaussian mixture model clustering with the total cluster number N (here, 6), similar to Fig. 7 (a2, b2). All velocities with the label n follow the same Gaussian distribution $N(\mu_n, \sigma_n^2)$. We then apply the expectation–maximization (EM) algorithm (McLachlan and Krishnan, 2007), an iterative method to find the MAP estimates of the parameters, which updates the Gaussian parameters μ_n and σ_n^2. The termination criterion is either reaching a predefined maximum number of iterations (here 10,000) or observing that the cumulative absolute difference across all pixels between consecutive iterations falls below an error threshold, whichever is achieved first. The algorithm is summarized in Table 1.

Summarizing the algorithm from an intuitive perspective, our PGM adjusts each point in the grid-based method, not only on the point itself as in many conventional approaches, but also on the values of the surrounding points. The model processes each pixel, adjusting its values to align more closely with its neighbors, resulting in smoother and more consistent results. Our approach is analogous to a diffusion process, similar to introducing ink into clear water, where the resulting patterns gradually spread throughout the entire system. In the context of image processing, the algorithm methodically traverses each pixel, recalibrating its coloration to achieve harmonious alignment with nearby pixels. This paradigm enhances overall smoothness and significantly reduces aberrations, thereby increasing the consistency of the entire model.

Experiments

An aggressive smoothing policy removes sharp boundaries, while potentially important details are lost. On the other hand, gentle smoothing preserves the details but leaves behind artificial boundaries between the LR and HR models. It is essential to achieve a trade-off between the two cases, and this is exactly the aim of our PGM method. To quantify this trade-off between presenting detailed information and minimizing artifacts we use the travel time between the stations and their residuals at 36 synthetic sensors (red X, 10 on each edge, see Fig. 1d on the border between the LR and HR areas. These residuals are then used to evaluate how much detailed information is preserved in the fused velocity model, compared to the HR maps.

Checkerboard Model

We used a 2D square checkerboard model with 100x100 pixels, each with 10 small squares on each edge, and each small square measures 10x10 pixels in size. The pattern on the board alternates circular high- and low-velocity pixels in each small square.

Ridgecrest Fault Zone Model

To demonstrate the efficacy of the proposed PGM model, we compare its performance with commonly used conventional methods (e.g. Gaussian filter and cosine-taper window) on both the synthetic checkerboard model and the real-data Ridgecrest model. We have selected the high-resolution model of the Ridgecrest, CA, region, obtained by ambient noise tomography, to test the efficacy of our proposed PGM. The Ridgecrest fault zone image consists of a shallow (representing a depth of approximately 0.5 km) high-resolution Rayleigh wave model (Zhou et al., 2022), from which the S-wave velocity is roughly approximated by dividing by 0.9. This model reveals a 3D flower-shaped low-velocity zone surrounding the M7.1 and M6.4 earthquakes that ruptured in the 2019 Ridgecrest sequence. Yeh and Olsen (2023) showed that including the fault zone model into the SCEC CVM-S V4.26M01 significantly improves the fit of simulations to strong motion data from the M7.1 Ridgecrest earthquake, including at stations more than 200 km away in Los Angeles. The improvement in the fit to data was caused by more accurately generated Love waves at the boundaries of the low-velocity zone around the faults, as compared to the low-resolution model without the fault zone model. Motivated by the results by Yeh and Olsen (2023), who used the cosine-window taper fusion method by Ajala and Persaud (2021), we compare the efficacy of our proposed PGM with other existing methods.

Optimal Parameter Selection

The number of clusters in GMM clustering has a significant impact on the results. Generally, the number of clusters can influence the complexity of the model and the interpretability of the results. More clusters result in a more complex model, which can better capture intricate data structures and lead to more detailed insights into the data. However, it also increases the risk of overfitting.

Selecting the optimal number of clusters is crucial in GMM and other clustering techniques. Several methods can help determine an appropriate number of clusters, such as the Bayesian Information Criterion (BIC), and the silhouette score (Neath and Cavanaugh, 2012). These methods balance the trade-off between the goodness of fit of the model and the complexity of the model. In this experiment, we tested and compared the cluster number sequences $N = 3, 5, 6, 7, 9$, which are commonly used in practical applications of MRFs. Figure 6 (a) shows that the larger the number of clusters, the more detailed information is preserved in the HR models, and the larger the computation is required, implying a trade-off between computational cost and performance. Figure 6(b) demonstrates the number of clusters versus RMSE (root mean square error), which is defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_t}(t_i - \hat{t}_i)}{N_t}}, \tag{13}$$

where N_t is the total number of ray paths, and t_i and $\hat{t}_i$ are the posterior and prior travel-times (left vertical axis). The right vertical axis shows the computation time for the Ridgecrest test. The RMSE decreases when the cluster number grows from 3 to 6, with a limited decrease for larger clusters, and there is a rapid growth in run time when the cluster number exceeds 6. For these reasons, we choose the cluster number as 6.

2D Performance

We demonstrate the fusion for the checkerboard and Ridgecrest velocity models described above, in both cases with directly superimposed HR and LR components, in Fig. 7 (a1, b1), both with an HR region in the center, surrounded by LR velocities. The dimensions of the models are summarized in Table 2.

Figure 7 (a2, b2) shows the checkerboard and Ridgecrest model station settings along with the ray-path density. For the checkerboard model, the stations are evenly distributed, whereas the stations for the Ridgecrest model are highly irregular, reflecting the pattern used in Zhou et al. (2022). Fig. 7 (a3, b3) shows the label mask maps generated in the final iteration of the PGM models. In these maps, pixels sharing the same label suggest that the corresponding areas may exhibit comparable velocity patterns, implying that they are likely sampled from a similar distribution. The smoothed fusion results with the 5×5 Gaussian smoothing filter (GF), dictionary learning (DL) (Yang et al., 2012), and our proposed probability graphical model (PGM) are shown in Fig. 7 (a4-a6) and (b4-b6). The outcomes suggest that the learning-based methods, e.g. DL and PGM, demonstrate a superior capacity to retain detailed information in comparison to the application of direct Gaussian smoothing. This enhanced performance is attributed to the adaptive nature of these learning methods in determining the optimal fusion parameters for accurate data representation. Conversely, Gaussian smoothing employs a fixed kernel to blend neighboring pixels, which does not allow for such adaptive optimization and may lead to a less detailed final output.

We evaluate the efficacy of our model fusion with multiple metrics: travel time Root-Mean-Squared-Error (RMSE, which measures the travel time misfits (Bianco et al., 2019), Naturalness Image Quality Evaluator, NIQE, a common-used measurement for image quality (Mittal et al., 2012), Peak Signal-to-Noise Ratio, PSNR, measuring the sharpness of images (Poobathy and Chezian, 2014), and the Fréchet inception distance, FID, capturing similarities between the original and fused models (Chong and Forsyth, 2020)) in Table 3. In the checkerboard test, due to the simplicity of the pattern and the uniform distribution of stations, all learning methods exhibit a comparable performance. In contrast, for the more complex and realistic Ridgecrest model, the PGMs outperform the DL model, as the latter is sensitive to the orientation of the patches while the graphical models are rotationally invariant.

Geological formations are often anisotropic, meaning their properties vary depending on the direction in which they are measured, e.g., laterally continuous and vertically stratified. Standard Markov Random Field (MRF) schemes, which assume homogeneous properties (same properties in all directions), can lead to errors when applied to such formations. PGMs, on the other hand, consider the anisotropic nature of geological formations, generally leading to more accurate results. Seismic inversion is an ill-posed problem, meaning it doesn't have a unique solution, and small changes in the input can lead to large changes in the output. Regularization is a technique used to stabilize the solution. Our PGM provides an edge-preserving regularization based on the information from neighboring pixels, which is effective for reconstructing subsurface models.

3D Ridgecrest Model Fusion Comparison

To assess the proficiency of the PGM fusion approach, we have expanded our methodological framework to the integration of 3D models. Analogous to the 2D fusion experiments, the S-wave velocity model was extracted from the top 5 km around the 2019 Ridgecrest, CA, earthquake sequence, from the SCEC CVM-S4.26, serving as the LR model, while the 3D S-wave velocity model derived from surface wave dispersion inversion by Zhou et al. (2022) represents the HR model. Both LR and HR models were interpolated into 100×100 (pixels) horizontal models for each specified depth and resampled to a depth resolution of 250 meters.

