

Uso de Atributos Sísmicos em Técnicas de SSL para a Segmentação Semântica de Fáceis Sísmicas

Gabriel A. T. Mendes

Carlos. A. Astudillo

Relatório Técnico - IC-PFG-25-48

Projeto Final de Graduação

2025 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Uso de Atributos Sísmicos em Técnicas de SSL para a Segmentação Semântica de Fáceis Sísmicas

Gabriel A. T. Mendes

Carlos. A. Astudillo*

Resumo

As limitações inerentes ao processo manual de interpretação sísmica demandam especialistas altamente treinados e frequentemente produzem segmentações com vieses e inconsistências. Aprendizado profundo é utilizado para lidar com esses problemas, automatizando esse processo. No entanto, o treinamento desses modelos tipicamente precisa de uma quantidade grande de dados rotulados, que, no contexto de dados sísmicos, essa rotulagem tem um altíssimo custo. Assim, a falta de dados rotulados ainda é um obstáculo significativo para a utilização de métodos de aprendizado de máquina supervisionado para processamento de dados sísmicos.

Técnicas de *self-supervised learning* (SSL) têm surgido como uma ferramenta para lidar com a escassez de dados rotulados, porém a maioria das técnicas SSL foi projetada no contexto de imagem natural ou em domínios específicos como dados médicos. Por outro lado, os atributos sísmicos desempenham um papel essencial na caracterização de padrões geológicos como falhas, horizontes e fácies sísmicas, mas não se exploram ainda em técnica SSL.

Este trabalho discute como técnicas de SSL podem ser empregadas para mitigar a escassez de rótulos e como o uso de diferentes atributos sísmicos impacta o desempenho desses métodos em tarefas de segmentação semântica de dados sísmicos. À luz de resultados anteriores envolvendo segmentação baseada em atributos, detecção de falhas e métodos não supervisionados, conjecturamos que a integração sistemática entre atributos sísmicos e pré-treinos auto-supervisionados pode oferecer avanços relevantes em diversas aplicações de interpretação sísmica, ao produzir representações capazes de generalizar melhor para regiões não vistas pelo modelo e de lidar com classes minoritárias.

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas

Agradecimentos

Início esta seção agradecendo à minha mãe, Elisa Maria Angélica T. Mendes, e ao meu pai, Felisberto A. O. dos Santos, por todos os anos que passamos juntos, pelo carinho e amor incondicional e por sempre me transmitirem o sentimento de que eu tinha um lugar seguro no mundo para estar. Agradeço aos meus irmãos, Raphael Felix T. Mendes, Ariane Nataly T. Mendes e Leticia Ashli T. Mendes, por fazerem parte da minha formação e por estarem ao meu lado nos bons e maus momentos. Agradeço ao meu cunhado, Guilherme Ronda, e à sua família por todos os anos vividos ao lado da minha irmã e da nossa família. E agradeço a Deus por todas essas pessoas.

Agradeço ao meu orientador, Carlos A. Astudillo, pelas horas em que conversamos até tarde sobre projetos, pela empatia e por sempre se mostrar interessado e disponível para a minha pesquisa e progresso acadêmico. Agradeço também aos professores Edson Borin, Lucas F. Wanner, Isaías B. Felzmann e demais docentes do Instituto de Computação por todos os ensinamentos, conselhos e oportunidades oferecidos ao longo da graduação.

Agradeço aos meus professores do ensino médio, Clovis, Edna, Fernando, Jamal, Fátima e Bel Moretto, pelas aulas, risadas e apoio durante os primeiros anos da minha formação. Agradeço a Sergio Floriano e Lizete P. Floriano por sempre acreditarem em mim e pelas oportunidades dadas através do Rotary Club, e agradeço a Kelly Dussán pela oportunidade do PIBIC Jr.

Agradeço aos amigos e colegas que fiz durante o ensino médio pelo tempo que passamos juntos. Em especial, agradeço a Francisco B. Albaricci pelas conversas durante as madrugadas, à Rebeca Vetuche por todos os conselhos, à Thainá N. Okada por ter sido uma grande fonte de admiração, ao Vinicius Aguiar pelo apoio ao longo dos anos e também a Luís F. Ulian, Lucas Mazzoni, Giovanne Palladino, Leonardo Cardoso, Maria Julia, Rafael Khatib, Leticia Bussola, Vitor Paiva e João S. Dantas por fazerem parte dessa jornada.

Agradeço aos membros da equipe Connected Earth Museum, Luc J. Ribas, Lucas N. Roberto, Pedro C. Cintra, Matheus Lozano e Riccardo C. Sofer, pelas conversas e risadas na faculdade e por me convidarem a participar deste incrível projeto, que nos garantiu o prêmio NASA Space Apps 2024 na categoria Art & Technology, e aos colaboradores Marco A. Linhares e Cristiane M. O. de Camargo pelo apoio à nossa viagem.

Agradeço aos membros do laboratório Discovery pelos últimos dois anos de pesquisa e, em especial, a Vinicius Leme Soares pela amizade e pela viagem que fizemos juntos, João A. M. Seródio pela grande disponibilidade em auxiliar nos problemas que apareciam, Gabriel B. Gutierrez pelos conselhos acadêmicos, Otávio Napoli pela grande positividade ao longo dos dias de trabalho, Fernando G. Marques pelos jogos em grupo, Filipe A. Sampaio pelas discussões sobre Machine Learning, Marcos Vinicius S. Freire pelas conversas e ensinamentos sobre sismica, Jorge L. B. Mesquita pela parceria e a Alonso M. P. Huachaca e Mauricio Cifuentes Ruiz pela amizade, ensinamentos culturais e pela visita ao aquário.

Agradeço a dois dos meus melhores amigos, João V. Barreira e Angelo R. P. Malaguti, pelos anos de amizade, pelas risadas, pelos jogos e pelos McDonald's após as aulas de eletrônica.

Agradeço às pessoas que conheci ao longo deste período de faculdade, nos corredores do IC e além. Agradeço à minha primeira namorada, Giovanna, pelo tempo que passamos juntos e pelos ensinamentos que levei comigo. Agradeço à Fernanda Sanches e ao Pedro Bordignon pelas noites de jogos, ao Antonio Junior pela positividade na vida, à Ana B. Mota pelo bom humor, à Thais A. S. Camacho, à Vitoria Dias e à Luana pelos dias de pizza e compiladores, à Cairê Otaldo pelos hackathons, ao Henrique Othavio pelas conversas em bares, ao Nicolas H. Silva e à sua namorada Maria Angélica pelas conversas durante o almoço, à Ana Margarida Borges pelo grande carisma durante os coffee breaks, ao Vitor Lago pelas recomendações culturais, ao Heitor Menezes pelo exemplo moral, à Anne pelas conversas, à Milena Furuta pela inspiração, ao Gabriel Eiji por ser um exemplo

a ser seguido, a Ana L. Holthausen, Larissa Fazolin, Enrico P. Fernandes, Luigi T., Pedro Gadêlha, Caio R. Coldebella, Luis, Carlos e Gabriel P. por estarem sempre presentes, e agradeço a Giovani Bianchini, Gabriel Vieira, Caio Rehder, Pedro Suguino e Rafael da Silva Lobato por serem meu primeiro grupo de amigos na faculdade.

Agradeço aos membros da Coimbralândia, Daniel C. Coimbra pela proximidade e pelas visões de mundo, à Francine Coelho por todas as conversas e reflexões, ao Augusto Zolet pelo bom humor e perfumes, ao Arthur Baia pelas resenhas e cafés, e ao Matheus H. Patrício pelos ensinamentos filosóficos e músicas compartilhadas.

Agradeço a Helena Inoue e a Vitor Alves por todas as aulas e recitais de piano, e a Luciano Berenhauser pelas aulas de inglês. Agradeço às pessoas que conheci na viagem ao Japão, em especial à Nicolly Marques, Jake Shuff, Chris Wang, Kye e Mavis pelos passeios. Agradeço às músicas de cantores que fizeram parte de toda essa jornada, como Aimer, Ed Sheeran, Beatles, Claude Debussy e vários outros.

Agradeço aos membros da CVLB, Guilherme Franco, Wellington Rufino e Henrique Almeida, por todas as dailys, conversas dentro e fora do grupo e pelos bons momentos vividos. Agradeço aos membros das equipes Siren, Recomendação, Integrações e Arquitetura da AR&CO e, em especial, a Marco Antonio, Jeffrey Maropo, Ton Carmargo, Cleiber Reis, Jorge Lemos, Vicente Agrizzi, Rodrigo Faillace H., Luzimere Junqueira, Thulio Gamarano, Henrique Fontenelle, Lucas Fontinele, Joselena Pires, Lucas Gulart, Joao V. Lima, Samuel Oliveira, Fernando Frizatti, Luan Souza, Pedro Vinicius e Pedro Policicchio por todas as horas que compartilhamos dentro e fora do trabalho. Agradeço a toda a equipe da Globalsys por confiarem em meu trabalho, e agradeço também a Tobias Viana, Zeonardo, Romário Arruda, Denis, Tai Costa, Diego Souza e demais membros pelo tempo que passamos na Original.io.

Agradeço a Samuel Goto e Alexandre Valdetaro pelas dicas de carreira, pela visão de vida, pelos direcionamentos e por tudo o que conseguiram transmitir para mim em nossas conversas. Agradeço aos gerentes que tive ao longo da carreira e, em especial, agradeço a Lucas Testa por sempre ser um líder compreensivo, a Lucas Coradini por demonstrar real interesse no meu crescimento, a Fernando Dias pelos conhecimentos compartilhados e, por fim, a Thiago Molino por sempre se mostrar um líder justo, empático e humano.

Agradeço à Petrobras pela oportunidade do projeto de pesquisa, ao Instituto de Computação por sua excelência e, por fim, à Unicamp, por ser o lugar que, além de computação, também me ensinou que a vida é sobre pessoas.

Sumário

1	Introdução	5
2	Objetivos e Escopo do Projeto	5
2.1	Objetivo Geral	6
2.2	Objetivos Específicos	6
3	Metodologia	6
3.1	Ferramental	6
3.1.1	Dataset	6
3.1.2	Atributos sísmicos	7
3.1.3	Backbone	7
3.1.4	Técnica de SSL	8
3.2	Estratégias de Pré-Treino e Downstream	8
4	Resultados	10
4.1	Pré-treino com BYOL	10
4.2	Incorporação dos atributos sísmicos	12
4.3	Análise do espaço latente com UMAP	18
4.3.1	UMAP dos backbones	18
4.3.2	UMAP dos modelos downstream	21
4.4	Segmentações	23
4.4.1	Generalização	24
4.5	Conclusão	27

1 Introdução

A interpretação de dados sísmicos desempenha um papel central em diversas áreas da geologia e da geofísica, permitindo a caracterização de estruturas na subsuperfície, como falhas, dobras, domos salinos e a identificação de recursos naturais, como minérios, gás natural e petróleo. Uma das abordagens mais utilizadas para esse tipo de análise é o uso de atributos sísmicos [1], os quais são transformações matemáticas aplicadas sobre o dado para realçar propriedades específicas como amplitude, frequência, continuidade e geometria dos refletores. Estas características são importantes na interpretação, por destacarem padrões geológicos que facilitam a anotação das fácies sísmicas [2].

No entanto, com o aumento no número de aquisições e, por consequência, no volume de dados, o processo de interpretação manual desses volumes tornou-se cada vez mais desafiador, exigindo um número maior de profissionais altamente especializados e também gerando segmentações com vieses e inconsistências ao longo dos volumes.

Por conta desses problemas, nos últimos anos técnicas de *machine learning* e *deep learning* têm sido aplicadas, e diversos estudos sobre quais são os melhores atributos para a segmentação de fácies sísmicas manualmente podem ser encontrados na literatura [3, 4], assim como o uso desses atributos para detecção de falhas em treinamentos não supervisionados [5], detecção de horizontes [6], treinamento supervisionado para segmentação [7, 8] e técnicas de *few-shot* para segmentação [9, 10] devido à grande falta de dados rotulados nessa área.

Apesar desses avanços, essa área sofre com a falta de dados anotados, pois o processo de anotação desses *datasets* é extremamente custoso, demorado e impreciso, como citado anteriormente. Além disso, as características do dado sísmico tornam tarefas como segmentação e detecção de falhas especialmente difíceis, e por conta disso os modelos apresentados nos trabalhos mencionados possuem grande dificuldade em generalizar para regiões não vistas durante o treinamento, errando nas interfaces entre duas classes e sofrendo com uma baixa precisão em regiões com classes minoritárias.

Nesse sentido, trabalhos que abordem características como a qualidade das representações que podem ser obtidas a partir do pré-treino com técnicas de aprendizado autosupervisionado (SSL, *Self Supervised Learning*) [11] ou diferentes abordagens de treinamento ao integrar o uso desses atributos tornam-se relevantes pelo potencial que possuem de favorecer diversos outros métodos que já são utilizados e que sofrem pela ausência de dados anotados.

