Submission code: R97BP54KZL

See this and other abstracts on our website: https://home.sbgf.org.br/Pages/resumos.php

Velocity downward continuation through generative models

Tariq Alkhalifah (KAUST), Randy Harsuko (KAUST), Shijun Cheng (KAUST)

SBGf Conference 18-20 NOV | Rio'25

Velocity downward continuation through generative models

Copyright 2025, SBGf - Sociedade Brasileira de Geofísica / Society of Exploration Geophysicist.

This paper was prepared for presentation during the 19th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 18-20 November 2025. Contents of this paper were reviewed by the Technical Committee of the 19th International Congress of the Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of the Brazilian Geophysical Society is prohibited.

Abstract Summary

Velocity model building (VMB) should comply with the physics involved in wave propagation and honor the location of the recording surface in which we extract the information. Thus, we propose using a generative model to predict velocity from shallow to deep. The shallow distribution act as priors to predict the deep, and in our implementation, with using the seismic image and the well information as guide. An application on offshore data from North west Australia demonstrated the versatility of this approach in predicinting an accurate velocity model. With the generative nature of the process, we can also quantify the uncertainty, which was well in agreement with what we expected. We will share more examples in the presentation of this work.

Introduction

Seismic data are recorded on the Earth's surface, yet the goal of seismic imaging is to resolve structures deep within the subsurface, requiring accurate velocity model building (VMB) to account for the complex travel paths of seismic waves (Yilmaz, 2001). A common strategy for velocity estimation is the top-down approach (layer stripping), where the model is constructed layer by layer from shallow to deep, guided by surface seismic data, well information, and stacking velocities (Biondi, 2006). A key tool in this process is ”downward continuation”, which extrapolates recorded wavefields from the surface to deeper levels using an estimated velocity model, in order to reduce the effects of velocity heterogeneity on imaging and improve reflector positioning (Claerbout, 1985). Thus, the natural progression of VMB should have a top-to-down succession. Nevertheless, most machine learning models for estimating the subsurface using surface recorded data ignore this fundamental fact (Taufik et al., 2024; Wang et al., 2024), as they borrow their models from image processing applications in computer vision that do not have this physical reality in its inverse formulation.

Recent advances in generative modeling have demonstrated the potential to learn and sample from complex distributions, making these tools invaluable for seismic applications, such as regularizing full waveform inversion (FWI, e.g., Taufik et al., 2024). Generative models, like Generative Adversarial Networks (GANs) and diffusion models, have emerged as state-of-the-art techniques, leveraging convolutional neural networks (CNNs) to generate high-quality samples (e.g., Wang et al., 2024). Despite their success, these methods often overlook the natural progression of velocity model building from surface recorded seismic data from shallow to deep. This oversight can limit the practical utility of these models in seismic workflows that inherently depend on spatial hierarchies and acquisition geometries. Thus, we need a generative framework that align with the spatially hierarchical nature of seismic velocity information.

Inspired by the success of Generative Pre-trained Transformer (GPT)-based models in natural language processing, VelocityGPT (Harsuko et al., 2024) was introduced as a generative framework tailored to seismic velocity models. By adopting an autoregressive approach, VelocityGPT predicts deeper subsurface layers conditioned on previously generated layers, mimicking the progression of traditional model-building practices. The original VelocityGPT demonstrated promising results, including the ability to incorporate velocity model classes during training, enabling conditional sampling for diverse geological scenarios (Harsuko et al., 2024).The other approach is a Diffusion model trained to estimate the velocity for a certain depth using the known velocity of the shallower depth as a condition. We will share applications based on both model in the top-down VMB process.

Theory

VelocityGPT was introduced as a two-stage framework: 1). VQ-VAE training to encode velocity models into discrete latent codes and 2). GPT training on latent codes to model the underlying distribution. The VQ-VAE training employs the following objective function (Van Den Oord et al., 2017):

$$\mathcal{L}_{VQ-VAE} = \log p(x|z_d(x)) + ||sg[z_e(x)] - e||_2^2 + \beta||z_e(x) - sg[e]||_2^2, \tag{1}$$

where x represents the input image, z_e is the encoder, z_d is the decoder, e denotes the discrete embedding space, and sg refers to the stop gradient operator, which acts as an identity function during the forward pass and zeros out gradients during the backward pass. Each term in the objective serves a specific purpose: the reconstruction loss (first term) optimizes the encoder and decoder to accurately reconstruct the input; the embedding loss (second term) adjusts the embedding dictionary; and the commitment loss (third term), ensures consistency between the encoder output and the embedding space. One of the caveats of using a multi-term objective function is the balancing of each term.