In Fig. 8, the LR CVM-S4.26 model centered on the Ridgecrest domain is shown in panel (a), while panel (b) depicts the model obtained by directly incorporating the HR model (from surface wave dispersion inversion, see Zhou et al., 2022) into the LR matrix. Employing the cosine-taper smoothing technique (with the three-dimensional window size set to $(108, 120, 21)$

and cosine fractions configured as $(0.75, 0.75, 0.9)$, representing the dimensions of south-north, west-east, and depth correspondingly) and 3D dictionary learning (with a kernel size of $7 \times 7 \times 5$) as benchmark methodologies, the resultant fusion models via benchmark methods, and PGM are shown in Fig. 8 panels (c)-(e), respectively. The cosine taper functions exclusively in the overlapping regions of HR and LR data. When there is a significant mismatch in the boundary areas, the cosine-taper smoothing function may not fully correct misaligned patterns (Fig. 8(c)). However, machine-learning-based methods (including dictionary learning and PGM) are adept at both overlapping regions and areas with only LR information. This capability enhances their effectiveness in successfully aligning unmatched patterns from both sides. It is also notable that the 3D PGM fusion methodology appears to retain enhanced details from the HR models compared to the 3D dictionary learning procedure. To quantify the performance in 3D, synthetic stations were placed between the LR and HR models at each depth level (similar to Fig. 1d in the 2D case), where the travel times were computed before and after the fusion methodologies were applied on the HR and LR directly-superimposed model. Using the travel time preceding the fusion as a reference, we calculated the root-mean-squared errors (RMSE) corresponding to the post-fusion travel time misfit. The calculated RMSEs of travel time misfit for cosine taper smoothing, 3D dictionary learning fusion, and our 3D PGM for depths from the surface to 5 km are listed in Table 4. As for the 2D fusion case, the results derived from machine-learning-based dictionary learning and PGM surpass those obtained through cosine taper smoothing. Notably, our 3D PGM approach yields the most significant improvement, achieving a 44% reduction in travel-time misfit relative to conventional cosine-taper methods. This substantial decrease indicates a minimal distortion of information from the HR model, underscoring the efficacy of the 3D PGM method in preserving data integrity.

Six dense sensor arrays were deployed across the faults that ruptured in the 2019 Ridgecrest earthquake sequence (see Fig. 9, left panel, A1, A2, B1 through B4). Owing to these densely distributed arrays, we computed surface wave dispersion inversion profiles for station pairs and subsequently aggregated them to derive HR 2D vertical S-wave velocity models (Zhou et al., 2022), as illustrated in Fig. 9 (top right). These derived models are compared with vertical cross-sections extracted from 3D models and combined with the LR background model (SCEC CVM-4.26) through various fusion methodologies. For instance, the B2 and B4 array panels (Fig. 9c, d) depict the 2D cross-sections extracted from the 3D dictionary learning fusion model and the 3D Probabilistic Graphical Model (PGM), respectively. The superior performance of our 3D PGM approach is evident in its ability to more precisely define and preserve the accuracy of the boundary of the low-velocity zone. This improved accuracy can be attributed to the PGM's strategy of assigning differential weights to edges, which are oriented in various directions. In contrast, the efficacy of 3D dictionary learning is somewhat limited due to its inherent rotational invariance and the constraints imposed by a fixed patch dimension.

Conclusions

We present a method for combining multiresolution seismic velocity maps using probabilistic graphical models (PGMs). The performance of our PGM algorithm is assessed through experiments, using both a checkerboard model and a complex fault zone model around the 2019 Ridgecrest earthquake sequence. The evaluation of the checkerboard model, which is characterized by its inherent simplicity and uniform station distribution, demonstrates that our PGM approach outperforms all tested established baseline techniques. The machine-learning-based methods employed for map synthesis, such as our PGM, distinctly outperform traditional methods, primarily due to adaptive parameter learning. In the context of the Ridgecrest model, the PGM technique produced a 44% reduction in the computed travel time residuals versus the conventional Gaussian smoothing methods in 3D exploration models. This is due to the limitations of traditional methods in addressing anisotropic patterns, in contrast to the PGM which learns weights consistent with the complex structure of the Ridgecrest model. In summary, our PGM fusion approach effectively minimizes the undesired sharp discontinuities often observed between LR and HR models, while simultaneously preserving detailed information inherent in the HR models. A prospective area of investigation in future work involves addressing the challenge of irregular model resolution within the HR domain, which is crucial for enhancing the fidelity and applicability of our models and potentially improves the understanding and application in various real-world models. Finally, we recommend that the efficacy of the PGM be tested directly through a comparison of synthetic and observed waveforms.

Data and Resources

The seismic and station data used in this research were obtained from the FDSN 3J:RAMP deployment of 3C nodal, collected after the July Searles Valley 2019 Earthquake (DOI: 10.7914/SN/3J_2019). Additionally, the low-resolution velocity model was accessed via the SCEC CVM version S4.26-M01 (DOI:10.1002/2014JB011346). The code supporting this study is openly available on GitHub https://github.com/zhz039/PGMfuionvm.

Declaration of Competing Interests

The authors acknowledge that there are no conflicts of interest recorded.

Acknowledgments

This research was supported by the Statewide California Earthquake Center (SCEC) based on Award DE-SC0016520 from the U. S. Department of Energy, with SCEC contribution number 13337.

References

Ajala, R. and P. Persaud (2021). Effect of merging multiscale models on seismic wavefield predictions near the southern San Andreas fault, *Journal of Geophysical Research: Solid*

Earth **126**(10), e2021JB021915.

Ajala, R. and P. Persaud (2022). Ground-Motion Evaluation of Hybrid Seismic Velocity Models, *The Seismic Record* **2**(3), 186–196.

Ajala, R., P. Persaud, and A. Juarez (2022). Earth model-space exploration in Southern California: Influence of topography, geotechnical layer, and attenuation on wavefield accuracy, *Frontiers in Earth Science* **10**, 964806.

Bianco, M. J. and P. Gerstoft (2018). Travel time tomography with adaptive dictionaries, *IEEE Transactions on Computational Imaging* **4**(4), 499–511.

Bianco, M. J., P. Gerstoft, K. B. Olsen, and F.-C. Lin (2019). High-resolution seismic tomography of Long Beach, CA using machine learning, *Scientific reports* **9**(1), 1–11.

Blake, A., P. Kohli, and C. Rother (2011). *Markov random fields for vision and image processing*. MIT press.

Carlo, C. M. (2004). Markov chain monte carlo and gibbs sampling, *Lecture notes for EEB* **581**(540), 3.

Cheung, G., E. Magli, Y. Tanaka, and M. K. Ng (2018). Graph spectral image processing, *Proceedings of the IEEE* **106**(5), 907–930.

Chong, M. J. and D. Forsyth (2020). Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6070–6079.

Das, P. and A. Ortega (2020). Graph-based skeleton data compression. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6. IEEE.

Dhamo, H., A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht (2020). Semantic image manipulation using scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5213–5222.

Dudik, M., S. J. Phillips, and R. E. Schapire (2004). Performance guarantees for regularized maximum entropy density estimation. In *International Conference on Computational Learning Theory*, pp. 472–486. Springer.

Edwards, B. and D. Fäh (2013). A stochastic ground-motion model for Switzerland, *Bulletin of the Seismological Society of America* **103**(1), 78–98.

Geman, S. and C. Graffigne (1986). Markov random field image models and their applications to computer vision. In *Proceedings of the international congress of mathematicians*, Volume 1, pp. 2. Berkeley, CA.

Ghosh, S. (2018). *Kernel smoothing: Principles, methods and applications*. John Wiley and Sons.

Koller, D. and N. Friedman (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Li, S. Z. (2012). *Markov random field modeling in computer vision*. Springer Science and Business Media.

McLachlan, G. J. and T. Krishnan (2007). *The EM algorithm and extensions*. John Wiley and Sons.

Melas, D. E. and S. P. Wilson (2002). Double Markov random fields and Bayesian image segmentation, *IEEE Transactions on Signal Processing* **50**(2), 357–365.

Mittal, A., A. K. Moorthy, and A. C. Bovik (2012). No-reference image quality assessment in the spatial domain, *IEEE Transactions on image processing* **21**(12), 4695–4708.

Mu, H.-Q. and K.-V. Yuen (2016). Ground motion prediction equation development by heterogeneous Bayesian learning, *Computer-Aided Civil and Infrastructure Engineering* **31**(10), 761–776.

Murphy, K. (2001). An introduction to graphical models, *Rap. tech* **96**, 1–19.