2 Objetivos e Escopo do Projeto

Durante a elaboração da proposta inicial, o objetivo central deste projeto era *avaliar o impacto do uso de atributos sísmicos em técnicas de Self-Supervised Learning (SSL)*, com a hipótese de que esses atributos, ao substituir transformações aleatórias comumente utilizadas nesses métodos, pudessem produzir representações mais robustas, estáveis e capazes de generalizar melhor para tarefas de *downstream*. A proposta previa ainda a investigação sobre como essas representações influenciariam a construção do espaço latente e o desempenho final em segmentação sísmica.

Entretanto, ao longo do desenvolvimento do trabalho, o escopo metodológico foi refinado para uma direção mais específica e compatível com o tempo e os recursos disponíveis. Dessa forma, o projeto passou a se concentrar na **incorporação de diferentes atributos sísmicos durante o processo de pré-treino e treinamento supervisionado**, utilizando exclusivamente o *Bootstrap Your Own Latent* (BYOL) [12] como técnica de SSL, analisando o comportamento dessa técnica no domínio sísmico e também como esses atributos podem funcionar como novas *views* tanto no pré-treino quanto no refino (*downstream*) para melhorarem o desempenho dos modelos de segmentação no *dataset* Parihaka / SEAM AI [13].

2.1 Objetivo Geral

O objetivo deste trabalho é **investigar o impacto da incorporação de atributos sísmicos no processo de pré-treino e treinamento supervisionado de modelos de *machine learning* para segmentação semântica**, avaliando como essas diferentes *views* influenciam a qualidade das representações aprendidas e o desempenho em tarefas de segmentação.

2.2 Objetivos Específicos

- Aplicar o método **BYOL** como técnica de pré-treino, analisando o comportamento dessa técnica no domínio sísmico.
- Incorporar atributos sísmicos como novas *views* (aumentações) durante o pré-treino com BYOL.
- Avaliar o impacto do uso de atributos sísmicos como forma de aumento de dados no refino.
- Comparar o comportamento do modelo pré-treinado com modelos *from scratch* e com pesos treinados de forma supervisionada com *ImageNet*, realizando o *finetuning* com diferentes regimes de dados.

3 Metodologia

3.1 Ferramental

3.1.1 Dataset

Para a análise de dados sísmicos, utilizou-se o dataset **SEAM AI Parihaka**, que consiste em imagens sísmicas 3D de alta resolução com anotações de fácies geológicas. O conjunto de dados foi desenvolvido para avançar o estado da arte em interpretação automática de estruturas subterrâneas, contendo informações essenciais para exploração de petróleo e gás, além do monitoramento de sequestro de carbono.

A interpretação de fácies sísmicas é uma tarefa complexa que requer conhecimento especializado. Para aplicações em exploração de recursos naturais, é crucial que essa análise seja precisa e eficiente. No dataset selecionado, são identificados vários tipos de fácies, sendo possível observar seis principais, como mostrado na Figura 1.

A escolha desse dataset se justifica pela sua representatividade de cenários geológicos complexos e pela qualidade das anotações especializadas, que permitem treinar modelos robustos para segmentação semântica. Além disso, a diversidade de padrões sísmicos presentes no SEAM AI o torna ideal para aplicações de aprendizado de máquina em geofísica.

Para a organização experimental, o volume foi dividido em três regiões distintas: a primeira parte do volume foi utilizada para o treino, contendo 200 *crosslines*; em seguida, a região intermediária foi destinada à validação, composta por 51 *crosslines*; por fim, todo o restante do volume, imediatamente após a região de validação, foi reservado para teste, totalizando 1121 *crosslines*. Essa divisão permite avaliar de forma mais realista a capacidade de generalização dos modelos ao longo do volume sísmico.

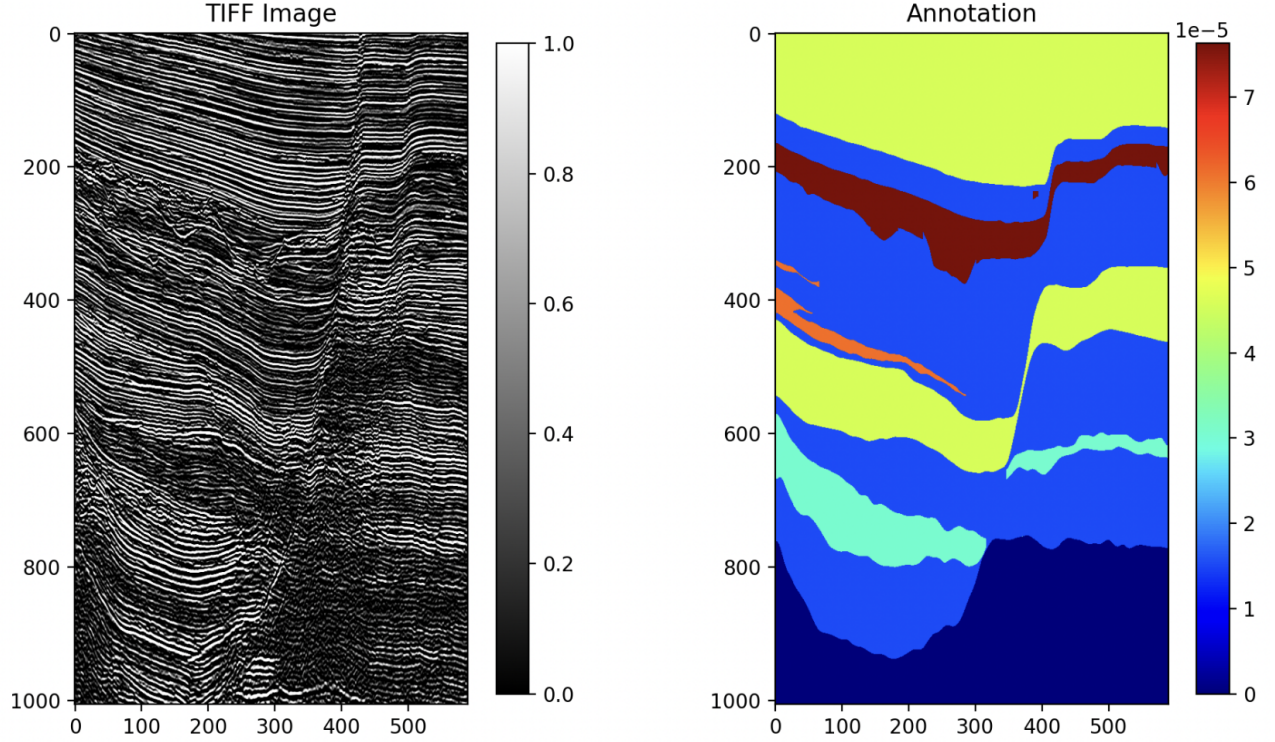


Figura 1: Exemplo de fácies sísmicas presentes no dataset SEAM AI

3.1.2 Atributos sísmicos

Para o cálculo dos atributos sísmicos, foi utilizada a biblioteca DASF [14], que otimiza a computação desses atributos em GPU.

Para os treinamentos, foram utilizados os atributos *Envelope*, *Instantaneous Phase*, *Coherence*, *GLCM Dissimilarity*, *LBP3D* e *GST3D-dip* [15, 16, 17, 18, 19, 20], todos normalizados por meio de *z-score*.

3.1.3 Backbone

Para a realização da tarefa, a arquitetura escolhida foi a DeepLabV3, um modelo avançado de redes neurais convolucionais para segmentação semântica, que emprega tipicamente o ResNet-50 como *backbone* para extração de características hierárquicas. Este modelo combina as vantagens das convoluções atóricas (*dilated convolutions*) e do *pooling* espacial piramidal (ASPP) com a capacidade do ResNet-50 de aprender representações profundas através de suas conexões residuais [21, 22].

No contexto sísmico, o SEAMAI-DeepLabV3 aproveita a estrutura do ResNet-50 para processar inicialmente os dados sísmicos, onde as camadas convolucionais iniciais capturam padrões locais (como reflexões e descontinuidades), enquanto as camadas mais profundas identificam estruturas em escala maior (como sistemas deposicionais ou falhas regionais). O módulo ASPP opera sobre essas características multiescala, permitindo a integração de contextos espaciais variados sem perda de resolução.

A combinação entre o *backbone* ResNet-50 e os mecanismos do DeepLabV3 oferece três vantagens principais para aplicações sísmicas:

- Robustez a variações de escala em estruturas geológicas

- Preservação de bordas e detalhes finos durante a segmentação
- Eficiência computacional no processamento de volumes sísmicos 3D

Esta arquitetura tem se mostrado particularmente eficaz em tarefas de interpretação automática de dados sísmicos, como a delimitação de corpos arenosos em reservatórios ou a identificação de zonas de fraturamento, onde a precisão na localização espacial é tão crítica quanto a classificação semântica.

3.1.4 Técnica de SSL

A técnica de SSL escolhida foi o Bootstrap Your Own Latent (BYOL), um método de aprendizado auto-supervisionado que aprende representações úteis sem utilizar pares negativos, diferentemente de abordagens contrastivas tradicionais [12]. O BYOL opera por meio de duas redes neurais idênticas (chamadas de *online* e *target*), onde a rede *online* é treinada para prever as representações da rede *target* a partir de uma vista aumentada da mesma entrada, ambas representadas na Figura 2.

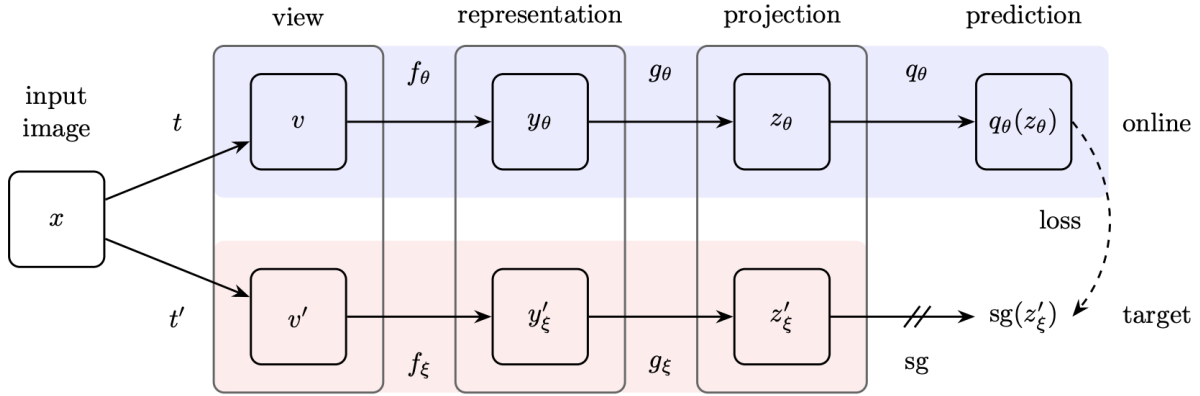


Figura 2: Arquitetura BYOL

Esta técnica foi selecionada para a tarefa de segmentação semântica dados sísmicos devido à sua capacidade única de aprender representações ricas sem a necessidade de dados rotulados durante o pré-treino.

Sua independência de pares negativos a torna particularmente robusto em cenários com distribuições de dados desbalanceadas, comuns em aplicações sísmicas onde algumas fácies como água, aparecem muito mais do que outras fácies sísmicas. Além disso, a natureza auto-supervisionada do método permite aproveitar grandes volumes de dados não rotulados, seguido por um *fine-tuning* eficiente com conjuntos reduzidos de anotações, sendo essa uma estratégia ideal para domínios onde a aquisição de rótulos especializados é custosa.

3.2 Estratégias de Pré-Treino e Downstream

A metodologia de pré-treino foi organizada em três etapas principais. A primeira etapa consistiu em caracterizar o comportamento da técnica BYOL no domínio sísmico, enquanto a segunda concentrou-se na adição dos atributos sísmicos de forma estruturada para a criação dos *backbones* e modelos finais. Por fim, a terceira etapa focou na análise das representações aprendidas pelos *backbones* para entender o impacto causado pelo uso desses atributos na tarefa de pré-treino e no refino.

Para determinar a melhor forma de empregar a técnica, foi realizado um estudo em artigos de SSL e, principalmente, no próprio artigo do BYOL. A partir dessa análise, observou-se que um dos

principais fatores que impactam o desempenho de modelos auto-supervisionados é o tamanho do *batch* utilizado durante o pré-treino. Embora o artigo do BYOL ressalte que a técnica é mais resistente a *batches* menores quando comparada a métodos como o SimCLR [23], conforme ilustrado na Figura 3, os próprios apêndices do trabalho evidenciam que o desempenho degrada em *batches* muito pequenos. Esse efeito pode ser visto na tabela da Figura 4, onde há uma queda de aproximadamente 10% quando o *batch size* é reduzido para 64.