VelocityGPT promotes adding of latent codes of an image at the corresponding location of the latent codes of the velocity model as a condition to impose the structural information. The patch-based generation of VelocityGPT allows for a lower training cost and freedom to extend to any velocity model size in the inference. However, for the latter, the network might lose the spatial context when the model size is quite large, and thereby, the generated samples might deteriorate in quality and continuity. Therefore, to preserve spatial coherence and maintain depth-wise continuity, we add a global positional encoding at the latent code level, which imposes information on the depth location in the actual domain after converting patches into latent codes. Specifically, consider a velocity model $M \in R^{h \times w}$. We patchify the velocity model into overlapping patches of size $p \times p$ with a stride s, which are then converted into latent codes. The depth of each latent code is denoted as d_i, where d_i corresponds to the relative depth of the latent code in the transformed latent code domain.

Data and training

We aim to train the VelocityGPT on synthetic velocity models and apply it to the target field data; thus, we design a workflow that ensures that the synthetic velocity models are representative of the subsurface structures in the field. The models are guided by well information, providing critical constraints on the velocity distributions and layering patterns. Initially, we generate 1D compressional wave velocity (V_p) profiles using velocity value ranges guided from well field measurements. These profiles are then laterally extended to form 2D layered velocity models, establishing a baseline structure. To introduce geological complexity, we apply random elastic transforms that simulate realistic subsurface features such as folding and intrusions, creating models that reflect the heterogeneity of the Earth. We created 2,048 samples of 2D V_p models using this approach. This choice of RTM velocity was used to demonstrate the versatility of the framework.

We trained two VQ-VAE networks: one responsible for encoding the velocity models and the other for the corresponding images. The two networks share a similar architecture, which is composed of two-layered encoder and two-layered decoder with 32 filters for each layer. Regarding the codebook size, we use 240 for both the velocities and the images. These VQ-VAEs are trained using an Adam optimizer with a batch size (bs) of 1024 and a learning rate (lr) of 4e-3 on a single NVIDIA A100 GPU each. Subsequently, we trained a GPT network to model the conditional distribution of the velocity models using well, depth position, and RTM images as the conditions. The depth position allows the network to focus on the distribution of velocity values and structures within a depth range, as velocity often increases with depth. The GPT network is configured as follows: 16 layers, 64 hidden dimensions, and 8 attention heads. We use an Adam optimizer with a bs = 128 and a lr = 1e-3 to train the network.

Results

Figure 1, top left, shows the top part of an FWI result of the field data (Figure 1, bottom left) used as input to the algorithm. However, the initial shallow segment could be derived from other sources of information, and it also could be limited to the water layer, if needed, as the well will help guide the velocity. Like in the training, we perform RTM using a smooth version of the velocity derived from the well (i.e., homogeneous laterally) for conditional generation purposes, taking into account that the training images were derived in a similar fashion. Figure 1, bottom row shows the generated samples with a well as an additional condition. This extra information provides constraints of the layering as well as the velocity variation with depth, which is reflected in the generated samples that appear closer to the FWI result, yet higher resolution thanks to the well information. We also achieve a high correlation between the generated samples and the FWI result, even at locations away from the well. The variance between the generated samples can provide a measure of uncertainty of the introduced process. Figure 2 compares the FWI results with the mean of the generated models using the same image and well condition (100 of them). The standard deviation computed from these models provides a map of uncertainty which increases away from the well and as we go deep in the model, which is expected.