Neath, A. A. and J. E. Cavanaugh (2012). The Bayesian information criterion: background, derivation, and applications, *Wiley Interdisciplinary Reviews: Computational Statistics* **4**(2), 199–203.

Ortega, A., P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst (2018). Graph signal processing: Overview, challenges, and applications, *Proceedings of the IEEE* **106**(5), 808–828.

Poobathy, D. and R. M. Chezian (2014). Edge detection operators: Peak signal to noise ratio based comparison, *IJ Image, Graphics and Signal Processing* **10**, 55–61.

Pulli, K., A. Baksheev, K. Kornyakov, and V. Eruhimov (2012). Real-time computer vision with OpenCV, *Communications of the ACM* **55**(6), 61–69.

Sambridge, M. and K. Mosegaard (2002). Monte Carlo methods in geophysical inverse problems, *Reviews of Geophysics* **40**(3), 3–1.

Shuman, D. I., S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, *IEEE signal processing magazine* **30**(3), 83–98.

Van Houtte, P., A. K. Kanjarla, A. Van Bael, M. Seefeldt, and L. Delannay (2006). Multiscale modelling of the plastic anisotropy and deformation texture of polycrystalline materials, *European Journal of Mechanics-A/Solids* **25**(4), 634–648.

Yang, J., Z. Wang, Z. Lin, S. Cohen, and T. Huang (2012). Coupled dictionary training for image super-resolution, *IEEE transactions on image processing* **21**(8), 3467–3478.

Yeh, T.-Y. and K. Olsen (2023). Fault damage zone effects on ground motions during the 2019 Mw 7.1 Ridgecrest, CA, earthquake, *Bull. Seismol. Soc. Am.* **113(4)**, 1724–1738.

Zhang, H. and Y. Ben-Zion (2023). Enhancing regional seismic velocity model with higher-resolution local results using sparse dictionary learning.

Zhang, S., Q. Deng, and Z. Ding (2022). Multilayer graph spectral analysis for hyperspectral images, *EURASIP Journal on Advances in Signal Processing* **2022**(1), 1–25.

Zhang, X., A. Lomas, M. Zhou, Y. Zheng, and A. Curtis (2023). 3-D Bayesian variational full waveform inversion, *Geophysical Journal International* **234**(1), 546–561.

Zhang, Z., Y. Zhang, Z. Zhou, and J. Luo (2018). Boundary-based image forgery detection by fast shallow cnn. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2658–2663. IEEE.

Zhao, X., A. Curtis, and X. Zhang (2022). Bayesian seismic tomography using normalizing flows, *Geophysical Journal International* **228**(1), 213–239.

Zhou, Z., M. Bianco, P. Gerstoft, and K. Olsen (2022). High-Resolution Imaging of Complex Shallow Fault Zones Along the July 2019 Ridgecrest Ruptures, *Geophysical Research Letters* **49**(1), e2021GL095024.

Zhou, Z., P. Gerstoft, and K. Olsen (2023). *Authorea Preprints*, Fusion of Multiresolution Seismic Tomography Maps Using Physics-informed Probability Graphical Models, doi:http://doi.org/10.22541/essoar.169841574.40243932/v1.

Full mailing address for each author

Corresponding author: Zheng Zhou zhz039@ucsd.edU Scripps Institution of Oceanography, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093. Peter Gerstoft, pgerstoft@ucsd.edu Scripps Institution of Oceanography, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093. Kim Bak Olsen: kbolsen@sdsu.edu San Diego State University, 5500 Campanile Dr, San Diego, CA 92182.

List of Figure Captions

Figure 1. (a) Excerpt of S-wave velocities from the SCEC CVM version S-4.26 (hereafter referred to as the low-resolution (LR) CVM) at 0.5 km depth around the Ridgecrest area. (b) High-resolution (HR) S-wave map from 1 Hz Rayleigh wave tomography from Zhou et al. (2022). (c) Direct superposition of the HR and LR models. (d) Synthetic stations ('X') are located on the boundaries between the HR and LR models for evaluation. These two models share some patterns in the low-velocity zones, but show many mismatched detailed patterns where the two models overlap, which results in sharp and misaligned boundaries in those areas. Our PGM is applied to the mismatched boundary areas between the two bounding boxes in (c). The pixels in this area belong to the effective vertices set.

Figure 2. A 6-cluster Gaussian Mixture model clustering is applied on the continuous velocity map $\boldsymbol{A}$ in (a). Each pixel is clustered to be a 6-cluster discrete label map $\boldsymbol{X}$ in (b). Pixels with similar velocity information have been assigned the same label.

Figure 3. (a) In a 2D graphical model, each pixel has a continuous velocity $A_{i,j}$ and a discrete label mask $X_{i,j}$. The objective function has two parts: (1) the data cost θ_0 (0th-order neighboring potential) that forces a pixel to have a $A_{i,j}$ specified by the Gaussian distribution of its label $X_{i,j}$, and (2) the smoothness cost θ_1 (1st-order neighboring potential) that promotes smoothness among neighboring labels (Koller and Friedman, 2009). (b) For a 3D graphical model, the framework closely aligns with 2D (a). The 3D model encompasses 6 adjacent nodes (while in 2D, it has 4 nodes).

Figure 4. (a) In a 2D graphical model, the neighborhood system $\mathcal{N}_{i,j}$ (marked in gray/black) of the given center node (i,j) (marked in black). The 1st- and 2nd- neighborhood systems of node (i,j), marked with numbers 1 and 2, is represented as $\mathcal{N}^1_{i,j}$ and $\mathcal{N}^2_{i,j}$. Number 0 denotes the center pixel.

(b) 3D graphical model similar to (a), where 0, 1, and 2 denote the center pixel and the 1st-, and 2nd- neighborhood systems, respectively.

Pipeline of the iterative update policy for both pixel labels (discrete class labels) and pixel values (continuous velocity values).

Figure 6. (a) Comparison of the fusion results using cluster numbers n= 3, 5, 6, 7, and 9. (left) fused velocity model, and (right) cluster distribution. (b) The number of clusters versus RMSE error (left vertical axis, corresponding to the solid line) and run time (right vertical axis, corresponding to the dashed line) for the Ridgecrest model. Generally, the larger the cluster number, the smaller the RMSE error with a longer run time.

Figure 7. (top 2 rows) Checkerboard and (bottom 2 rows) Ridgecrest models. (1) Superimposed HR and LR models. (2) Station location and ray density. (3) 6-class label mask maps for HR models (pixels with the same label are learned together). (4) Smoothing results with a 5×5 Gaussian filter (GF). (5) and (6) Fusion results with dictionary learning (DL) and with our PGM, respectively.

Figure 8. (a) The LR model extracted from CVM-S4.26 around the Ridgecrest area. (b) Direct superposition of the 3D HR surface wave dispersion inversion model and CVM LR models for the Ridgecrest area. (c-e) Combined LR and HR models, smoothed by (c) cosine-taper function, (d) dictionary learning, and (e) PGM. Figure 9. (a) Station locations (triangles) and main faults (lines)

surrounding the Ridgecrest area. There are six dense sensor arrays across the main faults (A1-2 and B1-4). (b) Vertical cross-sections of the shear wave velocity along the B1-4 station arrays from (top) surface wave dispersion inversion, (center) the 3D fusion model from dictionary learning, and (bottom) the PGM.

Tables

Table 1: Algorithm for 3D multiresolution velocity model fusion.

Algorithm 1 MCMC Method for MRF
1. Input: $\boldsymbol{A}_{LR}$ and $\boldsymbol{A}_{HR}$
2. Initialize velocity $\boldsymbol{A}$ by superimposing $\boldsymbol{A}_{HR}$ over $\boldsymbol{A}_{LR}$ Initialize $\boldsymbol{X}$, $\mu_{\mathbf{n}}$ and $\sigma_{\mathbf{n}}$ with GMM clustering
3. **for** each EM iteration **do**
4. Construct PGM
5. **for** $t = 1$ to max iteration T **do**
6. (E-Step) Gibbs Sampling
7. **for** pixel $(i,j,k) = (1,1,1)$ to the max index (I,J,K) **do**
8. $X_{i,j,k}^{(t+1)} \sim P\Big(X_{i,j,k} \,\vert\, X_{1,1,1}^{(t+1)}, \ldots, X_{i,j,k-1}^{(t+1)}, A_{i,j,k}^{(t)}, X_{i,j,k+1}^{(t)}, \ldots, X_{I,J,K}^{(t)}\Big)$
9. **end for**
10. **for** pixel $(i,j,k) = (1,1,1)$ to the max index (I,J,K) **do**
11. $A_{i,j,k} \sim \sum_{n=1}^{6} P(X_{i,j,k} = n) N(\mu_n, \sigma_n^2)$
12. **end for**
13. (M-Step) Update Gaussian parameters μ_n and σ_n^2 with sample means and variances of $\boldsymbol{A}^{(t+1)}$.
14. **end for**
15. **end for**
15. **return** $\boldsymbol{X}, \boldsymbol{A}$ (for each pixel)

Table 2: Model Coverage Range and Dimensions.