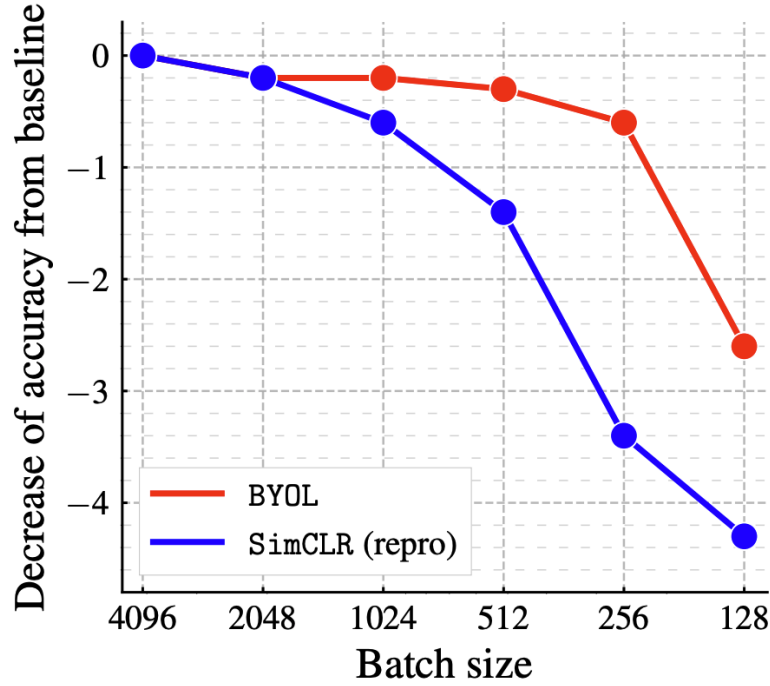


Figura 3: Tabela com o impacto detalhado do *batch size* no BYOL

Batch size	Top-1		Top-5	
	BYOL (ours)	SimCLR (repro)	BYOL (ours)	SimCLR (repro)
4096	72.5	67.9	90.8	88.5
2048	72.4	67.8	90.7	88.5
1024	72.2	67.4	90.7	88.1
512	72.2	66.5	90.8	87.6
256	71.8	64.3±2.1	90.7	86.3±1.0
128	69.6±0.5	63.6	89.6	85.9
64	59.7±1.5	59.2±2.9	83.2±1.2	83.0±1.9

Figura 4: Impacto do *batch size* no BYOL

Com base nessas observações, foram conduzidos experimentos variando o *batch size* no pré-treino, utilizando somente o dado sísmico. Além disso, avaliou-se o impacto de estratégias de congelamento (*freeze*) e não-congelamento (*full fine-tuning*) do *backbone*, bem como a comparação com modelos inicializados a partir da ImageNet. Essas análises permitiram compreender melhor a estabilidade do BYOL em diferentes condições de treinamento no domínio sísmico.

Na segunda etapa, um treino inicial foi realizado com atributos sísmicos para uma análise de comportamento geral, e após isso os hiperparâmetros mais promissores identificados nas análises anteriores foram utilizados para realizar novos pré-treinos de forma mais estruturada, agora incorporando diferentes atributos sísmicos. Esses modelos foram posteriormente comparados com modelos treinados do zero (*from scratch*) e com modelos que herdaram os pesos da ImageNet [24], permitindo avaliar a contribuição dos atributos e do pré-treino auto-supervisionado no desempenho final da tarefa de segmentação.

A terceira etapa concentrou-se em examinar as representações aprendidas pelos modelos pré-treinados e após o *downstream*. Para isso, foram geradas projeções UMAP dos conjuntos de treino, validação e teste, a fim de visualizar a organização das *features* no espaço latente do dado. Além disso, foram avaliadas as projeções de *patches* de diferentes classes da partição de treino, permitindo investigar a separabilidade entre elas tanto nos *backbones* quanto nos modelos ajustados para a tarefa de *downstream*.

4 Resultados

4.1 Pré-treino com BYOL

Foram feitos diversos treinamentos com a técnica *BYOL*, variando o *batch size* entre os tamanhos de 128, 200, 256 e 512 amostras. Além disso, como conjunto de transformações, foram utilizadas *RandomFlip* com probabilidade de 50%, *RandomRotation* de 25 graus com probabilidade de 50% e *RandomCrops* variados entre 128×128 e 256×256 . Outra alteração feita foi utilizar o parâmetro *Sync-BatchNorm* em alguns treinamentos para forçar uma mesma normalização no treinamento distribuído entre as GPUs, pois um dos problemas mencionados no artigo do *BYOL* foi a queda de desempenho do modelo causada por essa camada de normalização 3.

F.2 Batch size

We run a sweep over the batch size for both BYOL and our reproduction of SimCLR. As explained in Section 5, when reducing the batch size by a factor N , we average gradients over N consecutive steps and update the target network once every N steps. We report in Table 16, the performance of both our reproduction of SimCLR and BYOL for batch sizes between 4096 (BYOL and SimCLR default) down to 64. We observe that the performance of SimCLR deteriorates faster than the one of BYOL which stays mostly constant for batch sizes larger than 256. We believe that the performance at batch size 256 could match the performance of the large 4096 batch size with proper parameter tuning when accumulating the gradient. We think that the drop in performance at batch size 64 in table 16 is mainly related to the ill behaviour of batch normalization at low batch sizes [100].

Figura 5: Citação do problema de batches pequenos no paper do BYOL

Todos os treinamentos foram realizados com 500 ou 1000 épocas, a fim de garantir que o modelo aprendesse a representação necessária.

Para o *downstream*, foram realizados treinos com 50 épocas e *batches* de 8, utilizando amostras no formato 1008x592. Além disso, foram realizados *downstreams* com os *backbones* congelados e descongelados para entender o efeito de manter as *features* aprendidas durante o pré-treino. Por fim, foi feita uma avaliação dos modelos finais e dos modelos com melhor *loss* de validação, como apresentado na Figura 6.

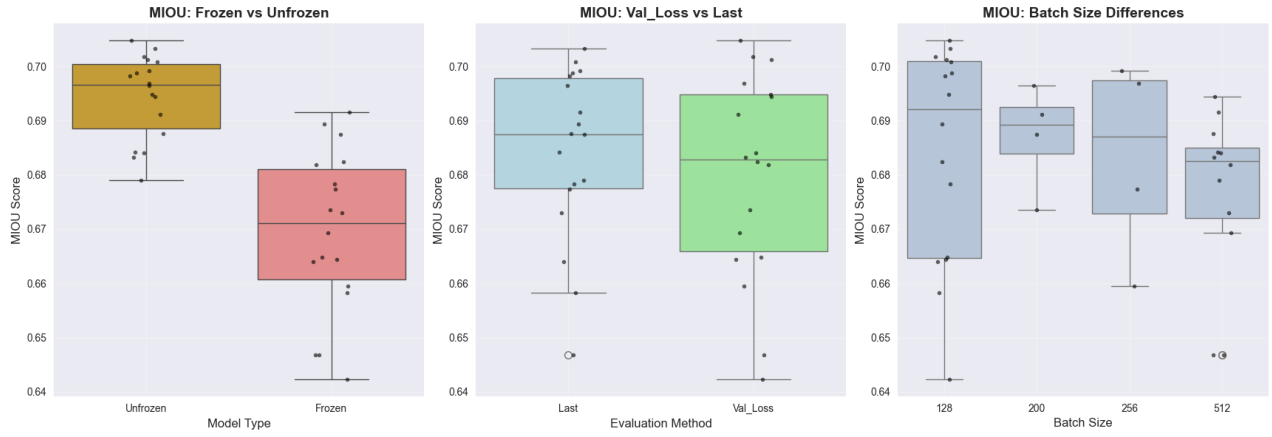


Figura 6: Comparação dos modelos com *backbone* congelado (*frozen*) e descongelado (*unfrozen*), último checkpoint vs checkpoint com melhor loss de validação

Os modelos *unfrozen* performaram melhor que os modelos *frozen*, mostrando que uma maior flexibilidade para a adaptação do modelo à tarefa de segmentação no *downstream* se mostrou favorável.

Além disso, também foi observado que tanto o último *checkpoint* dos modelos quanto os modelos com melhores *losses* de validação não apresentaram grande diferença.

Por fim, para entender melhor o comportamento do *batch size*, foram gerados mais dois gráficos, separando o desempenho dos modelos *frozen* e *unfrozen*, apresentados nas Figuras 7 e 8. A partir delas, podemos observar que, para os modelos com o *backbone* congelado, não houve um impacto notório causado pela variação do *batch size*, com todos os valores escolhidos apresentando intervalos de confiança similares.

Já para os modelos com o *backbone* descongelado, notamos que o melhor desempenho foi obtido pelo modelo com um tamanho de *batch* de 128 amostras, havendo uma degradação do desempenho conforme esse valor aumentava.

A um primeiro momento, isso parece ir contra o que foi reportado no artigo original do BYOL e também ao que é conhecido pela literatura de modelos de *SSL*, segundo a qual tamanhos de *batch* maiores tendem a melhorar o desempenho dessas técnicas. No entanto, ao analisar mais profundamente, notamos que esse comportamento de tamanhos de *batch* maiores degradarem a performance, mas trazerem maior estabilidade ao treinamento, é algo já conhecido em metodologias clássicas de *Machine Learning* [25], justamente por não fornecerem a variabilidade que *batches* menores oferecem ao longo do treinamento.

O problema citado no artigo para *batches* pequenos se refere à normalização na camada *Batch Norm* [26] se comportar de forma instável ao longo do treinamento. Diante dos resultados encontrados, teorizamos que isso ocorre quando o tamanho do *batch* leva a uma normalização pouco representativa para as demais amostras do *dataset*, como no caso de treinamentos realizados em *datasets* como a *ImageNet*. No caso de dados sísmicos, a variabilidade é consideravelmente menor, pois o dado tende a seguir uma distribuição semelhante ao longo do volume. Por conta disso, a instabilidade causada na normalização desses *batches* tende a ser reduzida. Sendo assim, *batches* menores fornecem uma maior variabilidade ao longo do pré-treino sem sofrer com a instabilidade mencionada.



Figura 7: Impacto do *batch_size* para modelos com *backbone* congelado



Figura 8: Impacto do *batch_size* para modelos com *backbone* descongelado

4.2 Incorporação dos atributos sísmicos

Para a incorporação dos atributos nos modelos, foram utilizadas 3 abordagens diferentes:

- dado x atributo: Pré-treinado com a *view* da rede *online* sendo o dado sísmico e a *view* da rede *target* um atributo sísmico dessa região. *Downstream* feito com dado e atributo.
- dado/atributo x dado/atributo: Pré-treinado com as *views* das redes *online* e *target* podendo ser tanto um atributo sísmico quanto o próprio dado. *Downstream* feito com dado e atributo.

- channels: Pré-treinado com amostras compostas por 3 canais (dado, atributo_1, atributo_2). *Downstream* utilizando o mesmo formato de amostras.

Todos os pré-treinos foram feitos por 500 épocas com *batch_size* de 128 e utilizaram as mesmas transformações apresentadas na Seção 4.1, mantendo o tamanho dos *crops* em 128×128 .

Para a avaliação dos modelos, foram calculadas as métricas de mIoU para os cenários **dado** \times **atributo** e **dado/atributo** \times **dado/atributo** utilizando a repartição de teste do dado sísmico, apresentadas nas Tabelas 1 e 2 e Figuras 9 e 10. Além disso, também foi calculada a métrica utilizando a repartição de teste dos respectivos atributos sísmicos, resultados apresentados nas Tabelas 3 e 4 e Figuras 11 e 12.

A partir disso, foi possível observar que todos os modelos testados nos atributos sísmicos obtiveram um desempenho muito inferior quando comparados às *baselines* apresentadas (*from_scratch*, ImageNet e BYOL). No entanto, para os modelos testados no dado sísmico, nota-se que quase todos superaram as *baselines* *from_scratch* e BYOL no regime de 128 amostras, e alguns superaram todas as *baselines* nesse regime. Já para o regime com todas as amostras, em ambos os formatos de treinamento, obteve-se pelo menos um modelo que superou as *baselines* com o uso de atributos, sendo o melhor o modelo **BYOL data-instantaneous-phase**, com mIoU de 0.7123.

Por fim, o experimento utilizando os atributos em diferentes canais não apresentou nenhum modelo que superasse as *baselines* no regime de poucos dados. Somente no *downstream* com todas as amostras, o modelo **BYOL data-coherence-instantaneous-phase** superou as *baselines*, com mIoU de 0.7131, como apresentado na Tabela 5 e Figura 13.