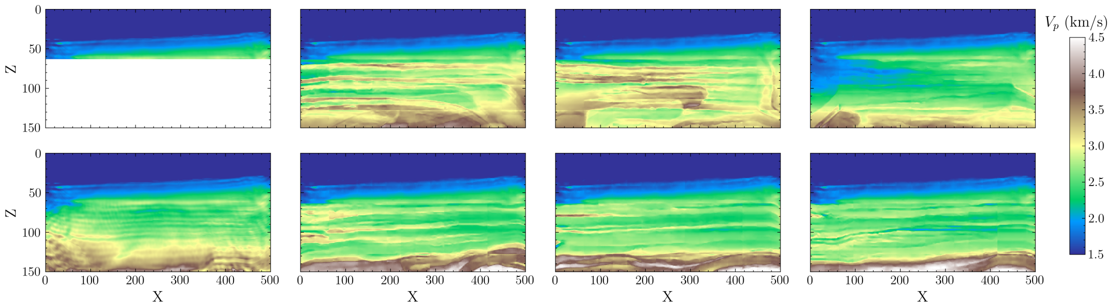

Figure 1: Input to the algorithm (top left), which is the top part of an FWI result (bottom left). Conditional generation results using: RTM and position (top row) and RTM, position, and well (bottom row).

Conclusions

VelocityGPT advances generative sampling for velocity model building by integrating RTM, global positional encoding, and well constraints to produce realistic and spatially coherent models. A simpler

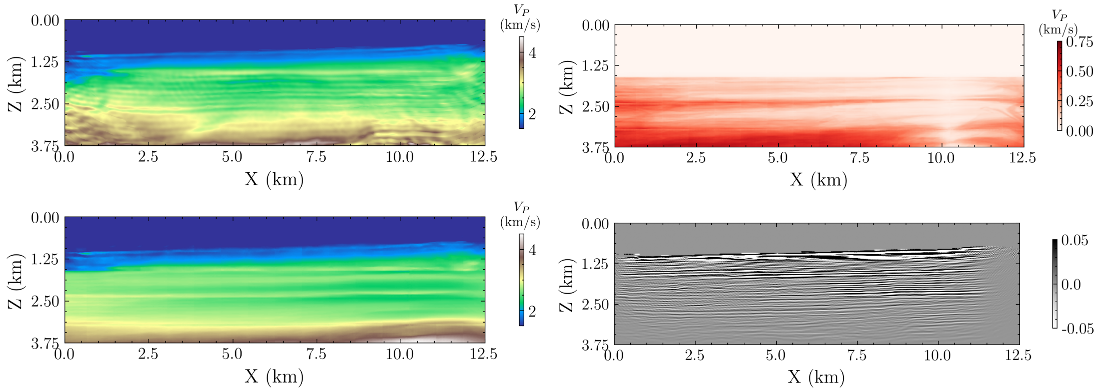

Figure 2: An FWI obtained velocity model (top left). The mean of 100 realization from the VelocityGPT with well and image condition (bottom left). The standard deviation correpsonding to the 100 realizations (top eight). The used RTM image (bottom right).

quantizer and the use of linear attention mechanism secured computational efficiency while maintaining quality, even for large models. The generated samples exhibit strong agreement with FWI results, demonstrating their potential as robust priors for inversion workflows.

Acknowledgements

The authors thank KAUST and the DeepWave Consortium sponsors for supporting this research. We also thank SWAG for the helpful discussions. This work utilized the resources of the Supercomputing Laboratory at King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia.

References

Biondi, B., 2006, 3d seismic imaging: Society of Exploration Geophysicists.

Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific Publications.

Harsuko, R., S. Cheng, and T. Alkhalifah, 2024, Propagating the prior from shallow to deep with a pre-trained velocity-model generative transformer network: arXiv preprint arXiv:2408.09767.

Taufik, M. H., F. Wang, and T. Alkhalifah, 2024, Learned regularizations for multi-parameter elastic full waveform inversion using diffusion models: Journal of Geophysical Research: Machine Learning and Computation, **1**, e2024JH000125.

Van Den Oord, A., O. Vinyals, et al., 2017, Neural discrete representation learning: Advances in neural information processing systems, **30**.

Wang, F., X. Huang, and T. Alkhalifah, 2024, Controllable seismic velocity synthesis using generative diffusion models: Journal of Geophysical Research: Machine Learning and Computation, **1**.

Yilmaz, , 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: Society of Exploration Geophysicists.