	LR		**HR**		**Fused**	
	Range (km)	Dimension	Range (km)	Dimension	Range (km)	Dimension
2D Checkerboard	100×100	40×40	40×40	40×40	100×100	100×100
2D Ridgecrest	100×100	50×50	58×64	192×224	100×100	330×350
3D Ridgecrest	$100 \times 100 \times 5$	$50 \times 50 \times 11$	$54 \times 60 \times 5$	$108 \times 120 \times 21$	$100 \times 100 \times 5$	$200 \times 200 \times 21$

* Range indicates the physical coverage of the models, and Dimension denotes the number of pixels used for computation. ‘LR’, ‘HR’, and ‘Fused’ denote the low-resolution, high-resolution, and fused models, respectively.

Table 3: 2D Evaluation Results.

	RMSE/s↓	**NIQE↓**	**PSNR/dB↑**	**FID↓**
Checkerboard GF	1.65	7.68	14.58	45.75
” DL	1.18	5.44	15.70	33.85
” PGM	1.14	5.40	16.14	32.49
Ridgecrest GF	3.52	12.41	21.80	61.39
” DL	2.61	7.29	22.36	54.25
” PGM	2.27	6.70	23.04	47.49

* Evaluation metrics are root-mean-square error (RMSE) of the travel time misfit (with unit s), naturalness image quality evaluator (NIQE), peak signal-to-noise ratio (PSNR), and Fréchet inception distance (FID). ↓ indicates smaller is better, and ↑ opposite.

Table 4: 3D Evaluation Results.

Depth	**Cosine Taper**	**DL**	**PGM**
0 km	1.67	1.57	0.86
1 km	1.52	1.39	0.73
2 km	1.73	1.38	1.04
3 km	1.58	1.53	0.96
4 km	1.57	1.62	1.13
5 km	1.79	1.43	1.04

* Evaluation metric is the root-mean-square error (RMSE) of the travel time misfits (with unit s).

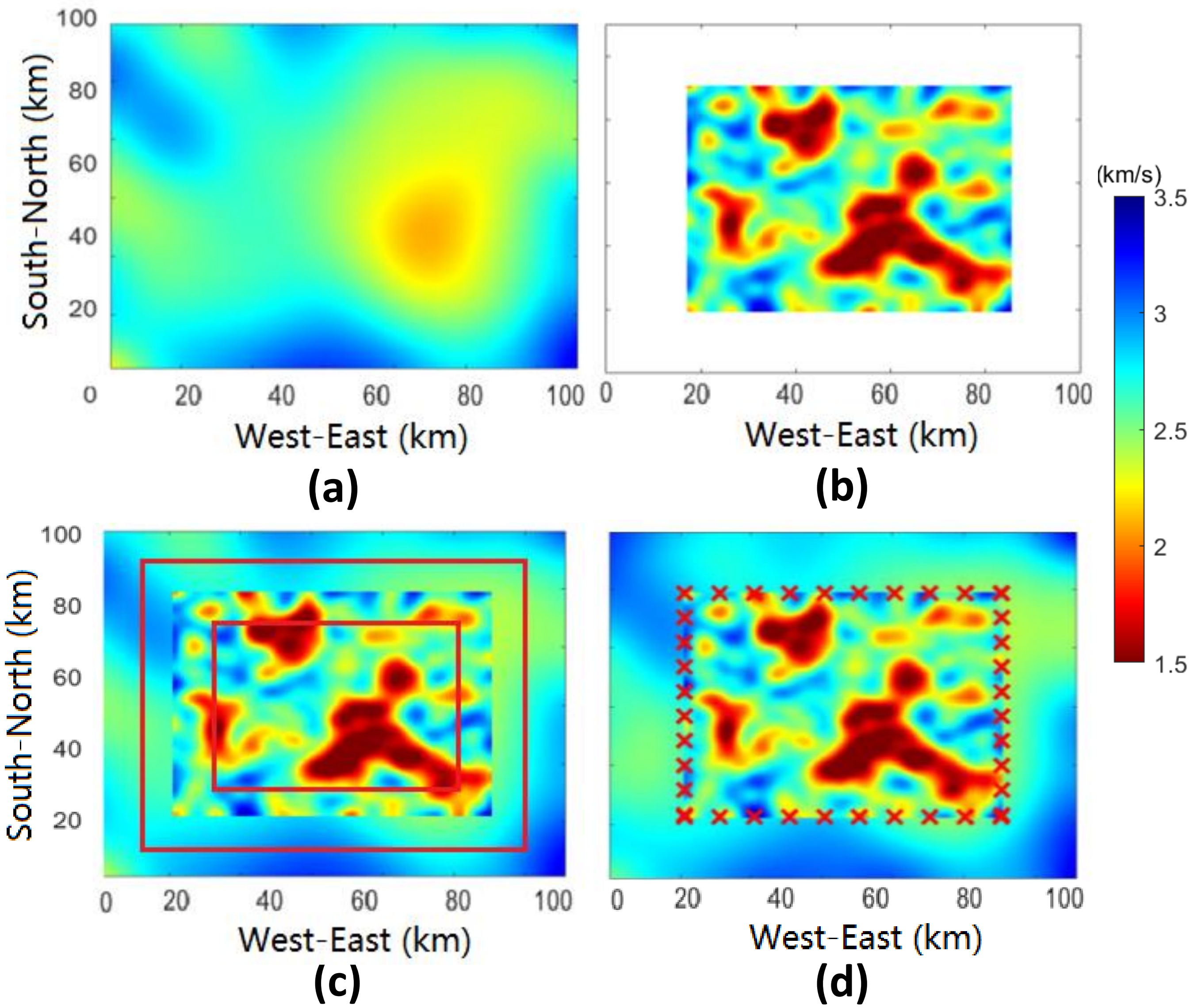


Figure 1: (a) Excerpt of S-wave velocities from the SCEC CVM version S-4.26 (hereafter referred to as the low-resolution (LR) CVM) at 0.5 km depth around the Ridgecrest area. (b) High-resolution (HR) S-wave map from 1 Hz Rayleigh wave tomography from Zhou et al. (2022). (c) Direct superposition of the HR and LR models. (d) Synthetic stations ('X') are located on the boundaries between the HR and LR models for evaluation. These two models share some patterns in the low-velocity zones, but show many mismatched detailed patterns where the two models overlap, which results in sharp and misaligned boundaries in those areas. Our PGM is applied to the mismatched boundary areas between the two bounding boxes in (c). The pixels in this area belong to the effective vertices set.