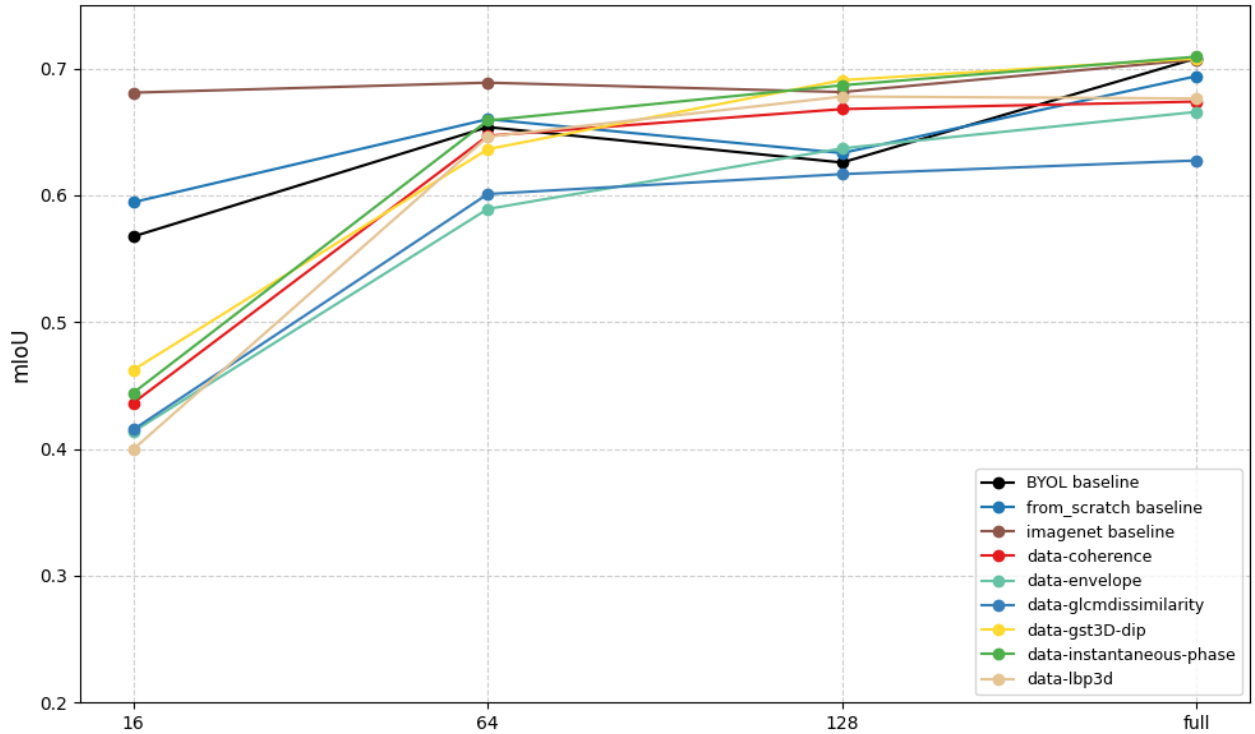


Figura 9: mIoU dos modelos dado \times atributo testados no dado sísmico

Tabela 1: mIoU dos modelos dado \times atributo testados no dado sísmico

Model/Attribute Set	16	64	128	Full
BYOL baseline	0.5675	0.6538	0.6259	0.7082
from scratch baseline	0.5946	0.6601	0.6334	0.6940
imagenet baseline	0.6810	0.6889	0.6815	0.7071
BYOL data-coherence	0.4363	0.6470	0.6680	0.6739
BYOL data-envelope	0.4136	0.5893	0.6371	0.6659
BYOL data-glcmdissimilarity	0.4152	0.6010	0.6168	0.6275
BYOL data-gst3D-dip	0.4623	0.6364	0.6908	0.7080
BYOL data-instantaneous-phase	0.4442	0.6590	0.6868	0.7094
BYOL data-lbp3d	0.3997	0.6462	0.6779	0.6764

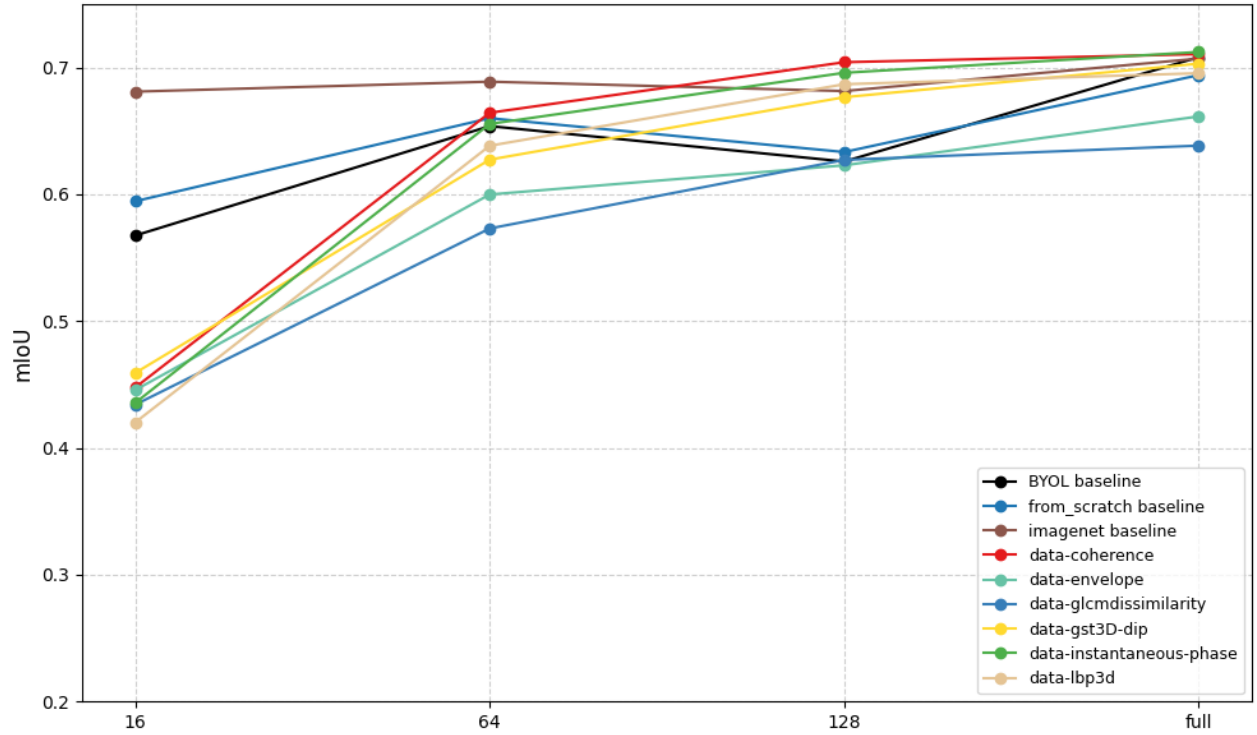
Figura 10: mIoU dos modelos dado/atributo \times dado/atributo testados no dado sísmico

Tabela 2: mIoU dos modelos dado/atributo \times dado/atributo testados no dado sísmico

Model/Attribute Set	16	64	128	Full
BYOL baseline	0.5675	0.6538	0.6259	0.7082
from scratch baseline	0.5946	0.6601	0.6334	0.6940
imagenet baseline	0.6810	0.6889	0.6815	0.7071
BYOL data-coherence	0.4477	0.6643	0.7042	0.7106
BYOL data-envelope	0.4457	0.5999	0.6229	0.6615
BYOL data-glcmdissimilarity	0.4341	0.5731	0.6272	0.6384
BYOL data-gst3D-dip	0.4592	0.6274	0.6766	0.7031
BYOL data-instantaneous-phase	0.4356	0.6555	0.6959	0.7123
BYOL data-lbp3d	0.4201	0.6383	0.6868	0.6956

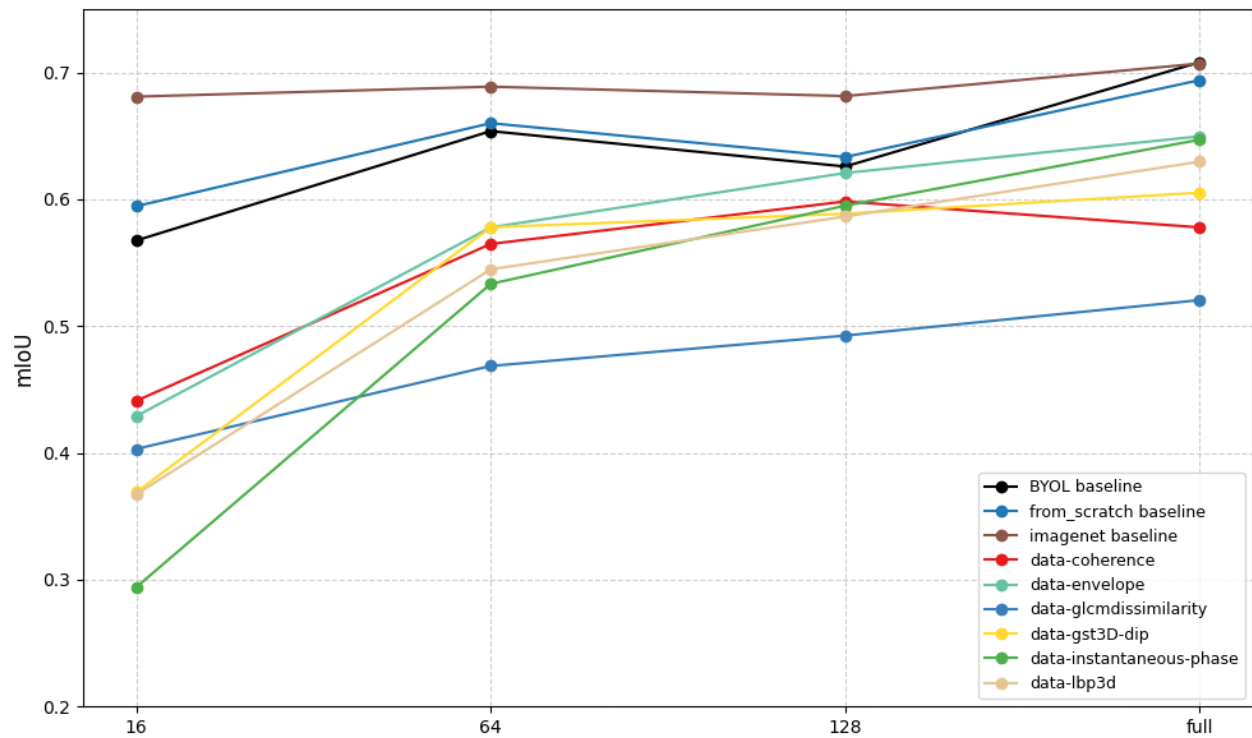

 Figura 11: mIoU dos modelos dado \times atributo testados no atributo sísmico

Tabela 3: mIoU dos modelos dado \times atributo testados no atributo sísmico

Model/Attribute Set	16	64	128	Full
BYOL baseline	0.5675	0.6538	0.6259	0.7082
from scratch baseline	0.5946	0.6601	0.6334	0.6940
imagenet baseline	0.6810	0.6889	0.6815	0.7071
BYOL data-coherence	0.4411	0.5648	0.5983	0.5779
BYOL data-envelope	0.4290	0.5779	0.6208	0.6497
BYOL data-glcmdissimilarity	0.4031	0.4686	0.4926	0.5205
BYOL data-gst3D-dip	0.3689	0.5780	0.5887	0.6053
BYOL data-instantaneous-phase	0.2941	0.5333	0.5950	0.6470
BYOL data-lbp3d	0.3674	0.5448	0.5865	0.6298

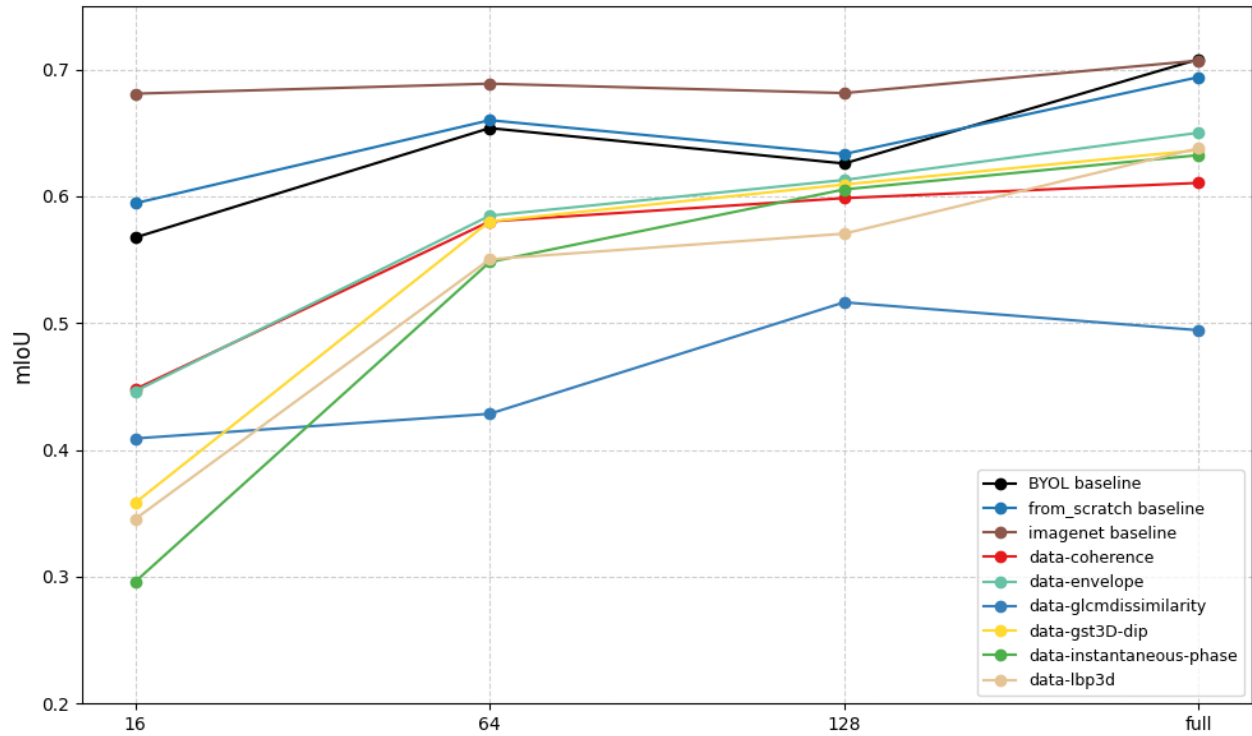
Figura 12: mIoU dos modelos dado/atributo \times dado/atributo testados no atributo sísmico