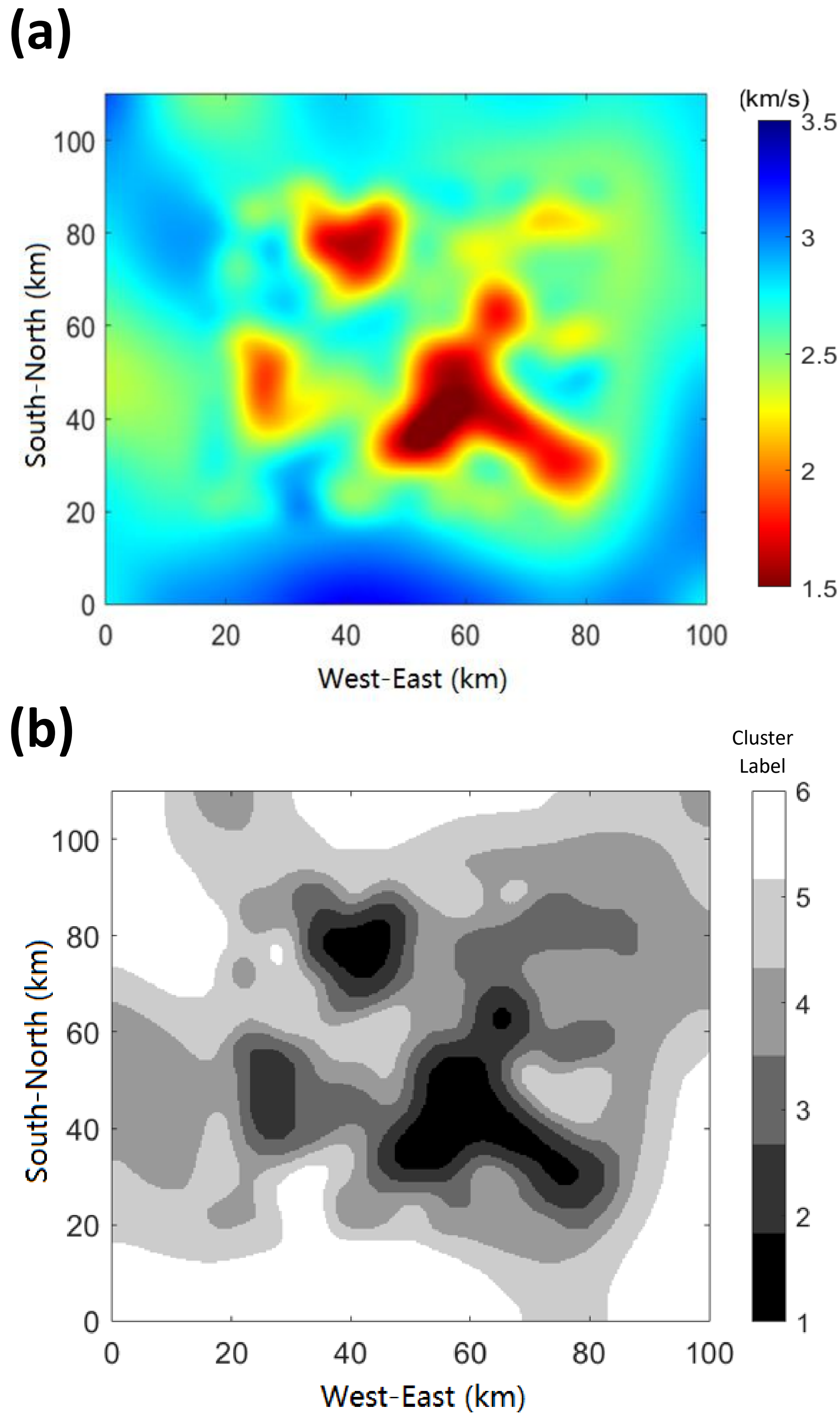


Figure 2: A 6-cluster Gaussian Mixture model clustering is applied on the continuous velocity map $\boldsymbol{A}$ in (a). Each pixel is clustered to be a 6-cluster discrete label map $\boldsymbol{X}$ in (b). Pixels with similar velocity information have been assigned the same label.

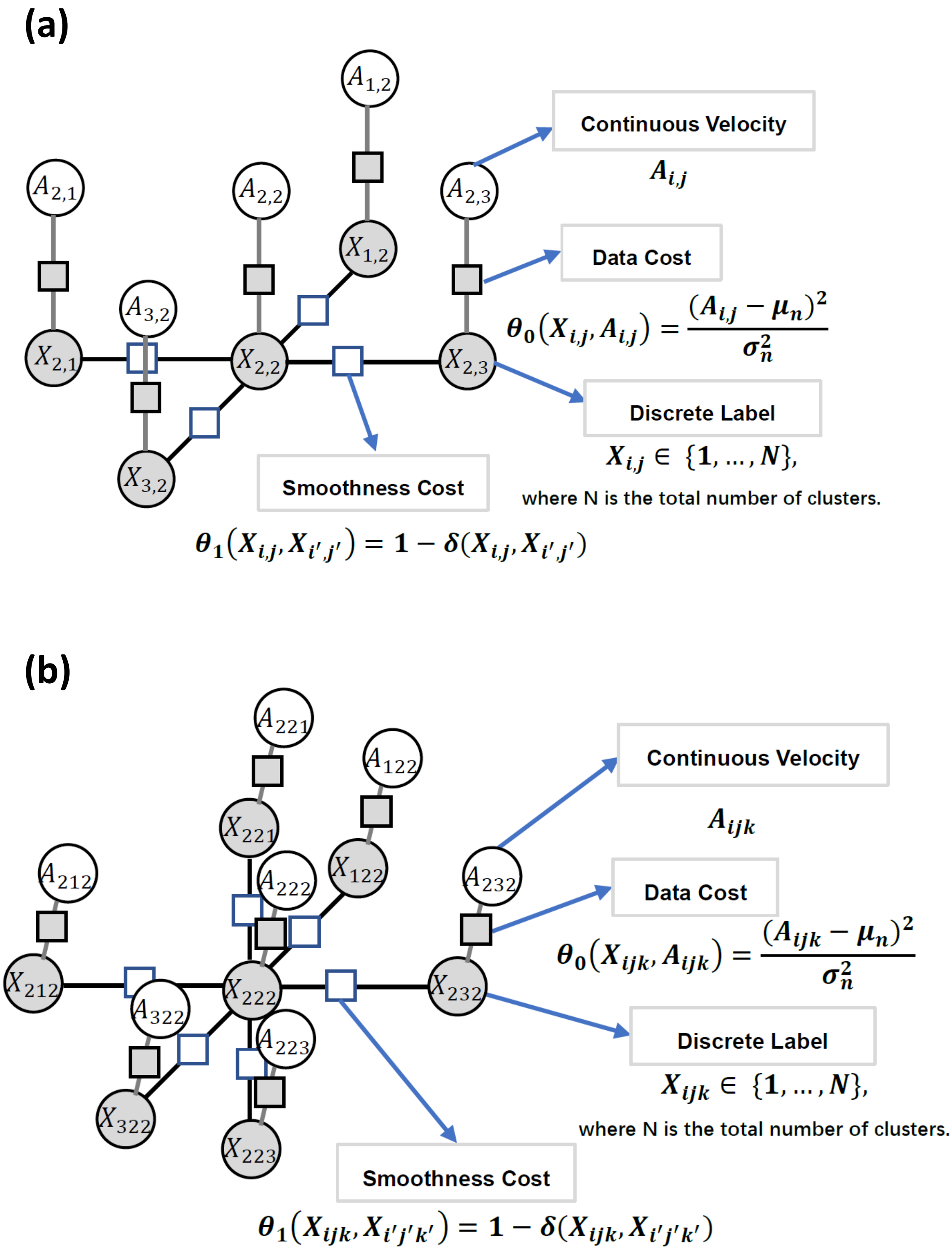
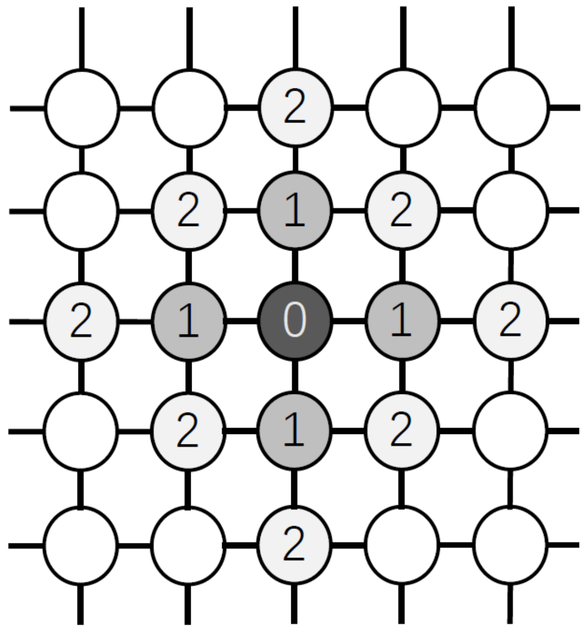


Figure 3: (a) In a 2D graphical model, each pixel has a continuous velocity $A_{i,j}$ and a discrete label mask $X_{i,j}$. The objective function has two parts: (1) the data cost θ_0 (0th-order neighboring potential) that forces a pixel to have a $A_{i,j}$ specified by the Gaussian distribution of its label $X_{i,j}$, and (2) the smoothness cost θ_1 (1st-order neighboring potential) that promotes smoothness among neighboring labels (Koller and Friedman, 2009). (b) For a 3D graphical model, the framework closely aligns with 2D (a). The 3D model encompasses 6 adjacent nodes (while in 2D, it has 4 nodes).

(a)



(b)

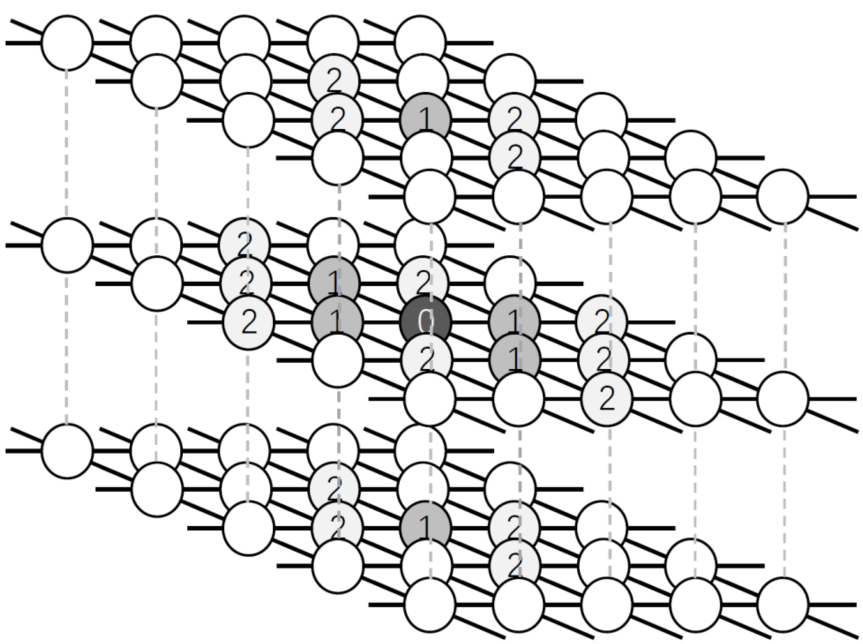


Figure 4: (a) In a 2D graphical model, the neighborhood system $\mathcal{N}_{i,j}$ (marked in gray/black) of the given center node (i, j) (marked in black). The 1st- and 2nd- neighborhood systems of node (i, j), marked with numbers 1 and 2, is represented as $\mathcal{N}_{i,j}^1$ and $\mathcal{N}_{i,j}^2$. Number 0 denotes the center pixel. (b) 3D graphical model similar to (a), where 0, 1, and 2 denote the center pixel and the 1st-, and 2nd- neighborhood systems, respectively.

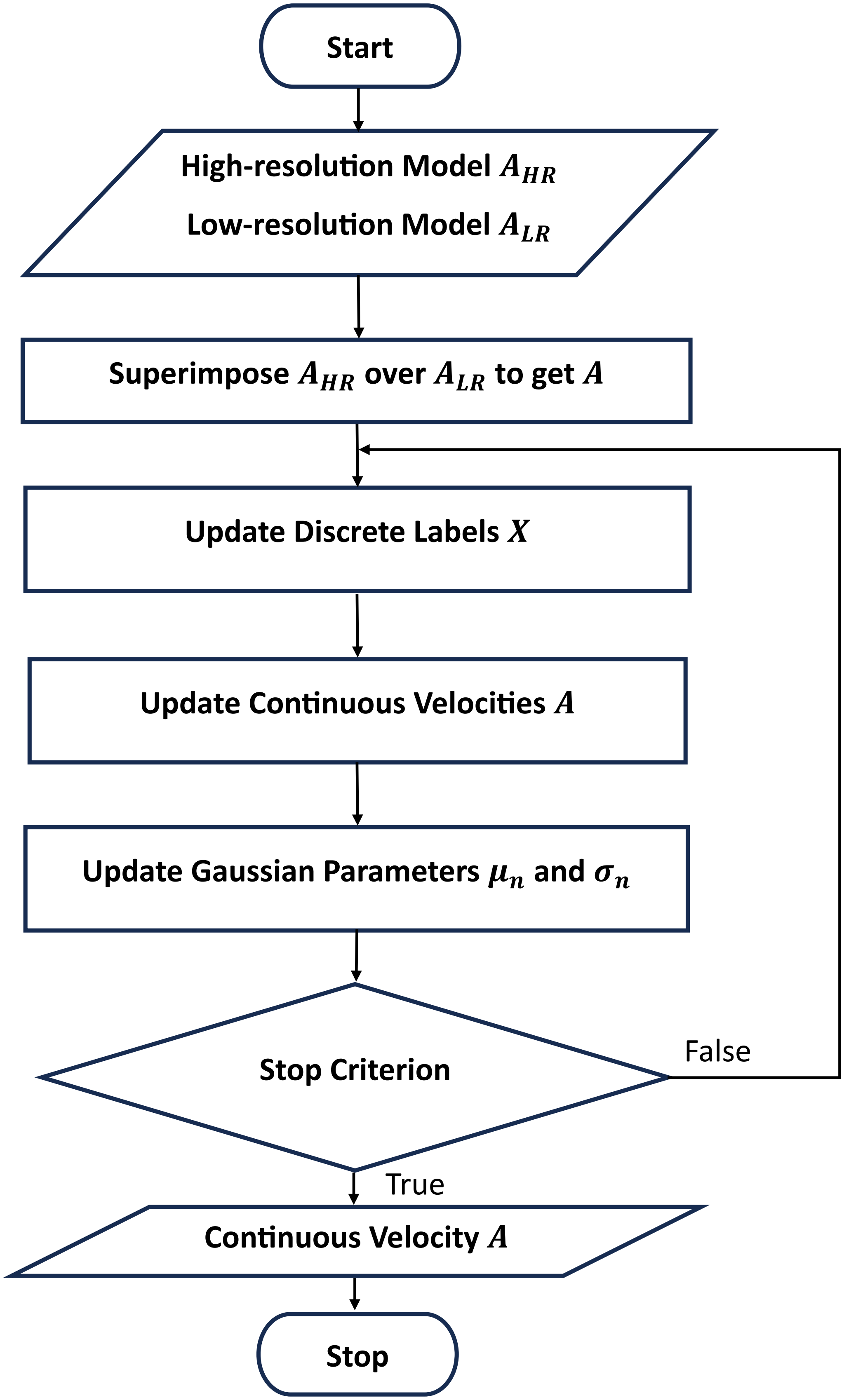


Figure 5: Pipeline of the iterative update policy for both pixel labels (discrete class labels) and pixel values (continuous velocity values).