Tabela 4: mIoU dos modelos dado/atributo \times dado/atributo testados no atributo sísmico

Model/Attribute Set	16	64	128	Full
BYOL baseline	0.5675	0.6538	0.6259	0.7082
from scratch baseline	0.5946	0.6601	0.6334	0.6940
imagenet baseline	0.6810	0.6889	0.6815	0.7071
BYOL data-coherence	0.4479	0.5802	0.5986	0.6106
BYOL data-envelope	0.4463	0.5847	0.6129	0.6502
BYOL data-glcmdissimilarity	0.4091	0.4285	0.5164	0.4944
BYOL data-gst3D-dip	0.3584	0.5803	0.6093	0.6364
BYOL data-instantaneous-phase	0.2961	0.5482	0.6055	0.6325
BYOL data-lbp3d	0.3456	0.5503	0.5706	0.6380

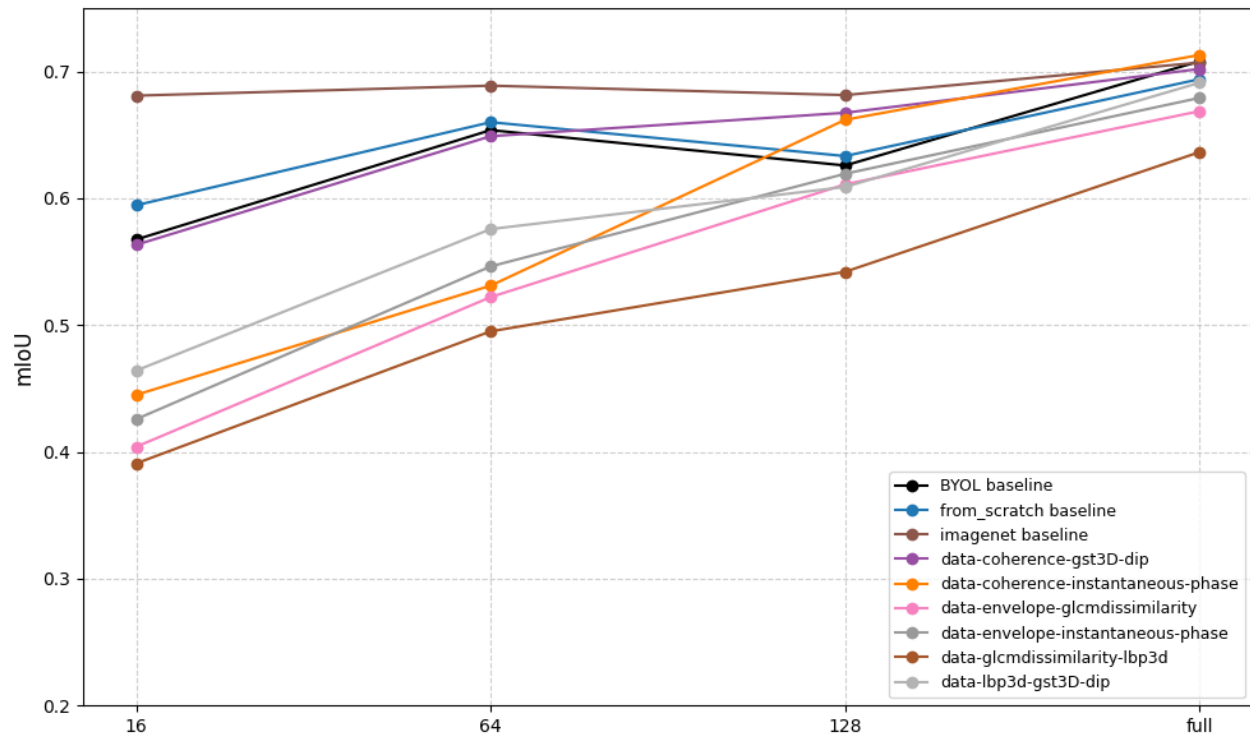


Figura 13: mIoU dos modelos com 3 canais, (dado, atributo_1, atributo_2)

Tabela 5: mIoU dos modelos com 3 canais, (dado, atributo_1, atributo_2)

Model/Attribute Set	16	64	128	Full
BYOL baseline	0.5675	0.6538	0.6259	0.7082
from scratch baseline	0.5946	0.6601	0.6334	0.6940
imagenet baseline	0.6810	0.6889	0.6815	0.7071
BYOL data-coherence-gst3D-dip	0.5634	0.6490	0.6675	0.7021
BYOL data-coherence-instantaneous-phase	0.4450	0.5312	0.6620	0.7131
BYOL data-envelope-glcmdissimilarity	0.4041	0.5222	0.6111	0.6688
BYOL data-envelope-instantaneous-phase	0.4259	0.5463	0.6194	0.6793
BYOL data-glcmdissimilarity-lbp3d	0.3910	0.4951	0.5421	0.6365
BYOL data-lbp3d-gst3D-dip	0.4642	0.5758	0.6090	0.6912

4.3 Análise do espaço latente com UMAP

Para entender melhor o comportamento do BYOL na construção de um espaço latente, foi realizado um *plot* bidimensional utilizando a técnica **UMAP** [27]. Esse *plot* foi gerado a partir dos *patches* do dado de treino, em que regiões de 64×64 foram caracterizadas por cores de acordo com as classes presentes em cada *patch*. Para regiões contendo somente uma classe, foi atribuída uma cor específica; para regiões com duas classes, uma nova cor foi utilizada; e, para regiões com três classes, outra cor distinta foi empregada. Esses *plots* foram produzidos para os *backbones from_scratch*, pré-treinados e com pesos da ImageNet, bem como para os modelos após o *downstream*, permitindo visualizar o comportamento de cada modelo nas duas etapas e compreender melhor o impacto do pré-treino.

4.3.1 UMAP dos backbones

Nessa etapa, foram gerados UMAPs para backbones com pesos aleatórios, pesos herdados da ImageNet, BYOL pré-treinado com o dado normal, e o melhor modelo obtido, sendo este o **BYOL data-coherence-instantaneous-phase**. Ao analisarmos o UMAP do modelo com pesos aleatórios, apresentado na Figura 14, percebemos que todas as cores estão espalhadas ao longo do gráfico de forma desordenada, tendo somente uma concentração dos *patches* da classe 0 no topo do gráfico. Já para o modelo com pesos da imagenet, apresentado na Figura 15, notamos que existe uma separação um pouco maior entre a classe 0 e a classe 3, representadas em extremos opostos do gráfico. Já para os modelos pré-treinados com BYOL, apresentados nas Figuras 16 e 17, vemos que todos os *patches* foram agrupados de uma forma mais compacta, independentemente da classe.