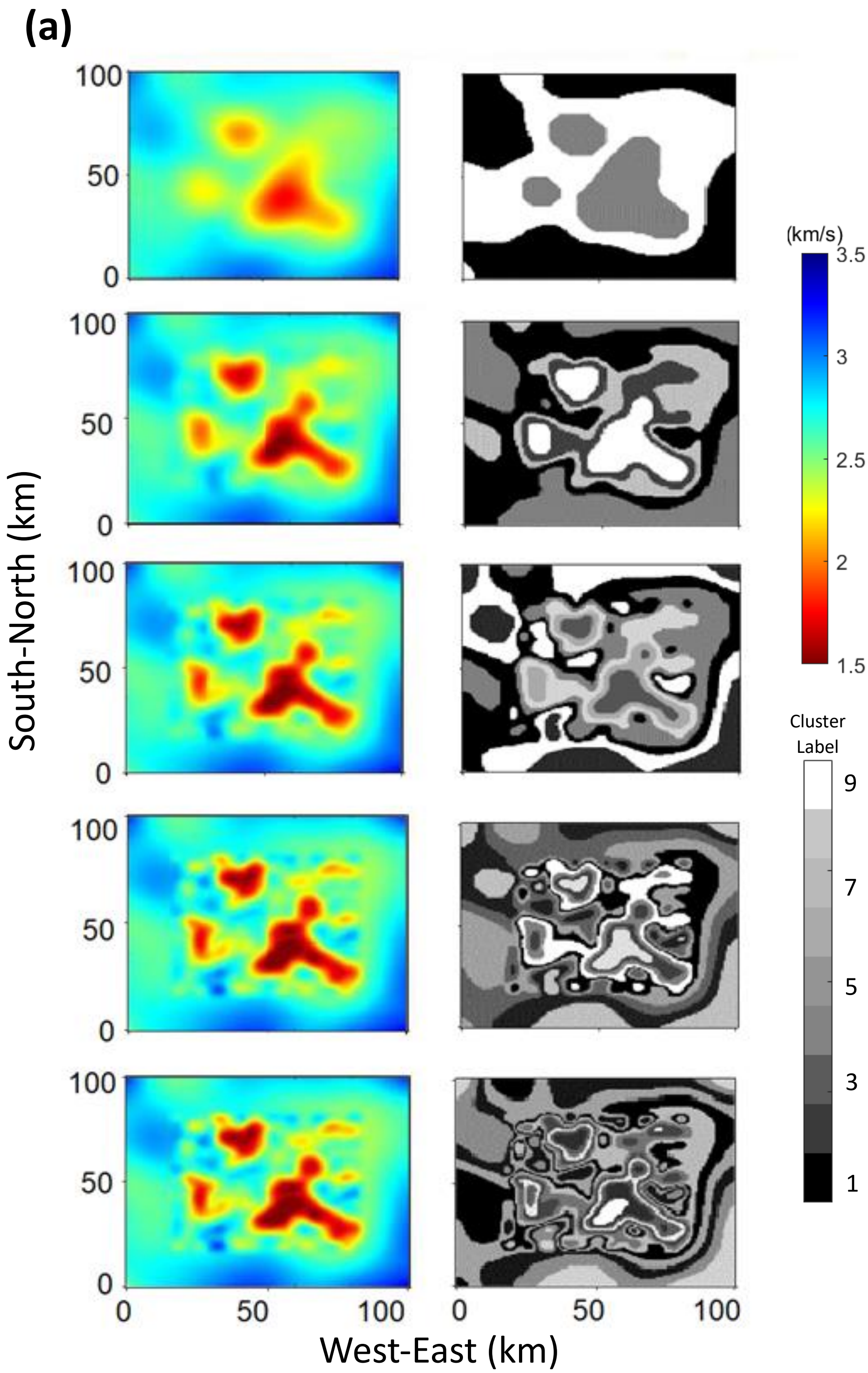


Figure 6: *(see next page)*

(b)

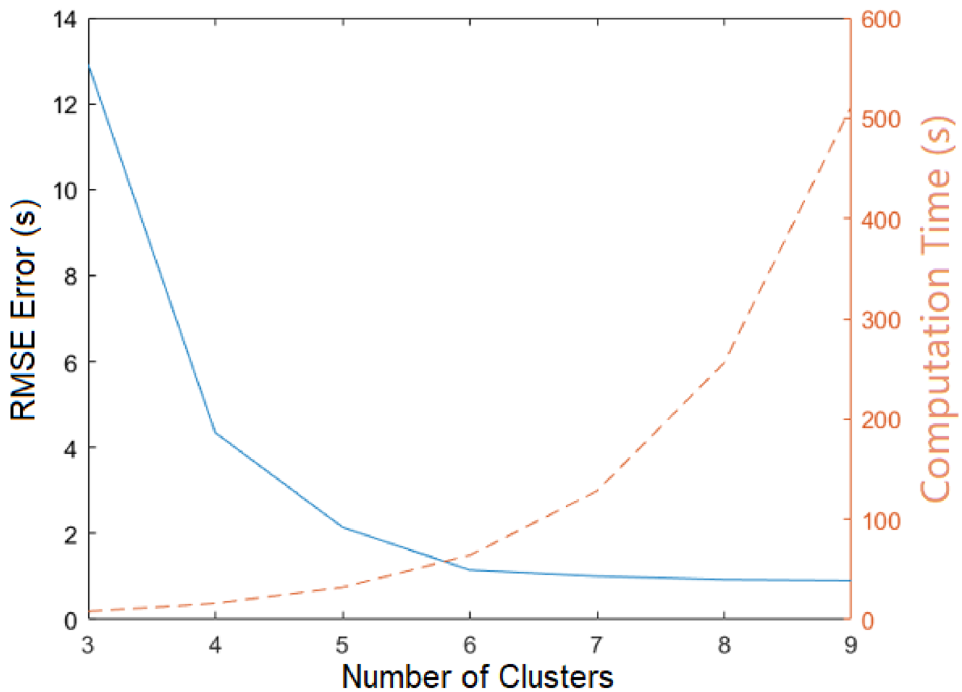


Figure 6: (a) Comparison of the fusion results using cluster numbers n= 3, 5, 6, 7, and 9. (left) fused velocity model, and (right) cluster distribution. (b) The number of clusters versus RMSE error (left vertical axis, corresponding to the solid line) and run time (right vertical axis, corresponding to the dashed line) for the Ridgecrest model. Generally, the larger the cluster number, the smaller the RMSE error with a longer run time.

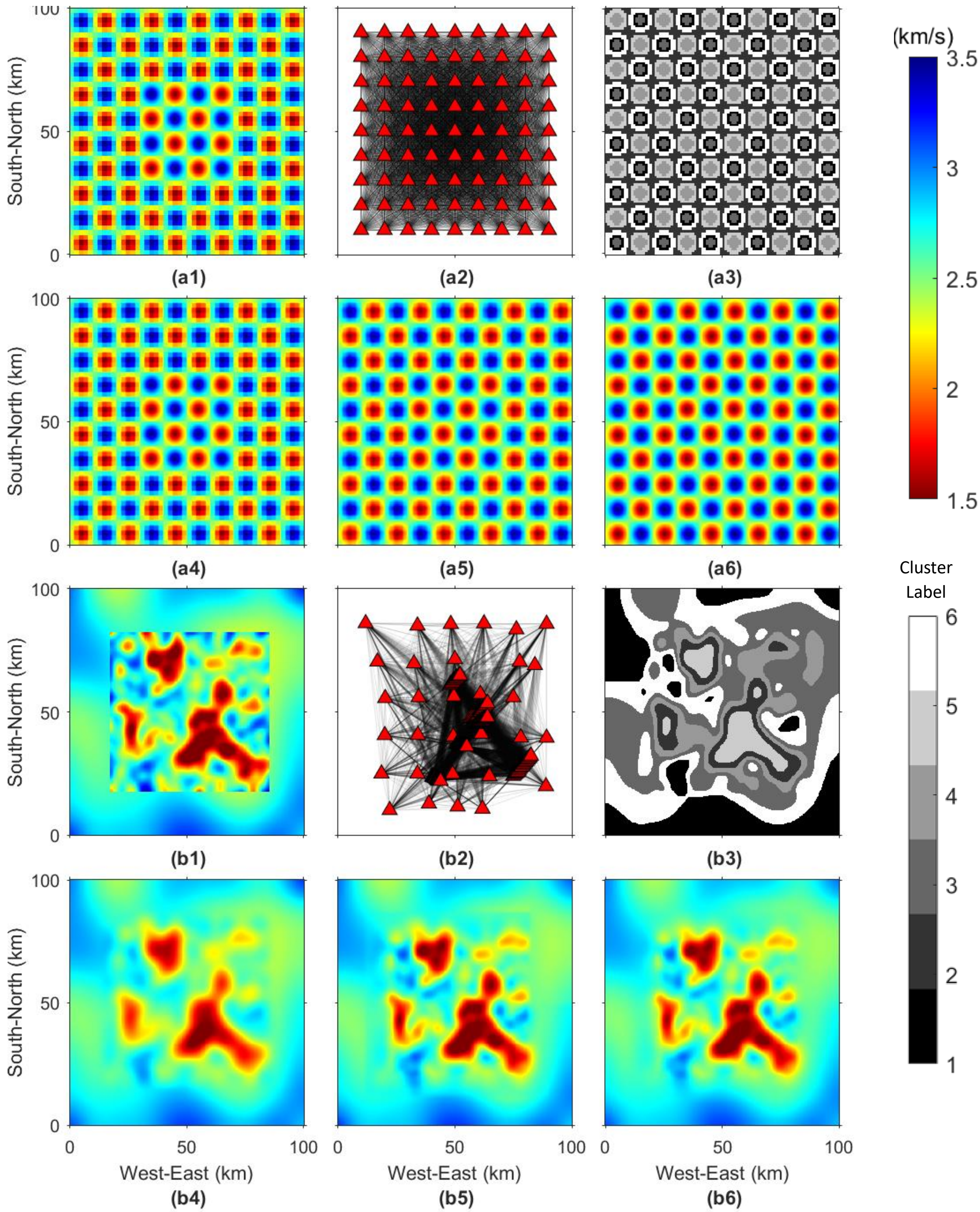


Figure 7: (top 2 rows) Checkerboard and (bottom 2 rows) Ridgecrest models. (1) Superimposed HR and LR models. (2) Station location and ray density. (3) 6-class label mask maps for HR models (pixels with the same label are learned together). (4) Smoothing results with a 5×5 Gaussian filter (GF). (5) and (6) Fusion results with dictionary learning (DL) and with our PGM, respectively.