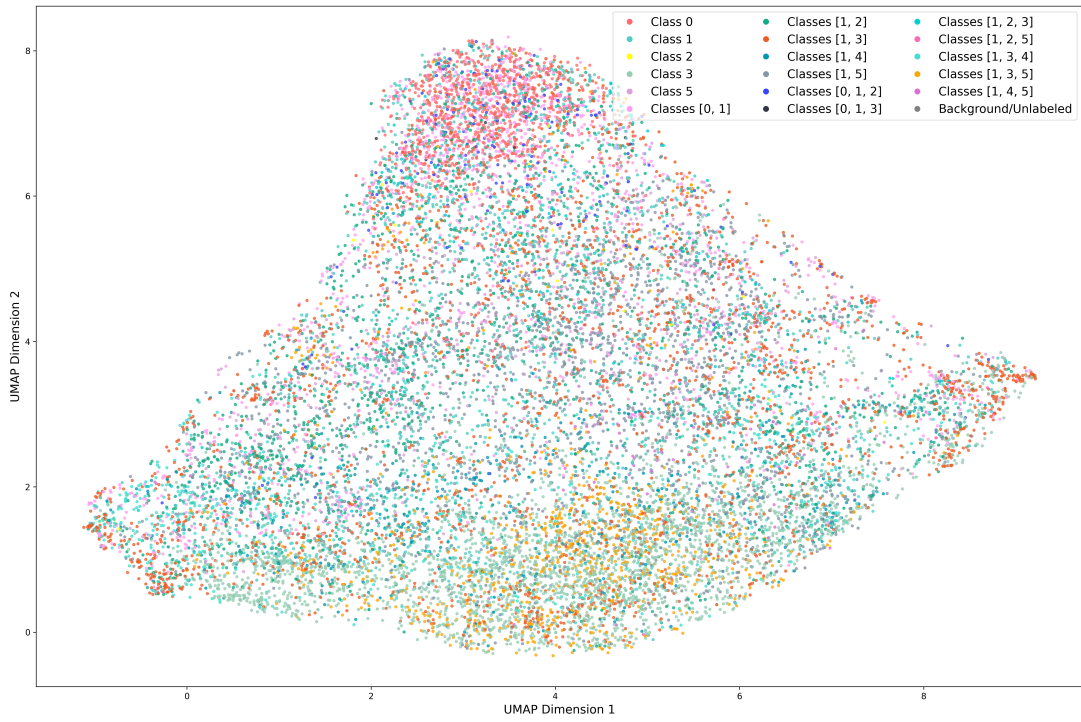


Figura 14: Features dos *patches* com pesos aleatórios no *backbone*

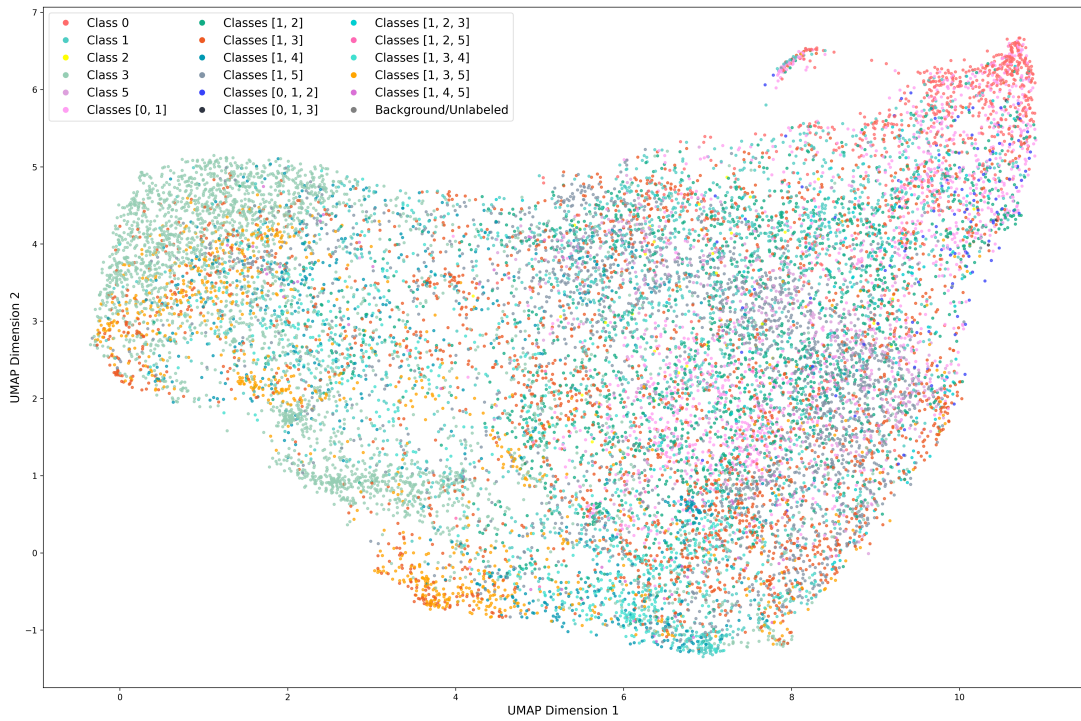


Figura 15: Features dos *patches* com pesos da ImageNet no *backbone*

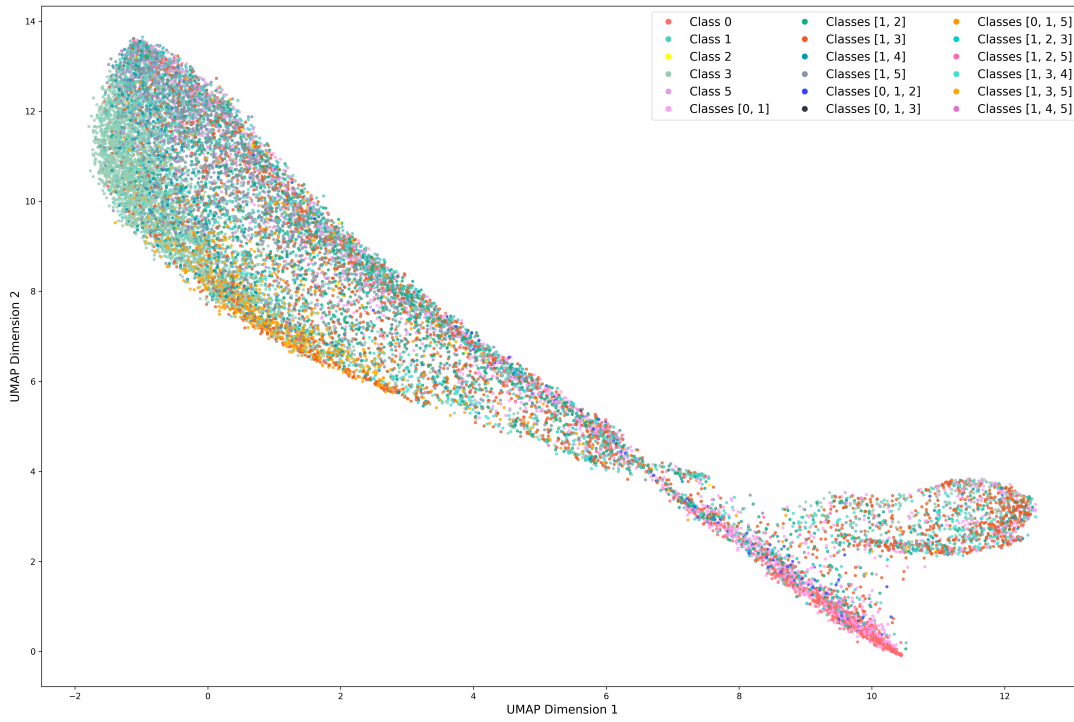


Figura 16: Features dos *patches* com *backbone* pré-treinado com BYOL

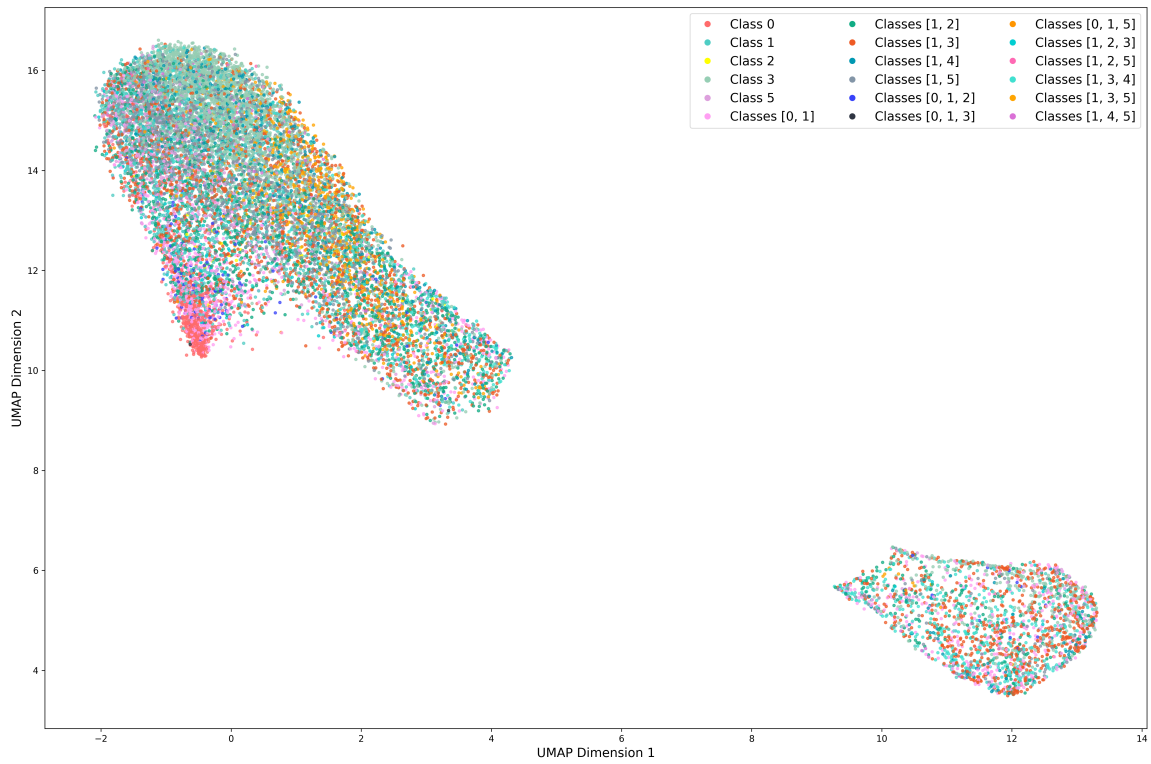


Figura 17: Features dos *patches* com *backbone* pré-treinado com BYOL, utilizando **dado sísmico e os atributos coherence e instantaneous-phase em 3 canais**

4.3.2 UMAP dos modelos downstream

Nessa etapa, foram gerados UMAPs para os modelos após o *downstream* dos modelos já citados. Para os modelos *from_scratch*, ImageNet e BYOL, apresentados nas Figuras 18, 19 e 20, ainda podemos ver uma aglutinação de vários patches, mas as regiões das classes 0 e classe 3 foram segregadas em regiões opostas do gráfico.

Já para o modelo 21, isso também ocorreu, mas podemos notar que patches com as classes 1, 3 e 5, representados pela cor laranja, foram melhor separados, dominando uma região própria no gráfico. Isso parece indicar que o modelo conseguiu separar melhor essas regiões durante a segmentação.

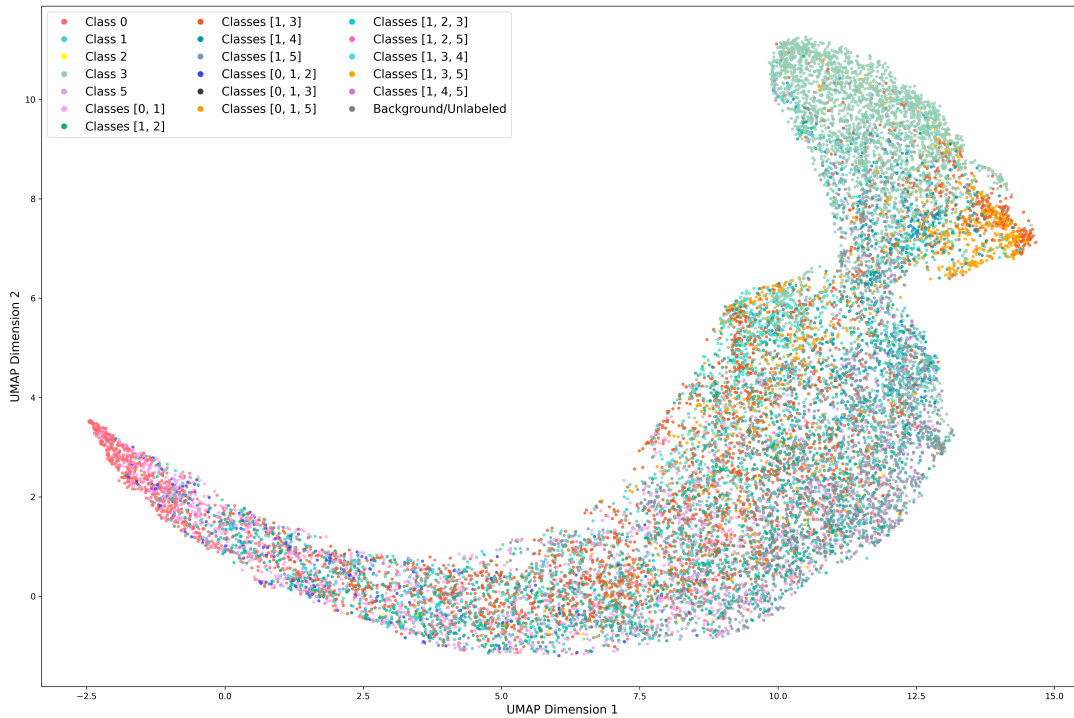


Figura 18: Features dos *patches* após *downstream from_scratch*

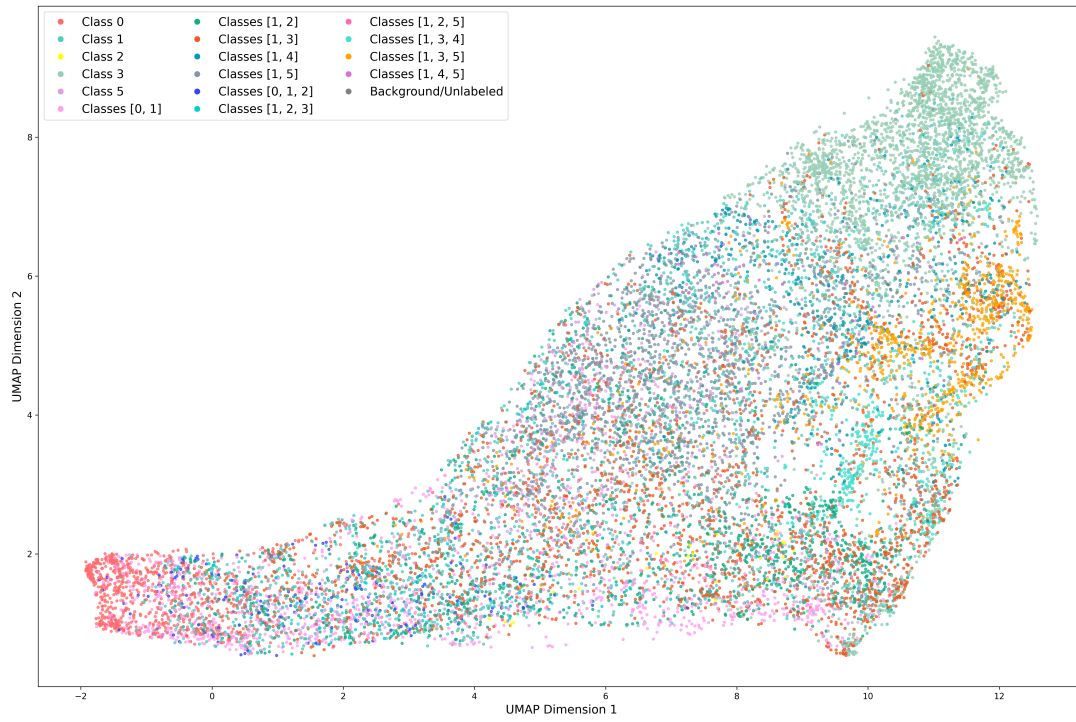


Figura 19: Features dos *patches* após *downstream* com *backbone* com pesos da ImageNet

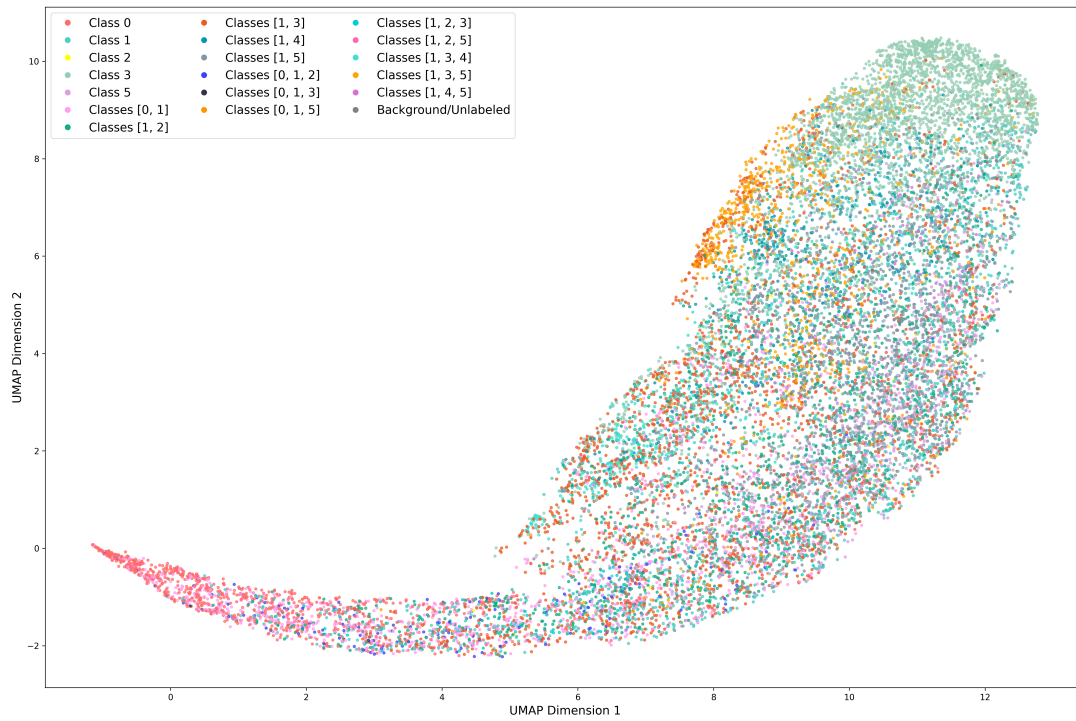


Figura 20: Features dos *patches* após *downstream* com *backbone* pré-treinado com *BYOL*