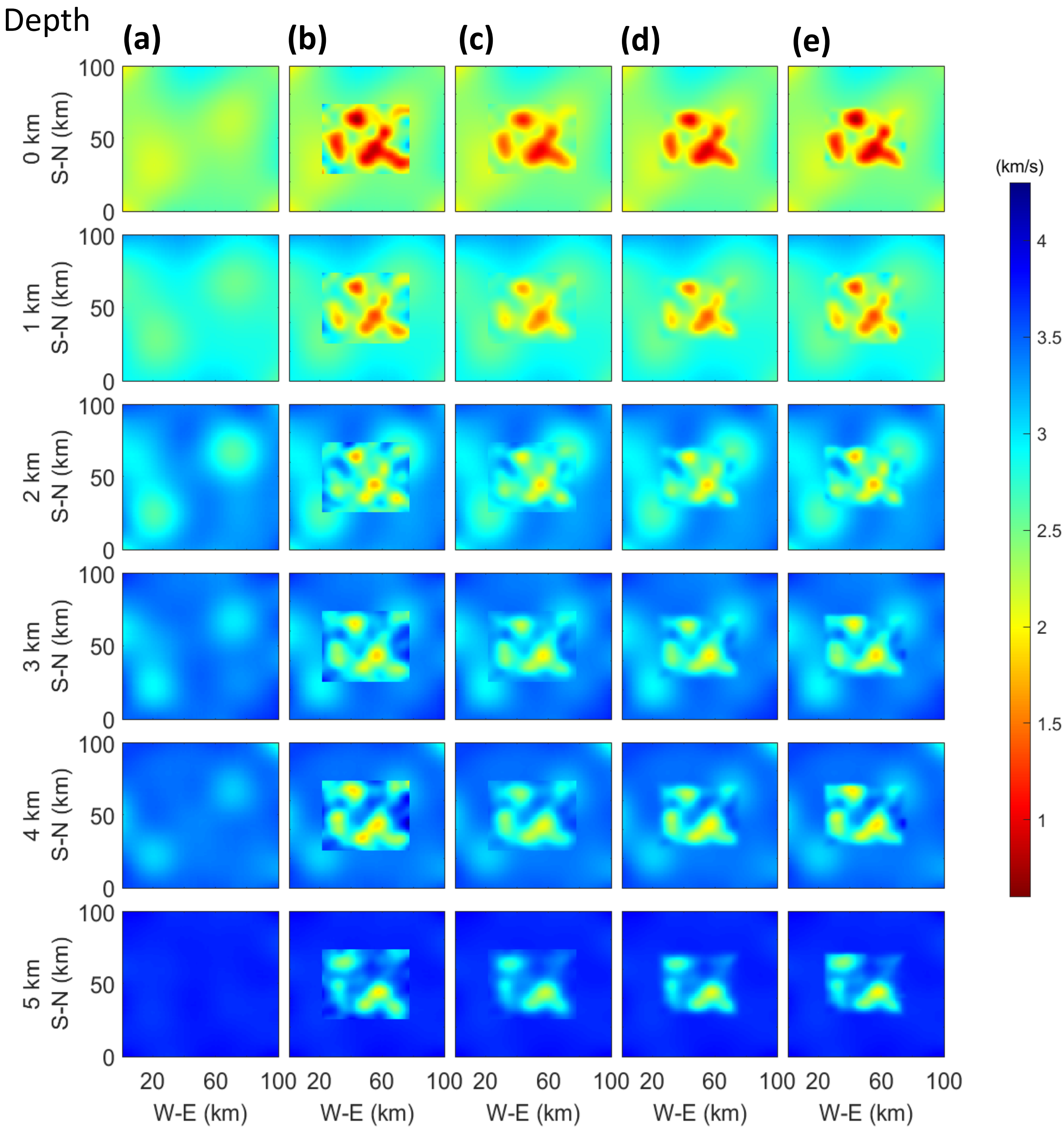


Figure 8: (a) The LR model extracted from CVM-S4.26 around the Ridgecrest area. (b) Direct superposition of the 3D HR surface wave dispersion inversion model and CVM LR models for the Ridgecrest area. (c-e) Combined LR and HR models, smoothed by (c) cosine-taper function, (d) dictionary learning, and (e) PGM.

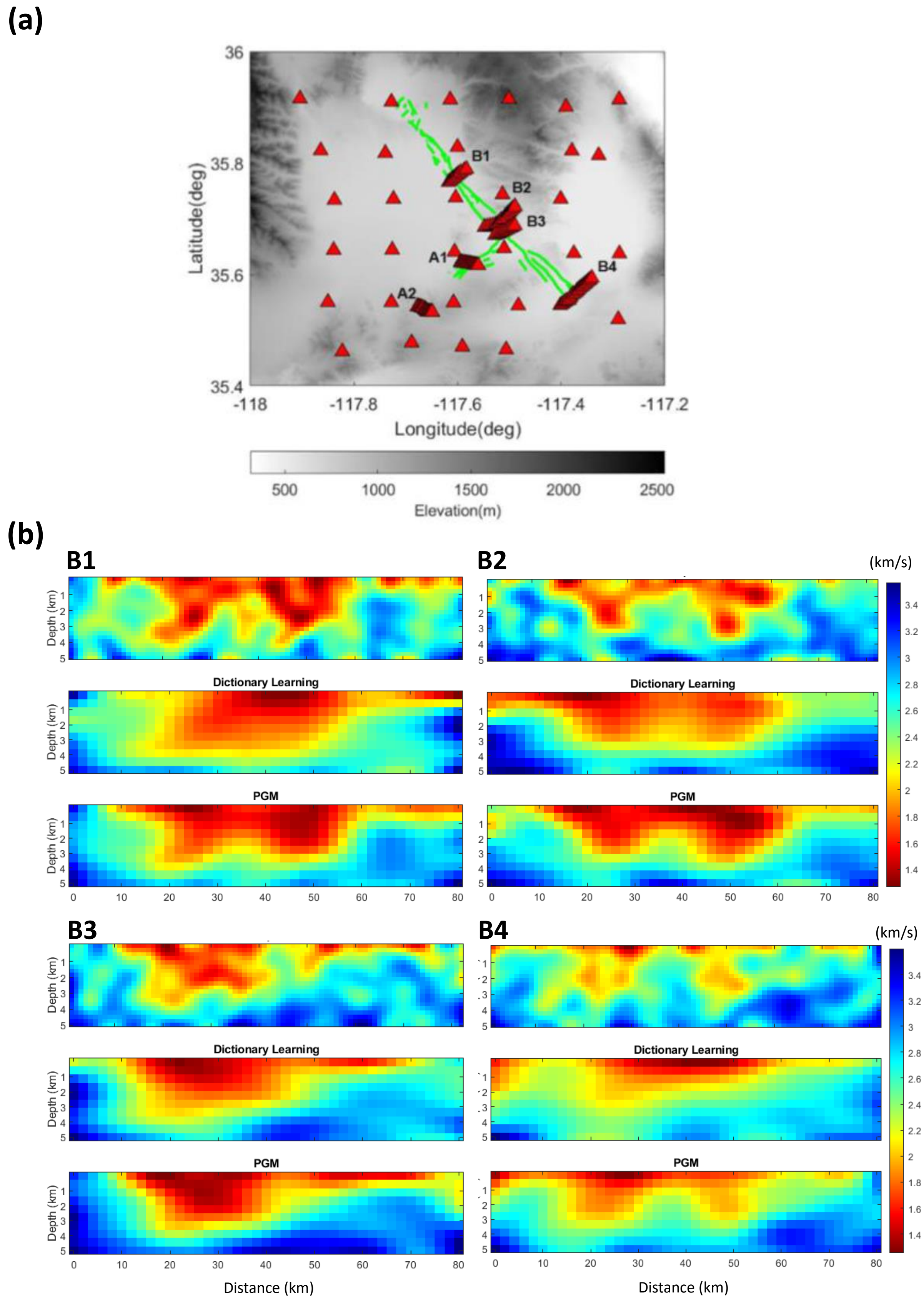


Figure 9: (a) Station locations (triangles) and main faults (lines) surrounding the Ridgecrest area. There are six dense sensor arrays across the main faults (A1-2 and B1-4). (b) Vertical cross-sections of the shear wave velocity along the B1-4 station arrays from (top) surface wave dispersion inversion, (center) the 3D fusion model from dictionary learning, and (bottom) the PGM.