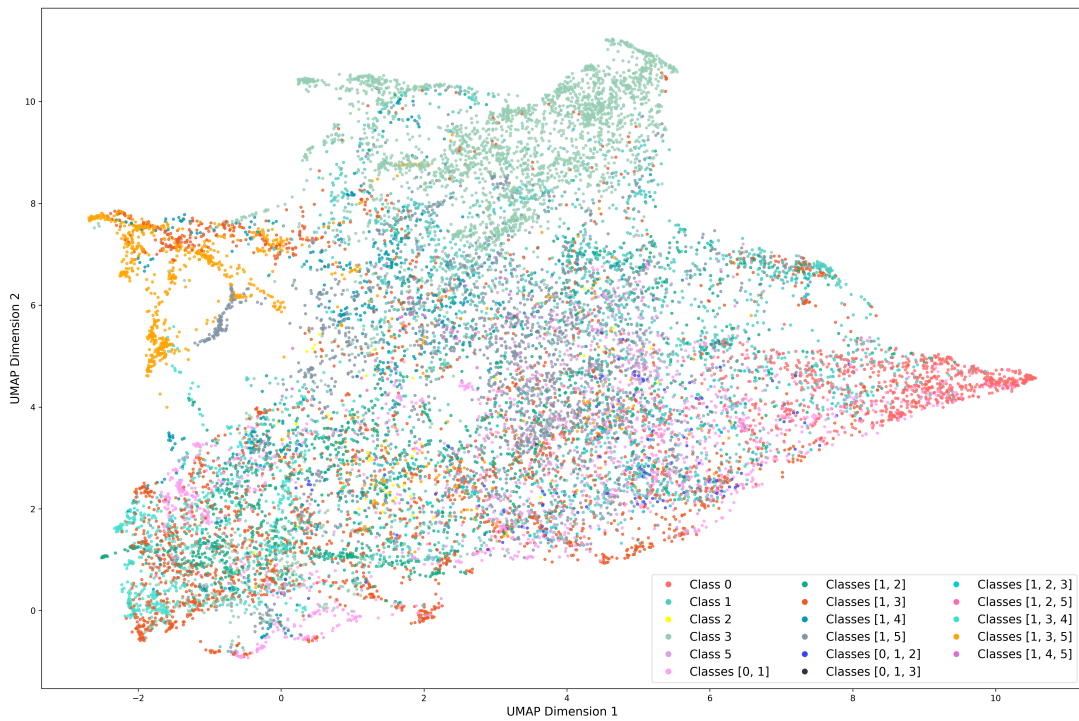


Figura 21: Features dos *patches* após *downstream* com *backbone* pré-treinado com *BYOL*, utilizando dado sísmico e os atributos *coherence* e *instantaneous-phase* em 3 canais

4.4 Segmentações

Analisando a segmentação gerada por cada um dos modelos, podemos ver que o modelo *from_scratch* possui uma tendência de segmentar o dado sísmico mantendo sempre as classes amarela e verde próximas, mesmo quando isso não deveria ocorrer, como apresentado na região mais à direita da segmentação da Figura 22. Já para o modelo treinado com os pesos da ImageNet, podemos ver que ele tende a manter o formato da segmentação para a classe amarela, mas tem dificuldade em separar a classe verde da azul e também em segmentar a classe laranja, como apresentado na região central da Figura 22.

Por fim, para os modelos pré-treinados com o BYOL, podemos ver que ambos apresentam uma grande mistura das classes verde e amarela. Apesar disso, podemos notar que o modelo pré-treinado com BYOL e atributos fez uma separação melhor da região amarela no topo da *crossline* do que os demais, mantendo a faixa azul existente naquela região, como podemos ver na Figura 23.

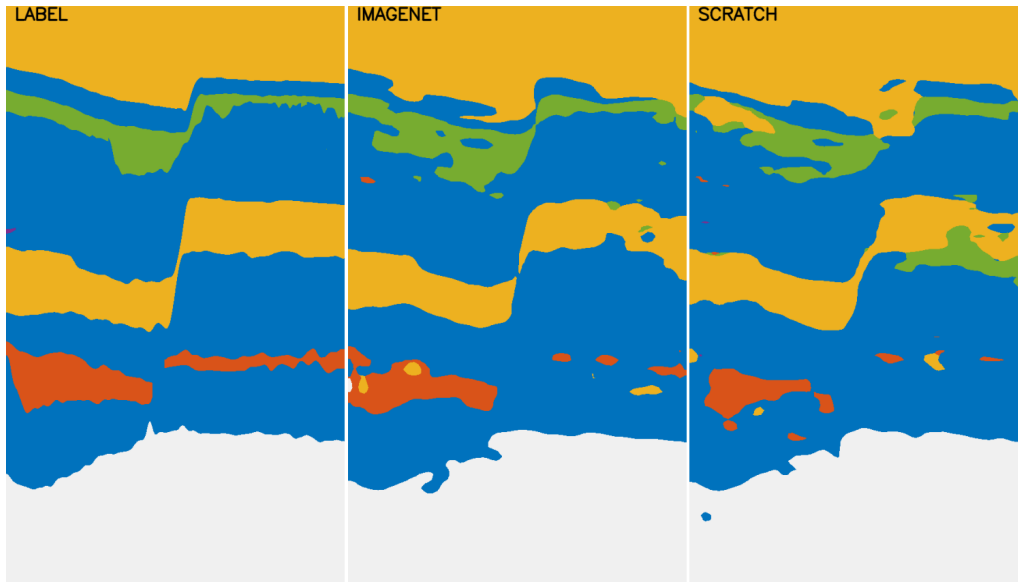


Figura 22: Segmentação da décima *crossline* (xl_10) do conjunto de teste, segmentação feita pelo modelo *from_scratch* e segmentação feita pelo modelo treinado com pesos da imagenet

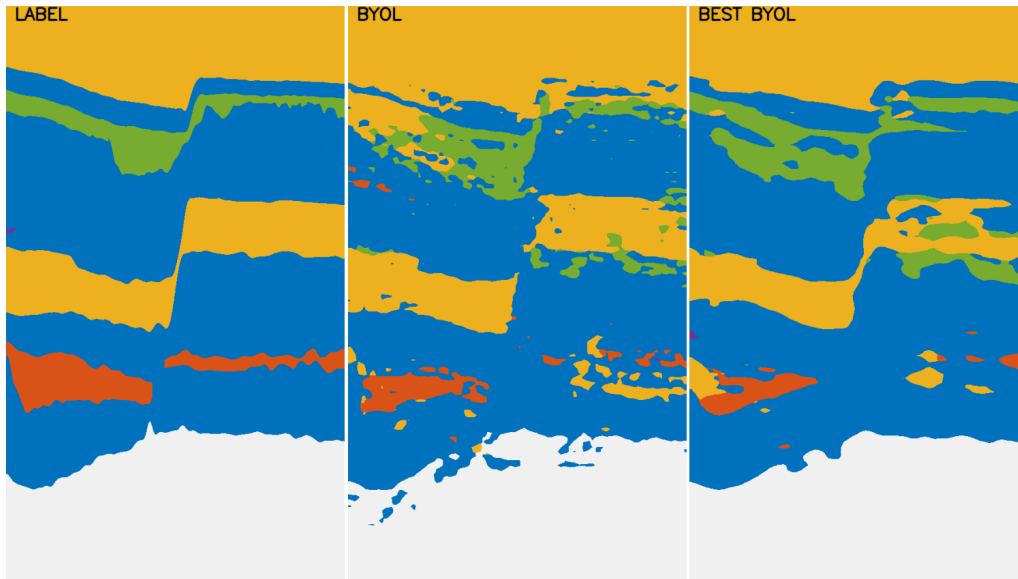


Figura 23: Segmentação da décima *crossline* (xl_10) do conjunto de teste, segmentação feita pelo modelo pré-treinado com BYOL e segmentação feita pelo modelo com melhor mIoU (BYOL data-coherence-instantaneous-phase)

4.4.1 Generalização

Em volumes de dado sísmico, *crosslines* mais próximas tendem a apresentar uma similaridade muito alta e, por conta disso, as segmentações de *crosslines* imediatamente vizinhas costumam ser muito parecidas, como mostrado na Figura 24. Nela, podemos ver a similaridade entre a segmentação da última *crossline* do particionamento de teste e as demais (a *crossline* mais próxima do conjunto de treino).

Devido a essa estrutura, uma boa forma de medir a generalização para esse problema é analisar

como as segmentações geradas pelos modelos tendem a se degradar ao longo das *crosslines* do particionamento de teste, de acordo com a distância destas em relação ao particionamento de treino. Como podemos observar nas Figuras 25, 26, 27 e 28, todos os modelos são afetados por esse problema mencionado.

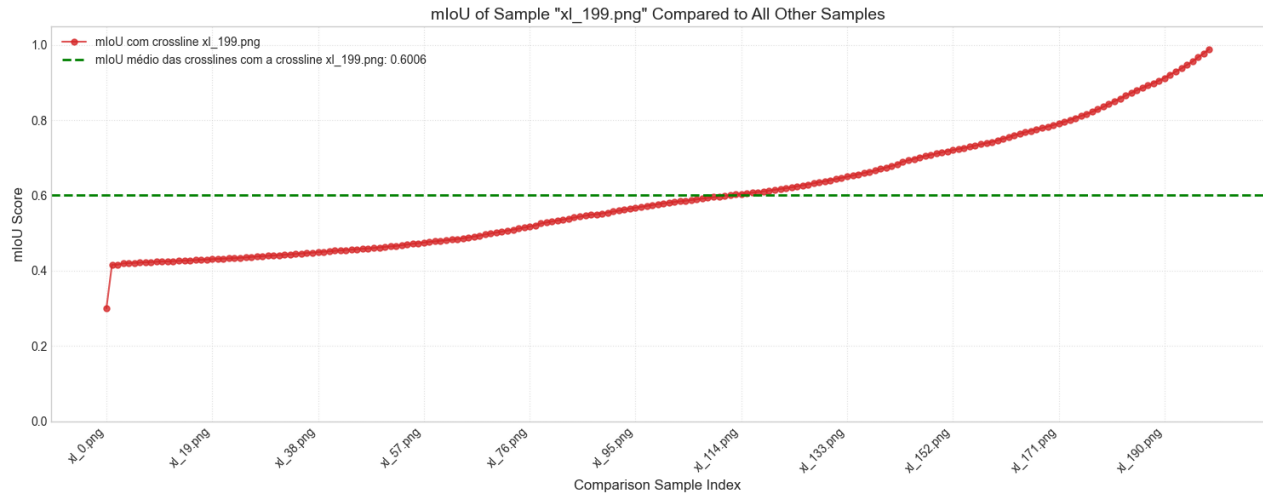


Figura 24: mIoU da última *crossline* da partição de teste com as demais

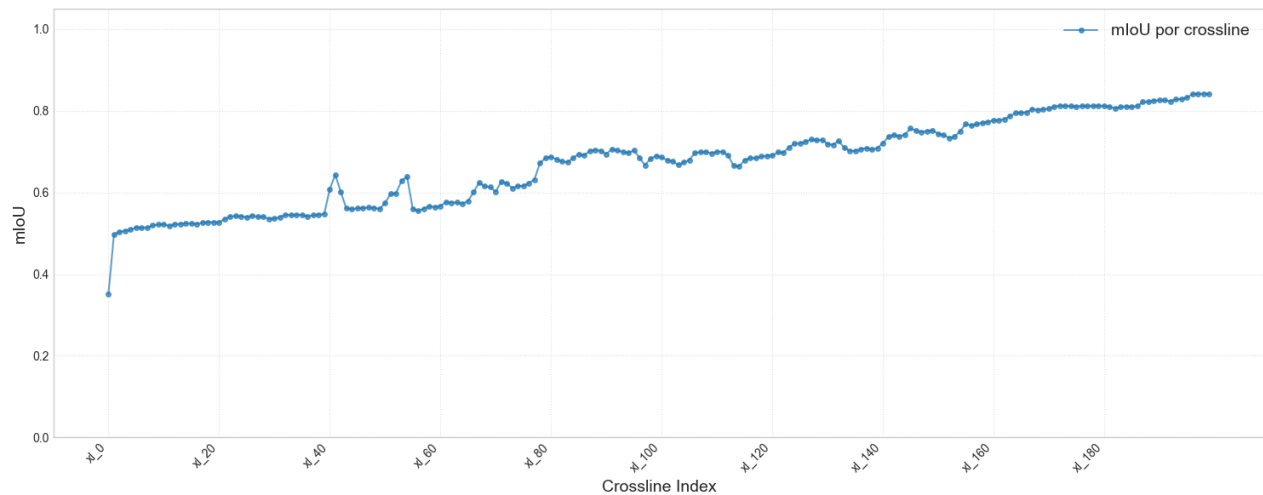


Figura 25: mIoU das predições do modelo from_scratch para as *crosslines* no particionamento de teste

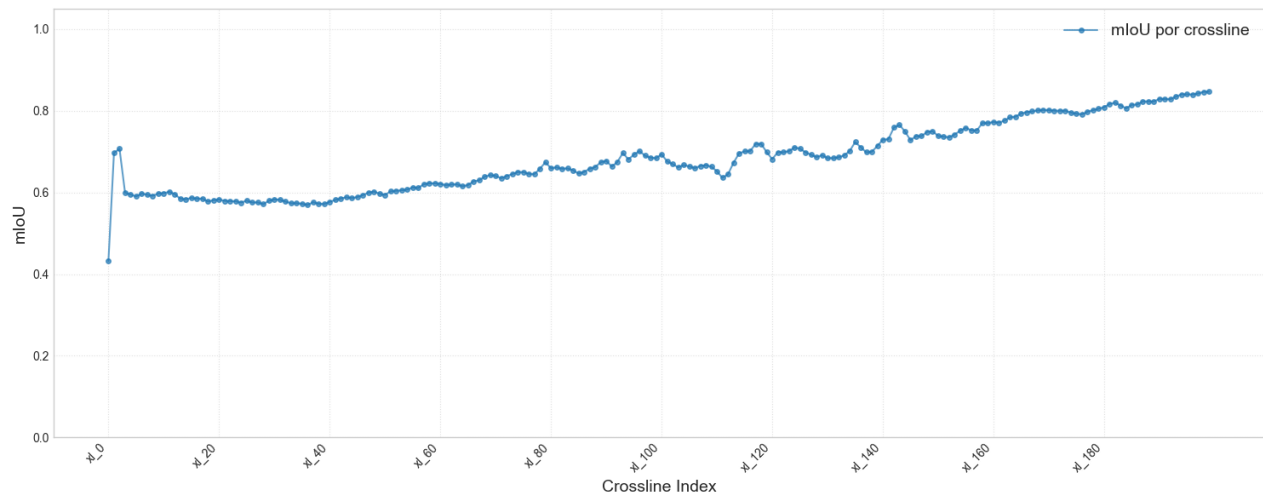


Figura 26: mIoU das predições do modelo from_scratch para as *crosslines* no particionamento de teste

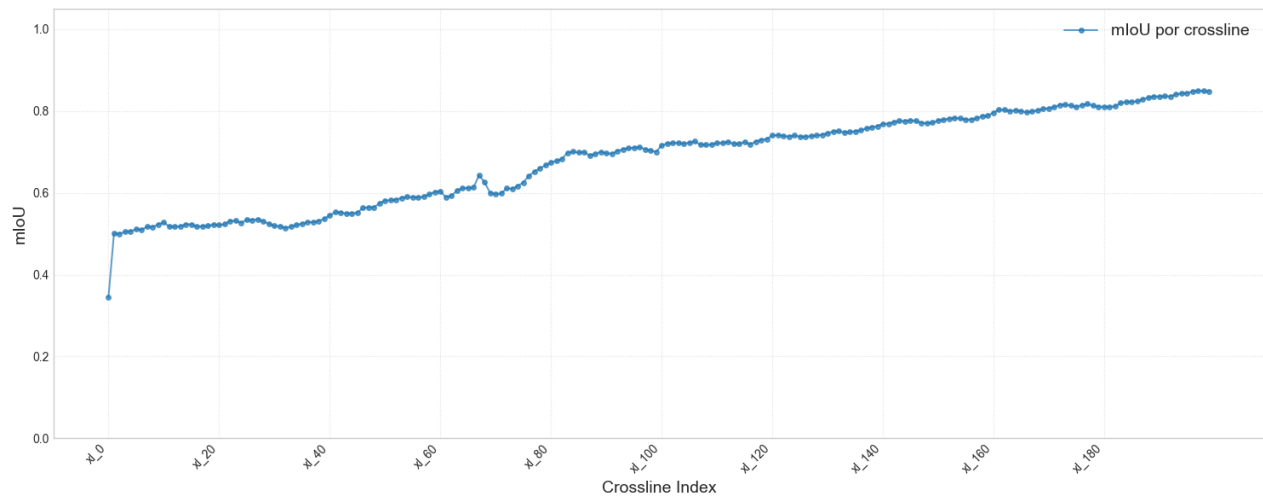


Figura 27: mIoU das predições do modelo from_scratch para as *crosslines* no particionamento de teste

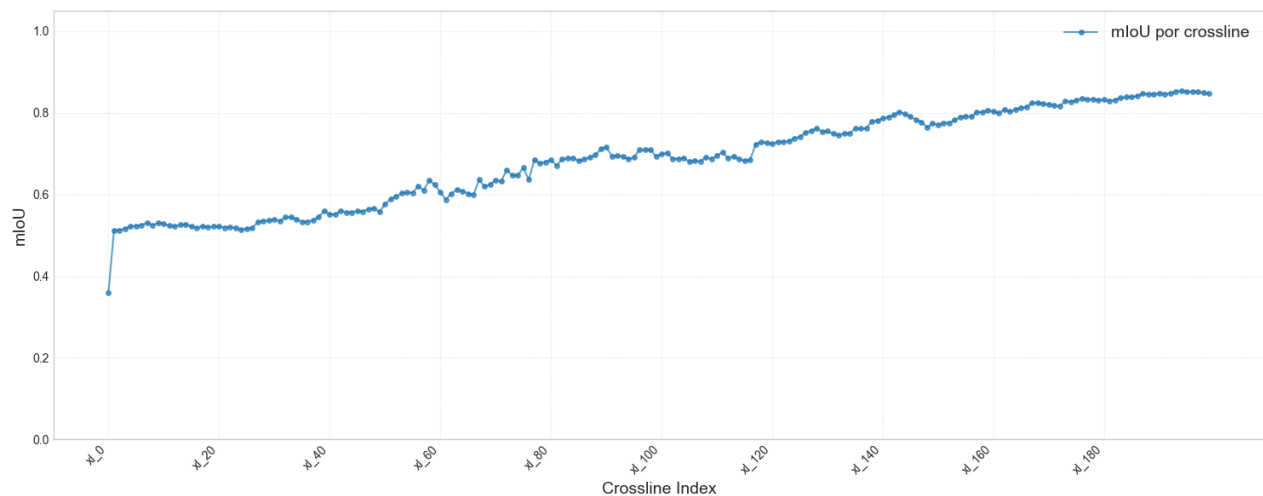


Figura 28: mIoU das predições do modelo `from_scratch` para as *crosslines* no particionamento de teste

4.5 Conclusão

No geral, notamos uma melhora no desempenho dos modelos pré-treinados com o uso de alguns atributos sísmicos, principalmente no regime de 128 amostras. Isso indica que o uso de atributos tem o potencial de evidenciar características do dado tanto na etapa de pré-treino quanto no *downstream*, sendo os principais atributos que beneficiaram o modelo o *instantaneous-phase*, *coherence* e *gst3d-dip*. Apesar disso, ainda é necessário um estudo mais aprofundado para determinar quais são os melhores atributos, o que cada um ajuda o modelo a destacar para a segmentação e qual é a combinação ótima de atributos para esse processo.

Além disso, por meio da análise das imagens, notamos que, apesar de apresentar maior mIoU, o modelo pré-treinado com BYOL parece performar pior nas regiões de interseção entre as classes do que o modelo pré-treinado com a ImageNet em algumas dessas regiões. Para conclusões mais robustas, mais experimentos precisam ser feitos devido aos fatores estocásticos presentes nos treinamentos, de forma a permitir a geração de um intervalo de confiança entre os resultados de cada modelo.

Trabalhos futuros podem explorar essa tendência de detecção de interseções entre as regiões, desenvolver modelos que caracterizem com mais precisão cada classe nas regiões centrais e, por fim, investigar otimizações no uso desses atributos.

Referências

- [1] S. Wiki, “Seismic attributes,” https://wiki.seg.org/wiki/Seismic_attributes, 2017, accessed: 2025-10-13.
- [2] —, “Seismic facies semantic classification,” https://wiki.seg.org/wiki/Seismic_Facies_Classification, 2019, accessed: 2025-10-13.
- [3] R. G. C. Adam D. Halpert and B. Biondi, “Seismic image segmentation with multiple attributes,” *ICLR*, 2009, accessed: 2025-10-13.
- [4] B. L. Jie Qi, Bo Zhang and K. Marfurt, “Seismic attribute selection for machine-learning-based facies analysis,” *Society of Exploration Geophysicists*, 2020, accessed: 2025-10-16.

- [5] M. L. A. A. . M. A. Amir Ismail, Ahmed A. Radwan, “Unsupervised machine learning and multi-seismic attributes for fault and fracture network interpretation in the kerry field, taranaki basin, new zealand,” *Geomechanics and Geophysics for Geo-Energy and Geo-Resources*, 2023, accessed: 2025-10-13.
- [6] D. Z. S. W. Junjie Ren, Hao Wu, “Automatic interpretation of seismic horizon based on multi-attribute and attention-unet,” *Journal of Geophysics and Engineering*, 2025, accessed: 2025-10-13.
- [7] E. R. K. A. . S. M. M. . A. A. Khayer, Keyvan; Hosseini Fard, “Integration of feature extraction, attribute combination and image segmentation for object delineation on seismic images,” *Acta Geophysica*, 2022, accessed: 2025-10-13.
- [8] B. A. A. M. H. O. J. A. dos Santos, “Self-supervised learning for seismic image segmentation from few-labeled samples,” <https://ieeexplore.ieee.org/document/9837909>, 2022, accessed: 2025-10-13.
- [9] S. S. R. Whitaker, “Adasemseg: An adaptive few-shot semantic segmentation of seismic facies,” *IEEE Transactions on Geoscience and Remote Sensing*, 2025, accessed: 2025-10-16.
- [10] D. G. Tobi Ore, “Seissegdiff: A label-efficient few-shot texture segmentation diffusion model for seismic facies classification,” *Computers Geosciences*, 2025, accessed: 2025-10-16.
- [11] J. G. T. C. J. Z. Q. C. Z. S. H. Luo, “A survey on self-supervised learning: Algorithms, applications, and future trends,” <https://ieeexplore.ieee.org/abstract/document/10559458>, 2024, accessed: 2025-10-13.
- [12] F. A. C. T. P. H. R. E. B. C. D. B. A. P. Z. D. G. M. G. A. B. P. K. K. R. M. M. V. Jean-Bastien Grill, Florian Strub, “Bootstrap your own latent: A new approach to self-supervised learning,” *NeuralIPS*, 2020, accessed: 2025-10-16.
- [13] S. Wiki, “Parihaka dataset 3d,” <https://wiki.seg.org/wiki/Parihaka-3D>, 2016, accessed: 2025-10-13.
- [14] J. C. Faracco, O. O. Napoli, J. Seródio, C. A. Astudillo, L. A. Villas, E. Borin, A. Souza, D. Miranda, and J. P. Navarro, “Dasf: A high-performance and scalable framework for large seismic datasets,” in *Fourth International Meeting for Applied Geoscience & Energy*. Society of Exploration Geophysicists, 2024.
- [15] S. W. contributors. (n.d.) Trace envelope. Acesso em: 10 Dec 2025. [Online]. Available: https://wiki.seg.org/wiki/Dictionary:Trace_envelope
- [16] ——. (n.d.) Instantaneous phase. Acesso em: 10 Dec 2025. [Online]. Available: https://wiki.seg.org/wiki/Dictionary:Instantaneous_phase
- [17] ——. (n.d.) Coherence. Acesso em: 10 Dec 2025. [Online]. Available: <https://wiki.seg.org/wiki/Coherence>
- [18] scikit-image development team. (2025) Glcm texture features — grey level co-occurrence matrix example. Acesso em: 10 Dec 2025. [Online]. Available: https://scikit-image.org/docs/0.25.x/auto_examples/features_detection/plot_glcm.html
- [19] scikit-image developers. (2025) Local binary pattern — scikit-image documentation. Acesso em: 10 Dec 2025. [Online]. Available: https://scikit-image.org/docs/0.24.x/auto_examples/features_detection/plot_local_binary_pattern.html

- [20] V. Karpushin and contributors. (2025) Anisotropic image segmentation by a gradient structure tensor. Acesso em: 10 Dec 2025. [Online]. Available: https://docs.opencv.org/4.x/d4/d70/tutorial_anisotropic_image_segmentation_by_a_gst.html
- [21] F. S. H. A. Liang-Chieh Chen, George Papandreou, “Rethinking atrous convolution for semantic image segmentation,” <https://arxiv.org/abs/1706.05587>, 2017, accessed: 2025-10-13.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [23] M. N. G. H. Ting Chen, Simon Kornblith, “A simple framework for contrastive learning of visual representations,” *ICML*, 2020, accessed: 2025-10-16.
- [24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.
- [25] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” *arXiv preprint arXiv:1609.04836*, vol. abs/1609.04836, 2016, published as a conference paper at ICLR 2017.
- [26] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, vol. abs/1502.03167, 2015, v3, revised 2 Mar 2015.
- [27] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018, *arXiv:1802.03426*.